# Reliability of Modular Fault-Tolerant Hypercube Networks

by

Abdul Hai Mohammed Abdulla

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

**MASTER OF SCIENCE**

In

**COMPUTER ENGINEERING**

January, 1995

# Reliability of Modular Fault-Tolerant Hypercube Networks

BY

Abdul Hai Mohammed Abdulla

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

**KING FAHD UNIVERSITY OF PETROLEUM & MINERALS**

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the

Requirements for the Degree of

**MASTER OF SCIENCE**

**In**

Computer Engineering

January 1995

UMI Number: 1362018

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS

DHAHRAN, SAUDI ARABIA

*This thesis, written by*

**Abdul Hai Mohammed Abdulla**

*under the direction of his Thesis Advisor, and approved by his Thesis committee, has been presented to and accepted by the Dean, College of Graduate Studies, in partial fulfillment of the requirements for the degree of*

**MASTER OF SCIENCE IN COMPUTER ENGINEERING**

***Thesis Committee:***

Chairman (Prof. Mostafa H. Abd-El-Barr)

Co-Chairman (Dr. Muhammad S. T. Benten)

Member (Dr. Sadiq M. Sait)

Member (Dr. Habib Youssef)

Dr. Samir H. Abdul-Jauwad
Department Chairman

Dr. Ala H. Al- Rabeh
Dean, College of Graduate Studies

Date: 4 . 2 . 95

Dedicated to

**my Mother and Father,**

## Acknowledgment

He has made subject to you, the night and the day, the sun and the moon. And the stars are in subjection: Verily in this are signs for men who are wise. It is He who sends down rain from the sky; From it ye drink, And out of it (grows), The vegetation on which Ye feed your cattle.

(The Holy Quran, Surah XVI)

All praise be to Almighty Allah Who gave me the patience and perseverance to carry out this work successfully.

Acknowledgement is due to King Fahd University of Petroleum and Minerals for providing support to this work.

I am deeply appreciative of my thesis committee chairman and academic advisor Prof. Mostafa H. Abd-El-Barr for his constant help, guidance and the countless hours of attention he devoted throughout the course of this work. My innumerable interruptions into his office always generated a cheerful response. Thank you Prof. Mostafa for introducing me to the world of fault-tolerant computing.

Thanks are due to my thesis committee Co-Chairman Dr. M. S. T. Benten and thesis committee members Dr. Sadiq M. Sait and Dr. Habib Youssef for their interest, co-operation, advice and constructive criticism.

I am also indebted to the department chairman, Dr. Samir H. Abdul-Jauwad and other faculty members for their support.

Lastly, but not the least, thanks are due to my family members and friends for their support and understanding, throughout my academic career and special thanks to my wife for her patience and encouragement throughout this work.

# Contents

# List of Tables

# List of Figures

# Abstract

**Name:** **Abdul Hai Mohammed Abdulla**

**Title:** **Reliability of Modular Fault-Tolerant Hypercube Networks**

**Major Field:** **Computer Engineering**

**Date of Degree:** **January, 1995**

*A new modular fault-tolerant hypercube architecture using Fault-Tolerant Basic Blocks (FTBBs) which can tolerate both node and link failures is proposed. A reconfiguration strategy to tolerate both node and link failures is given. In any hypercube network it is important to know the subcube reliability in the presence of node or/and link failures. We investigate subcube reliability of the proposed fault-tolerant hypercube architecture. The analysis shows that the proposed architecture has better subcube reliability than the basic hypercube. Further we extend our FTBB technique to hierarchical hypercubes which gives better system reliability than current techniques.*

**Master of Science Degree**

**King Fahd University of Petroleum and Minerals**

**Dhahran, Saudi Arabia**

**January 1995**

**جا معه الملك فهد للبترول والمعادن**

**كليه علوم وهندسه الحاسب الألى ـ قسم هندسه الحاسب الألى**

## موجز الرساله

**اسم الطالب** : عبد الحى محمد عبدا لله

**عنوان الرساله:** الاعتمادية للشبكات الفائقة التكعييب والمكونة من وحدات والمحتملة للاخطاء

**التخصص** : هندسه الحاسب الألى

**تاريخ الرساله :** يناير ١٩٩٥م

لقد حفّزت أهمية الشبكات الفائقة التكعيب العديد من الباحثين لاقتراح وتحليل وتصميم البناءات الفائقة التكعيب المتحملة للاخطاء. يمثل هذا العمل مسحاً لعدد من البناءات الفائقة التكعيب المتحملة للاخطاء المدروسة في مراجع البحث ولقد تم اقتراح بناء جديد فائق التكعيب ومحتمل للاخطاء ذو وحدات متنقلة باستخدام الوحدات الاساسية المتحملة للاخطاء (FTBBs) والذي يمكن ان يتحمل الانهيار العقدي والانهيار الوصلي وبالتالي فقد تم تقديم استراتيجية اعادة تشكيل لتحمل النهيار العقدي والوصلي. ففي اية شبكة فائقة التكعيب من الضروري معرفة احتمالية تضمين اكبر مكعب جزئي خالي من الاخطاء في حال وجود الانهيار العقدي والوصلي. لان ذلك يعتبر مقياساً لفعالية اجراء الخوارزميات الفائقة التكعيب على مختلف المكعبات الفائقة المتحملة للاخطاء.

ويتم في هذا البحث التحقق من اعتمادية المكعب الجزئي للبناء الفائق التكعيب المتحمل للتخطاء المقترح لاربعة نماذج انهيار : الانهيار العقدي, الانهيار الوصلي, الانهيار العقدي والوصلي المدمج ونماذج الانهيار للعقد فوق العادة. ويبين التحليل ان البناء الفائق التكعيب المتحمل للاخطاء الذي نقترحه هو اكثر مقاومة من المكعب الفائق الرئيسي من حيث قدرته على مساندة عدة مكعبات جزئية صغيرة في البناء المحطم ونذهب ابعد من ذلك بتوسيع اسلوب ال FTBB المقترح ليكون متضمناً في مكعبات فائقة هرمية مما يعطي اعتمادية للنظام اكثر مما تقدمه الاساليب الحالية.

**درجه الماجستير فى العلوم**

**جا معه الملك فهد للبترول والمعادن الظهران المملكه العربيه السعوديه**

**ينا ير ١٩٩٥م**

# Chapter 1

# Introduction

Hypercubes belong to a class of disjoint memory or distributed memory computers. Distributed memory computers have recently offered a cost-effective and feasible approach to supercomputing by connecting a large number of processors with local memory. Hypercube networks have received much attention over past few years since they offer a rich interconnection structure with large bandwidth, logarithmic diameter, and high degree of fault tolerance [20]. Many interconnection networks such as trees and multidimensional meshes can be embedded in the hypercube [35]. Popular networks such as PM21, Illiac,and shuffle exchange can be emulated by a hypercube with no hardware overhead. Another appealing feature of a hypercube is its homogeneity and symmetry. In contrast with trees and shuffle exchange networks, in a hypercube no node or link plays a special role. A number of hypercube machines have been implemented since the advent of the 64 processor Cosmic cube at Caltech

[38] such as Intel iPSC, Amtek's system/14 [19], NCUBE/10 [24] and the connection machine.

The probability of one or more processors failing in such a complex concurrent system is quite large. Further in a hypercube if a single processor or link fails, the physical topology is altered. The hypercube topology is ideally suited for running a large class of computationally intensive divide-and-conquer algorithms. Most of the algorithms for which the hypercube topology is ideally suited require that the topology does not change during their execution. Therefore some form of fault tolerance has to be introduced such that even in the presence of component (node and links) failures the topology is not altered. The importance of hypercube networks motivated several researchers to propose, analyze and design fault-tolerant hypercube based networks. In this thesis we survey a number of different fault-tolerant architectures proposed in the literature. We present a new modular fault-tolerant hypercube architecture using what are known as fault-tolerant basic blocks (FTBBs) and analyse its subcube reliability. Further we also extend our scheme for incorporating fault tolerance in hierarchical hypercube networks.

Section 1.1 of this introductory Chapter provides a brief introduction to the hypercube topology and its topological properties. Section 1.2 briefly introduces the reader to the previous work on fault-tolerant hypercube architectures, including the merits and demerits of the various fault-tolerant hypercube architectures investigated in the literature. Section 1.3 sets the motivation for our work. Section 1.4

defines the objectives of our work. In Section 1.5 we present the organization for rest of the thesis.

## 1.1 Hypercubes

In this section we introduce the hypercube topology and give a brief overview of the topological properties of the hypercube. A hypercube of dimension $d$ or a $d$-dimensional hypercube consists of $N = 2^d$ nodes placed at the vertices of a $d$-dimensional cube with $d2^{d-1}$ links interconnecting these nodes. A 3-cube is shown in Figure 1.1. Each node in a $d$-cube has a $d$-bit address consisting of $1s$ and $0s$. Two nodes whose addresses differ in a single bit position $i$, $0 \leq i \leq (d-1)$ are connected by a link which is said to span dimension $i$. Each of the $d$ dimensions (0 to $(d-1)$) have $N/2$ links spanning them [4, 35]. The minimum distance (number of links) between two nodes is the hamming distance between their addresses. The maximum internode distance is the maximum hamming distance between any two node addresses and is equal to $d$. The average internode distance is equal to $d/2$. Each node has direct links to $d$ other nodes i.e., to nodes whose addresses are at a hamming distance of one from it. A $j$-subcube of a $d$-cube is defined as a subgraph consisting of $2^j$ nodes obtained by choosing values ($0s$ and $1s$) for $d-j$ dimensions such that all node addresses in the $j$ subcube have the same values for these $d-j$ dimensions [1]. For example, there are two 2-cubes in a 3-cube, {0xx} and {1xx}

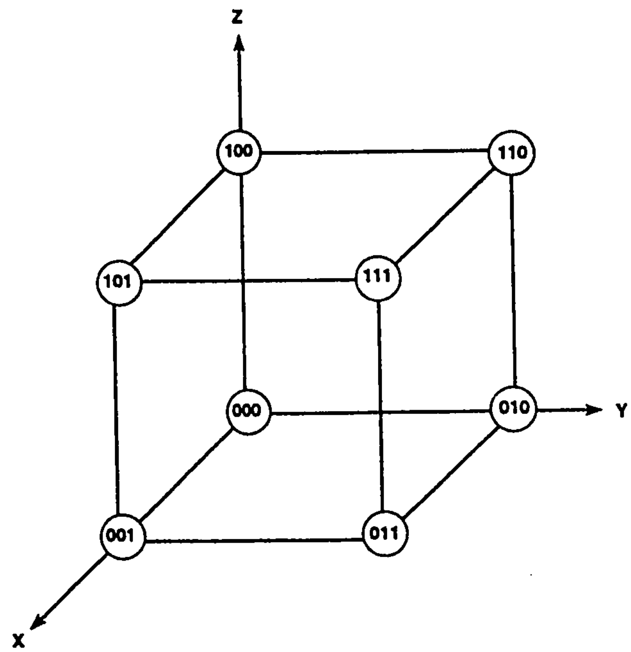


Figure 1.1: A hypercube of order d=3.

each containing four nodes, where x assumes a value either 0 or 1. In general we have $2^{d-j}$ disjoint $j$-subcubes in a $d$-cube. A $j$-subcube can be obtained by the following procedure. Split the $d$-cube across any dimension $i_1$ to get two $(d-1)$ subcubes (i.e., in one $(d-1)$ subcube all node addresses will have a '0' in dimension $i_1$ and in the other $(d-1)$ subcube all nodes addresses will have a '1' in dimension $i_1$). Next split the $(d-1)$ subcubes across any dimension $i_2$ to get four $d-2$ subcubes. Continue this procedure until each $j+1$ subcube has been split into two $j$ subcubes.

## 1.2 Previous Work

A number of fault-tolerant hypercube architectures have been proposed in the literature. In this Section we shall summarize the salient features of these architectures and then consider the motivation for our work. Chapter 3 considers each of the fault-tolerant architectures investigated in the literature in more detail.

The earliest fault-tolerant hypercube architecture is due to Rennels [34]. In Rennels's scheme a hypercube of dimension $d$ is constructed of $2^{d-m}$ fault-tolerant modules each of dimension $m$. A fault-tolerant module consists of a hypercube of dimension $m$ and some spares to replace faulty nodes in the module. Rennels uses two crossbar switches in each module to accomplish reconfiguration i.e., to switch in a spare node in place of faulty nodes. The demerit of Rennels's scheme is the high cost of crossbar switches. Further link failures are not tolerated by his scheme. Subcube reliability (explained in Section 2.2) is not analysed for Rennels's scheme.

Chau et al proposed another scheme similar to Rennels's idea of modular sparing [12]. But this scheme uses decoupling networks instead of crossbar switches to accomplish reconfiguration. This scheme is shown to have better reliability than Rennels's scheme for the same cost of hardware overhead [12]. Even this scheme does not tolerate link failures. Further link failures are not taken into consideration in the derivation of system reliability for this scheme. Subcube reliability is also not analysed.

Sultan and Melhem [42] proposed another modular sparing scheme using multiplexers and demultiplexers for reconfiguration within a module. Here the fault-tolerant module consists of $2^m$ active nodes and $k$ spare nodes. Unlike the fault-tolerant module of Rennels's scheme and the decoupling hypercube the failure of any active node in Sultan's scheme can be replaced only by two spare neighboring nodes in the same module. The advantage of Sultan's scheme is that it achieves better system reliability than that of Rennels's and Chau et al's schemes. But Sultan's scheme employs too many spare nodes and thus has a cost disadvantage.

Yang et al consider another modular fault-tolerant hypercube scheme [47]. In this scheme spare nodes in a fault-tolerant module can be used as local spares to replace the faulty nodes in the fault-tolerant module or as remote spares to replace the faulty nodes in other fault-tolerant modules via spare sharing links in the architecture. The advantage of this scheme is that it tolerates both node and link failures. The disadvantage is that it uses a lot of links in addition to the links in the hypercube. All the schemes above only evaluate the system reliability. Subcube reliability of their schemes is not investigated.

Banerjee and Peercy proposed another fault-tolerant hypercube architecture [7]. Their scheme uses spare processors attached to certain processors in the hypercube. The hypercube is composed of two class of nodes $P$ nodes which are ordinary nodes and $S$ nodes which have spare nodes attached to them. The $S$ nodes are embedded in a hypercube at appropriate places such that a $P$ node is adjacent to exactly one

$S$ node in the hypercube. Failures of the $P$ nodes are tolerated by adjacent $S$ nodes.

## 1.3 Motivation

In this Section we set the motivation for this thesis. In general there are two ways to introduce spare nodes and links in a hypercube system: using either *global* sparing or *modular* sparing. In *global* sparing, spare nodes or links can replace any failed primary node or link respectively in the system. *Global* sparing is difficult to implement due to increased amount of logic needed for reconfiguration i.e., a spare node or link should have the capability to replace any node or link in the hypercube. In *modular* sparing a hypercube is built of basic modules. This is feasible due to the recursive construction property exhibited by the hypercube. Specifically, a $d$-cube can be constructed from $2^m$ fault-tolerant basic subcubes each of dimension $d-m$. These fault-tolerant subcubes have in general some spare nodes and links to tolerate node and link failures within the basic block. In this thesis we adopt the modular sparing approach because of its many merits listed in Section 3.1.2. We also consider building hypercubes from basic blocks referred to as fault-tolerant basic blocks (FTBBs) because of their capability to tolerate node and/or link failures within the block.

Most of the fault-tolerant hypercubes described in literature tolerate only node failures [34, 12, 42]. Only overall system reliability is analysed for these fault-tolerant schemes. Subcube reliability which is an important issue for hypercubes (Section 2.2) is not investigated for any of the fault-tolerant hypercube architectures. However, subcube reliability analysis for the basic hypercube is provided in [1]. Link failures are not accounted for in the system reliability analysis in [34, 12, 42]. It should also be noted that most of them use hardware switches like crossbar switches [34] or multiplexers [42] or decoupling networks [12]. Until recent years, proposals for fault tolerance in distributed memory machines such as hypercubes and meshes assumed that only processors that are connected directly by links could communicate easily with each other. The hardware solutions to the problem focussed on direct replacement of nodes through extra links and switches [34, 12, 42, 8]. Schemes such as these carry a high cost due to ports and switches required to bring in spare links and nodes [7]. The scheme which we shall introduce avoids hardware switches. Further our scheme tolerates both node and link failures. More importantly we analyse subcube reliability of our scheme extending the analysis in [1] for fault-tolerant hypercube architectures.

In either global or modular sparing there are in general two techniques to replace the failed primary nodes or links by spare nodes or links respectively: i.e., reconfiguration by hardware switches or use of fault-tolerant routing. In modular sparing schemes [34, 12, 42] hardware switches are used for reconfiguration. As stated earlier hardware switches are extra overhead (cost) and they also introduce additional links into the system. Fault-tolerant routing schemes [7, 42] have the advantage of not introducing any hardware switches or links into the system. The disadvantage is that they alter the physical topology even though the logical topology is maintained. This could mean introducing a few extra link delays between some nodes. If these delays do not represent too much a penalty then, fault-tolerant routing schemes are excellent both cost and performance wise. Recently routing strategies based on circuit-switching concepts have been proposed for hypercubes and meshes where the message delays for multiple hop messages are only slightly greater than those for single hop messages. Hence it is possible to have logical links that span a number of physical links without much extra cost in the message delays. We adopt fault-tolerant routing into our architecture. The reconfiguration scheme which we shall describe uses the notion of logical links spanning one or more physical links to implement the desired topology [15, 22, 5, 31]. A penalty paid due to increased congestion in the physical links might induce delays in the presence of increased traffic. Hence even though multilink mapping of logical links is feasible with these technologies, it should be kept to a minimum.

The logical complexity of the link is less than that of the node. This implies that the failure rate of link should be lower than that of node. However link failures cannot be neglected as discussed in [1] since there are $d/2$ times more links than nodes in a $d$ dimensional hypercube. Subcube reliability analysis of the basic hypercube in [1] shows that if identical failure rates were assumed for the nodes and links then link failures dominate the reliability of the hypercube over the nodes. System reliability evaluation of most of the fault-tolerant schemes only considers node failures.

We propose a fault-tolerant hypercube architecture which combines several features of previous architectures. We use modular sparing i.e., spare nodes and links can replace failed primary node and links respectively within the fault-tolerant basic blocks. We define a FTBB of order $m$ or a $m$-FTBB to be a basic building block consisting of an $m$-cube and a spare node connected to all the primary nodes (nodes of the $m$-cube) with $2^m$ spare links. The scheme which we propose tolerates both node and link failures within a FTBB. No hardware switches are used for reconfiguration, instead we use fault-tolerant routing. Larger cubes are built from FTBBs using the recursive construction property of the hypercube. We analyse subcube reliability of our scheme permitting both node and link failures in the analysis. Further, we extend our technique to be incorporated in hierarchical hypercubes which gives better system reliability than current techniques.

## 1.4 Objectives of our Work

This section defines the objectives of our work. This thesis addresses the problem of fault-tolerance in hypercubes. The primary objective is to study the merits and demerits of existing fault-tolerant schemes in hypercubes and then propose a new fault-tolerant scheme or a modification to the existing one such that both node and link failures can be tolerated. Then we are required to develop a reconfiguration strategy for the proposed fault-tolerant hypercube architecture such that both node and link failures are tolerated. Our objective also includes the subcube reliability analysis of the proposed fault-tolerant scheme we propose for four failure models:

1. **Node failure model:** where only node failures are permitted.

2. **Link failure model:** where only link failures are permitted.

3. **Combined node and link failure model:** where both node and link failures are permitted.

4. **Supernode failure model:** an approximation to the combined node and link failure model.

adopting a methodology similar to the one considered in [1] to evaluate the subcube reliability of ordinary hypercube. We propose a new fault-tolerant scheme which is a modification to the Rennels's scheme and Sultan & Melhem's scheme so that both node and link failures are tolerated. We investigate the subcube reliability of the proposed scheme and compare it with the subcube reliability of the ordinary hypercube. Our secondary objective is to study the current fault-tolerant schemes in hierarchical hypercubes and then investigate the incorporation of modular fault-tolerance concepts in hierarchical hypercubes. We extend our FTBB technique to hierarchical hypercubes and build analytical models to evaluate the system reliability of fault-tolerant schemes including the FTBB scheme, in hierarchical hypercubes.

## 1.5 Organization of the Thesis

The thesis is organized as follows. Chapter 2 presents the required background material in fault-tolerance used throughout the thesis, the importance of subcube reliability in evaluating hypercubes & fault-tolerant hypercubes and the subcube reliability analysis of the basic hypercube for four failure models: node failure model, link failure model, combined node and link failure model and the supernode failure models. Chapter 3 presents a survey of a number of fault-tolerant hypercube and hierarchical hypercube architectures investigated in the literature. Chapter 4 presents the proposed modular fault-tolerant hypercube architecture utilizing FTBBs and also present the reconfiguration strategy adopted for this architecture in the presence of node or/and link failures.In Chapter 5, we investigate the reliability of the FTBBs. In Chapter 6, we investigate the subcube reliability of the proposed FTBB based hypercube architecture and compare its reliability with the ordinary hypercube. In the same chapter we also investigate the system reliability of fault-tolerant schemes in hierarchical hypercubes. Finally, Chapter 7 presents our contributions, conclusions and suggestions for future research.

# Chapter 2

# Subcube Reliability of Hypercube

Section 2.1 of this Chapter provides the background material in fault-tolerant computing used throughout the thesis. Section 2.2 discusses the subcube reliability importance in comparing different fault-tolerant hypercube architectures. The remaining sections discuss the subcube reliability of the basic hypercube for four failure models: node failure model, link failure model, combined node & link failure model and the supernode failure model [1].

## 2.1 Background Material

Fault Tolerance is an attribute of the system which aids the system to continue functioning in the presence of hardware failures or software errors [27]. System reliability ($R(t)$) at time $t$ is defined as the conditional probability that the system is operational during the interval [0,t] given that it was operational at time $t = 0$. System availability is defined as the probability that the system is operational at time $t$. Reliability and availability are important goals of fault tolerance. Mean Time To Failure (MTTF) is defined as the average time a system will continue to function before encountering a failure. Mathematically

$$MTTF = \int_0^\infty R(t)dt$$

The reciprocal of the MTTF i.e., the number of failures of a component or system per unit time is known as the failure rate of that component or system and is usually denoted by $\lambda$. Therefore $MTTF = \frac{1}{\lambda}$. A widely used realistic assumption in fault tolerance is the exponential failure law which relates the reliability '$R(t)$' and failure rate '$\lambda$' of a component and is given as $R(t) = e^{-\lambda t}$. Reliability evaluation of any system first consists of building a model for the system which is defined below.

A model for the hypercube or any system is an abstract representation of a system derived using various assumptions about the system and involving three steps: definition, parameterization and evaluation [17].

1. Definition: Here the system designer defines the system and the objective of modelling. The objective could be reliability, availability or performance related reliability or availability. The designer also defines precisely what constitutes an operational system. The minimum connectivity of the system for an operational system could be based on the following:

(a) *Terminal Reliability*: The system works as long as two specified nodes are working and connected.

(b) *Multiterminal Reliability*: The system works as long as connection between a specific set of input nodes and output nodes exist.

(c) *Network Reliability*: The system works as long as all nodes in the system are working and connected. This is also referred to as system reliability with the added constraint that no degradation is permitted in the working of the system.

(d) *Task Based Reliability*: The system works as long as some minimum number of connected nodes are available on the system for task execution.

(e) *Subcube Reliability*: The system works as long as some functional minimum degree (order) subcube exists. This is a special case of task based reliability.

The reliability of a hypercube based multi-computer is generally computed by using the above measures [1, 3, 11, 18, 20]. In hypercube structures measures $(a)$ and $(c)$ are useful for packet switching applications because they verify the sturdiness of the topology and depict the probability of successful flooding (for route setup or packet transmission). $(b)$ is good measure for reliability of dynamic multistage interconnection networks. The other two measures incorporate graceful degradation into reliability metrics. Measure $(e)$ is particularly important in hypercubes because most of the algorithms for hypercubes can be executed on various sizes of hypercubes by setting parameters appropriately.

2. Parameterization: The designer has to specify his input parameters for the model and output parameters in which he is interested. For example, for the hypercube the failure rate of the nodes and the links, their repair rate etc., can be input parameters. The model's output parameters include measures such as reliability, availability, MTTF(mean time to failure) etc.

3. Evaluation: The designer finally evaluates the model using analytical techniques or simulation. Analytical techniques include reliability block diagrams (combinatorial modelling), fault trees, markov models or petri nets.

A conventional reliability modelling approach considers a stochastic graph model for the hypercube system and evaluates reliability measures by assuming that nodes and/or links can fail independently with known probabilities. We us markov and combinatorial models to evaluate the reliability of the proposed architecture. Failure rates of nodes and links are input parameters and system reliability and subcube reliability are output parameters for our models.

