NASA Technical Memorandum 4380

Generalized Hypercube Structures and Hyperswitch Communication Network

Steven D. Young

JUNE 1992

NASA Technical Memorandum 4380

Generalized Hypercube Structures and Hyperswitch Communication Network

Steven D. Young
Langley Research Center
Hampton, Virginia

National Aeronautics and
Space Administration
Office of Management
Scientific and Technical
Information Program
1992

Abstract

One of the Grand Challenges of the Federal High Performance Computing and Communications (HPCC) Program is in remote exploration and experimentation (REE). The goal of the REE Project is to develop a space-borne computing technology base that will enable the next generation of missions to explore the Earth and the Solar System. This paper discusses an ongoing study that uses a recent development in communication control technology to implement hybrid hypercube structures. These architectures are similar to binary hypercubes, but they also provide added connectivity between the processors. This added connectivity increases communication reliability while decreasing the latency of interprocessor message passing. Because these factors directly determine the speed that can be obtained by multiprocessor systems, these architectures are attractive for applications such as REE, where high performance and ultrareliability are required. This paper describes and enumerates these architectures and discusses how they can be implemented with a modified version of the hyperswitch communication network (HCN). The HCN is analyzed because it has three attractive features that enable these architectures to be effective: speed, fault tolerance, and the ability to pass multiple messages simultaneously through the same hyperswitch controller.

1. Introduction

One of the Grand Challenges of the Federal High Performance Computing and Communications (HPCC) Program is in the area of remote exploration and experimentation (REE). The goal of the REE Project is to develop a space-borne computing technology base that will enable high-performance, fault-tolerant, adaptive space systems for a new generation of missions to explore the Earth and the Solar System. The specific objectives of the REE Project are to demonstrate that a thousandfold increase in performance is feasible and to identify a parallel, scalable architecture that can incorporate new technologies to meet a broad range of requirements. As described in The Remote Exploration and Experimentation Project Plan by the Jet Propulsion Laboratory, the architecture must also provide affordable fault tolerance and long-term reliability in an environment of limited power and weight, high radiation, and no maintainability. To meet these objectives, new architectures must be investigated with consideration given to REE-type applications.

This paper discusses an ongoing study that attempts to use a recent development in hypercube communications control technology, the hyperswitch communication network (HCN) chip set (ref. 1), to implement a variety of generalized and hybrid hypercube architectures. These architectures are similar to binary hypercubes; but they also provide added connectivity between the processors. This added connectivity increases communication reliability while decreasing the latency incurred when passing messages between processors. Because these factors directly determine the speed that can be obtained with multiprocessor systems, these architectures are attractive for applications such as REE, where high performance and ultrareliability are required.

This paper describes and enumerates these architectures and discusses how they can be implemented with a modified version of the HCN chip set developed at the Jet Propulsion Laboratory. The HCN chip set is analyzed here because it has three attractive features that enable these architectures to be effective: speed, fault tolerance, and ability to pass multiple messages simultaneously through the same hyperswitch controller.

This paper is organized as follows. Section 2 describes generalized interconnection networks: both their organization and their relation to binary hypercube implementations. Expressions are given for the number of links, the number of disjoint paths between nodes, and other characteristic indices. Section 3 describes the hyperswitch communication network chip set: both its capabilities and its limitations. Section 4 describes and enumerates the possible generalized hypercubes that become feasible when hyperswitch technology is used in the network input/output (I/O) elements. Section 5 describes how the HCN chips can be modified to implement these architectures. Section 6 presents the benefits of these networks when used for multiple instruction multiple data (MIMD) architectures and how these networks can be used to increase system performance and reliability.

features: the ability to pass multiple messages simultaneously through the same hyperswitch (up to 11), the ability to reroute around busy channels and most importantly, the ability to reroute these messages quickly (less than 200 μsec for 512 byte messages).