## 2.2 Subcube Reliability

In this section we present the importance of subcube reliability for fault-tolerant hypercube architectures. A $d$-cube can be built recursively from smaller dimensional cubes. Take two 1-cubes and connect the corresponding nodes by links to get a 2-cube. Take two 2-cubes and connect the corresponding nodes by links to get a 3-cube and so on until the corresponding nodes of two $(d-1)$ cubes are connected to get a $d$-cube. This recursive construction of a hypercube from smaller subcubes proves very useful in task allocation and partitioning the binary $d$-cube for various applications. The subcubes have all the architectural properties of the larger cube so that many of the algorithms designed for hypercubes can use the size of the available cube as a runtime parameter. For example, in the NCube Multiprocessor (a binary 10 cube), its operating system called AXIS, permits the main cube array to be shared among many tasks, allocating subcubes of appropriate size to each task [24, 25, 13]. Because the subcubes are disjoint from each other, allocation of the partitions is easy since each task considers itself as working on an $i$-cube with nodes relabelled accordingly. It is possible to view an incoming task as a set of interacting modules that have to be assigned to the nodes of a subcube with adjacencies between modules in the task graph being preserved in the subcube. Algorithms have been developed to determine the size of the subcube required for each task under this condition [14]. In addition, efficient algorithms for many applications are designed to exploit the subcube partitioning ability of the hypercube, quite often in a recursive or divide-and-conquer fashion [28, 44]. It is useful to see how much of this ability is lost when failures begin to occur in a hypercube. Therefore evaluation of the subcube reliability - the probability of having functional subcubes of different sizes even in the presence of failures for any hypercube based architecture is important [39, 10]. This provides a basis for comparison of various fault-tolerant hypercube architectures in their efficacy in running the various hypercube algorithms.

Subcube reliability for the basic hypercube has been investigated in [1, 37] under four failure models, namely:

1. Node Failure Model - where only node failures are assumed to occur.

2. Link Failure Model - where only link failures are assumed to occur.

3. Combined Node and Link Failure Model - where both node and link failures are assumed to occur.

4. Supernode Failure Model - which is an approximation to the combined node and link failure model.

The reliability is computed in terms of the number of disjoint subcubes that can be embedded in a $d$-cube in the presence of node and/or link failures. In particular, the ability of a hypercube to embed a $(d-1)$ subcube (the largest fault-free subcube) in the presence of failures is considered. When component failures (node and link) occur the topology of the hypercube changes. For example, when a single node failure occurs in a $d$-cube it is no longer possible to have a functional $d$-cube but a functional $(d-1)$-subcube could be embedded. Thus some form of fault tolerance has to be incorporated such that at least the first few component (node and link) failures are tolerated and still a $d$-cube embedding is possible. Algorithms for operational subcube identification even in the presence of failures is treated in [2, 46]. A lower bound on hardware redundancy that is necessary to tolerate a single node and link failure is established in [39]. For all the failure models considered in this thesis we assume that both node and link failures obey exponential failure law with node failure rate denoted by $\lambda_n$ and link failure rate denoted by $\lambda_l$. Therefore evaluation of subcube reliability of any hypercube or fault-tolerant hypercube involves the evaluation of probabilities of existence of subcubes of different sizes inside a damaged or undamaged hypercube structure.

## 2.3 Node Failure Model

In this section subcube reliability of the basic hypercube is considered under the node failure model. We assume that a single node fails at any given time. Consider the embedding of a fault free $(d-1)$ subcube in a $d$-cube in the presence of node failures. A single node failure will always leave an undamaged $(d-1)$ subcube but two node failures could destroy all $(d-1)$ subcubes. For example, if node 0 and node $N-1$ fail there is no way of embedding a $(d-1)$ subcube. A fault free $(d-1)$ subcube exists if and only if all failures occur such that they can be enclosed in an $i$ subcube and $i < d$. This could be easily seen as follows. Split the $d$ cube along a dimension such that all failed nodes are left in an $i$ subcube $(i < d)$. This is not possible when nodes 0 and $N-1$ fail because both these failed nodes can only be enclosed in a $d$-cube. We shall now define the system states.

Let $S_i$ be defined as the system state signifying that all failures that have occurred could be enclosed in a maximal $i$ subcube i.e., a lower order subcube that encloses all the failed nodes does not exist. In terms of functional cubes state $S_i$ can be characterized as embedding $(d-i)$ disjoint fault free subcubes of order $(d-1), (d-2), (d-3), \cdots, i$ respectively. This could be seen as follows. Split the $d$ cube into two subcubes of order $(d-1)$ such that the faulty $i$ subcube is enclosed in one $(d-1)$ subcube. The other $(d-1)$ subcube is fault free. Now split the faulty $(d-1)$ subcube into two $(d-2)$ subcubes such that the faulty $i$ subcube is enclosed in one $(d-2)$ subcube. Continue this procedure until a faulty $(i+1)$ subcube is split into two $i$ subcubes such that one is faulty and the other one is fault free. By this procedure we would have obtained $(d-i)$ disjoint subcubes mentioned above. Although we cannot embed a fault free subcube of order greater than or equal to $(i-1)$ it could be possible to enclose subcubes of order $(i-2)$ or less inside the faulty $i$ subcube. The state diagram along with the transition rates is shown in Figure 2.1. State $S_d$ means that all failures have occurred in a maximal $d$ cube so that no embedding of $(d-1)$ subcube is possible. $S_*$ represents the fault free initial state or perfect state.

We shall now describe the state transitions. Consider the state transition from $S_*$ to $S_0$. Any node failure out of the $N$ nodes can cause a transition from $S_*$ to $S_0$. Therefore the transition rate is $N\lambda_n$. Now consider the state transition from $S_i$ to

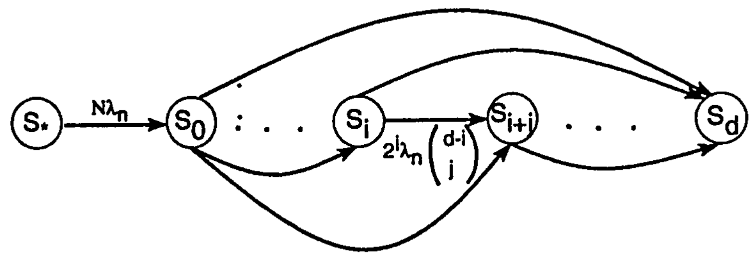


Figure 2.1: System state diagram for the node failure model.

$S_{i+j}$. $S_i$ signifies that all failures have occurred within a $i$ subcube. This takes up $i$ dimensions of the $d$. Label these dimensions as faulty. $(d-i)$ fault free dimensions remain. $S_{i+j}$ signifies that all failures have occurred in a $(i+j)$ subcube. This labels $(i+j)$ dimensions faulty. The transition from $S_i$ to $S_{i+j}$ then means additional $j$ dimensions become faulty out of the fault-free $(d-i)$ dimensions. $j$ dimensions out of $(d-i)$ can be chosen in $\binom{d-i}{j}$ ways. Considering one of these ways we see that the original faulty $i$ dimensions could take up values either $0s$ or $1s$ (total $2^i$ ways). Each of the different ways represents the address of the node which could fail to cause the transition. Therefore the transition rate is $\binom{d-i}{j}2^i\lambda_n$. Let the probability of being in state $S_i$ be $P_i(t)$ at time $t$. The state equations for this system are given as follows:

$\delta P_*/\delta t = -\lambda_n N P_*$,

$\delta P_0/\delta t = -\lambda_n(N-1)P_0 + \lambda_n N P_*$,

$\delta P_i/\delta t = -\lambda_n(N-2^i)P_i + \sum_{j=0}^{i-1}\lambda_n 2^j\binom{d-j}{i-j}P_j, \ \ 0 < i \leq d.$

The initial conditions are $P_*(0) = 1$ and $P_i(0) = 0$ for all $i$. It can be shown by induction on $i$ that the solution to this system of equations is [1]

$$P_i(t) = (-1)^{i+1}\binom{d}{i}2^{d-i}e^{-\lambda_n Nt} + \binom{d}{i}\sum_{m=0}^{i}(-1)^{i-m}\binom{i}{m}2^{d-m}e^{-(N-2^m)\lambda_n t} \quad (2.1)$$

Let us define the reliabilities as follows

$R_*(t) = P_*(t)$,

$R_0(t) = P_0(t) + R_*(t)$ and

$R_i(t) = P_i(t) + R_{i-1}(t)$, for all $i$.

Thus $P_i(t)$ is the probability that all failures are enclosed in an $i$ subcube. $R_i(t)$ is the probability that all failures have occurred in a subcube of order less than or equal to $i$. So the probability $R_i(t)$ is larger than $P_i(t)$. The system's mean time to failure can be evaluated by integrating $R_i(t)$. The MTTF is given by the expressions

$T_* = 1/N\lambda_n$,

$T_0 = T_* + 1/(N-1)\lambda_n$ and

$$T_i = T_{i-1} + (-1)^{i+1}\binom{d}{i}\frac{1}{2^i\lambda_n} + \binom{d}{i}\sum_{m=0}^{i}(-1)^{i-m}\binom{i}{m}\frac{2^{d-m}}{(N-2^m)\lambda_n} \quad (2.2)$$

The curves for $R_i(t)$ and the values of MTTF are shown in Figure 2.2 and Table 2.1 respectively. We see from Figure 2.2 that values of $R_i(t)$ increase with $i$. This is because with larger $i$ more failures can be tolerated. Let us interpret the MTTF numbers taking the 10 cube as an example. The mean time to the first node failure

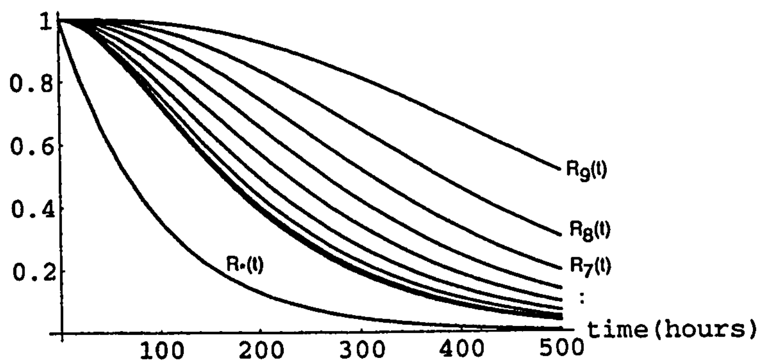


Figure 2.2: 10 cube reliabilities $R_i(t)$ for the node failure model.

Table 2.1: MTTF in hours for node failure model ($\lambda_n = 10^{-5}/hour$).

| $d$ | $N$ | $T_*$ | $T_0$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ | $T_8$ | $T_9$ | $T_{10}$ | $T_{11}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 6 | 64 | 1563 | 3150 | 3303 | 3726 | 4468 | 5611 | 7867 | | | | | | |
| 7 | 128 | 781 | 1569 | 1612 | 1750 | 2014 | 2400 | 2968 | 4094 | | | | | |
| 8 | 256 | 391 | 783 | 795 | 839 | 934 | 1079 | 1272 | 1554 | 2118 | | | | |
| 9 | 512 | 195 | 391 | 394 | 408 | 442 | 498 | 573 | 668 | 808 | 1090 | | | |
| 10 | 1024 | 98 | 195 | 196 | 201 | 212 | 234 | 264 | 301 | 348 | 418 | 559 | | |
| 11 | 2048 | 49 | 98 | 98 | 99 | 103 | 111 | 124 | 139 | 157 | 180 | 216 | 286 | |
| 12 | 4096 | 24 | 49 | 49 | 49 | 51 | 54 | 59 | 65 | 73 | 81 | 93 | 111 | 146 |

is 98 hours. If the system can stay operational with disjoint subcubes of $9, 8, 7, 6, 5$ and 4 (i.e., all failures could be enclosed in a 4 cube ), then the MTTF increases to about 10 days. Much of this increase materializes from just the ability to tolerate first node failure ($T_0 = 195$ hours).

## 2.4 Link Failure Model

We now consider the effect of link failures. As before we are interested in embeddings which have a $(d-1)$ subcube. We again assume that the link failures follow the exponential failure law and that the failure rate of a link is $\lambda_l$. Each dimension $i$ $(0 \leq i < d)$ has $N/2$ links spanning them. We can split a $d$ cube across any of its dimensions and get two $(d-1)$ cubes. By splitting a $d$ cube across any dimension we mean that all links spanning that particular dimension are cut. So there are a total of $2d$ possible $(d-1)$ subcube embeddings. Not all these subcubes are disjoint. Let us label the $(d-1)$ subcubes as follows. $C_{i,j}$ ( $0 \leq i < d, 0 \leq j \leq 1$) refers to a subcube with all nodes having a $j$ in their $i$th bit position. Only $C_{i,0}$ and $C_{i,1}$ are disjoint for all $i$. Any other pair of subcubes will have a common $d-2$ subcube i.e., $C_{i,j}$ and $C_{k,l}$ will share a $d-2$ subcube with node addresses having a $j$ in their $i$th bit position and a $l$ in their $k$th bit position. The first link to fail will destroy $(d-1)$ of the $2d$ possible $(d-1)$ subcube embeddings. This can be seen as follows. Without loss of generality assume that the first link to fail is the link connecting node 0 and node 1. Then the link spans dimension 0. When the $d$-cube is split across $0th$ dimension we get two fault-free $(d-1)$ subcubes since anyway the failed link (link between node 0 and node 1) is cut and is not part of any $(d-1)$ subcube obtained by the split. If the $d$-cube is split across any other dimension we get one fault-free subcube and the other which contains the failed link, since in this case the faulty link is not split. The second link failure may or may not destroy the remaining subcubes depending on whether it belongs to the $0th$ dimension or not. To characterize the subcube reliability we have to now define the system states for every link failure just like the node failure model.

Consider again the failure of the link between node 0 and node 1. This damages only subcubes $C_{i,0}, i > 0$ leaving the remaining $d+1$ subcubes undamaged. While any link in the system is equally likely to fail, for the purpose of analysis we relabel all the nodes in the cube such that the faulty link is mapped to the link between node 0 and node 1. Since only $0th$ dimension is fixed by this relabeling we have $(d-1)!$ ways of relabeling the remaining dimensions. When additional links fail an appropriate labelling for the remaining dimensions may be chosen. Only if the subsequent link failure does not belong to the already labelled dimension, do we have to choose an additional labelling for the new failed link. We effectively assume therefore, that the cubes are damaged in order i.e., $C_{1,1}$ is damaged first then $C_{2,1}$ and so on.

We now define the system states in terms of the undamaged cubes remaining in the system as follows. The possible system states ( $0 \leq i < d$) are

$S_* = \{C_{k,j}, 0 \leq k < d, 0 \leq j \leq 1\}$

$S_{2,i} = \{C_{0,0}, C_{0,1}, C_{j,1}, i < j < d\}$

$S_{1,i} = \{C_{0,1}, C_{j,1}, i < j < d\}$

$S_{0,i} = \{C_{j,1}, i < j < d\}$.

The states $S_{2,i}$ each support two disjoint $(d-1)$ subcubes and these are the only states which do not have equivalent states in the node failure model. State $S_{1,i}$ embeds $d-i$ fault free subcubes of order $d-1, d-2, \cdots, i$ respectively. We can summarize the equivalence between the states for the node and the link failure models as follows. State $S_{2,i}$ does not have any equivalent state in the node failure model. $S_{1,0}$ is equivalent to state $S_0$ in the node model. State $S_{0,d-1}$ is equivalent to $S_d$, the state in which no $(d-1)$ subcube embedding is possible. The state diagram for the link failure model is shown in Figure 2.3. Now we shall explain the transition rates. The transition rate from $S_*$ to $S_{2,0}$ is $\lambda d 2^{d-1}$. This is because the failure of any link in the fault free system will result in the system state of $S_{2,0}$.

A detailed description of the transition rates between the remaining states now follows. Let us first consider the transitions from state $S_{2,0}$ to state $S_{2,j}$; $C_{0,0}$ and $C_{0,1}$ are both undamaged in these transitions. Any link failure contributing to this transition must span dimension 0. Without loss of generality, let the new failed link be incident to node $x$, $x$ belongs to $C_{0,0}$. Consider such nodes with exactly $j$ bits of value 1 in the node address. We may remap all the nodes in the cube so that the $j$

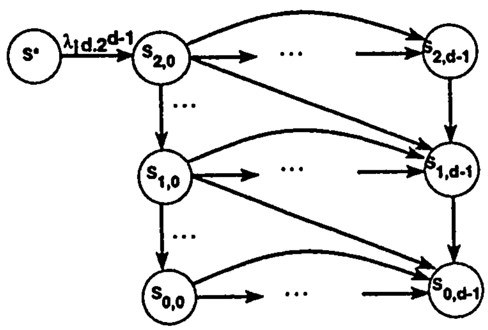


Figure 2.3: System state diagram for the link failure model.

bits of value 1 in the address of node $x$ will be in bit positions $1, 2, ..., j$. Note that $x$ and its new faulty link (across dimension 0) are now in $C_{k,l}$ for all $k$, $0 < k \leq j$. Therefore, exactly $j$ additional subcubes have been damaged. We may count the number of nodes $x$ by counting the nodes in $C_{0,0}$ with exactly $j$ 1 bits; this results in a transition rate from state $S_{2,0}$ to state $S_{2,j}$ of $\lambda\binom{d-1}{j}$.

To generalize this for the $S_{2,j}$ to $S_{2,i+j}$ transition, $(0 \leq i < d, 0 < j < d-i)$ we are again concerned only with faulty links that span the 0th dimension. Since the sub-cubes $C_{k,l}$, $i < k \leq d$ are already damaged, only the $d-1-i$ sub-cubes $C_{k,l}$, $i < k < d$ need be considered. Let us consider a faulty link incident to node $x$, $x$ belongs to $C_{0,0}$, with a node address containing exactly $j$, 1 bits in the bit positions greater than $i$. Since the only dimension labels fixed from previous mappings are those that are $\leq i$, we may remap all the nodes in the cube so that the $j$ 1 bits just described are in bit positions $i+1, i+2, ..., i+j$. Clearly $x$ and its new faulty link are in $C_{k,l}$ for all $k$, $i < k \leq i + j$. Thus, we count the number of nodes in $C_{0,0}$ with an address containing exactly $j$ 1 bits in bit positions $k$, $i < k < d$ and obtain a transition rate of $\lambda\binom{d-1-i}{j}2^i$. This rate (among others) is shown in Figure 2.4. The remainder of the transitions from state $S_{2,j}$ involve the links which do not span dimension 0. Failure of any of the nonzero dimensioned links will damage either $C_{0,0}$ or $C_{0,1}$ plus $j$ additional subcubes $0 \leq j < d - i$. Since we may remap the cube so that $C_{0,0}$ is always damaged first, we may consider links in $C_{0,0}$ without loss of generality. To account for link failures in $C_{0,1}$ the final rate will be doubled.

Consider some node $x$ in $C_{0,0}$ with an address containing $j$ bits with value 1 in nodes in the cube so that the $j$ bits with value 1 just described are in bit positions $i+1, i+2, \cdots, i+j$. We need to determine all the links incident to $x$ (other than the link spanning dimension 0) such that $x$ and this link will be in $C_{k,l}$, for all $k$, $i < k \leq i + j$. The only links which fit this description span the dimensions $1, 2, \cdots, i, i+j+1, i+j+2, \cdots, d-1$. We may determine the number of nodes $x$ by counting the number of nodes in $C_{0,0}$ with an address containing exactly $j$ 1 bits in bit positions $k$, $i < k < d$. However, we may not simply multiply this figure by $d - 1 - j$ to obtain the number of links since the links which span dimensions $1, 2, ..., i$ are incident to two nodes with exactly $j$ 1 bits in bit positions $k$, $i < k < d$. Thus, half of these links must be subtracted out so that the total number of links is $2^i\lambda\binom{d-1-i}{j}(d - 1 - i/2 - j)$. To take into account links in $C_{0,1}$, we double this figure to obtain the rate for the $S_{2,i}$ to $S_{1,i+j}$ transition as $2^{i+1}\lambda\binom{d-1-i}{j}(d - 1 - i/2 - j)$

for $0 \leq i < d$, $0 \leq j < d-i$.

The transitions from state $S_{1,i}$, $0 \leq i < d$ (See Figure 2.4(b)), are similar to the transitions out of state $S_{2,i}$. For the $S_{1,i}$ to $S_{1,i+j}$,$(0 < j < d-i)$ transition, start with the rate $2^i\lambda\binom{d-1-i}{j}$ for links spanning dimension 0. $2^i\lambda\binom{d-1-i}{j}(2d-1-i-2j)$. The transitions rates are indicated in the Figure 2.4.

From the state diagram we can write the state equations. Let $P_{i,j}(t)$ be the probability of being in state $S_{i,j}$ at time $t$. The solution to the system of equations is obtained numerically. We can define the following probabilities.

$P_*(t)$ =Probability that no failures have occurred.

$P_{2-(d-1)cubes}(t)$ = probability that link failures have occurred, but two disjoint subcubes can be embedded.

$P_i(t)$ =probability that link failures have occurred leaving $d-i$ functional disjoint subcubes of order $d-1, d-2, ..., i$ respectively. Reliability measures are then given by:

$R_*(t) = P_*(t)$

$R_{2-(d-1)cubes}(t) = R_*t + P_{2-(d-1)cubes}(t)$

$R_0(t) = R_{2-(d-1)cubes}(t) + P_0(t)$

$R_i(t) = R_{i-1}(t) + P_i(t)$, $0 \leq i \leq d$.

The mean time to failure $T$, corresponding to these reliability measures are shown in Table 2.2.

Note that we have used a lower failure rate for the links than that for the nodes,

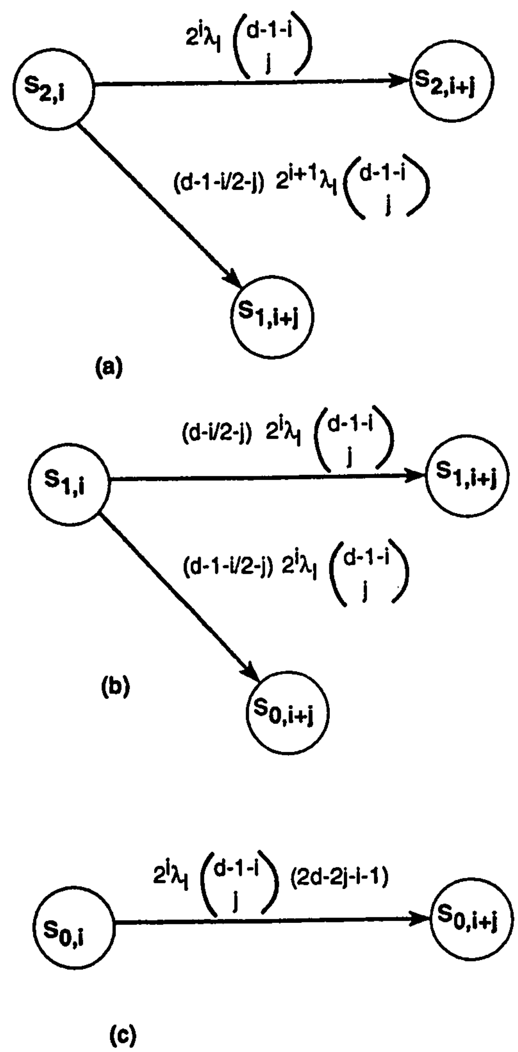


Figure 2.4: Transition rates for the link failure model.

Table 2.2: MTTF in hours for link failure model ($\lambda_l = 10^{-6}/hour$).

| $d$ | $N$ | $T_*$ | $T_{2-}$ | $T_0$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ | $T_8$ | $T_9$ | $T_{10}$ | $T_{11}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 6 | 64 | 5208 | 11458 | 11740 | 12957 | 15298 | 18552 | 23434 | 33361 | | | | | | |
| 7 | 128 | 2232 | 4836 | 4897 | 5213 | 5924 | 6945 | 8273 | 10284 | 14353 | | | | | |
| 8 | 256 | 977 | 2093 | 2106 | 2188 | 2406 | 2753 | 3184 | 3745 | 4601 | 6324 | | | | |
| 9 | 512 | 434 | 922 | 925 | 947 | 1013 | 1132 | 1288 | 1471 | 1715 | 2088 | 2835 | | | |
| 10 | 1024 | 195 | 412 | 413 | 419 | 438 | 479 | 537 | 605 | 685 | 793 | 958 | 1287 | | |
| 11 | 2048 | 89 | 186 | 187 | 188 | 194 | 207 | 229 | 256 | 285 | 321 | 369 | 443 | 591 | |
| 12 | 4096 | 41 | 85 | 85 | 85 | 87 | 92 | 100 | 110 | 122 | 135 | 151 | 173 | 206 | 273 |

to account for their lower logical complexity.

## 2.5 Combined Node and Link Failure Model

In this section, the effect of both node and link failures on the system reliability is explored. Under the node failure model, the failure of links can be ignored because link failures may be considered as a failure of one of its terminal nodes. Thus the node failure rate under that model encompassed the failure of the node and all of its incident links. However, to apply this technique to the link failure model would necessitate modeling a node failure as the simultaneous failure of all its incident links. This would violate the assumption that failures are independently distributed. For this reason a combined node and link failure model is developed.

Let the node failure rate be $\lambda_n$ and the link failure rate be $\lambda_l$ . We again assume that both the node and link failures obey the exponential failure law. Both failure rates are assumed to be constant and independent. We start from the link failure model. We shall see that this model and the state transitions accurately model link failures under the combined model. It only remains to know what happens to the links of a failed node. In the link failure model the failure of a link is significant only if it damages an as yet undamaged $(d-1)$ subcube. Since links incident to failed nodes are not part of any undamaged subcube, they can be ignored in the analysis. Thus the state diagram and link failure transitions (with $\lambda = \lambda_l$) of Figures 2.3

and 2.4 are valid for the link failure transitions for the combined model. All that now remains is to add the node failure transitions rates. These rates are developed below.

When a node fails in the fault free system it can be mapped to node 0 of the system. This failure damages $d$ subcubes ($C_{i,0}$ for $0 \leq i < d$) and corresponds to state $S_{1,0}$. Thus we have new transition from $S_*$ to state $S_{1,0}$ with rate $\lambda_n 2^d$ (see Figure 2.5(a)). Any node failure in state $S_{2,i}$, $0 \leq i < d$ puts the system into state $S_{1,i+j}$, $0 \leq j < (d-i)$, since it damages $C_{0,0}$ or $C_{0,1}$ and $j$ subcubes. The rates of these transitions may be derived in a manner similar to the way link failures were derived i.e., counting the number of nodes with addresses containing $j$ bits with value 1 in the $d-1-i$ positions greater than $i$. The rate must be doubled to account for failures in $C_{0,0}$ and $C_{0,1}$, leading to the transitions in Figure 2.5(a). In state $S_{1,i}$, $0 \leq i < d$, a node failure will damage all, some or none of the remaining subcubes. First considering the nodes in $C_{0,0}$, we see that only those nodes with addresses containing 1 bits in bit positions greater than $i$ will damage additional subcubes. These are the transitions from state $S_{1,i}$, to state $S_{1,i+j}$, $0 < j < (d-i)$ shown in Figure 2.5(b). Next if the faulty node is in $C_{0,1}$, at least one subcube ($C_{0,1}$) will be damaged; j additional subcubes $0 \leq j < (d-i)$ will be damaged for nodes with addresses containing $j$ 1 bits in bit positions greater than $i$. These are the transitions from $S_{1,i}$ to $S_{0,i+j}$ depicted in Figure 2.5(b). Finally the transitions from state $S_{0,i}$, $0 \leq i < d$ (see Figure 2.5(c)) correspond to the failure of nodes with

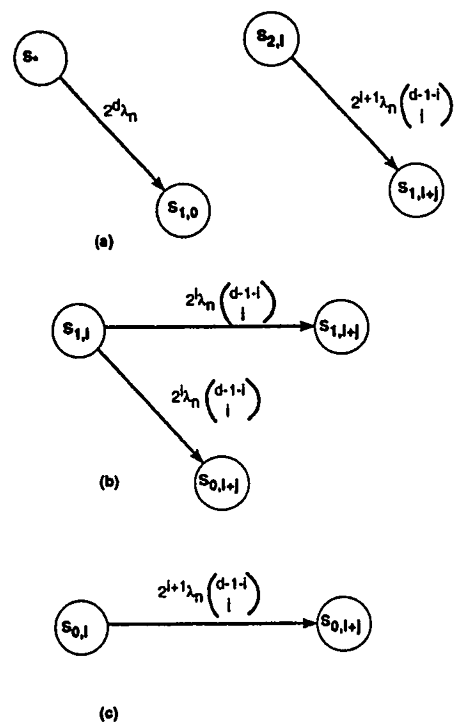


Figure 2.5: Additional failure rates for the combined model.

addresses containing $j$ bits with value 1, $0 < j < (d - i)$ in bit positions greater than $i$.

The node transition rates can now be combined with the link transition rates to derive the system of equations. We can solve the system of equations numerically and the reliability measures defined in the previous section are computed and shown in Table 2.3. The reliability and the MTTF are seen to be lower than those under the node and link failure models because now both link and node failures are taken into account.

Table 2.3: MTTF in hours for combined node and link failure model ($\lambda_l = 10^{-6}/hour, \lambda_n = 10^{-5}/hour$).

| $d$ | $N$ | $T_*$ | $T_{2-}$ | $T_0$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ | $T_8$ | $T_9$ | $T_{10}$ | $T_{11}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 6 | 64 | 1202 | 1490 | 2443 | 2599 | 2982 | 3603 | 4532 | 6369 | | | | | | |
| 7 | 128 | 579 | 735 | 1170 | 1215 | 1341 | 1559 | 1862 | 2305 | 3187 | | | | | |
| 8 | 256 | 279 | 362 | 563 | 575 | 616 | 695 | 806 | 950 | 1162 | 1587 | | | | |
| 9 | 512 | 135 | 178 | 271 | 285 | 288 | 316 | 359 | 412 | 481 | 483 | 787 | | | |
| 10 | 1024 | 65 | 88 | 131 | 132 | 136 | 146 | 163 | 184 | 209 | 242 | 291 | 390 | | |
| 11 | 2048 | 32 | 43 | 63 | 64 | 65 | 68 | 75 | 83 | 93 | 105 | 121 | 145 | 193 | |
| 12 | 4096 | 15 | 21 | 31 | 31 | 31 | 32 | 35 | 38 | 42 | 47 | 53 | 61 | 72 | 95 |

## 2.6 Supernode Failure Model

We present now a simplified model for the combined failure model known as the supernode failure model. This is an approximation to the combined node and link failure model. We define a supernode to be a node and half its incident links i.e., each link is associated with a node. The failure of any of the $1 + d/2$ components of the supernode would lead to its failure. The failure rate for the supernode is given by $\lambda = \lambda_n + d\lambda_l/2$. The supernode is treated as though it is an independent component. Substituting this value of $\lambda$ into the node failure model we derive the MTTF given in Table 2.4. This provides results close to the exact combined model.

Table 2.4: MTTF in hours for supernode failure model ($\lambda = \lambda_n + \lambda_l d/2$).

| $d$ | $N$ | $T_*$ | $T_0$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ | $T_8$ | $T_9$ | $T_{10}$ | $T_{11}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 6 | 64 | 1202 | 2423 | 2541 | 2866 | 3437 | 4316 | 6051 | | | | | | |
| 7 | 128 | 579 | 1162 | 1194 | 1296 | 1491 | 1778 | 2198 | 3033 | | | | | |
| 8 | 256 | 279 | 559 | 568 | 600 | 667 | 771 | 909 | 1110 | 1513 | | | | |
| 9 | 512 | 135 | 270 | 272 | 282 | 305 | 344 | 395 | 461 | 558 | 752 | | | |
| 10 | 1024 | 65 | 130 | 131 | 134 | 142 | 156 | 176 | 201 | 232 | 279 | 373 | | |
| 11 | 2048 | 32 | 63 | 63 | 64 | 67 | 72 | 80 | 90 | 101 | 116 | 139 | 185 | |
| 12 | 4096 | 15 | 31 | 31 | 31 | 32 | 34 | 37 | 41 | 45 | 51 | 58 | 69 | 91 |

## 2.7 Discussion of the Results

In this Section we discuss the results i.e., the reliability and the MTTF obtained for each of the models above. From Figure 2.2 we see that the reliabilities $R_i(t)$ increase with the value of $i$. $R_i(t)$ is the probability that the failures are contained in a subcube of order $i$ or less. For large $i$ the permissible subcube in which to encompass all failures is bigger and so is easier to satisfy since now more node failures are tolerable. Hence the probability is large. $R_*(t)$ signifies the perfect cube and so is most difficult to satisfy . Therefore this is the lowermost curve in Figure 2.2. Table 2.1 shows the MTTF values for the node failure model. For a fixed $d$ we see that the MTTF $T_i$ increases with $i$. For example, with $d = 10$ the MTTF for the first node failure ($T_*$) is 98 hours and for the second node failure ($T_1$, all failures could be enclosed in a 0 subcube) is 195 hours. The MTTF for subsequent failures does not increase to a large extent until $T_5$ (all failures could be enclosed in a 5 subcube). We notice that the any MTTF $T_i$ decreases with increase in the order ($d$) of the hypercube. In fact $T_*$, the MTTF to the first node failure (to leave the perfect state), nearly gets half as the order of the hypercube increases by one (i.e., the nodes are doubled). For large practical hypercubes for example, $d = 11$ or $d = 12$ we see that the MTTF to leave the perfect state is rather very small just 49 and 24 hours respectively. Again consider the MTTF values for a $d = 10$ cube. The MTTF to the first node failure is 98 hours. If the system can stay operational with disjoint subcubes of order $9, 8, 7, 6, 5$ and 4 (i.e., all failures could be enclosed in a 4-cube), then the MTTF increases to about 10 days [1]. Note that much of this increase materializes from just the ability to tolerate one node failure ($T_0 = 195$ hours). The key point is then this. If we could tolerate the first node failure using fault masking by having a spare node i.e., the first failed node can be replaced by the spare node, then the MTTF to leave the perfect state ($T_*$) and $T_0$ can be improved. Other subcube MTTFs ($T_i$'s) will also be improved. Similar remarks hold for the link and combined node and link failure models. The reliability and the MTTF for the link failure model could be improved by the same technique as for the node model i.e., we can have spare links to replace failed links in the system. We use the above markov models for to evaluate the subcube reliability of the proposed fault-tolerant hypercube architecture for each of the failure models considered above.

# Chapter 3

# Existing Fault-Tolerant Hypercubes

In this chapter we shall review in detail different fault-tolerant hypercube and hierarchical hypercube architectures investigated in the literature. Section 3.1 presents the various fault-tolerant hypercube architectures while Section 3.2 considers fault-tolerant hierarchical hypercube architectures.

## 3.1 Existing FTBB Based Hypercubes

This section reviews the following fault-tolerant hypercube architectures

1. Rennels's Scheme.

2. Chau et al's Scheme.

3. Sultan and Melhem's Scheme.

4. Banerjee and Peercy's Scheme.

5. Yang et al's Scheme.

### 3.1.1 Rennels's Scheme

Rennels [34] suggests two schemes that use spare nodes to tolerate node failures in a hypercube. The first scheme can be used for systems that do not require very high reliability. Here an $n$-cube is divided into $2^s$ subcubes each of dimension $m$ where $n = s + m$. One spare node is used to back up the nodes in each subcube. The spare node which may be used to replace any failed node in the subcube is connected to every node in the subcube and to each of its neighbors by means of two crossbar switches. The first crossbar has $2^m + s$ inputs and $n$ outputs. The second crossbar has $2^m$ inputs and $s$ outputs. In addition, each node requires an extra port to connect to the crossbar switches.

For very high reliability systems, a second hierarchical scheme is proposed [34]. Here one spare node is hooked up to each subcube of four nodes via a high speed bus. The scheme is applied recursively. For example, a spare group of five nodes (one spare and four active) is used to back up four groups of five nodes each via a high speed bus. This multilevel redundancy provides high reliability. Rennels did not evaluate the overall system reliability or the subcube reliability of his schemes.

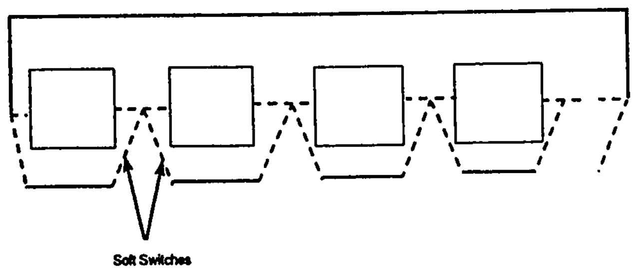


Figure 3.1: A fault-tolerant module with $k$ spares.

Further Link failures are not tolerated by his schemes.

### 3.1.2 Chau et al's Scheme

Chau *et al* [12] suggest another fault-tolerant hypercube architecture using similar fault-tolerant modules as Rennels's but using decoupling networks and soft switches instead of crossbar switches. Here a fault-tolerant module consists of $m$ active nodes and $k$ spare nodes. When $m = 4$ and with k spares the four active nodes are connected in a cycle to model a 2-cube. Since only four nodes are active at any given time in a fault-tolerant module, the spare and faulty nodes have to be bypassed. This can be done using soft switches as shown in Figure 3.1. Such 2-cubes are connected together to form a $n$-cube. The connections between the 2-cubes are realized by decoupling networks such that only active nodes in a module are connected. Decoupling networks have been previously used in fault-tolerant binary tree architectures [41]. A group of $k$ level decoupling networks as shown in Figure 3.2 is used to connect one fault-tolerant module to another. When none of

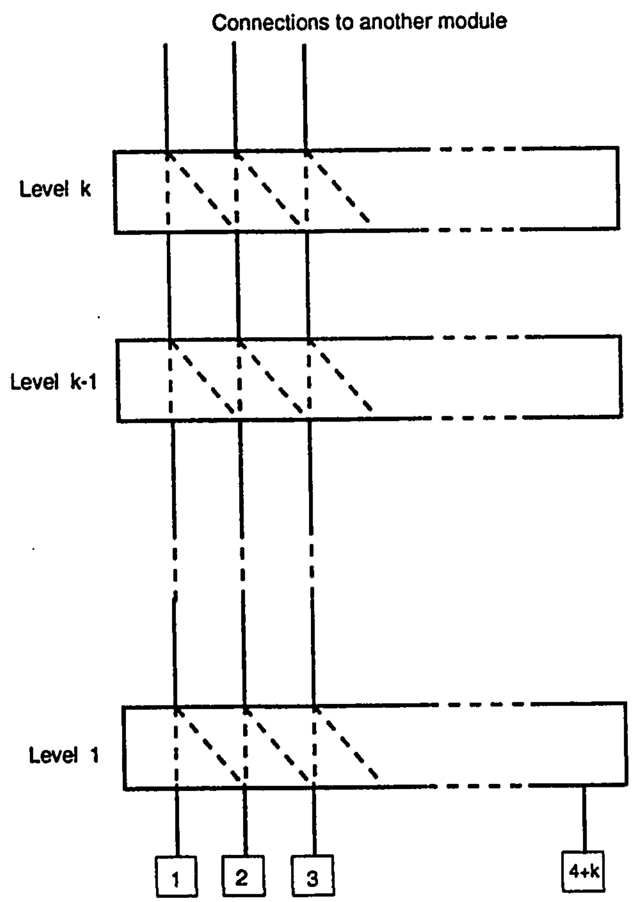


Figure 3.2: Connecting one fault-tolerant module to another using decoupling networks.

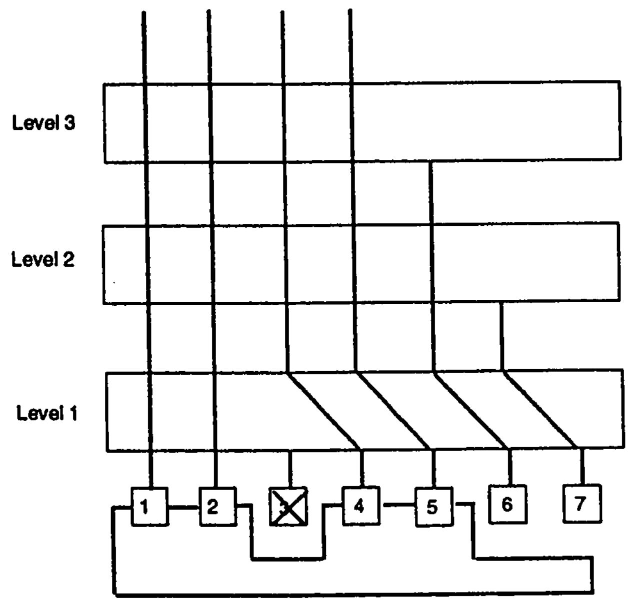


Figure 3.3: Connections after node 3 has become faulty.

the nodes have failed, dotted connections shown in Figure 3.2 are not used. When a node fails the fault-tolerant module is reconfigured by the soft switches so as to bypass the failed node and replace it with a spare node. Now connections between nodes of one module and another are established as follows. Given a module with $k$ spare nodes, let $i-1$ be the number of nodes that have failed and replaced at some fixed time during the operation of the system. When another active node fails, the level $i$ decoupling network is switched. The $i$th level of the decoupling network is reconfigured by switching the link that connects to the failed node and all links to the right of it, one position to the right. For example, with three spares we show the connections in Figure 3.3 after node 3 has failed. If the fault-tolerant module is a $m$-cube ($m > 2$) with $k$-spare nodes, connections among the nodes can be done using decoupling networks instead of soft switches. For a $m$-cube fault-tolerant module with $k$ spares we need $m$ groups of $k$-level decoupling networks to establish the connections among the nodes in a module [12]. Chau *et al* generalize this scheme when $m = n$ i.e., global sparing and the entire network is a single fault-tolerant module. The system reliability of the local sparing scheme for an $n$-cube with fault-tolerant modules consisting of a $m$-cube and $k$ spares is given by

$$RS_{n,m,k} = (RM_{m,k})^{2^{n-m}}$$

where,

$$RM_{m,k} = RM_{m,k-1} + \binom{2^m + k - 1}{k} r^{2^m} (1-r)^k c^k$$

and $c$ is the fault coverage factor. Fault coverage is defined as the probability of successful fault detection and reconfiguration. The system reliability of their global sparing scheme is given by

$$RS_{n,n,k} = RS_{n,n,k-1} + \binom{2^n + k - 1}{k} r^{2^n} (1-r)^k c^k$$

where $RS_{n,n,0} = r^{2^n}$ for an $n$-cube with $k$ spares [12]. This global sparing scheme requires far less number of spares than Rennels's basic scheme to achieve the same level of system reliability. Further Chau *et al*'s scheme requires less amount of hardware than Rennels's hierarchical scheme to achieve the same level of reliability. Comparing Chau *et al*'s modular sparing and global sparing techniques it is shown in [12] that global sparing is preferable when $n \leq 8$ and modular sparing is better when $n > 8$ in terms of the hardware overhead and reliability. Further, modular sparing has the following advantages [42],

- Local and fast fault detection and reconfiguration.
- Ease of construction, scalability and module replacement and
- Simple fault-tolerant routing algorithms

The drawback of Chau *et al*'s scheme is that it takes longer to reconfigure, since on average the states of half the nodes have to be shifted to neighboring nodes. Furthermore, link failures have not been taken into consideration in the derivation of the system reliability. Link failures are not tolerated by their schemes. Only the system reliability is evaluated and subcube reliability of the given schemes is not analyzed.

### 3.1.3 Sultan and Melhem's Scheme

Sultan and Melhem introduce two schemes for reconfiguration in fault-tolerant modules: one using hardware switches-multiplexers & demultiplexers and the other one using fault-tolerant routing [42]. Their scheme using hardware switches does not have the overhead of shifting processor states. This leads to reduction in reconfiguration time and overhead significantly. The scheme using hardware switches works as follows. Consider an FTBB with $M$ primary nodes $P_1, \cdots, P_M$ and $K$ spare nodes, $S_1, \cdots, S_K$ using full spare utilization i.e., any spare can replace any primary

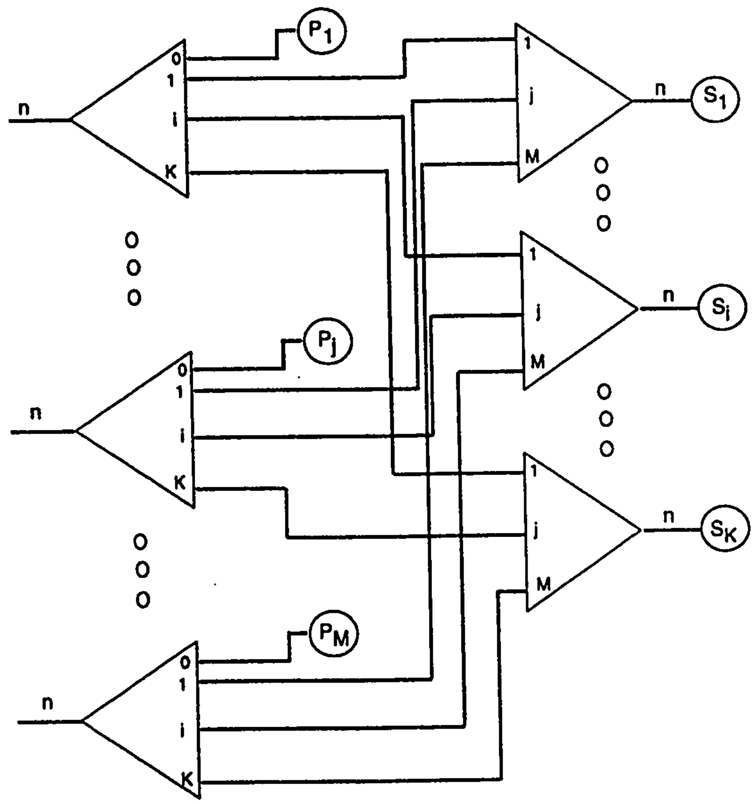


Figure 3.4: Switching logic within an FTBB.

node within an FTBB. The switching logic for reconfiguration is in terms of multiplexers and demultiplexers as shown in Figure 3.4. In this figure it is assumed for simplicity that multiplexers and demultiplexers can multiplex and demultiplex duplex lines respectively. Specifically, a $1-to-(K+1)$ demultiplexer is used for each $P_j$ to divert when needed the links of $P_j$ to any $S_i$. Also an $M-to-1$ multiplexer is used for each $S_i$ to connect it to the appropriate links. If a primary node $P_j$ is nonfaulty then the demultiplexer is set to select the $0th$ output. In case $P_j$ is faulty, the replacement of $P_j$ by $S_i$ requires that the demultiplexer associated with $P_j$ be set to its *ith* output and the multiplexer associated with $S_i$ be set to select its *jth* input. This scheme can also support spare failures.

The second technique uses a general two phase routing algorithm. Here the logical topology of the hypercube is preserved by modifying the routing algorithm rather than preserving the physical topology in the presence of failures. A node failure in this scheme is assumed to be the failure of the processor, the router and the links of the node. If a node $P$ fails then the spare node that replaces $P$ inherits the address of $P$. The routing algorithm is distributed and requires local fault knowledge, in the sense that only the neighbors of the failed nodes need to know about the fault in order to take corrective action. Sultan and Melhem calculated only the system reliability of their schemes and subcube reliability is not investigated. Link failures are also not considered.

### 3.1.4 Banerjee and Peercy's Scheme

Banerjee and Peercy consider two fault-tolerant hypercube schemes [7]. The first scheme called the *node spare scheme* uses spare nodes attached to specific nodes in the cube using a novel embedding technique. The hypercube in this scheme consists of two types of nodes called $P$ nodes and $S$ nodes. The $P$ and $S$ nodes have different internal architectures. Figure 3.5(a) shows the internal details of a normal processing node (called a $P$ node) of a hypercube. The $P$ node consists of a computation processor (CPU) connected through an internal bus to a local memory

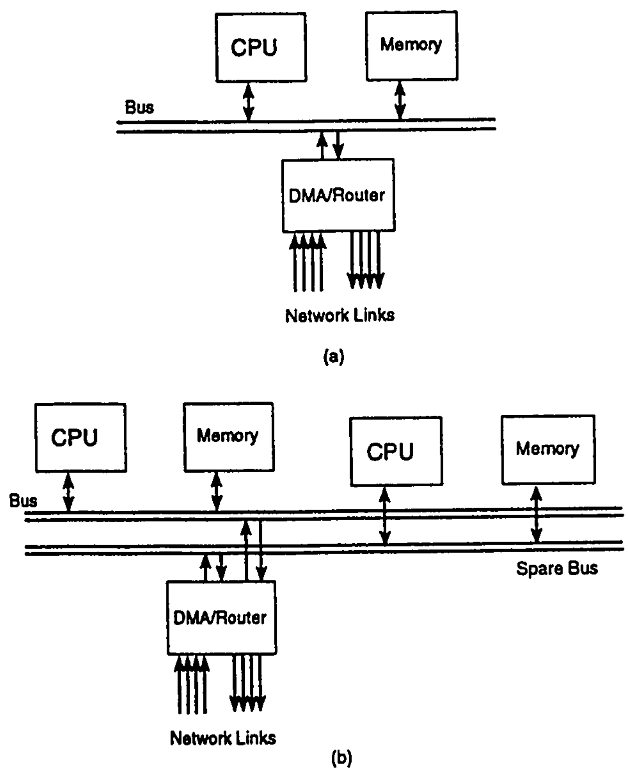


Figure 3.5: Architecture of two types of nodes. (a) $P$-node architecture (b) $S$-node architecture.

and message routing logic consisting of the DMA unit and a $(d+1) \times (d+1)$ crossbar switch for a $2^d$ processor hypercube. Figure 3.5(b) shows an $S$-node consisting of two copies of the CPU and local memory connected to two internal busses. The DMA and message routing logic is shared between the two processing units, one of which is active under normal conditions; the other is a standby spare. Under failure of any processing element either within the $S$ node or in a nearby $P$ node, the spare processor/memory/bus from the corresponding $S$-node is brought on line.

From trace driven studies of a large number of parallel CAD and numeric applications on the hypercube it has been determined that typical applications running on a node of the hypercube only use the DMA/router less than 10% of the time [7, 6, 33]. The router is actually used less than 25% of the time by other parallel data transfers between channels not involving the node. Hence a single DMA unit could be shared by both the CPU units.