The hyperswitch chip set (HSP) (fig. 5) consists of a custom hyperswitch (crossbar) element (HS), a hyperswitch I/O element (HSIO), and a message dispatch processor element (DP) (ref. 5). The HSP interfaces with other HSP's through 11 bidirectional channels (Ch0 to Ch10). These chips were designed specifically to provide fast dynamic circuit- and packet-switching capabilities in binary hypercube architectures.

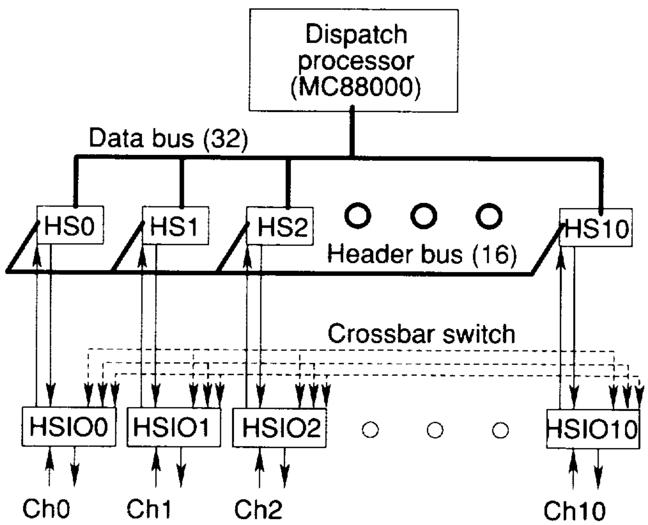

Figure 5. Hyperswitch processor.

In circuit-switching mode, the HSP establishes a path from source to destination before message transmission. This path is established by emitting a circuit probe (1 to 4 bytes) from the source node. The probe contains the destination node address, message length information, distance information, and some history information in case backtracking is required to establish the virtual link. The probe is then sent through intermediate nodes to the destination and the virtual link is established. At this time, the message itself can be transmitted across the virtual link at a rate equal to the link bandwidth. For circuit-switching mode, the message transmission latency T_{ckt} is

$$T_{\text{ckt}} = (S_{\text{probe}} H B_{\text{link}}) + (S_{\text{msg}} B_{\text{link}}) \tag{4}$$

where S_{probe} is the size of the probe, H is the number of hops in the virtual link, B_{link} is the bandwidth of the links, and S_{msg} is the size of the message.

In packet-switching mode, the HSP passes an entire message as a packet or set of packets, just as it passes a probe in circuit-switching mode. For packet-switching mode, the message transmission latency T_{pkt} is

$$T_{\text{pkt}} = S_{\text{pkt}} N H B_{\text{link}} \tag{5}$$

where S_{pkt} is the size of each packet, and N is the number of packets required to send the entire message.

In busy networks, both equations (4) and (5) must be appended to include the effects of encountering busy or failed links when establishing a path from source to destination. When a busy or failed link is encountered, one of three options is available: buffer the message until the link becomes available, drop the transaction and try again at a later time, or detour around the link. Each of these options increases the overall message latency.

Each HSP has 11 hyperswitch elements that act as the I/O ports for each node in the hypercube. Therefore, for binary hypercubes, the maximum number of nodes is 2^{11} (2048) because only one port is needed for each dimension. For nonbinary (e.g., generalized) hypercubes, a slightly different interpretation is discussed in section 4. For each hyperswitch, an HSIO performs the parallel-to-serial–serial-to-parallel conversion of the 8-bit data that travel between the hyperswitch and serial links that connect to neighboring HSP's (up to 11 serial links connect every node).

The DP is a Motorola MC88000 32-bit reduced instruction set computer (RISC), which can provide 17 million instructions per second. The DP performs transfers to and from system memory and acts as the interface between the HSP and the application processor. This processor also controls all crossbar settings in the hyperswitches of the HSP when establishing paths from source to destination during message transmission. The DP can act as the application processor as well.