A perfect embedding is considering for allocation of $S$ nodes in the cube i.e., each $P$ node is adjacent to exactly one $S$ node in the cube. A perfect embedding in a 3 cube is shown in Figure 3.6. An algorithm for reconfiguration in the presence of primary node failures is given in [7]. The second scheme called the *link spare scheme* introduces spare nodes ($L$) into the hypercube by inserting them along links connecting particular pairs of processors. The hardware of the node is not modified here. Figure 3.7 shows an embedding of spares in a 16 node hypercube. Under normal operation, nodes 0 and 8 are connected to each other through the spare

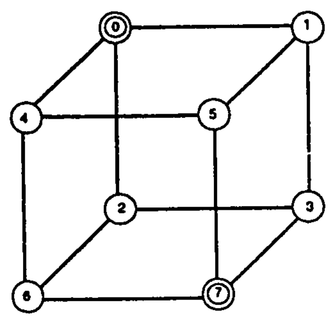


Figure 3.6: A perfect embedding in a 3-cube.

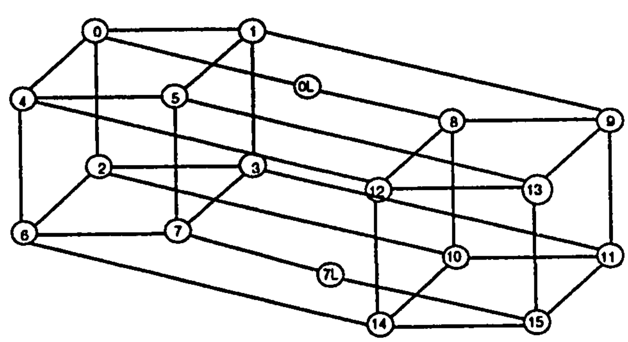


Figure 3.7: Spares embedded in a four dimensional cube.

node $0L$ whose connections are permanently set to minimize the overhead of setting the connections. When node $0L$ fails its task can be taken by the spare node $0L$. All messages that were originally routed to node 0 will now be routed to $0L$ via node 0 or node 8. In this example, the logical link $(4, 0)$ gets mapped to a path of length 2 $(4 - 0 - 0L)$ in the case of partial failures and a path of length 3 $(4 - 12 - 8 - 0L)$ in the case of total failure of node 0. The algorithm for allocation of $L$ nodes in the hypercube and reconfiguration is given in [7].

### 3.1.5 Yang et al's Scheme

Yang et al propose another fault-tolerant scheme consisting of fault-tolerant modules (FTMs) [47]. The FTMs are then interconnected with additional links called the spare-sharing links (SSLs) in the modular hypercube. The main characteristic of this scheme is that spare nodes within a FTM can be used as local spares , or as remote spares to replace faulty nodes in other FTMs. The structure of each FTM is built by $2^m$ active nodes, $p$ local spare nodes, an internal switch connection (ISC), $(n - m)$ external switch connections (ESCs), $n.p$ input SSLs and $n.p$ output SSLs. as shown in Figure 3.8. In this figure the ISC is used to connect the first $2^m$ fault-free nodes from a total of $2^m + p$ nodes to form the $m$ subcube topology and the $(n - m)$ ESCs are used to connect this FTM with other FTMs. The $n.p$ input SSLs will be used by this FTM to receive at most $p$ remote spare nodes from other FTMs and the $n.p$ output SSLs will used to send at most $p$ idle spare nodes to other

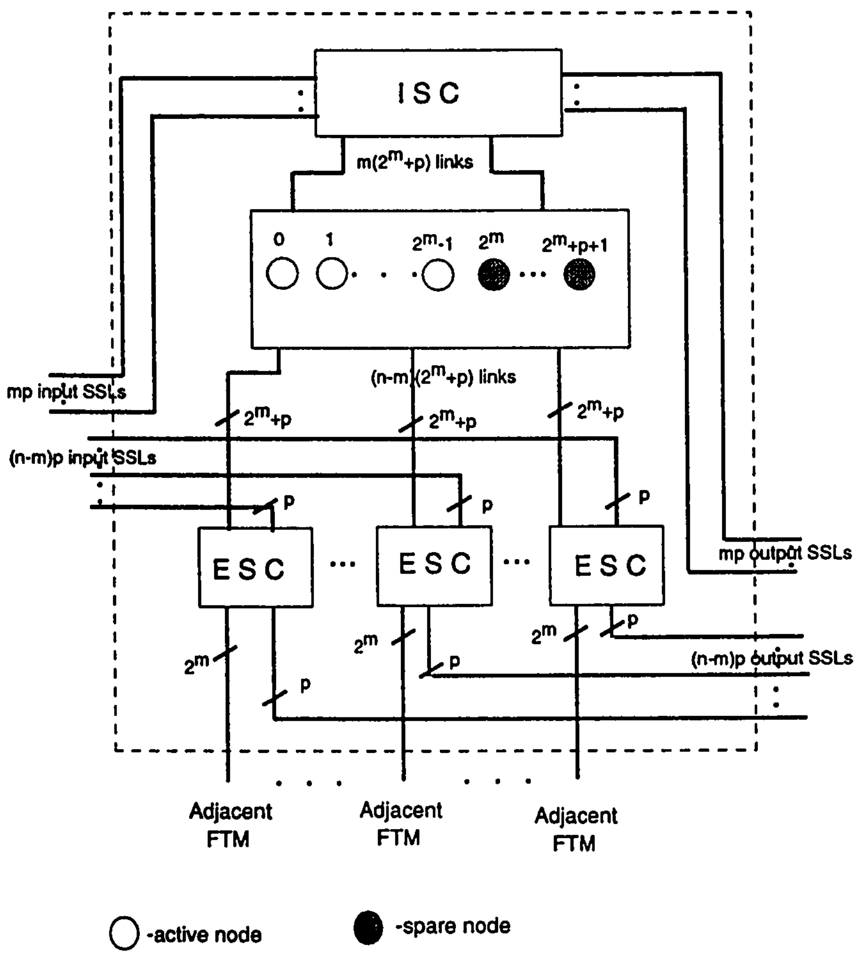


Figure 3.8: The structure of the FTM.

FTMs. A reconfiguration algorithm to tolerate node and link failures within the FTM is given in [47]. This scheme achieves better system reliability than earlier schemes and tolerates both node and link failures. But it uses hardware switches. The reconfiguration strategy is complex. Subcube reliability of this scheme is not investigated.

## 3.2 Existing Hierarchical Hypercubes

This Section presents fault-tolerant schemes in hierarchical hypercube networks. Hypercubes present a good tradeoff between the number of links required (cost) and the worst case message delay when a moderate to large number of processors (around $10^2$ to $10^3$) are to be connected. For connecting very large number of processors, the link cost of a hypercube becomes prohibitively large and hierarchical interconnection network (HINs) schemes are to be used [16]. There are two main motivations for HINs:

1. to reduce link cost, and

2. to provide a framework for integrating several network topologies.

A homogeneous two level Binary Hypercube/Binary Hypercube (BH/BH) HIN is shown in Figure 3.9. Each level 1 network is called a cluster. A cluster size of 8 and two levels have been found to be most optimal under a cost/performance tradeoff for connecting very large number of processors in a BH/BH network. Although we can

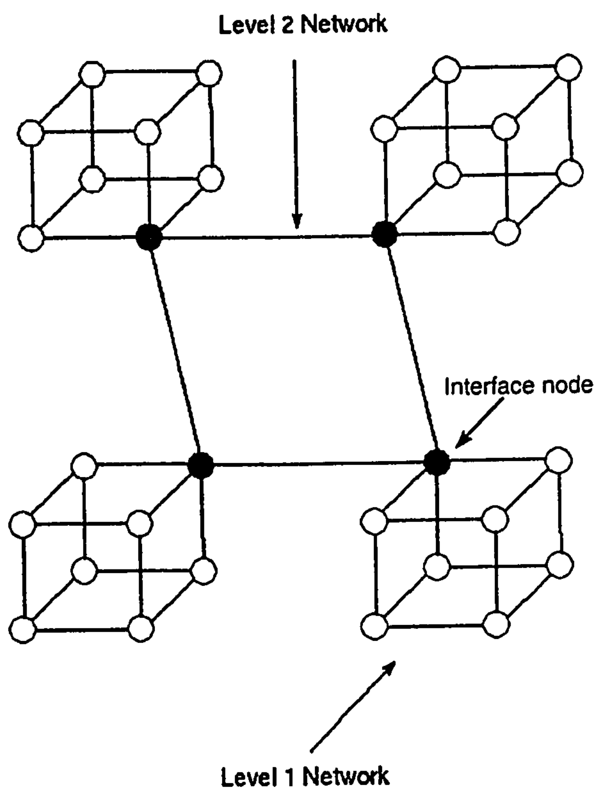


Figure 3.9: A BH/BH HIN with a cluster size of 8.

reduce the number of links, the HIN does not seem to possess good fault tolerance characteristics. Major disadvantages of HINs include the potential for high traffic rates on inter-cluster links, and thus the potential degradation in performance, and the potential for diminished fault tolerance due to the special role played by the interface nodes. However standard fault tolerance techniques can be applied to the interface nodes to improve the reliability of the BH/BH network close to that of a BH network. We shall elaborate on two techniques using hardware redundancy to improve the fault tolerance of the BH/BH network suggested in [16]. In one technique, the level two network is duplicated. In the other technique, a standby spare interface node is provided to reduce the impact of interface node failures on reliability. The two techniques are described below:

1. Replication Technique: Here each cluster uses two nodes as interface nodes. The level 2 network is duplicated for each interface node. This network referred to as BH/BH-RS and is shown in Figure 3.10. It is preferred to keep the two interface nodes in a cluster as far apart as possible.

2. Standby Spare Interface node Technique: Since the interface nodes are the most vulnerable, a standby spare for each interface node in the network is used. One node is normally operational and the other serves as a spare. If a failure of the interface node occurs, the spare takes over. This technique is referred to as BH/BH-SI. We see that the BH/BH-RS technique uses more

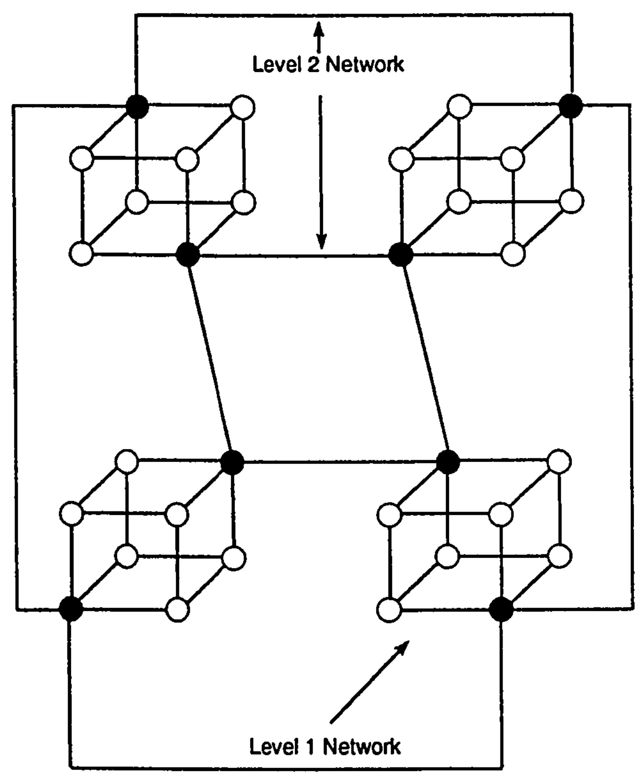


Figure 3.10: A BH/BH-RS network with a cluster size of 8.

links than the BH/BH-SI but the BH/BH-SI requires fault detection of the interface node and extra spare nodes. We shall see later that the performance of the BH/BH-RS is substantially better justifying the increased link cost.

We present simulation results for the task based reliability of fault-tolerant schemes hierarchical hypercubes below from [16]. With task based reliability, a system is considered to be working as long as some minimum number '$I$' of nodes are fault-free and available for task execution. The simulation model considered is as follows. The nodes and links are assumed to follow the exponential failure law with failure rates $\lambda_n$ and $\lambda_l$ respectively. Therefore, the node and link failure rates for the whole network are $N_a\lambda_n$ and $L_a\lambda_n$ respectively where $N_a$ is the number of active nodes and $L_a$ is the number of active links in the network. The given network is represented by its adjacency matrix. The adjacency matrix of an interconnection network with $N$ nodes is a $N \times N$ symmetric binary matrix $A = [a_{ij}]$ such that $a_{ij} = 1$ if there is a link between the $i$th and $j$th nodes, otherwise $a_{ij} = 0$. A fault-free node is represented by making the diagonal element $a_{ii} = 1$. The network connectivity is shown by using the reachability matrix. Two nodes $i$ and $j$ are considered reachable if there is a path between them. The reachability matrix can be computed from the adjacency matrix by standard algorithms. The adjacency matrix is modified every time a node or a link failure occurs. The failure of the link connecting nodes $i$ and $j$ is represented by making both $a_{ij}$ and $a_{ji}$ 0s. The failure of a node $i$ is represented by making all entries in the $i$th row and $i$th column 0s. This modification of the adjacency matrix is reflected in the reachability matrix. The reachability matrix then determines whether or not there is connectivity among at least $I$ of the $N$ nodes in the network. We also assume that a single repair facility is available for both the nodes and links with the repair rates of the nodes and links $\mu_n$, $\mu_l$ respectively. The failed component is put in a FIFO queue waiting for the repair facility. When the component is repaired the corresponding entries in the adjacency matrix are updated.

Reliability results are now given for the simulation model presented above [16]. The node failure and link failure rates used are $100/10^6$ hours and $20/10^6$ hours respectively. Figure 3.11 shows the reliability results with cluster sizes of 8 and 4. For comparison purposes the reliabilities of the bidirectional ring (BR) and complete connection (CC) networks are also included. It can be seen from this figure that the BH network reliabilities are as good as a complete connection network. The BR network is at the other extreme. The BH/BH network provides reliabilities in between these two extremes. The impact of the two fault-tolerant schemes BH/BH-SI and BH/BH-RS are also shown. Both these networks provide similar reliabilities . We see that the reliabilities are better for the smaller cluster size. This is mainly due to the fact that the penalty associated with an interface node failure decreases as the cluster size decreases. A weakness of both the BH/BH-SI and BH/BH-RS networks is that their reliability is uniformly poorer than the BH network. This drawback can be remedied by using both the schemes together called BH/BH-SI&

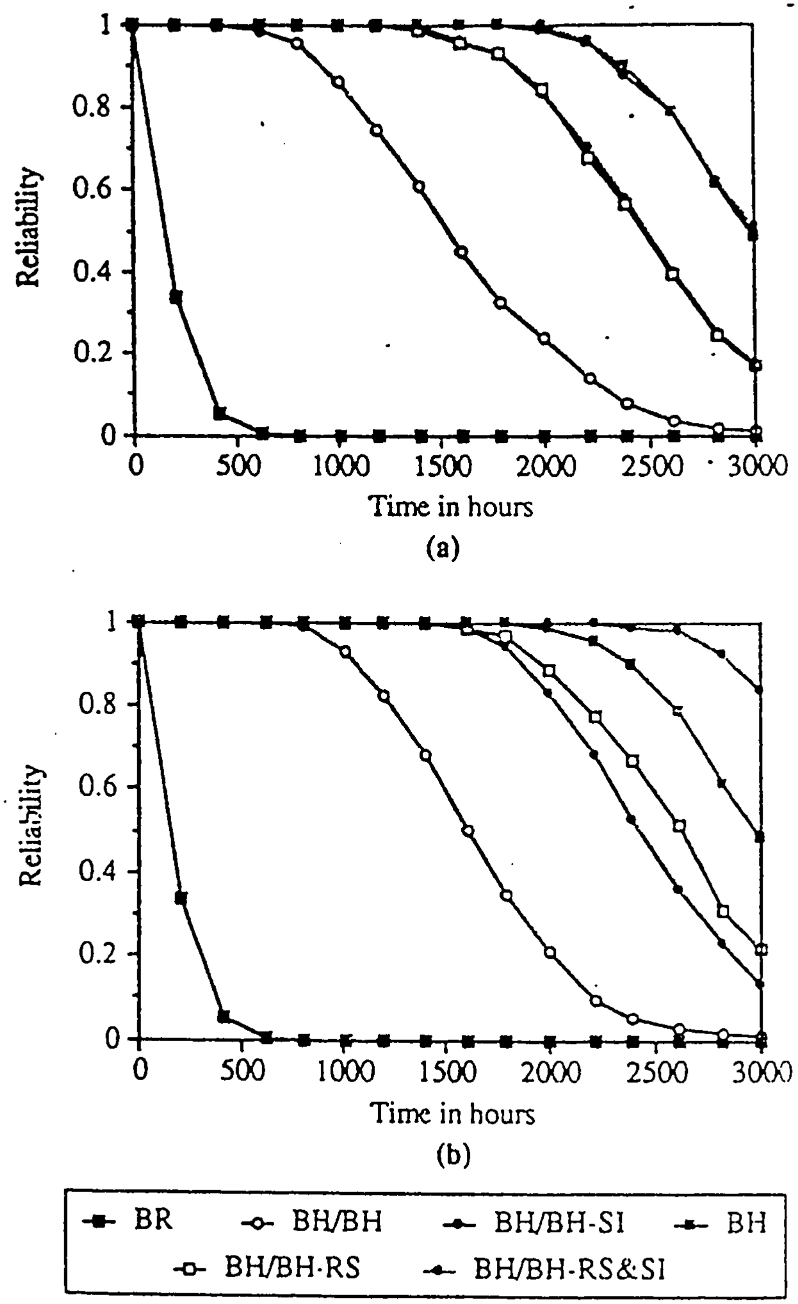


Figure 3.11: Reliability results of various interconnection networks (N=128, I=96) (a) cluster size = 8 (b) cluster size = 4.

RS scheme and its results are also shown in the Figure 3.11.

We are going to use the fault-tolerant basic blocks for level one clusters instead of the spare interface node scheme . This architecture is expected to perform better in terms of reliability than the schemes given in [16] since now the spare node can replace any node in the cluster rather than just the interface node.

# Chapter 4

# The Proposed FTBB Based Architecture

In this chapter we introduce the proposed FTBB based architecture. In Section 4.1 of this Chapter we present the new fault-tolerant hypercube architecture utilizing fault-tolerant basic blocks (FTBB). Section 4.2 presents reconfiguration strategy used by our architecture to tolerate node and/or link failures. Section 4.3 presents the methodology used by us to analyse the subcube reliability of the proposed fault-tolerant hypercube architecture.

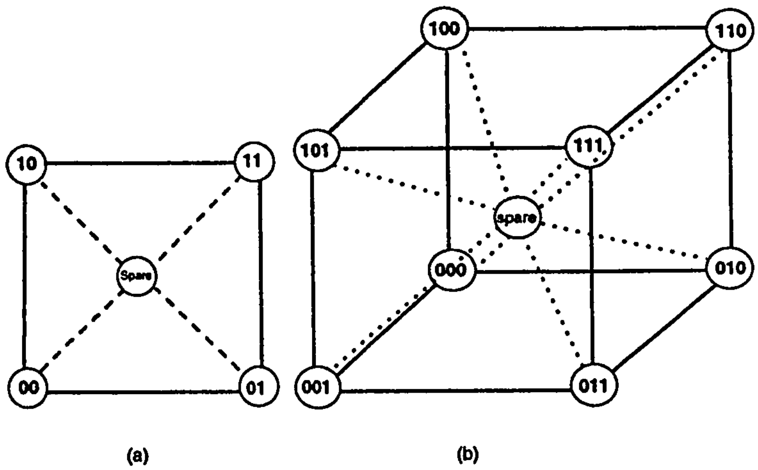


Figure 4.1: (a)A FTBB of order 2. (b)A FTBB of order 3.

## 4.1 Architecture

In this section we shall describe the proposed fault-tolerant hypercube architecture. The architecture of our fault-tolerant module is similar to the modular scheme in [34] in that it contains a spare node. It however is different in that no hardware switches are used for reconfiguration. We define a FTBB of order $m$ or a $m$-FTBB to be a basic building block consisting of a $m$-cube and a spare node connected to all the primary nodes (nodes of the $m$-cube) with $2^m$ spare links. A FTBB of order 2 and a FTBB of order 3 are shown in Figure 4.1a and Figure 4.1b respectively. A 3-FTBB consists of a 3-cube with a spare node connected to all the nodes of the 3-cube by $2^3$ spare links. Each node contains an extra port to connect to the spare. In real systems using hypercube topology, the processors are manufactured with the maximum allowable links. Very often not all these links are used in the

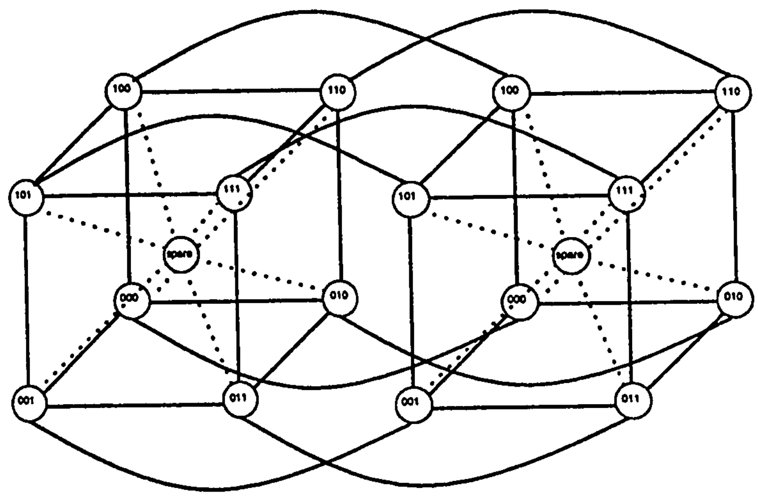


Figure 4.2: A 4-cube built of 3-FTBBs.

regular structure of the system [45]. This gives the motivation of utilizing the extra connections of the nodes for spare links. Therefore the proposal of having extra spare links is a very practical one because these links are equipped as the processors are fabricated, would be unused otherwise, and the implementation does not pose difficulty [45]. Such cubes with extra links are called enhanced hypercubes and they achieve noticeable improvements in many measurements such as mean internode distance, diameter and traffic density compared to regular hypercubes [43].

Larger hypercubes can be built using FTBBs by utilizing the recursive construction property of the hypercube. For example, to get a 4-cube, we take two FTBBs of order three and join the corresponding nodes in the two FTBBs with links (see Figure 4.2). A 5-cube would be obtained by taking the two previously constructed 4-cubes and joining the corresponding nodes. Higher order hypercubes could be obtained in a similar manner. In general a $d$-cube can be constructed of $2^b$ fault-tolerant basic blocks each of dimension $m = d - b$. It should be noted that the spare nodes are not connected directly to each other in this architecture. We refer to such a cube as an *augmented* $b$-cube i.e., constructed of FTBBs to distinguish from a $d$-cube which is the ordinary hypercube consisting of $2^d$ nodes. A $b$-cube consists of $2^b$ basic blocks. If each basic block in a $b$-cube is of dimension $m$ then it has $2^d$ nodes where $d = b + m$. For example, if a $b$-cube is constructed of 3-FTBBs then $d = b + 3$.

The spare nodes are no different than the primary nodes in a FTBB in terms of their hardware structure. Spare nodes within a $m$ FTBB will have $m$ links attached to each of them. For any augmented $d$ cube built of $m$ FTBBs if $d \geq 2^m$ than the spare nodes have same or lesser links attached to them than the primary nodes. For cubes built of 2 & 3 FTBBs and for practical cubes ($d > 8$) the spare nodes will have lesser links attached to them than the primary nodes. Therefore the spare node can be identical to the primary node except that its extra link connections will be unused.

## 4.2 Reconfiguration Strategy

In this Section we shall present the reconfiguration strategy used by the FTBB to tolerate node and link failures. Consider the 3-FTBB and consider the failure of any primary node. The failure can be tolerated in the following manner. Whenever a primary node fails, the failure is detected in the FTBB by some diagnostic strategy [45]. Low cost schemes for achieving such diagnosis and recovery are presented in [29]. The spare node takes over the functions of the failed node. Nodes adjacent (directly connected) to the failed node can now communicate with the replaced spare via spare links connecting them to it. The physical topology of a 3-cube is still maintained. For example, if node 4 fails (see Figure 4.3(a)) the spare node takes over the functions of node 4. Nodes $0, 5$ and 6 which were adjacent to the failed node in the FTBB now start communicating with the spare node which inherits the address of node 4. In effect links $(0, S), (5, S)$ and $(6, S)$ replace links $(0, 4), (5, 4)$ and $(6, 4)$ respectively. $S$ represents the spare node. Effectively a spare node and three spare links replace a failed node. The links connected to a failed node are deemed to be failed. This is because of the failure model for a node we are adopting here. We have assumed that when a node fails, it loses both its computational and communication capabilities. Another model could assume that a node can fail on computation but its communication capability is intact. Under this assumption links connected to the failed node could be utilized. We assume throughout that a node failure means the failure of both its computation and communication capabilities.

Now consider the case where the 3-FTBB were part of a larger cube and node 4 had failed. The spare within the FTBB would have replaced it. But node 4 was adjacent to some other nodes in addition to 3-nodes within its FTBB. The spare

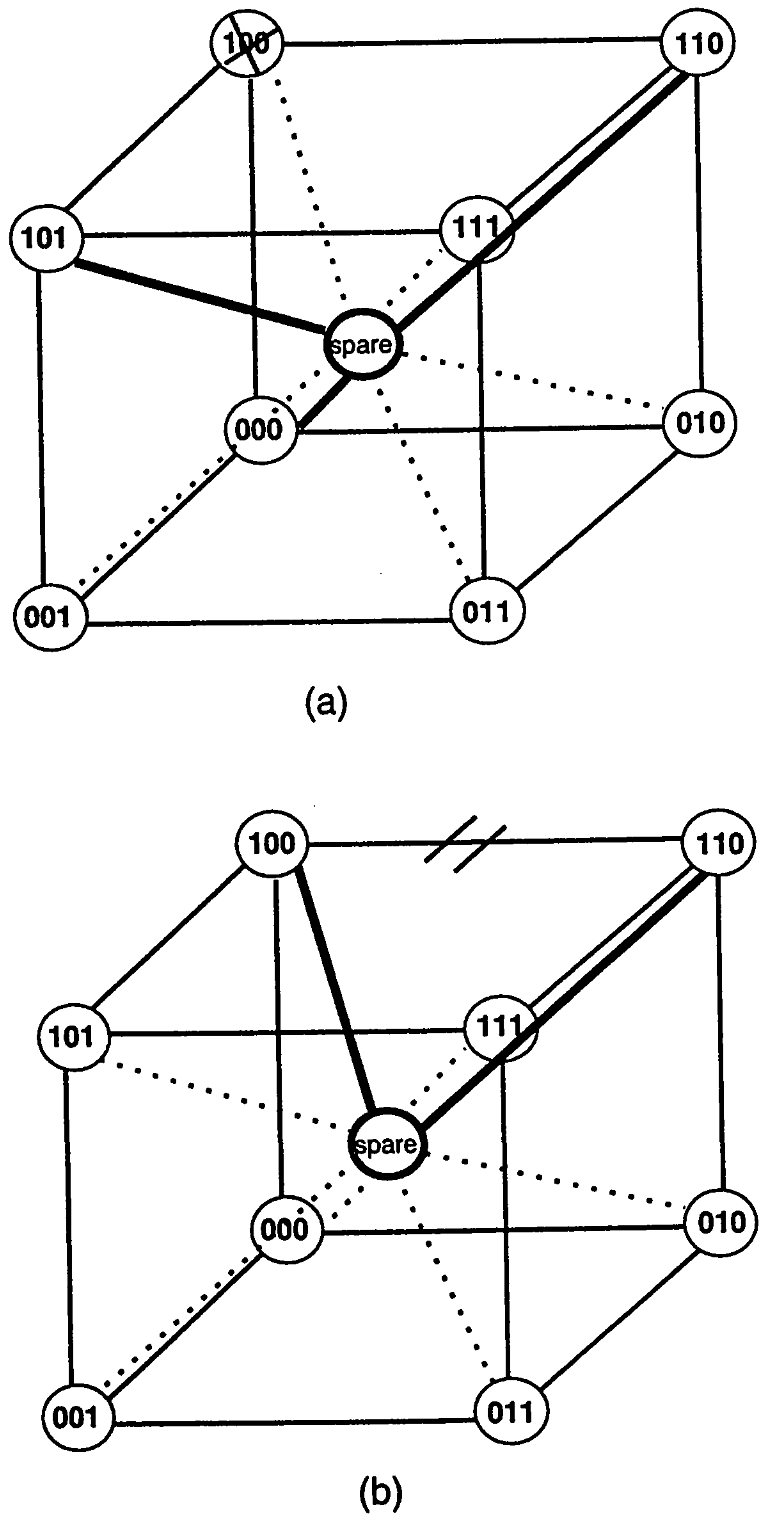


Figure 4.3: (a)A single node (4) has failed (b)A single primary link has failed.

which has replaced node 4 has direct links only to nodes which were adjacent within the FTBB and does not have direct links to other adjacent nodes of node 4. This is resolved in the following manner. Before its failure node 4 was either computing or sending messages or receiving messages from other nodes. Now the replaced spare performs the computation. The messages which node 4 would have sent via the adjacent nodes (other than the adjacent nodes within its FTBB) will now be sent to either node $0, 5$ and 6. All messages destined to node 4 will now be received by either node $0, 5$ and 6 and passed on to the replaced spare node. Therefore the routing algorithm should be suitably altered depending on the list of failed nodes in the system. Only nodes adjacent to the failed node need to know about its failure. More than one node failure cannot be tolerated, within a FTBB as a single spare node is present.

Consider now the failure of any primary link within a FTBB. When a primary link fails, communication between the nodes connected by the failed link is disrupted. This can be tolerated in the following manner. The nodes connected by a failed link establish connection between themselves by using the spare links connecting them to the spare node. The spare node relays messages between the two nodes, acting effectively as a short circuiting element between these nodes. For example, if link $(4, 6)$ fails in (see Figure 4.3(b)), links $(4, S)$ and $(5, S)$ replace link $(4, 6)$. Note that the spare does not take part as a computing element but only as a communicating element. But now nodes 4 and 6 have an extra link delay between them i.e., the

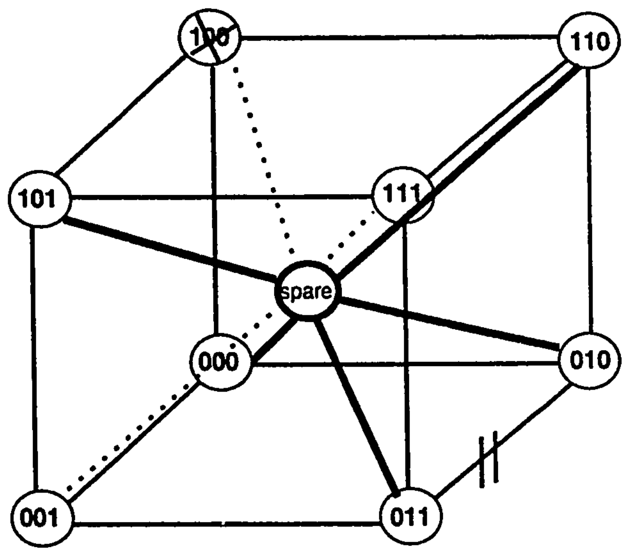


Figure 4.4: Both a primary node and link have failed.

logical topology of a 3-cube is still maintained though the physical topology is not. More than one link failure can also be tolerated. Even if all the primary links fail, they can still be tolerated by using the spare node and all the spare links. In this case the physical topology will be equivalent to a star network. In general only combinations of link failures which have both a spare and a primary link connected to the same node cannot be tolerated. All other multiple link failures can be tolerated.

In addition combinations of certain node and link failures could be tolerated. For example the combined failure of node 4 (see Figure 4.4) and any link or links other than $(0, S), (5, S)$ and $(6, S)$ can be tolerated. All combinations of node and their direct link failures can be tolerated. All combinations of node and link failures in which any one or more links in the combination is a link which is supposed to replace the failed node in the combination, cannot be tolerated.

The various combinations of node, link and combined node and link failures that can be tolerated within a FTBB are captured in the reconfiguration strategy presented in Figure 4.5 in the form of rules ($0 \leq i, j < 2^m$).

The reconfiguration strategy is divided into two steps. Step 1 includes the case where the spare node has failed. Step 2 includes the case where the spare node within the FTBB is working. In step 1 where the spare node has failed no primary node/link failures can be tolerated as the spare node is critical to tolerate any primary node/link failures. Step 2 is divided into three cases: a. All primary nodes functioning. b. One primary node has failed and 3. More than one primary node has failed. In the reconfiguration strategy we call a spare link which replaces a failed link or node as a switched spare link. We only permit spare links replacing primary node/link failures. No spare link is permitted to replace another failed spare link.

When none of the nodes have failed the spare node is functional and can enhance the performance by taking part in the computation within the FTBB. It can serve to reduce the maximum distance between the nodes of an $m$ FTBB from $m$ to 2. This will reduce the average internode distance for the whole network.

It has to be noted that if the spare node fails before the failure of any component in the FTBB then no further failures of primary nodes or links can be tolerated. So it would be advantageous if the spare node is made more reliable than the primary node. We have made a realistic assumption above, that the primary nodes and spare

**Reconfiguration Strategy;**

**Begin**

1. **If** the spare node $s$ has failed **then;**
   No primary node/link failures can be tolerated.
   FTBB continues functioning.
   a. **If** a spare link $(i, s)$, for any $i$ fails **then;**
      FTBB continues functioning.
   b. **If** any primary node $i$ or link $(i, j)$, for any $i, j$ fails **then;**
      FTBB unreconfigurable.
      **Else**
      FTBB continues functioning.
2. **If** the spare node $s$ is working **then;**
   a. **If** all primary nodes are working **then;**
      i. **If** any primary link $(i, j)$ for any $i, j$ fails **then;**
         Replace it with spare links $(i, S)$ and $(j, S)$
         FTBB continues functioning.
      ii. **If** a switched in spare link $(i, S)$, for any $i$ fails **then;**
         FTBB unreconfigurable.
      iii. **If** a non-switched spare link $(i, S)$ and a
         primary link $(i, j)$ for any $i, j$ fail **then;**
         FTBB unreconfigurable.
         **Else**
         FTBB continues functioning.
   b. **If** any single primary node $i$ fails **then;**
      Replace it with the spare node
      Replace primary link $(i, j)$ for all $j$ with link $(j, S)$
      i. **If** any primary link $(i, j)$ for any $i, j$ fails **then;**
         Replace it with spare links $(i, S)$ and $(j, S)$
         FTBB continues functioning.
      ii. **If** a switched in spare link $(i, S)$, for any $i$ fails **then;**
         FTBB unreconfigurable.
      iii. **If** a non-switched spare link $(i, S)$ and a
         primary link $(i, j)$ for any $i, j$ fail **then;**
         FTBB unreconfigurable.
         **Else**
         FTBB continues functioning.
   c. **If** more than one primary node fails **then;**
      FTBB unreconfigurable

**End.**

Figure 4.5: Reconfiguration strategy for any $m$-FTBB.

nodes are identical in their hardware architecture and this assumption is maintained throughout our analysis. A different sort of hardware architecture is considered for the spare nodes to make them more reliable in [7] and explained in Section 3.1.4. The idea is to adopt the $S$ node architecture for the spare nodes within our FTBB. In the $S$ nodes the CPU and memory section of the node is duplicated and so will be more reliable (will have lesser failure rate) than ordinary nodes. We can estimate the failure rate of the $S$ nodes given the failure rate of ordinary nodes ($\lambda_n$) as explained below. We assume that the each CPU and memory section has a failure rate of a ordinary node-$\lambda_n$. Assume that the DMA section of the node is perfect. Then the $S$ node will function correctly if one of the pairs of the CPU and memory section is working. It is equivalent to a parallel system of two modules where one module has to work correctly for the system to function correctly. Given that the failure rate of the CPU and the memory section is $\lambda_n$ we can easily calculate the failure rate of the $S$ node to be $2\lambda_n/3$.

Throughout our analysis we assume that the spare node is identical to the primary node i.e., they have the same failure rate. We shall mention it specifically when we relax this assumption.

## 4.3 Methodology

In this Section we shall describe the methodology we adopt to analyse the reliability of the FTBBs and the cubes built of FTBBs (augmented cubes). We have seen that the FTBB can tolerate a variety of node/link failure conditions. Therefore it is expected that the FTBB will have better reliability than the corresponding ordinary basic cube, cubes built of FTBBs are expected to be more reliable than corresponding cubes built of ordinary basic blocks.

The foremost objective of our proposed work is to analyse and evaluate the subcube reliability of the proposed fault-tolerant hypercube built of FTBBs for the four failure models listed below.

1. Node Failure Model - where only node failures are assumed to occur.

2. Link Failure Model - where only link failures are assumed to occur.

3. Combined Node and Link Failure Model - where both node and link failures are assumed to occur.

4. Supernode Failure Model - which is an approximation to the combined node and link failure model

We consider the possibility of using two basic building blocks either the 2-FTBBs or the 3-FTBBs to build larger cubes. To achieve this objective we have done the following. We first analyse the system reliabilities of the 2-FTBB and the 3-

FTBB by using the definition of a functioning FTBB given in the reconfiguration rules in Section 4.2 for each of the failure models listed above. By this analysis we obtain MTTF $T_*$ for 2 and 3 FTBBs for all the failure models. By extending the results given in [1] we obtain MTTF $T_*$ for 2-cube and 3-cubes and compare the results with the corresponding FTBBs. This shall give us an idea of how good the FTBB is in terms of reliability over the corresponding ordinary cubes. By using the failure rates of the 2 or 3-FTBB and the corresponding ordinary cubes (reciprocal of their $T_*$ values) , we can build markovian models for the cubes constructed from FTBBs or corresponding ordinary basic blocks for each of the failure models listed above extending the subcube reliability analysis given for the ordinary hypercube in chapter 2 [1]. Again we can compare the subcube reliabilities and the various MTTFs of the cubes built of 3 or 4-FTBB with the corresponding values for cubes built of ordinary basic blocks.

As a secondary objective, we investigate the possibility of using the 3-FTBB or 2-FTBBs for the level 1 clusters instead of the BH/BH-SI scheme. We denote this scheme as BH/BH-FTBB. We build analytical models to analyse the system reliability of hierarchical hypercubes such as BH/BH, BH/BH-SI, BH/BH-RS, BH/BH-SI & RS and BH/BH-FTBB.

# Chapter 5

# Reliability of Fault-Tolerant Basic Blocks

In this Chapter we present the reliability and MTTF analysis of 2 and 3 FTBBs for node, link, combined node & link and the supernode failure models. We shall compare the reliability and MTTF of the 2 and 3 FTBBs with those of ordinary 2 and 3 basic blocks respectively for each of the failure models.

## 5.1 Node Failure Model

In this section we shall develop a state transition model for the FTBBs assuming only node failures can occur and derive expressions for reliability and MTTF. We then compare the reliability and MTTF figures of the 2 and 3 FTBBs with corresponding

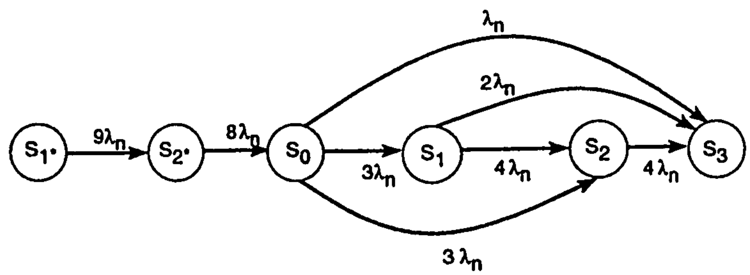


Figure 5.1: System state diagram for 3 FTBB under node failure model.

values of ordinary 2 and 3 cubes respectively.

Consider the 3-FTBB shown in Figure 4.1(b). The first node to fail is replaced by the spare node. We assume here perfect fault coverage i.e., the probability that a failure is detected and appropriately reconfigured is unity. The state transition diagram is shown in Figure 5.1. $S_{1*}$ is the perfect state i.e., the state where none of the nodes, including the spare node have failed. $S_{2*}$ is a state where one node has failed and has been replaced by the spare node. The spare node could also fail but in this case no reconfiguration takes place. The hypercube structure is still maintained. The structure is now an ordinary 3-cube. Since in state $S_{1*}$ any node out of the 9 nodes within the FTBB can fail, the transition rate from $S_{1*}$ to $S_{2*}$ is $9\lambda_n$. The transitions out of $S_{2*}$ are similar to the transitions out of $S_*$ in Figure 2.1. The transition rates can be calculated with $d = 3$ in Figure 2.1. The transition rates are indicated in Figure 5.1. We write the state transition equation in the usual form [27].

$$P(t + \Delta t) = A.P(t) \tag{5.1}$$

$$\text{where } P(t+\Delta t) = \begin{bmatrix} P_{1*}(t+\Delta t) \\ P_{2*}(t+\Delta t) \\ P_0(t+\Delta t) \\ P_1(t+\Delta t) \\ P_2(t+\Delta t) \\ P_3(t+\Delta t) \end{bmatrix}, P(t) = \begin{bmatrix} P_{1*}(t) \\ P_{2*}(t) \\ P_0(t) \\ P_1(t) \\ P_2(t) \\ P_3(t) \end{bmatrix}$$

and the state transition matrix $A$ using a discrete markov model is given by

$$A = \begin{bmatrix} 1-9\lambda_n\Delta t & 0 & 0 & 0 & 0 & 0 \\ 9\lambda_n\Delta t & 1-8\lambda_n\Delta t & 0 & 0 & 0 & 0 \\ 0 & 8\lambda_n\Delta t & 1-7\lambda_n\Delta t & 0 & 0 & 0 \\ 0 & 0 & 3\lambda_n\Delta t & 1-6\lambda_n\Delta t & 0 & 0 \\ 0 & 0 & 3\lambda_n\Delta t & 4\lambda_n\Delta t & 1-4\lambda_n\Delta t & 0 \\ 0 & 0 & \lambda_n\Delta t & 2\lambda_n\Delta t & 4\lambda_n\Delta t & 1 \end{bmatrix}$$

The differential equations obtained from Equation 5.1 are as follows.

$$P'_{1*}(t) = -9\lambda_n P_{1*}(t)$$

$$P'_{2*}(t) = 9\lambda_n P_{1*}(t) - 8\lambda_n P_{2*}(t)$$

$$P'_0(t) = 8\lambda_n P_{2*}(t) - 7\lambda_n P_0(t)$$

$$P'_1(t) = 3\lambda_n P_0(t) - 6\lambda_n P_1(t)$$

$$P'_2(t) = 3\lambda_n P_0(t) + 4\lambda_n P_1(t) - 4\lambda_n P_2(t)$$

$$P'_3(t) = \lambda_n P_0(t) + 2\lambda_n P_1(t) + 4\lambda_n P_2(t).$$

With the initial conditions $P_{1*}(0) = 1$, $P_{1*}(0) = P_i(0) = 0$ for all $i$, the solutions for the above set are

$P_{1*}(t) = e^{-9\lambda_n t}$

$P_{2*}(t) = -9e^{-9\lambda_n t} + 9e^{-8\lambda_n t}$

$P_0(t) = 36e^{-9\lambda_n t} - 72e^{-8\lambda_n t} + 36e^{-7\lambda_n t}$

$P_1(t) = -36e^{-9\lambda_n t} + 108e^{-8\lambda_n t} - 108e^{-7\lambda_n t} + 36e^{-6\lambda_n t}$

$P_2(t) = (36/5)e^{-9\lambda_n t} - 54e^{-8\lambda_n t} + 108e^{-7\lambda_n t} - 72e^{-6\lambda_n t} + 54e^{-4\lambda_n t}$

$P_3(t) = 1 + (4/5)e^{-9\lambda_n t} + 9e^{-8\lambda_n t} - 36e^{-7\lambda_n t} + 36e^{-6\lambda_n t} - 54e^{-4\lambda_n t}$

$P_i$'s for a 2-FTBB and 4 FTBB could be derived in a similar manner. We just give the solutions for a 2 FTBB below

$P_{1*}(t) = e^{-5\lambda_n t}$

$P_{2*}(t) = -5e^{-5\lambda_n t} + 5e^{-4\lambda_n t}$

$P_0(t) = 10e^{-5\lambda_n t} - 20e^{-4\lambda_n t} + 10e^{-3\lambda_n t}$

$P_1(t) = (-20/3)e^{-5\lambda_n t} + 20e^{-4\lambda_n t} - 20e^{-3\lambda_n t} + (20/3)e^{-2\lambda_n t}$

$P_2(t) = 1 + (2/3)e^{-5\lambda_n t} - 5e^{-4\lambda_n t} + 10e^{-3\lambda_n t} - (20/3)e^{-2\lambda_n t}$

We define the reliabilities as follows

$R_*(t) = P_{1*}(t) + P_{2*}(t)$, $R_0(t) = P_0(t) + R*(t)$ and $R_i(t) = P_i(t) + R_{i-1}(t)$, for $i > 0$. The system's MTTF can be evaluated by integrating $R_i(t)$ [27]. The variation of subcube reliabilities $R_i(t)$ for all $i$, with $t$, for a 3 FTBB and a 3-cube are shown in Figure 5.2. To get $R_i(t)$ for a 3 cube we have used the reliability equations in

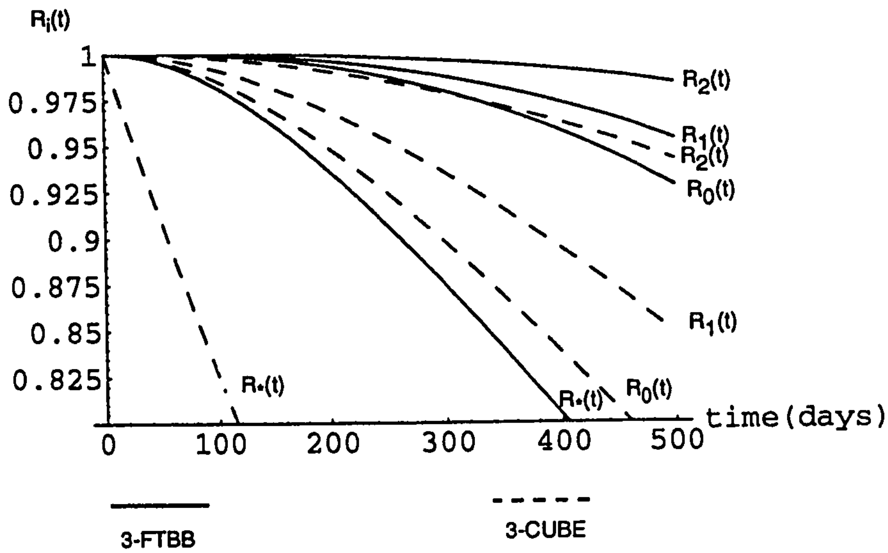


Figure 5.2: Reliabilities for 3-FTBB and Ordinary 3-cube.

Section 2.3 with $d = 3$. The subcube reliabilities of the 3-FTBBs are seen to be superior to the ordinary 3-cube. Similar reliability curves for the 2-FTBB and 2-cube are shown in Figure 5.3. Integrating the subcube reliability expressions $R_i(t)$ for any $i$, we get the corresponding MTTF. The comparison of the MTTF for the 2 and 3 FTBBs with ordinary 2 and 3 cubes is shown in Table 5.1 and Table 5.2 respectively.

We observe that the MTTF to leave the state of complete hypercube structure ($T_*$

Table 5.1: MTTF in hours for 2-FTBB and 3-FTBB for node failure model.

| $d$ | $N$ | $T_*$ | $T_0$ | $T_1$ | $T_2$ |
|---|---|---|---|---|---|
| 2 | 4 | 45000 | 78333 | 111667 | |
| 3 | 8 | 23611 | 37896 | 45040 | 62897 |

value) of the 2-FTBB or 3-FTBB is nearly twice the value of the ordinary 2-cube

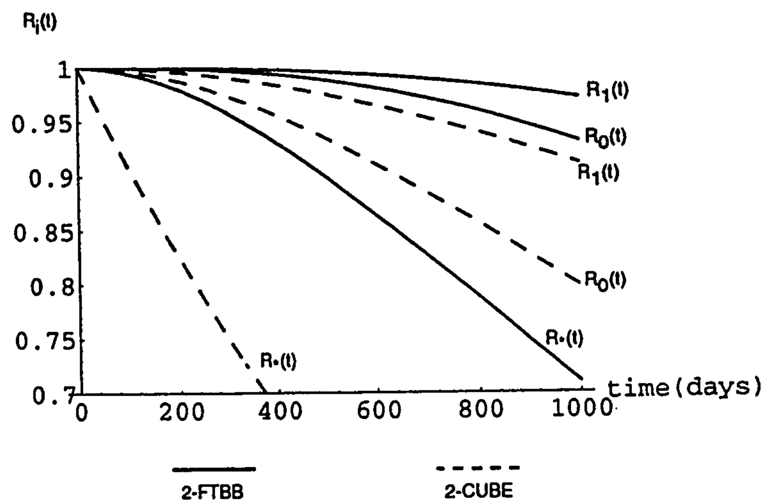


Figure 5.3: Reliabilities for 2-FTBB and Ordinary 2 cube.

Table 5.2: MTTF in hours for 2-cube and 3-cube for node failure model.

| $d$ | $N$ | $T_*$ | $T_0$ | $T_1$ | $T_2$ |
|---|---|---|---|---|---|
| 2 | 4 | 25000 | 58333 | 91666 | |
| 3 | 8 | 12500 | 26785 | 33929 | 51786 |

or 3-cube respectively. This has been achieved with a spare node and additional links. We take $\frac{1}{T_*}$ as the failure rate of the FTBB or the corresponding basic block under the node failure model. Then subcube reliability for larger cubes built using FTBBs or corresponding ordinary blocks is evaluated hierarchically with $\frac{1}{T_*}$ taken as the failure rate of the basic blocks.