Message routing latency is reduced with an adaptive backtracking algorithm implemented in the DP. This algorithm automatically avoids congested links based on its current knowledge of congestion in the network. When a message encounters a busy link, it does not wait for the link to become idle; instead, it tries to reach the destination by backtracking to the previous intermediate node and departing from another port. Virtual links between nodes are established by the switching elements in the HSP's of each node. This dynamic routing method has been shown to significantly reduce message routing overhead as well as increase the communication reliability because of the ability to backtrack and avoid busy or faulty network links (ref. 4).

4. Generalized Structures and the HCN

Using an HSP as the I/O controller at each node of a generalized hypercube architecture allows a wide variety of configurations to be implemented. As discussed previously, each HSP has 11 I/O ports that can be used to interconnect a number of processing sites. The chip set specification denotes that one of these ports should be used for diagnostic purposes; that is, it should be connected to itself and periodically have test data run through the port. The other 10 ports are then free to be interconnected to the HSP's of other nodes in the system.

Therefore, we can now calculate the number of possible generalized hypercube architectures that can be constructed with a maximum of 10 ports per node. This number equals the number of unique integer partitions of 10 as well as any integer less than 10. An integer partition of an integer r is the division of r into a number of integers whose sum is r. Thus, the list of generalized hypercubes that can be implemented with the hyperswitch can be represented by any set of integers whose sum is less than or equal to 10. For example, the partition $\{2, 2, 3, 3\}$ is an integer partition of 10. The corresponding four-dimensional generalized hypercube is a (3,3,4,4) configuration consisting of 144 nodes. The integers in the partition correspond to the number of ports required in each dimension.

From reference 6, the number of unique integer partitions of a number r is obtained from the coefficient of x^r in the following generating function:

$$G(x) = \prod_{m=1}^{\infty} \sum_{k=0}^{\infty} x^{km} \tag{6}$$

Specifically, for $r \leq 10$,

$$\begin{aligned} G(x) = & (1 + x + x^2 + ... + x^8 + x^9 + x^{10}) \\ & \times (1 + x^2 + x^4 + x^6 + x^8 + x^{10}) \\ & \times (1 + x^3 + x^6 + x^9)(1 + x^4 + x^8) \\ & \times (1 + x^5 + x^{10})(1 + x^6)(1 + x^7) \\ & \times (1 + x^8)(1 + x^9)(1 + x^{10}) \end{aligned} \tag{7}$$

or

$$\begin{aligned} G(x) = & 1 + x + 2x^2 + 3x^3 + 5x^4 + 7x^5 \\ & + 11x^6 + 15x^7 + 22x^8 + 30x^9 + 42x^{10} \end{aligned} \tag{8}$$

Where in equations (7) and (8), all terms with powers larger than 10 have been eliminated, because 10 is the maximum r we are interested in for this example. Furthermore, the generating function in equation (8) indicates the number of possible architectures with respect to the number of ports required per node (table 3). Finally, we can calculate the total number of generalized hypercube architectures possible by simply adding the coefficients of equation (8) as follows:

$$1 + 1 + 2 + 3 + 5 + 7 + 11 + 15 + 22 + 30 + 42 = 139$$

Table 3. Possible Generalized Hypercubes

Number of ports/node	0	1	2	3	4	5	6	7	8	9	10
Number of architectures	1	1	2	3	5	7	11	15	22	30	42

These architectures are listed in the appendix (with the exception of the trivial architecture that has 0 ports per node) and grouped according to the number of dimensions. The one-dimensional architectures in the appendix represent the fully connected systems that can be implemented. In addition to the list in the appendix, a large number of hyperrectangular and hybrid hypercubes can be constructed. Again, the only constraint imposed is the number of I/O ports required per node.