## 5.2 Link Failure Model

In this Section, we shall derive the expressions for system reliability of a 2 and 3 FTBB using the definition of a working FTBB in the reconfiguration strategy presented in Section 4.2 assuming only link failures can occur. We then calculate the MTTF figures ($T_*$) of the 2 and 3 FTBB and compare them with corresponding values of ordinary 2 and 3 cubes respectively under link failure model. From $T_*$ we obtain failure rates of basic blocks for the link failure model, similar to the node failure case in Section 5.1.

Consider again the 3-FTBB in Figure 4.1(b). There are a total of 20 links in the system. Using the combinatorial model we calculate the probability of the 3-FTBB functioning when any $1, 2..., 20$ links fail in the 3-FTBB. The state of a functioning FTBB is defined in the reconfiguration rules in Section 4.2. Using this definition of a working FTBB we give the system reliability expressions for the functioning 2 and

3 FTBBs below for the link failure model.

$$Rl_{2-FTBB}(t) = r_l(t)^8 + 8r_l(t)^7(1-r_l(t)) + 20r_l(t)^6(1-r_l(t))^2 + 16r_l(t)^5(1-r_l(t))^3 + 2r_l(t)^4(1-r_l(t))^4 \quad (5.2)$$

Consider the expression for a 2-FTBB i.e., $Rl_{2-FTBB}(t)$. The first term includes the state where all the 8 links are working. The second term signifies that there are 8 ways of one link failing and still the FTBB successfully reconfigures. Other terms could be verified in a similar manner. For a 2-cube we have $Rl_{2-cube}(t) = r_l(t)^4$ which means all the links should function in a 2-cube.

$$\begin{aligned} Rl_{3-FTBB}(t) = {} & r_l(t)^{20} + 20r_l(t)^{19}(1-r_l(t)) + \\ & 166r_l(t)^{18}(1-r_l(t))^2 + 744r_l(t)^{17}(1-r_l(t))^3 + \\ & 1969r_l(t)^{16}(1-r_l(t))^4 + 3184r_l(t)^{15}(1-r_l(t))^5 + \\ & 3198r_l(t)^{14}(1-r_l(t))^6 + 2056r_l(t)^{13}(1-r_l(t))^7 + \\ & 914r_l(t)^{12}(1-r_l(t))^8 + 304r_l(t)^{11}(1-r_l(t))^9 + \\ & 74r_l(t)^{10}(1-r_l(t))^{10} + 12r_l(t)^9(1-r_l(t))^{11} + \\ & r_l(t)^8(1-r_l(t))^{12} \end{aligned} \quad (5.3)$$

For a 3-cube we have $Rl_{3-cube}(t) = r_l(t)^{12}$ Where $r_l(t) = e^{-\lambda_l t}$ is the reliability of a link.

Integrating Equations 5.2 and 5.3 we get the MTTF $T_*$ for the 2 and 3 FTBBs. $T_*$ for the 2 and 3 FTBBs are found to be 451190 and 237402 hours respectively for the link failure model ($\lambda_l = 10^{-6}/hour, \lambda_n = 10^{-5}/hour$). Corresponding values for the 2 and 3 cubes are 250000 and 83333 hours respectively. We observe that the MTTF to leave the state of working FTBB ($T_*$ value) of the 2-FTBB or 3-FTBB is superior to the value of the ordinary 2-cube or 3-cube respectively using the link failure model. We again consider $\frac{1}{T_*}$ as the failure rate of the FTBB or the corresponding ordinary basic block under the link failure model tp evaluate the subcube reliability of augmented cubes hierarchically.

## 5.3 Combined Node and Link Failure Model

In this Section, we shall derive the expressions for system reliability of a 2 and 3 FTBB using the definition of a working FTBB in the reconfiguration strategy presented in Section 4.2 assuming both node and link failures can occur. We then calculate the MTTF figures ($T_*$) of the 2 and 3 FTBB and compare them with corresponding values of ordinary 2 and 3 cubes respectively. From $T_*$ we obtain failure rates of basic blocks for the combined node and link failure model, similar to the node failure case in Section 5.1.

Here we use the bayes theorem to derive the expressions for the system reliability again using combinatorial approach [40]. Consider again the 3-FTBB in Figure 4.1(b). We separate the working state of the FTBB to two cases: the spare node is in working state and the spare node is in failed state. We again separate the working FTBB state where the spare node is working to two cases: Case 1 Where all primary nodes within the FTBB are working and case 2 where one of the primary nodes has failed and has been replaced. Following this approach we give the system reliability expressions for the functioning 2 and 3 FTBBs below for the combined node and link failure model.

$$Rc_{2-FTBB}(t) = r_n(t)[r_n(t)^4 Rl_{2-FTBB} + 4r_n(t)^3(1-r_n(t))R_{2f}(t)] + (1-r_n(t))[r_n(t)^4 r_l(t)^4] \tag{5.4}$$

For a 2-cube we have

$$Rc_{2-cube}(t) = r_n(t)^4 r_l(t)^4 \tag{5.5}$$

Expressions for the 3-FTBB and 3-cube are given below.

$$Rc_{3-FTBB}(t) = r_n(t)[r_n(t)^8 Rl_{3-FTBB} + 8r_n(t)^7(1-r_n(t))R_{3f}(t)] + (1-r_n(t))[r_n(t)^8 r_l(t)^{12}] \tag{5.6}$$

$$Rc_{3-cube}(t) = r_n(t)^8 r_l(t)^{12} \tag{5.7}$$

Where $r_n(t) = e^{-\lambda_n t}$ is the reliability of a node.

$R_{2f}$ is the probability of the 2 FTBB working when the spare node is working, any primary node within the FTBB has failed and has been replaced.

$$R_{2f} = r_l(t)^5 + 3r_l(t)^4(1 - r_l(t)) + r_l(t)^3(1 - r_l(t))^2 \tag{5.8}$$

$R_{3f}$ is the probability of the 3 FTBB working when the spare node is working, any primary node within the FTBB has failed and has been replaced.

$$R_{3f} = r_l(t)^{16} + 13r_l(t)^{15}(1-r_l(t)) + 66r_l(t)^{14}(1-r_l(t))^2 + 169r_l(t)^{13}(1-r_l(t))^3 + 240r_l(t)^{12}(1-r_l(t))^4 + 204r_l(t)^{11}(1-r_l(t))^5 + 111r_l(t)^{10}(1-r_l(t))^6 + 40r_l(t)^9(1-r_l(t))^7 + 9r_l(t)^8(1-r_l(t))^8 + r_l(t)^7(1-r_l(t))^9 \quad (5.9)$$

Integrating Equations 5.4 and 5.6 we get the MTTF $T_*$ for the 2 and 3 FTBBs. $T_*$ for the 2 and 3 FTBBs are found to be 42320 and 22394 hours respectively for the combined node and link failure model. Corresponding values for the 2 and 3 cubes are 22727 and 10869 hours respectively. We observe that the MTTF to leave the state of working FTBB ($T_*$ value) of the 2-FTBB or 3-FTBB is superior to the value of the ordinary 2-cube or 3-cube respectively using the combined node and link failure model. We again take $\frac{1}{T_*}$ as the failure rate of the FTBB or the corresponding ordinary basic block under the combined node and link failure model.

## 5.4 Supernode Failure Model

In this Section, we present reliability and MTTF values of 2-FTBB and 3-FTBB for the supernode failure model which is an approximation to the combined node and link failure case. We then compare the MTTF values with corresponding values of 3-cube and 4-cube respectively. A supernode is defined as a node and half its incident links such that every link is associated with a node. The 3-FTBB has 20 links and 9 nodes. We associate each link with a node so that the failure rate of the supernode in a 3-FTBB is $\lambda = \lambda_n + 20\lambda_l/9$. For a $2 - FTBB$, $\lambda = \lambda_n + 8\lambda_l/4$. Similarly, $\lambda = \lambda_n + 3\lambda_l/2$ and $\lambda = \lambda_n + 4\lambda_l/2$ for a 3-cube and 4-cube respectively. Substituting the value of $\lambda$s for the failure rate in the node failure model analysis in Section 5.1 we get the MTTF values for a 2-FTBB and 3-FTBB for the supernode failure model in Table 5.3. The MTTF values for a 2-cube and 3-cube are similarly obtained and are shown in Table 5.4. From Table 5.3, we observe that the MTTF for the FTBBs are superior to the ordinary cubes of the same order using the supernode failure model. We again take $\frac{1}{T_*}$ as the failure rate of the FTBB or the corresponding ordinary basic block under the supernode failure model. In Table 5.5 we show $T_*$ i.e., MTTF to leave the working state of an ordinary cube or a FTBB for 2 & 3 cubes and 2 & 3 FTBBs for all the failure models. From this table we see that the FTBBs are superior in reliability to the corresponding ordinary basic blocks under all the failure models.

Table 5.3: MTTF in hours for 2-FTBB and 3-FTBB for supernode failure model.

| $d$ | $N$ | $T_*$ | $T_0$ | $T_1$ | $T_2$ |
|---|---|---|---|---|---|
| 2 | 4 | 38793 | 67529 | 96264 | |
| 3 | 8 | 19318 | 31006 | 36850 | 51461 |

Table 5.4: MTTF in hours for 2-cube and 3-cube for supernode failure model.

| $d$ | $N$ | $T_*$ | $T_0$ | $T_1$ | $T_2$ |
|---|---|---|---|---|---|
| 2 | 4 | 22727 | 53030 | 83333 | |
| 3 | 8 | 10870 | 23292 | 29503 | 45031 |

Table 5.5: MTTF $T_*$ in hours for 2 & 3-cubes and 2 & 3-FTBBs with and without $S$ nodes, for all failure models.

| $FailureModel$ | $2-cube$ | $2-FTBB$ | $2-FTBB_S$ | $3-cube$ | $3-FTBB$ | $3-FTBB_S$ |
|---|---|---|---|---|---|---|
| Node | 25000 | 45000 | 48377 | 12500 | 23611 | 24540 |
| Link | 250000 | 451190 | 451190 | 83333 | 237402 | 237402 |
| Combined N & L | 22727 | 42320 | 45417 | 10869 | 22394 | 23304 |
| Supernode | 22727 | 38793 | 41260 | 10869 | 19318 | 19933 |

If we assume the $S$ node architecture for the spare nodes in our FTBBs then the failure rate of the spare node is $\frac{2\lambda_n}{3}$ where $\lambda_n$ is the failure rate of the primary node as explained at the end of Section 4.2. We denote the FTBBs with $S$ nodes as $FTBB_S$. In Table 5.5 we also show $T_*$ i.e., MTTF to leave the working state of a 2 or 3 FTBB with $S$ nodes for all the failure models. We observe from Table 5.5 that $T_*$ for FTBBs with $S$ nodes is marginally better than the corresponding FTBBs without $S$ nodes. Note that $T_*$ for FTBBs with and without $S$ nodes has the same value under link failure model.

# Chapter 6

# Reliability of Augmented Hypercubes

In this Chapter, we investigate the reliability and MTTF of larger cubes built of 2 and 3 FTBBs for node, link, combined node and link and the supernode failure models. We then compare the MTTFs obtained with corresponding cubes built of 2 and 3 cubes for each of the failure model. Sections 6.1, 6.2, 6.3 and 6.4 consider the reliability of the augmented cubes built of 2 and 3 FTBBs for node, link, combined node & link and the Supernode failure models respectively. Section 6.5 discusses the results obtained in the previous sections. Section 6.6 presents analytical models and results for augmented hierarchical hypercubes.

We refer to the cubes built of 2 and 3 FTBBs as *augmented* hypercubes. We also refer to the cube built of basic blocks either FTBB or the corresponding ordinary basic block, as the $b$-cube to distinguish from the cube built using single nodes referred to before as the $d$-cube as explained in Section 4.1. For cubes built of 3-FTBBs or 3-cubes as basic blocks we have $d = b + 3$. We define $N_b = 2^b$ to be the number of basic blocks in a $b$-cube. $N$ as before is the total number of nodes in the cube ($N = 2^d$). For example, a $b =$ 7-cube ($N_b = 2^7$ basic blocks ) built using 3-FTBBs or 3-cubes is actually a $d =$ 10-cube (contains $N = 2^{10}$ nodes).

We shall compare the reliability and MTTF of larger cubes built using 2 and 3 FTBBs as basic blocks with the cubes built using ordinary 2 and 3 cubes as basic blocks respectively for each of the failure models.

## 6.1 Node Failure Model

In this section, we present the reliability and MTTF analysis of augmented hypercubes built using 2 and 3 FTBBs as basic blocks for the node failure model. Augmented hypercubes can be built from FTBBs using the recursive construction property of hypercubes as explained in Section 4.1. The approach we use to evaluate the reliability and MTTF of augmented hypercubes is explained below.

Consider the cubes built of 3-FTBBs. From Table 5.5 we see that the MTTF to leave the working FTBB state ($T_*$) for the 3-FTBB under node failure model is 23611 hours. It's failure rate is therefore $1/T_* = 4.24 \times 10^{-5}/hour$ under node failure model. We consider the 3 FTBB as a basic block i.e., treat it as a single component with the failure rate = $4.24 \times 10^{-5}/hour$. Now we can use the markov model of Section 2.3 with $\lambda_n$ considered as the failure rate of the 3 FTBB under node failure model. The various subcube reliabilities and MTTF for a $b$ cube constructed of 3 FTBBs can be evaluated using the expressions in Section 2.3. Similar procedure could be adopted for cubes built of 3 cubes as basic blocks. Subcube reliabilities defined in 2.3 are evaluated and plotted for a $b =$ 7-cube ($2^{10}$ nodes) constructed of 3-FTBBs or 3-cubes as basic blocks in Figure 6.1.(a) and 6.1.(b) respectively. For comparison purposes we show each $R_i(t)$ for all $i$ in Figure 6.2. We observe that the reliability curves for a $b =$ 7-cube constructed of 3-FTBBs are superior to the curves for the same cube built of ordinary 3-cubes as basic blocks. Let us interpret the subcube reliabilities in Figure 6.1.(a) and 6.1.(b). $R_6^{ftbb}(t)$ for example, in the curves represents the probability that all failures are contained within a $b \leq$ 6-cube. $R_0^{ftbb}(t)$ in the curves represents the probability that all failures are contained within a $b =$ 0-cube, which means a single node has failed in cubes constructed of ordinary 3-cubes or two nodes have failed in a single 3-FTBB in cubes constructed of 3-FTBBs .

We have shown MTTF values for cubes of varying sizes built using 3-FTBBs or 3-ordinary cubes in Tables 6.1 and 6.2 respectively. We note that the MTTF $T_*$ for cubes built of 3-FTBBs is almost twice as much the value of the corresponding cubes built using ordinary 3-cube basic blocks. All other MTTFs ($T_i$s) of cubes built of 3 FTBBs are superior to the corresponding MTTFs of cubes built of 3 cube basic

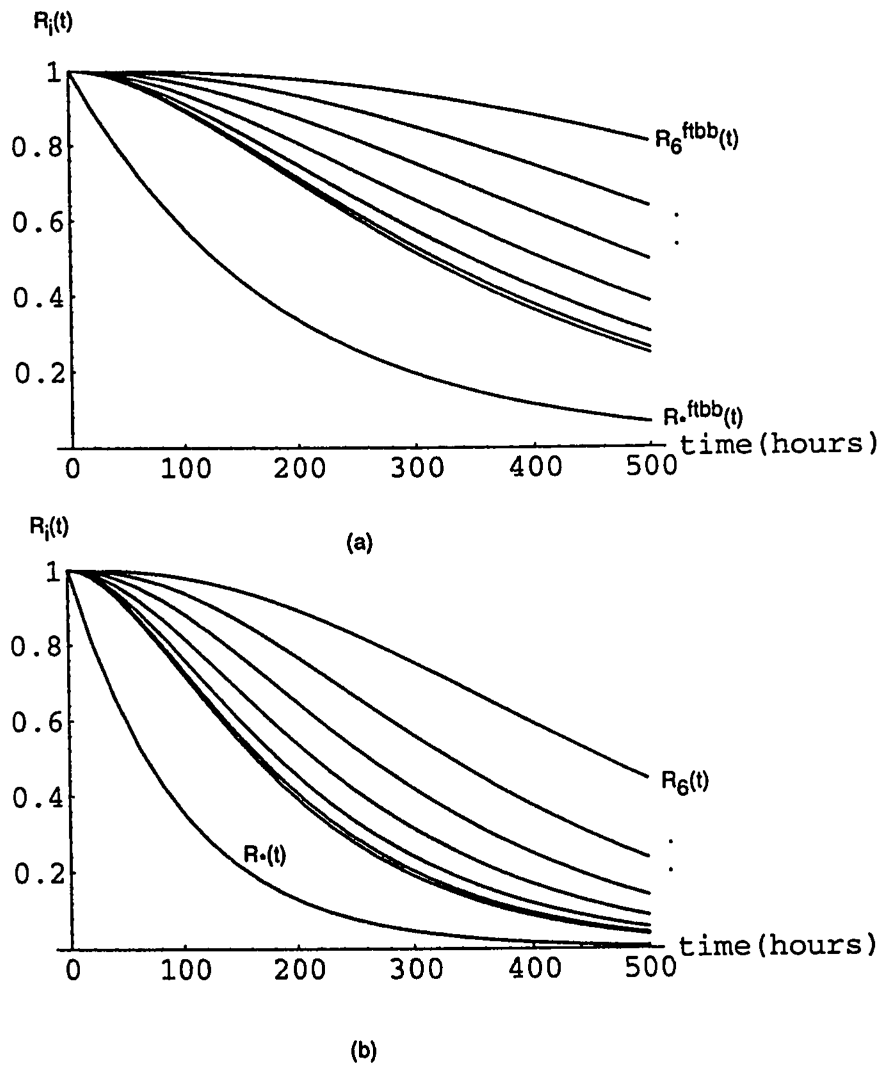


Figure 6.1: Reliabilities for a $b$ = 7-cube built (a)using 3-FTBBs (b) Ordinary 3-cubes.

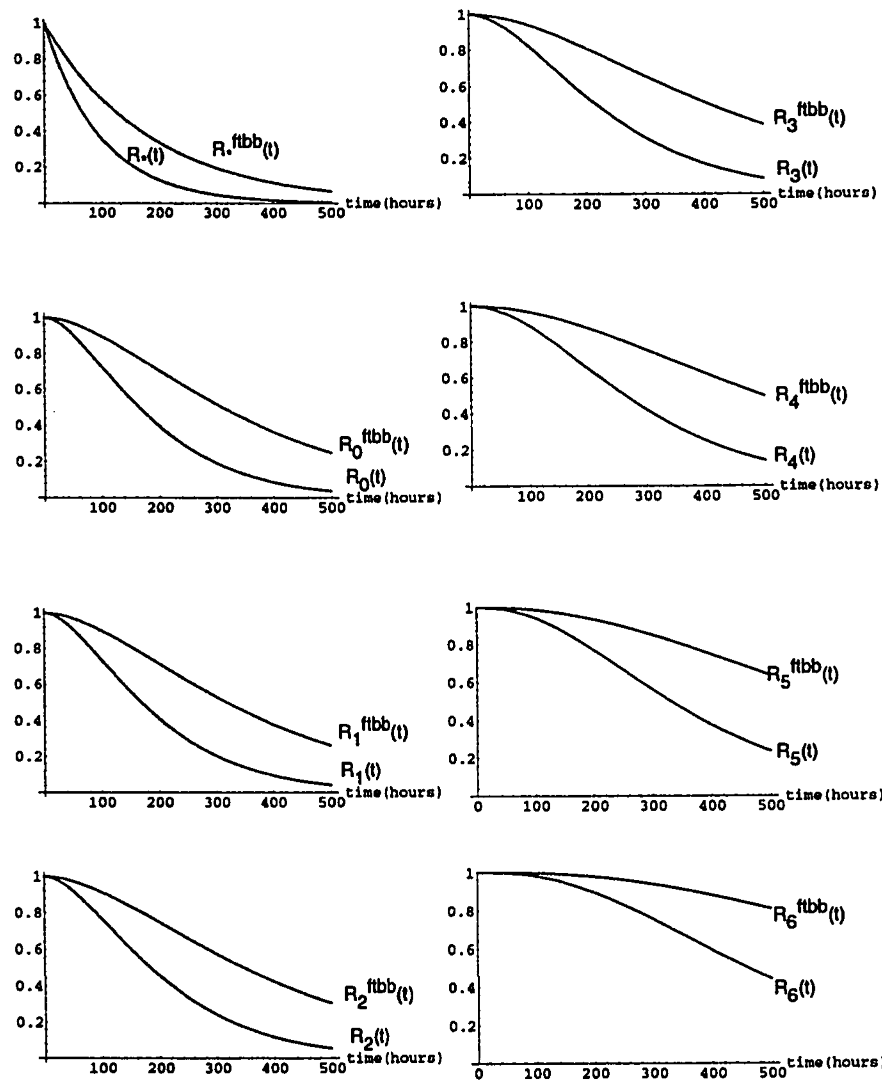


Figure 6.2: Comparison of individual reliabilities for a 10-cube built using 3-FTBBs and 3-cubes.

blocks. MTTFs for the cubes built of 2 FTBBs and 2 cubes as basic blocks for the node failure model can be obtained similarly and are shown in Tables 6.3 and 6.4 respectively. We observe from Tables 6.3 and 6.4 that the MTTFs of cubes built of 2 FTBBs are superior to the MTTFs of cubes built of 2 cubes.

Table 6.1: MTTF in hours for node failure model (basic block is a 3-FTBB).

| $b$ | $N_b$ | $N$ | $T_*$ | $T_0$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ | $T_8$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 8 | 64 | 2948 | 6317 | 8002 | 12214 | | | | | | |
| 4 | 16 | 128 | 1474 | 3046 | 3496 | 4506 | 6640 | | | | | |
| 5 | 32 | 256 | 737 | 1498 | 1625 | 1933 | 2467 | 3533 | | | | |
| 6 | 64 | 512 | 369 | 743 | 779 | 879 | 1054 | 1323 | 1855 | | | |
| 7 | 128 | 1024 | 184 | 370 | 380 | 413 | 475 | 566 | 700 | 966 | | |
| 8 | 256 | 2048 | 92 | 185 | 188 | 198 | 220 | 255 | 300 | 367 | 499 | |
| 9 | 512 | 4096 | 46 | 92 | 93 | 96 | 104 | 118 | 135 | 158 | 191 | 257 |

Table 6.2: MTTF in hours for node failure model (basic block is a 3-cube).

| $b$ | $N_b$ | $N$ | $T_*$ | $T_0$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ | $T_8$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 8 | 64 | 1562 | 3348 | 4241 | 6473 | | | | | | |
| 4 | 16 | 128 | 781 | 1615 | 1853 | 2388 | 3519 | | | | | |
| 5 | 32 | 256 | 391 | 794 | 861 | 1024 | 1307 | 1873 | | | | |
| 6 | 64 | 512 | 195 | 394 | 413 | 466 | 559 | 701 | 983 | | | |
| 7 | 128 | 1024 | 98 | 196 | 202 | 219 | 252 | 300 | 371 | 512 | | |
| 8 | 256 | 2048 | 49 | 98 | 99 | 105 | 117 | 135 | 159 | 194 | 265 | |
| 9 | 512 | 4096 | 24 | 49 | 49 | 51 | 55 | 62 | 72 | 83 | 101 | 136 |

Table 6.3: MTTF in hours for node failure model (basic block is a 2-FTBB).

| $b$ | $N_b$ | $N$ | $T_*$ | $T_0$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ | $T_8$ | $T_9$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 16 | 64 | 2812 | 5812 | 6670 | 8598 | 12670 | | | | | | |
| 5 | 32 | 128 | 1406 | 2858 | 3100 | 3687 | 4707 | 6742 | | | | | |
| 6 | 64 | 256 | 703 | 1417 | 1487 | 1677 | 2011 | 2525 | 3540 | | | | |
| 7 | 128 | 512 | 352 | 706 | 726 | 787 | 906 | 1080 | 1335 | 1842 | | | |
| 8 | 256 | 1024 | 176 | 352 | 358 | 378 | 420 | 486 | 572 | 699 | 953 | | |
| 9 | 512 | 2048 | 88 | 176 | 178 | 184 | 199 | 224 | 258 | 301 | 364 | 491 | |
| 10 | 1024 | 4096 | 44 | 88 | 88 | 90 | 96 | 105 | 119 | 135 | 157 | 188 | 252 |

Table 6.4: MTTF in hours for node failure model (basic block is a 2-cube).

| $b$ | $N_b$ | $N$ | $T_*$ | $T_0$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ | $T_8$ | $T_9$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 16 | 64 | 1562 | 3229 | 3705 | 4777 | 7039 | | | | | | |
| 5 | 32 | 128 | 781 | 1588 | 1722 | 2049 | 2615 | 3745 | | | | | |
| 6 | 64 | 256 | 391 | 787 | 826 | 931 | 1117 | 1403 | 1967 | | | | |
| 7 | 128 | 512 | 195 | 392 | 403 | 437 | 503 | 600 | 742 | 1024 | | | |
| 8 | 256 | 1024 | 98 | 196 | 199 | 210 | 234 | 270 | 318 | 389 | 529 | | |
| 9 | 512 | 2048 | 49 | 98 | 99 | 102 | 111 | 125 | 143 | 167 | 202 | 273 | |
| 10 | 1024 | 4096 | 24 | 49 | 49 | 50 | 53 | 59 | 66 | 75 | 87 | 105 | 140 |

## 6.2 Link Failure Model

In this section, we present the reliability and MTTF analysis of augmented hypercubes built using 2 and 3 FTBBs as basic blocks for the link failure model. The approach we use to evaluate the reliability and MTTF of augmented hypercubes for the link failure model is slightly different than the node failure model and is explained below.

Consider the cubes built of 3-FTBBs. From Table 5.5 we see that the MTTF to leave the working FTBB state ($T_*$) for the 3-FTBB under link failure model is 237402 hours. It's failure rate is therefore $1/T_* = 4.21 \times 10^{-6}/hour$ under link failure model. We consider the 3 FTBB as a basic block i.e., treat it as a single node with the failure rate given above. Therefore $\lambda_n = 4.21 \times 10^{-6}/hour$. A 3 FTBB has 8 links coming out of it. We consider all these links as a superlink with failure rate

8 times the failure rate of a single link. Therefore $\lambda_l = 8 \times 10^{-6}/hour$. Now we can use the markov model of Section 2.4 (combined node and link failure model results) with $\lambda_n$ considered as the failure rate of the 3 FTBB under link failure model and $\lambda_l = 8 \times 10^{-6}/hour$. Note that although we are using the results of the combined node and link failure model only link failures are taken into account above in calculating $\lambda_n$ and $\lambda_l$. The various subcube reliabilities and MTTF for a $b$ cube constructed of 3 FTBBs can be evaluated using the expressions in Section 2.4. Similar procedure could be adopted for cubes built of 3 cubes as basic blocks.

We have shown MTTF values for cubes of varying sizes built using 3-FTBBs or 3-ordinary cubes under link failure model in Tables 6.5 and 6.6 respectively. We note that all other MTTFs ($T_i$s) of cubes built of 3 FTBBs are superior to the corresponding MTTFs of cubes built of 3 cube basic blocks even under the link failure model. MTTFs for the cubes built of 2 FTBBs and 2 cubes as basic blocks for the node failure model can be obtained similarly and are shown in Tables 6.7 and 6.8 respectively. We observe from Tables 6.7 and 6.8 that the MTTFs of cubes built of 2 FTBBs are superior to the MTTFs of cubes built of 2 cubes for the link failure model.

Table 6.5: MTTF in hours for link failure model (basic block is a 3-FTBB).

| $b$ | $N_b$ | $N$ | $T_*$ | $T_{2-}$ | $T_0$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ | $T_8$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 8 | 64 | 7700 | 16300 | 21800 | 30800 | 48600 | | | | | | |
| 4 | 16 | 128 | 3100 | 6600 | 7900 | 9950 | 13350 | 19800 | | | | | |
| 5 | 32 | 256 | 1290 | 2730 | 3120 | 3630 | 4530 | 5850 | 8400 | | | | |
| 6 | 64 | 512 | 555 | 1170 | 1290 | 1425 | 1695 | 2070 | 2610 | 3660 | | | |
| 7 | 128 | 1024 | 240 | 520 | 552 | 592 | 672 | 792 | 952 | 1184 | 1620 | | |
| 8 | 256 | 2048 | 106 | 228 | 244 | 252 | 280 | 320 | 372 | 440 | 540 | 760 | |
| 9 | 512 | 4096 | 49 | 104 | 110 | 112 | 120 | 135 | 154 | 177 | 206 | 251 | 339 |

Table 6.6: MTTF in hours for link failure model (basic block is a 3-cube).

| $b$ | $N_b$ | $N$ | $T_*$ | $T_{2-}$ | $T_0$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ | $T_8$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 8 | 64 | 5125 | 8625 | 13375 | 18500 | 29375 | | | | | | |
| 4 | 16 | 128 | 2200 | 3950 | 5350 | 6600 | 8850 | 13250 | | | | | |
| 5 | 32 | 256 | 960 | 1770 | 2250 | 2580 | 3210 | 4170 | 6030 | | | | |
| 6 | 64 | 512 | 430 | 810 | 990 | 1080 | 1270 | 1550 | 1960 | 2890 | | | |
| 7 | 128 | 1024 | 192 | 368 | 432 | 456 | 520 | 616 | 736 | 920 | 1272 | | |
| 8 | 256 | 2048 | 84 | 168 | 196 | 204 | 224 | 256 | 300 | 352 | 432 | 592 | |
| 9 | 512 | 4096 | 41 | 79 | 90 | 91 | 98 | 110 | 126 | 145 | 169 | 206 | 277 |

Table 6.7: MTTF in hours for link failure model (basic block is a 2-FTBB).

| $b$ | $N_b$ | $N$ | $T_*$ | $T_{2-}$ | $T_0$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ | $T_8$ | $T_9$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 16 | 64 | 6100 | 12900 | 15600 | 19600 | 26300 | 39500 | | | | | | |
| 5 | 32 | 128 | 2550 | 5400 | 6200 | 7200 | 8950 | 11600 | 16900 | | | | | |
| 6 | 64 | 256 | 1100 | 2325 | 2575 | 2850 | 3375 | 4100 | 5175 | 7250 | | | | |
| 7 | 128 | 512 | 480 | 1020 | 1095 | 1170 | 1335 | 1575 | 1890 | 2355 | 3255 | | | |
| 8 | 256 | 1024 | 210 | 455 | 485 | 505 | 555 | 640 | 745 | 875 | 1075 | 1470 | | |
| 9 | 512 | 2048 | 96 | 206 | 218 | 220 | 240 | 268 | 306 | 350 | 408 | 498 | 684 | |
| 10 | 1024 | 4096 | 44 | 94 | 99 | 100 | 105 | 115 | 130 | 146 | 166 | 192 | 232 | 310 |

Table 6.8: MTTF in hours for link failure model (basic block is a 2-cube).

| $b$ | $N_b$ | $N$ | $T_*$ | $T_{2-}$ | $T_0$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ | $T_8$ | $T_9$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 16 | 64 | 5150 | 9950 | 12850 | 16000 | 21450 | 32150 | | | | | | |
| 5 | 32 | 128 | 2220 | 4360 | 5260 | 6100 | 7580 | 9820 | 14660 | | | | | |
| 6 | 64 | 256 | 970 | 1940 | 2250 | 2470 | 2930 | 3570 | 4510 | 6360 | | | | |
| 7 | 128 | 512 | 432 | 870 | 984 | 1044 | 1188 | 1404 | 1674 | 2082 | 2886 | | | |
| 8 | 256 | 1024 | 190 | 395 | 440 | 455 | 500 | 575 | 670 | 790 | 970 | 1320 | | |
| 9 | 512 | 2048 | 84 | 180 | 196 | 200 | 216 | 240 | 276 | 316 | 368 | 452 | 608 | |
| 10 | 1024 | 4096 | 41 | 83 | 91 | 91 | 96 | 105 | 119 | 134 | 152 | 176 | 213 | 284 |

## 6.3 Combined Node and Link Failure Model

In this section we present the reliability and MTTF analysis of augmented hypercubes built using 2 and 3 FTBBs as basic blocks for the combined node and link failure model. The approach we use to evaluate the reliability and MTTF of augmented hypercubes for the combined node and link failure model is same as that for the link failure model considered above. We use the markov model of Section 2.4 (combined node and link failure model results) as for the link failure model except that now $\lambda_n$ is considered as the failure rate of the 3 FTBB under combined node and link failure model. Therefore $\lambda_n = \frac{1}{22394} = 4.47 \times 10^{-5}$. $\lambda_l = 8 \times 10^{-6}/hour$ is same as before.

We have shown MTTF values for cubes of varying sizes built using 3-FTBBs or 3-ordinary cubes under combined node and link failure model in Tables 6.9 and 6.10 respectively. We note that all other MTTFs ($T_i$s) of cubes built of 3 FTBBs are superior to the corresponding MTTFs of cubes built of 3 cube basic blocks even under the combined node and link failure model. MTTFs for the cubes built of 2 FTBBs and 2 cubes as basic blocks for the combined node and link failure model can be obtained similarly and are shown in Tables 6.11 and 6.12 respectively. We observe from Tables 6.11 and 6.12 that the MTTFs of cubes built of 2 FTBBs are superior to the MTTFs of cubes built of 2 cubes.

Table 6.9: MTTF in hours for combined node and link failure model (basic block is a 3-FTBB).

| $b$ | $N_b$ | $N$ | $T_*$ | $T_{2-}$ | $T_0$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ | $T_8$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 8 | 64 | 2200 | 2750 | 5250 | 7000 | 10850 | | | | | | |
| 4 | 16 | 128 | 1025 | 1350 | 2325 | 2800 | 3700 | 5475 | | | | | |
| 5 | 32 | 256 | 480 | 660 | 1065 | 1200 | 1470 | 1905 | 2715 | | | | |
| 6 | 64 | 512 | 220 | 320 | 490 | 530 | 620 | 760 | 950 | 1350 | | | |
| 7 | 128 | 1024 | 105 | 155 | 235 | 240 | 270 | 320 | 385 | 475 | 660 | | |
| 8 | 256 | 2048 | 50 | 76 | 110 | 114 | 122 | 140 | 164 | 192 | 236 | 322 | |
| 9 | 512 | 4096 | 24 | 37 | 53 | 53 | 56 | 63 | 72 | 82 | 96 | 117 | 157 |

Table 6.10: MTTF in hours for combined node and link failure model (basic block is a 3-cube).

| $b$ | $N_b$ | $N$ | $T_*$ | $T_{2-}$ | $T_0$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ | $T_8$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 8 | 64 | 1175 | 1375 | 2800 | 3650 | 5650 | | | | | | |
| 4 | 16 | 128 | 550 | 675 | 1275 | 1525 | 2000 | 2950 | | | | | |
| 5 | 32 | 256 | 270 | 330 | 600 | 675 | 810 | 1050 | 1515 | | | | |
| 6 | 64 | 512 | 130 | 165 | 290 | 310 | 355 | 435 | 550 | 770 | | | |
| 7 | 128 | 1024 | 63 | 81 | 138 | 144 | 159 | 189 | 225 | 279 | 387 | | |
| 8 | 256 | 2048 | 31 | 40 | 67 | 69 | 74 | 84 | 98 | 116 | 142 | 194 | |
| 9 | 512 | 4096 | 15 | 20 | 33 | 33 | 35 | 38 | 44 | 51 | 59 | 72 | 96 |

Table 6.11: MTTF in hours for combined node and link failure model (basic block is a 2-FTBB).

| $b$ | $N_b$ | $N$ | $T_*$ | $T_{2-}$ | $T_0$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ | $T_8$ | $T_9$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 16 | 64 | 1980 | 2550 | 4500 | 5370 | 7110 | 10200 | | | | | | |
| 5 | 32 | 128 | 925 | 1250 | 2050 | 2300 | 2825 | 3725 | 5225 | | | | | |
| 6 | 64 | 256 | 440 | 600 | 960 | 1020 | 1200 | 1460 | 1840 | 2580 | | | | |
| 7 | 128 | 512 | 210 | 300 | 450 | 468 | 528 | 618 | 744 | 918 | 1260 | | | |
| 8 | 256 | 1024 | 98 | 147 | 215 | 220 | 238 | 271 | 317 | 373 | 457 | 623 | | |
| 9 | 512 | 2048 | 47 | 71 | 102 | 104 | 108 | 120 | 138 | 159 | 186 | 225 | 303 | |
| 10 | 1024 | 4096 | 22 | 34 | 49 | 49 | 50 | 54 | 62 | 70 | 80 | 92 | 111 | 148 |

Table 6.12: MTTF in hours for combined node and link failure model (basic block is a 2-cube).

| $b$ | $N_b$ | $N$ | $T_*$ | $T_{2-}$ | $T_0$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ | $T_8$ | $T_9$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 16 | 64 | 1170 | 1380 | 2670 | 3150 | 4170 | 6180 | | | | | | |
| 5 | 32 | 128 | 575 | 675 | 1250 | 1400 | 1700 | 2200 | 3150 | | | | | |
| 6 | 64 | 256 | 274 | 340 | 600 | 640 | 740 | 900 | 1120 | 1580 | | | | |
| 7 | 128 | 512 | 132 | 168 | 288 | 300 | 336 | 390 | 468 | 582 | 784 | | | |
| 8 | 256 | 1024 | 65 | 84 | 140 | 144 | 154 | 175 | 204 | 240 | 294 | 401 | | |
| 9 | 512 | 2048 | 32 | 41 | 66 | 68 | 72 | 80 | 90 | 105 | 122 | 149 | 200 | |
| 10 | 1024 | 4096 | 15 | 21 | 33 | 33 | 34 | 37 | 41 | 47 | 53 | 62 | 74 | 99 |

## 6.4 Supernode Failure Model

In this section we present the reliability and MTTF results of augmented hypercubes built using 2 and 3 FTBBs as basic blocks for the Supernode failure model.

Consider again cubes built of 3-FTBBs as basic blocks . The failure rates of the basic blocks are taken to be $(\frac{1}{T_*})$ as before. From Table 5.5, the failure rate of 3-FTBB considered as basic blocks are $1/T_* = 1/19318 = 5.17\times10^{-5}$. The failure rates of the supernode for cubes built of 3-FTBBs is given by $\lambda = 5.17 \times 10^{-5} + 2^3\lambda_l b/2$. The various subcube reliabilities and MTTF for a $b$ cube constructed of 3 FTBBs for the supernode failure model can be evaluated using the expressions in Section 2.3 and the failure rate of the supernode. Similar procedure could be adopted for cubes built of 3 cubes as basic blocks. We have shown MTTF values for cubes of varying sizes built using 3-FTBBs or 3-cubes in Table 6.13 and Table 6.14 respectively. We observe that all MTTFs ($T_i$s) of cubes built of 3 FTBBs are superior to the corresponding MTTFs of cubes built of 3 cube basic blocks. MTTFs for the cubes built of 2 FTBBs and 2 cubes as basic blocks for the supernode failure model can be obtained similarly and are shown in Tables 6.15 and 6.16 respectively. We observe from Tables 6.15 and 6.16 that the MTTFs of cubes built of 2 FTBBs are superior to the MTTFs of cubes built of 2 cubes.

## 6.5 Discussion of the Results

In this Section we shall discuss the subcube reliability and MTTF results presented in the the previous sections in this Chapter for each of the failure models. Observing Table 6.1 and Table 6.2 we see that the MTFFs for the cubes built from 3-FTBBs are nearly twice as much as the corresponding values of cubes built from 3-cubes under the node failure model. In fact the MTTFs scale with the failure rate ($\frac{1}{T_*}$) of the corresponding basic block (3-FTBB or 3 cube) given in Table 5.5 under the node failure model. Similar remarks hold for the MTTFs of cubes built from 2-FTBBs and 2-cubes shown in Table 6.3 and Table 6.4 respectively.

Table 6.13: MTTF in hours for supernode node failure model (basic block is a 3-FTBB).

| $b$ | $N_b$ | $N$ | $T_*$ | $T_0$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ | $T_8$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 8 | 64 | 1962 | 4205 | 5326 | 8130 | | | | | | |
| 4 | 16 | 128 | 923 | 1908 | 2189 | 2822 | 4159 | | | | | |
| 5 | 32 | 256 | 436 | 886 | 961 | 1143 | 1459 | 2090 | | | | |
| 6 | 64 | 512 | 206 | 416 | 436 | 492 | 590 | 741 | 1039 | | | |
| 7 | 128 | 1024 | 98 | 197 | 202 | 220 | 253 | 301 | 372 | 514 | | |
| 8 | 256 | 2048 | 47 | 94 | 95 | 100 | 112 | 129 | 152 | 186 | 253 | |
| 9 | 512 | 4096 | 22 | 45 | 45 | 47 | 50 | 57 | 65 | 76 | 92 | 124 |

Table 6.14: MTTF in hours for supernode failure model (basic block is a 3-cube).

| $b$ | $N_b$ | $N$ | $T_*$ | $T_0$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ | $T_8$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 8 | 64 | 1202 | 2576 | 3262 | 4979 | | | | | | |
| 4 | 16 | 128 | 579 | 1196 | 1372 | 1769 | 2607 | | | | | |
| 5 | 32 | 256 | 279 | 567 | 615 | 732 | 934 | 1338 | | | | |
| 6 | 64 | 512 | 135 | 272 | 285 | 321 | 385 | 484 | 678 | | | |
| 7 | 128 | 1024 | 65 | 131 | 134 | 146 | 168 | 200 | 247 | 341 | | |
| 8 | 256 | 2048 | 32 | 63 | 64 | 68 | 75 | 87 | 103 | 125 | 171 | |
| 9 | 512 | 4096 | 15 | 31 | 31 | 32 | 35 | 39 | 45 | 52 | 63 | 85 |

Table 6.15: MTTF in hours for supernode failure model (basic block is a 2-FTBB).

| $b$ | $N_b$ | $N$ | $T_*$ | $T_0$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ | $T_8$ | $T_9$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 16 | 64 | 1496 | 3092 | 3548 | 4574 | 6739 | | | | | | |
| 5 | 32 | 128 | 683 | 1387 | 1505 | 1790 | 2285 | 3273 | | | | | |
| 6 | 64 | 256 | 314 | 633 | 664 | 748 | 898 | 1127 | 1580 | | | | |
| 7 | 128 | 512 | 145 | 292 | 300 | 325 | 374 | 446 | 552 | 761 | | | |
| 8 | 256 | 1024 | 68 | 135 | 138 | 145 | 162 | 187 | 220 | 269 | 367 | | |
| 9 | 512 | 2048 | 32 | 63 | 64 | 66 | 72 | 81 | 93 | 108 | 131 | 176 | |
| 10 | 1024 | 4096 | 15 | 30 | 30 | 31 | 32 | 36 | 40 | 46 | 53 | 64 | 85 |

Table 6.16: MTTF in hours for supernode failure model (basic block is a 2-cube).

| $b$ | $N_b$ | $N$ | $T_*$ | $T_0$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ | $T_8$ | $T_9$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 16 | 64 | 1042 | 2153 | 2470 | 3184 | 4692 | | | | | | |
| 5 | 32 | 128 | 488 | 992 | 1076 | 1280 | 1634 | 2341 | | | | | |
| 6 | 64 | 256 | 230 | 463 | 486 | 548 | 657 | 825 | 1157 | | | | |
| 7 | 128 | 512 | 109 | 218 | 224 | 243 | 280 | 333 | 412 | 569 | | | |
| 8 | 256 | 1024 | 51 | 103 | 105 | 110 | 123 | 142 | 167 | 205 | 279 | | |
| 9 | 512 | 2048 | 24 | 49 | 49 | 51 | 55 | 62 | 72 | 83 | 101 | 136 | |
| 10 | 1024 | 4096 | 12 | 23 | 23 | 24 | 25 | 28 | 31 | 36 | 41 | 50 | 67 |

Further note that the MTTF $T_*$ for for cubes built of 3-cubes or 2-cubes has the same value for the cubes built of single nodes (refer Table 2.1) for the same value of $N$ (number of nodes) under the node failure model. This is expected as the MTTF $T_*$ is the Mean Time To Failure to leave the perfect state (MTTF to the first node failure). Comparing the MTTFs for cubes built from 3-FTBBs and 2 FTBBs under node failure model (Tables 6.1 and 6.3) we see that for the same value of $N$, MTTF $T_*$ for the cubes built from 3-FTBBs is superior to the corresponding value for the cubes built from 2-FTBBs. This is due to the following reason. Cubes built from 2-FTBBs have twice the number of spare nodes than cubes built from 3-FTBBs. These additional spare nodes (their possibility of failure) in the cubes built of 2 FTBBs account for the lower value of $T_*$. We remind that $T_*$ is the mean time to first failure. Other $T_i$s cannot be compared as they denote different states in terms of the number of failures that can occur leading to them. However it should be noted that cubes built from 2 FTBBs will be more fault-tolerant than cubes built from 3 FTBBs, for the same value of $N$ due to the following reason. Consider a 3-FTBB and a 3-cube built using two 2-FTBBs. Two node failures cannot be tolerated in a 3-FTBB but in a 3-cube built from two 2-FTBBs two node failures can be tolerated if they occur in different 2-FTBBs. Conclusions for the MTTFs under the supernode failure model are similar to the node failure model case.

Now consider the MTTFs for the cubes built from 3 FTBBs and 3-cubes under the link failure model (Tables 6.5 and 6.6). The MTTFs for the cubes built from

3-FTBBs are superior to the corresponding MTFFs for cubes built using 3-cubes, but unlike the node failure model, the MTTFs do not scale with failure rate of the corresponding basic block (3-FTBB and 3-cube) under link failure model. This is explained as follows. In the node failure model failure of any node in a cube built from 3-FTBB is tolerated but this is not the case in the link failure model. In particular only intra-FTBB link failures are tolerated but not inter-FTBB links in our reconfiguration strategy. In other words no spare link can replace any failed link which connects the nodes of different FTBBs. Similar remarks hold for the cubes built from 2-FTBBs and 2-cubes under the link failure model. Again Comparing the MTTFs for cubes built from 3-FTBBs and 2 FTBBs under link failure model (Tables 6.5 and 6.7) we see that for the same value of $N$, MTTF $T_*$ for the cubes built from 3-FTBBs is superior to the corresponding value for the cubes built from 2-FTBBs. This is due to the following reason. For example, consider a 3-FTBB and a 3-cube. In a 3 FTBB failure of any first link can be tolerated but in a 3-cube built using 2-FTBBs not all first link failures can be tolerated. In particular if links interconnecting the two 2-FTBBs fail, no spare link can replace them. Further for the same reason, cubes built from 3 FTBBs will be more fault-tolerant than cubes built from 2 FTBBs, for the same value of $N$ unlike the node failure model. Conclusions for the MTTFs under the combined node and link failure model are similar to the link failure model case.

### 6.5.1 Cost/Fault Tolerance Tradeoff

In this section we shall calculate the percentage node and link redundancy in our scheme over a regular hypercube. Let us first calculate the node and link redundancy in our scheme over a regular hypercube. This redundancy is a measure of the cost of our scheme to achieve higher reliability over the regular hypercube. In general for cubes built of $m$ FTBBs the percentage node redundancy is $\frac{100}{(2^m+1)}$ and the link redundancy is $\frac{100}{(1+\frac{d}{2})}$. Note that the node redundancy does not depend on the order of the augmented cube and the link redundancy does not depend on the order of the FTBB. For practical hypercubes such as $d = 10$ and $d = 12$ the node redundancy is approximately 11% if 3 FTBBs are used and 20% if 2 FTBBs are used. The link redundancy is 17% for a $d = 10$ cube and approximately 14% for a $d = 12$ cube.

### 6.5.2 Runtime Performance of Hypercube Applications

In this section we shall comment on the degradation in performance suffered by actual hypercube applications when embedded on hypercubes using Banerjee and Peercy's fault-tolerant schemes [7]. Since even our architecture uses fault-tolerant routing like Banerjee and Peercy's scheme the following general comments hold for our architecture too.

Banerjee and Peercy have done work on evaluating the reliability of their reconfiguration schemes [7] on actual hypercube applications. Their work shows that given a bounded number of faults the dilation of logical links is bounded. More specifically they quantify the loss in performance of the reconfigured system by examining the runtimes of standard applications on iPSC/2. They used the simulator developed by Hsu and Banerjee [26]. This simulator was used to investigate communication characteristics of the commercially-available Intel iPSC/2 hypercube. The simulator accurately models all the steps of communication, both hardware and software. Banerjee and Peercy augmented the simulator to allow for faults and the tolerance of them by the reconfiguration schemes outlined above. The following applications were run on their simulator, three numerical programs and three CAD programs.

1. **FFT** [21]: The FFT algorithm maps perfectly on a hypercube. Each node performs identical computation on its points and then exchanges these with its neighbors in each dimension in turn. All communication in the programming of the FFT algorithm on a hypercube is nearest neighbor and very synchronous.

2. **QR**[32]:(128 × 128 matrix) The QR factorisation application maps columns of the matrix on a ring of processors and performs orthogonal factorisation on them. It is used as an example of a numerical algorithm carefully and successfully tuned to execute of the hypercube.

3. **TRED2**[23]: A subroutine taken from the EISPACK routines for linear algebra, TRED2 does not map well to distributed processing. As a result this

program is very communication intensive, and it provides a good example for examining the reconfiguration under high network load.

4. **EXTRACT**[9]:VLSI circuit extraction. This program runs in two distinct phases. In the first, communication intensive phase, the circuit is sent to the nodes in a tree fashion. In the second phase, the circuits are locally extracted and the results are merged, a computation intensive activity.

5. **PLACE**[36]:cell placement. This CAD program uses a simulated annealing algorithm to place cells on a VLSI chip so that connecting wirelength is minimizing. Cells allocated geographically to processors are exchanged in moves at every iteration. Hence it requires fine-grain synchronization.