Architectures can now be chosen based on the characteristics of the application. For example, consider an application with three distinct distributed components: A, B, and C. Each component has increasing levels of communication bandwidth requirements. Choose a three-dimensional architecture with the processors in dimension 1 connected in a ring, processors in dimension 2 connected in a mesh, and processors in dimension 3 fully connected. Finally, map component A onto the processors in dimension 1, component B onto the processors in dimension 2, and component C onto the processors in dimension 3. Choosing the number of processors in each dimension now depends on the amount of parallelism inherent in the corresponding distributed components of the application.

5. Modifying HSP Element

To implement generalized hypercubes with the hyperswitch network element (fig. 5), two issues must be addressed. The first issue relates to the header information within the probes and message packets. The second issue requires changes in the coding of the DP as well as any hardwired functions pertaining to the architecture being configured (neighbor addresses) and the routing algorithm used.

Appendix

Generalized Hypercubes With the HCN

Tables A1 to A10 list the generalized hypercubes that can be implemented with a modified version of the hyperswitch communication network (HCN). Architectures are described by the generalized hypercube representation (which conveys the number of nodes in each dimension and the number of dimensions d), the number of I/O ports required for each node P, the number of bits required to represent the node addresses B_g, and the total number of nodes in the topology N.

Table A1. Ten-Dimensional Generalized Hypercubes

Configuration	P	B_g	N
2,2,2,2,2,2,2,2,2,2	10	10	1024

Table A2. Nine-Dimensional Generalized Hypercubes

Configuration	P	B_g	N
2,2,2,2,2,2,2,2,2	9	9	512
2,2,2,2,2,2,2,2,3	10	10	768

Table A3. Eight-Dimensional Generalized Hypercubes

Configuration	P	B_g	N
2,2,2,2,2,2,2,2	8	8	256
2,2,2,2,2,2,2,3	9	9	384
2,2,2,2,2,2,3,3	10	10	576
2,2,2,2,2,2,2,4	10	10	512

Table A4. Seven-Dimensional Generalized Hypercubes

Configuration	P	B_g	N
2,2,2,2,2,2,2	7	7	128
2,2,2,2,2,2,3	8	8	192
2,2,2,2,2,3,3	9	9	288
2,2,2,2,2,2,4	9	8	256
2,2,2,2,3,3,3	10	10	432
2,2,2,2,2,3,4	10	9	384
2,2,2,2,2,2,5	10	9	320

Table A5. Six-Dimensional Generalized Hypercubes

Configuration	P	B_g	N
2,2,2,2,2,2	6	6	64
2,2,2,2,2,3	7	7	96
2,2,2,2,3,3	8	8	144
2,2,2,2,2,4	8	7	128
2,2,2,3,3,3	9	9	216
2,2,2,2,3,4	9	8	192
2,2,2,2,2,5	9	8	160
2,2,3,3,3,3	10	10	324
2,2,2,3,3,4	10	9	288
2,2,2,2,4,4	10	8	256
2,2,2,2,3,5	10	9	240
2,2,2,2,2,6	10	8	192

Table A6. Five-Dimensional Generalized Hypercubes

Configuration	P	B_g	N
2,2,2,2,2	5	5	32
2,2,2,2,3	6	6	48
2,2,2,3,3	7	7	72
2,2,2,2,4	7	6	64
2,2,3,3,3	8	8	108
2,2,2,3,4	8	7	96
2,2,2,2,5	8	7	80
2,3,3,3,3	9	9	162
2,2,3,3,4	9	8	144
2,2,2,4,4	9	7	128
2,2,2,3,5	9	8	120
2,2,2,2,6	9	7	96
3,3,3,3,3	10	10	243
2,3,3,3,4	10	9	216
2,2,3,4,4	10	8	192
2,2,3,3,5	10	9	180
2,2,2,4,5	10	8	160
2,2,2,3,6	10	8	144
2,2,2,2,7	10	7	112