6. **TEST**[30]:test generation. In this application, processor 0 distributes brand-and-bound tasks to all other processors. Thus every communication is sourced or sinked at processor 0, and communication in general is very asynchronous-processor 0 being able to source only one message at a time.

Each of the six experiments was run on four (simulated) hypercube machines, each with node-sparing and link-sparing, reconfigured from varying degrees of fault injection. To standardize the results all four machines use the iPSC/2 communications protocols and timings. Runtime overhead is the most important overhead in the design of fault-tolerant schemes for hypercubes. The simulation results given in [7] demonstrate that logical mapping of links i.e., replacing single physical link delays with multiple link delays in their schemes does not degrade the runtime performance of the above applications. Specifically the runtime overheads are small for both proposed schemes i.e., the *node spare scheme* and the *link spare scheme.* For the 4-fault cases (the most faults which can possibly be tolerated in a 4-spare 16 node machine) the mean magnitudes of overheads run from under 0.04% for TEST to under 7.4% for TRED2.

## 6.6 Augmented Hierarchical Hypercubes

In this Section we shall consider the possibility of using FTBBs for the level 1 clusters in hierarchical hypercubes instead of BH/BH-SI scheme. We denote this scheme as BH/BH-FTBB and called such cubes as augmented hierarchical hypercubes.

A two level hierarchical hypercube network using 3-FTBBs is shown in Figure 6.3. Note that the spare node in BH/BH-FTBB network can replace any node, including the interface node in the level 1 network unlike the BH/BH-SI scheme where the spare can replace only the interface node. Further, link failures within the level 1 cluster are also tolerated by using the reconfiguration strategy given in Section 4.2. Even in the BH/BH-FTBB scheme the level 2 network could be duplicated like the BH/BH-RS scheme where two nodes within the cluster act as interface nodes. Observing Figure 6.3 we see that when the interface node fails, the spare node within the level 1 network replaces it, but the spare node now is not connected to the level

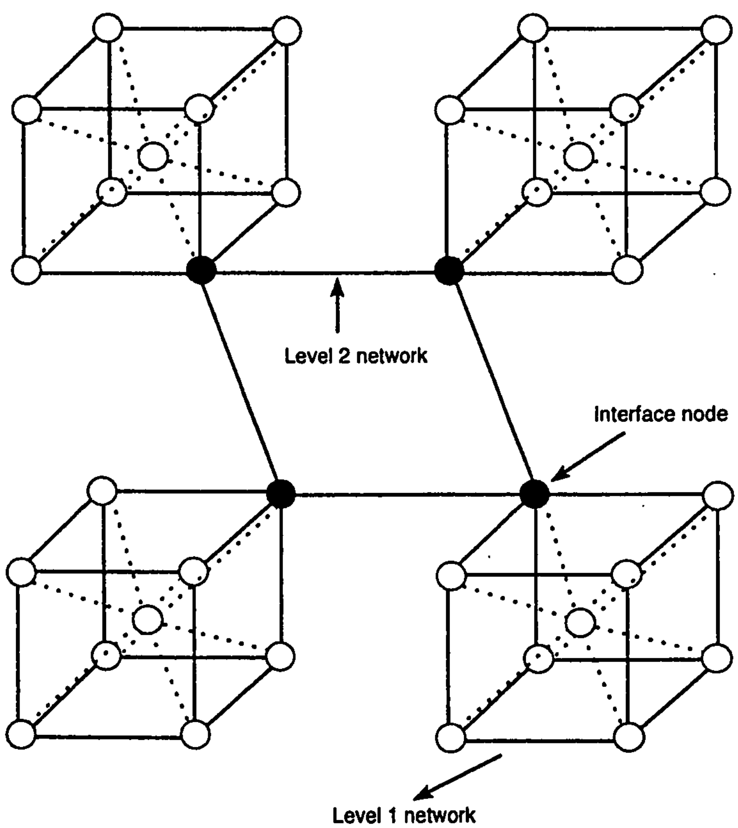


Figure 6.3: A two level BH/BH-3FTBB network.

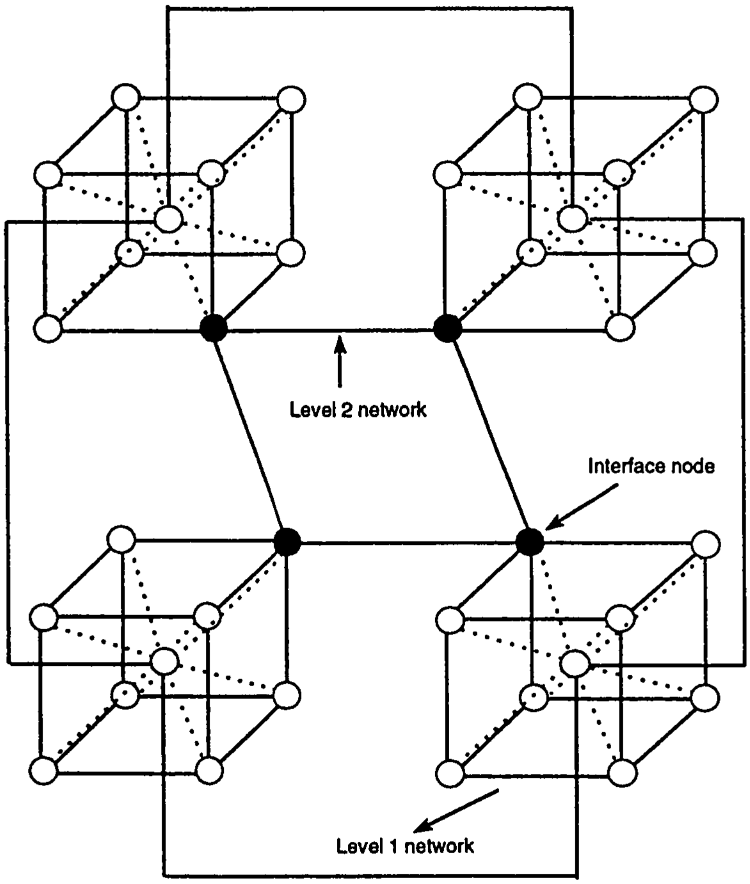


Figure 6.4: A two level BH/BH-3FTBB-RS network.

two network. To avoid this we suggest to replicate the level 2 network on the spare nodes of level 1 clusters. We call this scheme BH/BH-FTBB-RS similar to the BH/BH-RS scheme and is shown in Figure 6.4. We shall now use an analytical model to evaluate the system reliability of the BH/BH-FTBB-RS scheme and also the hierarchical schemes given in Section 3.2 such as BH/BH, BH/BH-SI and BH/BH-RS. The analytical model is explained below.

Let the level cluster be a 3-cube or a 3-FTBB. Consider a 64 node network. So the level 2 network is a 3-cube. We have derived expressions for system reliability for a 3 cube and 3-FTBB in Section 5.3 (equations 5.7 and 5.6). We use these system reliability expressions to calculate the system reliability of each level 1 cluster. Now consider the level 1 clusters as single nodes connected by the level 2 network, which is a 3-cube if 64 nodes are to be connected. We can again use the same expressions to calculate the system reliability of the level 2 network, with level 1 clusters considered as single nodes. This gives the system reliability for the whole network. Using this analytical model and using equations 5.7 and 5.6, the system reliability for a 64 node BH/BH network is

$R_{BH/BH}(t) = r_l^{12} R_{c3-cube}^8$. $R_{c3-cube}$ is the reliability of a $3 - cube$ cluster and all 8 such clusters should function ($R_{c3-cube}^8$) alongwith all the 12 links ($r_l^{12}$) of the level 2 network for the 64 node BH/BH network to function. Other expressions given below could be explained in a similar manner.

For a 64 node BH/BH-SI we have

$R_{BH/BH-SI}(t) = r_l^{12} R_{SI1}^8$ where

$R_{SI1} = r_l^{12}(r_n^9 + 2r_n^8(1 - r_n))$.

is the system reliability of a level 1 cluster with a spare for the interface node. For a 64 node BH/BH-RS we have

$R_{BH/BH-RS}(t) = r_{l*}^{12} R_{c3-cube}^8$ where,

$r_{l*} = e^{-2\lambda_l t/3}$ is the reliability of each pair of replicated links in the level 2 network.

For a 64 node BH/BH-RS & SI we have

$R_{BH/BH-RS\&SI}(t) = r_{l*}^{12} R_{SI2}^{8}$ where,

$R_{SI2} = r_n^{10} + 4r_n^9(1 - r_n) + 4r_n^8(1 - r_n)^2$.

For a 64 node BH/BH-3FTBB-RS we have

$R_{BH/BH-3FTBB-RS}(t) = r_{l*}^{12} R_{c3-ftbb}^{8}$.

$r_n$ and $r_l$ are the reliability of a node and link respectively. In the BH/BH-3FTBB-RS if we do not replicate the level 2 network, instead provide a spare for the spare within the FTBB (a spare node for a spare node) and the level 2 network only connects the spares within the FTBB we get the BH/BH-FTBB-SI network. The BH/BH-FTBB network saves on links but uses extra spare nodes. Similarly expressions for system reliability can be derived if 2 cubes or 2-FTBBs are used for level 1 clusters. The system reliability of each of these 64 node hierarchical networks is shown in Figure 6.5 From this figure we see that using FTBBs for level 1 clusters improves the system reliability considerably. This is due to the following. There are a large number of primary link failure configurations which are tolerated. This can be seen in the reconfiguration strategy given in Figure 4.5 used to tolerate node and link failures within the FTBB. Consider Figure 4.1. Specifically when a primary link fails, two spare links replace it. This is accepted and the FTBB is considered to be working. Trace driven performance studies on hypercubes have shown that 2 hop message delays are only slightly greater than single hop message delays [15, 5, 31, 26].

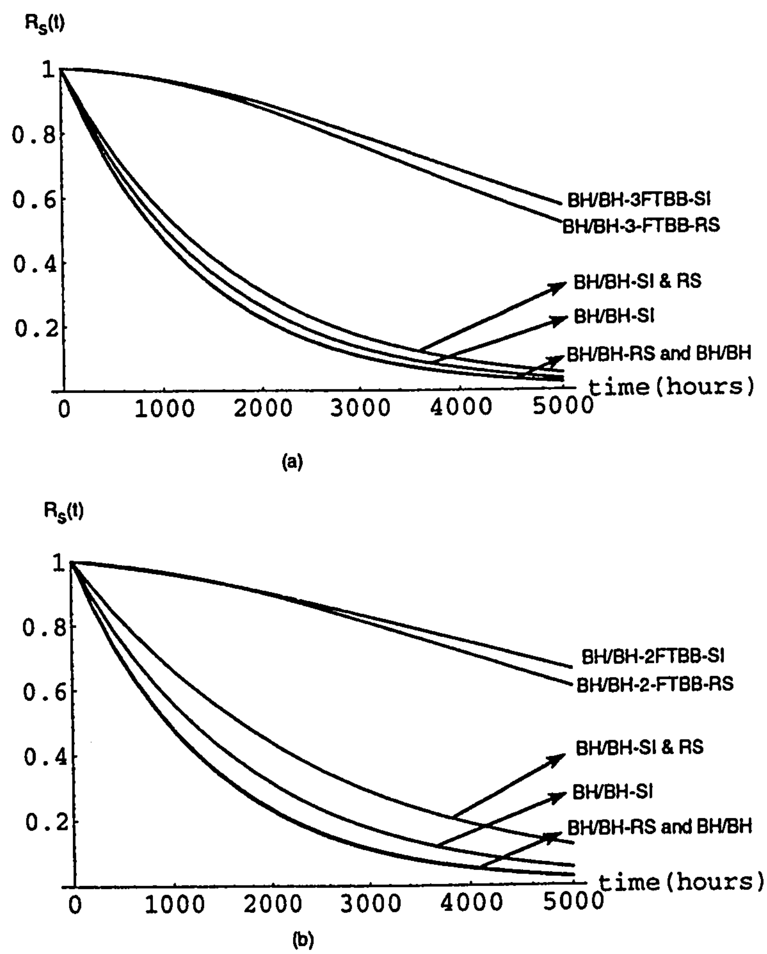


Figure 6.5: System Reliability for 64 node BH/BH, BH/BH-SI, BH/BH-RS, BH/BH-SI&RS and BH/BH-FTBB-RS (a) Cluster size 3-cube or 3-FTBB (b) Cluster size 2-cube or 2-FTBB.

# Chapter 7

# Conclusions and Future Research

This final chapter summarizes the results of the thesis and its contributions to fault-tolerance in hypercubes. Future research based on this thesis and general directions in related areas are also described.

## 7.1 Contributions

In this Section we present the main contributions of this research in the area of fault-tolerant hypercube architectures.

- This thesis presented a new modular fault-tolerant hypercube architecture capable of tolerating both nodes and link failures in a hypercube.
- This thesis has given a set of rules for reconfiguration in the event of node and link failures in a hypercube for the proposed fault-tolerant hypercube

architecture.

- This thesis presented subcube reliability analysis of the proposed fault-tolerant hypercube architecture for four failure models:

  1. Node failure model: where only nodes are permitted to fail
  2. Link failure model: where only links are permitted to fail
  3. Combined node and link failure model: where both nodes and links are permitted to fail
  4. Supernode failure model: an approximation to the more realistic combined node and link failure model

  The analysis showed that the architecture has better subcube reliabilities and Mean Time To Failures (MTTF) than ordinary hypercube. We have analysed subcube reliability of our architecture. This analysis could be extended to other fault-tolerant hypercube architectures. Specifically our subcube reliability analysis for the node failure model holds for Rennels's, Chau et al's and Sultan and Melhem's schemes with the fault-tolerant modules consisting of a single spare node. Note that the Rennels's, Chau et al's and Sultan and Melhem's schemes do not provide spare links and do not tolerate link failures with spare links.

- This thesis extended the FTBB idea to be incorporated in hierarchical hypercubes which has been shown to have better system reliability than previous fault-tolerant techniques in hierarchical hypercubes.

## 7.2 Summary and Conclusions

This research has proposed a modular fault-tolerant hypercube architecture which can tolerate both node and link failures in a hypercube. The scheme we have proposed has extended the ideas for modular fault tolerance in hypercubes investigated in [34, 12, 42, 7]. We have analysed the subcube reliability of the proposed modular fault-tolerant hypercube architecture extending the the subcube reliability analysis of hypercube presented in [1]. The analysis shows that the fault-tolerant hypercube architecture we propose is more resilient than the basic hypercube in terms of its ability to support several smaller subcubes in the damaged structure. Specifically, the reliabilities and the MTTF values of the FTBBs have been shown to be superior to those of the ordinary cubes of the same order under the node, link, combined node and link and supernode failure models. The reliabilities and MTTF of augmented cubes built of FTBBs have been shown to be superior to corresponding cubes built using ordinary basic blocks under the node, link, combined node and link and supernode failure models. We have extended the idea of FTBBs to be incorporated in hierarchical hypercube networks which gives better reliability than current fault-tolerant techniques.

## 7.3 Future Research

In this Section we outline some future research topics in the area of fault-tolerant hypercube architectures prompted by the work presented herein. We have analysed subcube reliability of the modular fault-tolerant architecture we presented. Other measures that characterize the fault-tolerant hypercube systems are performance related dependability and diagnosability. Performance related dependability of a fault-tolerant hypercube architecture is a combined measure of the reliability and performance. Performance is related to the average diameter between two nodes which effects the message delay in a hypercube network. The introduction of spare nodes and links in the architecture we presented serves to reduce the average diameter between nodes in a hypercube. The spare nodes and links could be therefore utilized to increase the performance. Performance related dependability is treated in [17, 16]. Performance related dependability is another important measure in fault tolerance and could be investigated for modular fault-tolerant hypercube architectures. Diagnosability of a hypercube architecture is a measure of how effectively nodes in a hypercube can test their adjacent nodes to find for node failures. In [45] diagnosability of a fault-tolerant hypercube architecture which has spare links between certain nodes is investigated. In general diagnosability of the modular fault-tolerant hypercube architectures including the architecture we presented could be investigated.

The modular fault-tolerant hypercube architecture together with the reconfiguration strategy we presented tolerate link failures only within a FTBB i.e., intra-FTBB link failures are tolerated but not inter-FTBB links. The reconfiguration strategy could be extended to tolerate inter-FTBB links. The spare links within a FTBB could be used to replace link failures outside the FTBB as well. This needs to be investigated.

Several advantages are associated with modular sparing. Global sparing uses less spare overhead but in general the reconfiguration strategies become complex. Specifically in hypercubes the question is how large a FTBB has to be to achieve high reliability as well as performance. Current VLSI technology also affects this. Various factors like the average message delay, reliability, reconfiguration strategy etc., have to be considered on deciding the size of the FTBB. This issue needs further investigation.

# Bibliography

[1] S. Abraham and K. Padmanabhan. Reliability of the hypercube. *International Conference on Parallel Processing*, II(4):7–19, oct. 1988.

[2] El-Amawy Ahmed and R. Raja. Split sequence generation algorithms for efficient identification of operational subcubes in faulty hypercubes. *Parallel Computing*, 19(7):789–805, July 1993.

[3] R. M. Ahmed and J. L. Trahan. Two terminal reliability of hypercubes. *Proc. Southeastcon'91*, pages 427–431, 1991.

[4] J. R. Armstrong and F. G. Gray. Fault diagnosis in a boolean n-cube array of microprocessors. *IEEE Transactions on Computer*, C-30(8):587–590, Aug. 1981.

[5] H. Badr and S. Podar. An optimal shortest path routing policy for network computers with regular mesh connected topologies. *IEEE Transactions on Computers*, 38(3):284–294, 1989.

[6] P. Banerjee and Michael Peercy. Design and evaluation of hardware strategies for reconfiguring hypercubes and meshes under faults. *Coordinated Science lab Tech, Illinois, Rep*, 1992.

[7] P. Banerjee and Michael Peercy. Design and evaluation of hardware strategies for reconfiguring hypercubes and meshes under faults. *IEEE Transactions on Computers*, 43(7):841–848, July. 1994.

[8] P. Bannerjee et al. An evaluation of system level fault tolerance on the intel hypercube multiprocessor. *Proc. 18th Int. Symp. Fault-Tolarant computing*, pages 362–367, 1988.

[9] K. P. Belkhale and P. Banerjee. Pace2: An improved parallel vlsi extractor with parametric extraction. *Proc.Int. Conf. Computer Aided design*, pages 526–530, 1989.

[10] Bruck et al. Tolerating faults using subcube partitioning. *IEEE Transactions on Computers*, 41(5):599–605, May. 1992.

[11] D. Bulka and J. B. Dugan. A lower bound on the reliability of an n-dimesional hypercube. *Proc. 9th Symp. Reliable Distributed Systems*, pages 44–53, 1990.

[12] Siu-Cheung Chau and A. L. Liestman. A proposal for a fault-tolerant binary hypercube architecture. *19th International Symposium on Fault Tolerant Computing*, pages 323–331, June. 1989.

[13] T. F. Chau and Y. Saad. Multigrid algorithms on the hypercube multiprocessors. *IEEE Transactions on Computers*, c-35:969–977, 1986.

[14] M. Chen and K. Chin. Message routing in an injured hypercube. *Proc. Fourth Hypercube Conf*, pages 312–317, 1988.

[15] M. S. Chen and K. G. Shin. Depth first search approach for fault tolerant routing in hypercube multicomputers. *IEEE Transactions on Parallel and Distributed Systems*, 2:152–159, April. 1990.

[16] S. P. Dandamudi. *Heirarchial Hypercube Multicomputer Interconnection Networks.* Ellis Horwood series, 1991.

[17] C. R. Das et al. Dependability modelling for multiprocessors. *IEEE Computer*, II(4):7–19, Oct. 1990.

[18] C. R. Das and J. Kim. A unified task based dependability model for hypercube computers. *IEEE Transactions on Parallel and Distributed systems*, 3:312–324, May. 1992.

[19] J. J. Dongorra and I. S. Dutt. Advanced architecture computers. *Tech Memo 57, Argonne National Lab*, 1985.

[20] A. El-Amawy and S. Latifi. Properties and performance of folded hypercubes. *IEEE Transactions on Parallel and Distributed Systems*, 2:31–42, Jan. 1991.

[21] G. C. Fox et al. *Solving problems on concurrent processors.* Englewood Cliffs, Prentice-Hall, 1989.

[22] J. M. Gordon and Q. F. Stout. Hypercube message routing in the presence of faults. *Proc. 3rd conf. Hypercube Concurrent computers and applications*, pages 318-327, Jan. 1988.

[23] M. Gupta and P. Banerjee. Demonstration of automated data partitioning techniques in parallelizing compilers for distributed memory multiprocessors. *IEEE transactions on Parallel and Distributed Systems*, 3:179–193, Mar. 1992.

[24] J. P. Hayes. A microprocessor based hypercube supercomputer. *IEEE Micro*, II(4):7–19, Oct. 1986.

[25] M. T. Heath. *Hypercube Multiprocessors.* SIAM, Philadelphia, 1987.

[26] J. Hsu and P. Banerjee. Performance measurements and trace-driven simulation of parallel cad and numeric applications on hypercube multicomputers. *IEEE Transactions on Parallel and Distributed systems*, pages 451–464, July. 1992.

[27] B. W. Johnson. *Design and Analysis of Fault-Tolerant Digital systems.* Addisson-Wesley, 1989.

[28] J. E. Lee, E. Shragowitz, and S. Sahni. A hypercube algorithm for 0/1 knapsack problem. *Proc. Int. conf. on Parallel Processing*, pages 699–706, Aug. 1987.

[29] H. Vaidya Nitin. *Low Cost Schemes for Fault Tolerance (Modular redundancy, Multiprocessors).* Ph.D Dissertation, University of Massachussetts, 1993.

[30] S. Patil and P. Banerjee. a parallel branch and bound approach to test generation. *IEEE Transactions on Computer Aided design*, 9:313–322, mar. 1990.

[31] Michael Peercy and P. Banerjee. Distributed algorithms for shortest path deadlock free routing and broadcasting in arbitrarily faulty hypercubes. *Proc. 20th Int. Symp Fault-Tolerant Computing*, pages 218–225, June. 1990.

[32] A. Pothen et al. Orthogonal factorisation on a distributed memory multiprocessor. *Proc.2nd SIAM Conf. Hypercube Comput. and applications*, pages 587–596, 1987.

[33] A. L. N. Reddy et al. I/o emdedding in hypercubes. *Proc. Int. Conf on Parallel Processing*, pages 331–338, 1988.

[34] D. A. Rennels. On implementing fault tolerance in binary hypercubes. *Digest of Papers of the International Symposium on Fault-Tolerant Computing*, pages 344–349, 1986.

[35] Y. Saad and M. H. Schultz. Topological properties of hypercubes. *Yale University, Department of Computer Science, Research Report 389*, June. 1985.

[36] J. Sargent and P. Banerjee. A parallel row based algorithm for standard cell placement with integrated error control. *proc. 26th Design Automation Conf*, pages 590–594, June. 1989.

[37] Soh Seiteng. *Reliability Analysis of the hypercube architecture (Multiprocessor).* The Louisiana state University, 1994.

[38] C. L. Seitz. The cosmic cube. *Communications of the ACM*, 28(1):7–19, Jan. 1985.

[39] P. Sharma and El-Amawy Ahmed. Fault-tolerant hypercube networks with minimal redundancy. *Proc. 27th Asilomar Conference on Signals, Systems and Computers*, 2:1603–1607, 1993.

[40] D. P. Siewiorek and R. S. Swarz. *Reliable Computer Systems.* Digital Press, 1992.

[41] Adit. D. Singh. A reconfigurable modular fault tolerant binary tree architecture. *Digest of Papers of the International Symposium on Fault Tolerant Computing*, pages 298–304, 1987.

[42] Sultan and Melhem. An efficient modular spare allocation scheme and its application to fault-tolerant binary hypercubes. *IEEE Transactions on Parallel and Distributed Systems*, 2(1):117–126, Jan. 1991.

[43] N. F. Tzeng and S. Wei. Enhanced hypercubes. *IEEE Transactions on Computers*, 40:1362–1371, Oct. 1991.

[44] B. Wagar. Hyperquicksort:a fast sorting algorithm for hypercubes. *Hypercube Multiprocessors*, pages 292–299, 1987.

[45] Dajin Wang. Diagnosibility of enhanced hypercubes. *IEEE Transactions on Computers*, 43(9):1054–1061, Sep. 1994.

[46] Mao T. Wen and Yang C. B. Prefix computation on faulty hypercubes. *TENCON' 93*, pages 142–145, 1993.

[47] C. S. Yang et al. A reconfigurable modular fault-tolerant hypercube architecture. *IEEE Transactions on Parallel and Distributed Systems*, 5:1018–1032, Oct. 1994.

# Vita

- Abdul Hai Mohammed Abdulla
- Born in Bangalore, India on September 27, 1969.
- Received Bachelor's degree in Electronics and Communication Engineering from the Karnataka Regional Engineering College, Mangalore University, India in June, 1992.
- Worked as Production Engineer for Wipro Infotech, Bangalore, India (July 92-July 93).
- Joined King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia in August 1993.
- Completed Master's degree requirements at King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia in January, 1995.