Table A7. Four-Dimensional Generalized Hypercubes

Configuration	P	B_g	N
2,2,2,2	4	4	16
2,2,2,3	5	5	24
2,2,3,3	6	6	36
2,2,2,4	6	5	32
2,3,3,3	7	7	54
2,2,3,4	7	6	48
2,2,2,5	7	6	40
3,3,3,3	8	8	81
2,3,3,4	8	7	72
2,2,4,4	8	6	64
2,2,3,5	8	7	60
2,2,4,5	9	7	80
2,2,3,6	9	7	72
2,2,2,7	9	6	56
3,3,4,4	10	8	144
3,3,3,5	10	9	135
2,4,4,4	10	7	128
2,3,4,5	10	8	120
2,3,3,6	10	8	108
2,2,5,5	10	8	100
2,2,4,6	10	7	96
2,2,3,7	10	7	84
2,2,2,8	10	6	64

References

1. Chow, E.; Madan, H.; Peterson, J.; Grunwald, D.; and Reed, D.: Hyperswitch Network for the Hypercube Computer. *The 15th Annual International Symposium on Computer Architecture,* IEEE Catalog No. 88CH2545-2, IEEE Computer Soc. Press, 1988, pp. 90–99.

2. Bhuyan, Laxmi N.; and Agrawal, Dharma P.: General Class of Processor Interconnection Strategies. *The 9th Annual Symposium on Computer Architecture,* IEEE Catalog No. 82CH1754-1, IEEE Computer Soc. Press, 1982, pp. 90–98.

3. Bhuyan, Laxmi N.; and Agrawal, Dharma P.: Generalized Hypercube and Hyperbus Structures for a Computer Network. *IEEE Trans. Comput.,* vol. C-33, no. 4, Apr. 1984, pp. 323–333.

4. Peterson, J.; Chow, E.; and Madan, H.: A High-Speed Message-Driven Communication Architecture. *Conference Proceedings—1988 International Conference on Supercomputing,* Assoc. for Computing Machinery, 1988, pp. 355–366.

5. *Hypercube Project—Hyperswitch Communication Network Chip Set.* JPL D-5956, California Inst. of Technology, May 1989.

6. Liu, C. L.: *Introduction to Combinatorial Mathematics.* McGraw-Hill, Inc., c.1968.

7. Young, Steven D.; and Yalamanchili, Sudhakar: Adaptive Routing in Generalized Hypercube Architectures. *Proceedings of the Third IEEE Symposium on Parallel and Distributed Processing,* IEEE Catalog No. 91TH0396-2, IEEE Computer Soc. Press, 1991, pp. 564–571.

REPORT DOCUMENTATION PAGE			*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1992	Technical Memorandum

4. TITLE AND SUBTITLE
Generalized Hypercube Structures and Hyperswitch Communication Network

5. FUNDING NUMBERS
WU 509-10-04

6. AUTHOR(S)
Steven D. Young

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, VA 23665-5225

8. PERFORMING ORGANIZATION REPORT NUMBER
L-16984

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA TM-4380

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified–Unlimited

Subject Category 62

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
One of the Grand Challenges of the Federal High Performance Computing and Communications (HPCC) Program is in remote exploration and experimentation (REE). The goal of the REE Project is to develop a space-borne computing technology base that will enable the next generation of missions to explore the Earth and the Solar System. This paper discusses an ongoing study that uses a recent development in communication control technology to implement hybrid hypercube structures. These architectures are similar to binary hypercubes, but they also provide added connectivity between the processors. This added connectivity increases communication reliability while decreasing the latency of interprocessor message passing. Because these factors directly determine the speed that can be obtained by multiprocessor systems, these architectures are attractive for applications such as REE, where high performance and ultrareliability are required. This paper describes and enumerates these architectures and discusses how they can be implemented with a modified version of the hyperswitch communication network (HCN). The HCN is analyzed because it has three attractive features that enable these architectures to be effective: speed, fault tolerance, and the ability to pass multiple messages simultaneously through the same hyperswitch controller.

14. SUBJECT TERMS
Hypercube; Hyperswitch, Integer partitions; Adaptive routing

15. NUMBER OF PAGES
13

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		