J. Indian Inst. Sci., Mar.–Apr. 1991, 71, 117–123
© Indian Institute of Science

Performance studies of a transputer-based extended hypercube

J. MOHAN KUMAR* AND L. M. PATNAIK*,†
Indian Institute of Science, Bangalore 560012, India.

Received on September 14, 1990; Revised on March 8, 1991.

Abstract

We discuss the implementation aspects and performance studies of a transputer-based extended hypercube whose design is aimed at reducing cost and increasing utilization factor. A comparative study of the performance of extended hypercube (EH) and hypercube in executing communication-intensive tasks is carried out.

Key words: Multiprocessors, hypercube, extended hypercube, transputers.

1. Introduction

Multiprocessor systems are characterized by two vital factors, *viz.*, i) the cost factor[1] defined as the product, (degree of node)* (diameter), and ii) the utilization factor[2] defined as the ratio of computation time to communication time. It is desirable to have a multiprocessor system with low cost and high utilization factor. In this paper, we discuss the implementation aspects and performance studies of a transputer-based extended hypercube (EH) whose design is aimed at reducing cost and increasing the utilization factor. EH topology has a low-cost factor[2] and is suitable for constructing communication-efficient parallel computing systems. EH comprises two sets of processors, one for performing computation tasks—processor elements(PEs), and another for performing communication tasks—network controllers (NCs)[3]. The basic module of EH consists of a k-cube of PEs and an NC. All the nodes of the k-cube are connected to NC by means of separate links. All communication tasks within any k-cube are performed by employing procedures similar to those of binary hypercubes[4]. NCs handle communication among all pairs of nodes residing in different modules. An EH can be constructed incrementally with identical predefined building blocks called the basic modules. In a basic module, PEs are at the zeroth level of hierarchy and the NC is at the first level of hierarchy. In general, an EH is defined by EH(k, l): where k is the dimension of the hypercube and l the hierarchical level of the NC at the top most level. The basic module can be defined as EH$(k, 1)$. An EH$(k, 2)$ has three levels of hierarchy:

* Microprocessor Applications Laboratory.
† Supercomputer Education and Research Centre and Department of Computer Science and Automation.

one NC at level 2; 2^k NCs at level 1 which form a k-cube among themselves; 2^k k-cubes at level 0. Likewise, one can build EHs of greater dimensions by interconnecting the basic blocks. The degree of a node of the EH is always a constant, for a given value of k, being equal to $(k+1)$ for PEs and (2^k+k+1) for the NCs.

IMS T800, a 32-bit transputer with a built-in floating point processor, is used as the PE. A T-800 transputer in conjunction with a programmable crossbar switch, IMS C004, is used as the NC. Reconfiguration of transputer networks can be done with the help of the crossbar switch and some low-level software support. We have used the 3-cube with an NC as the basic building block. For $k = 3$, the degree of a node for each PE is four, hence the four links of the transputer (PE) can be connected to the three neighbours and an NC.

2. The extended hypercube

In this section, a formal introduction to the EH architecture is presented. EH architecture discussed earlier[2,3] is suited for hierarchical expansion of multiprocessor systems. The basic module of EH consists of a k-cube and an additional node for handling communication, the network controller. There are a total of $[2^k \times (k/2+1)]$ links in the basic module, consisting of the $[(2^k \times k)/2]$ links of the hypercube and 2^k links between the processor elements and the NC as indicated in Fig. 1. Message passing between neighbouring nodes of the hypercube is *via* the direct communication links among them whereas between non-neighbouring nodes it is *via* the NC.

An EH consisting of a k-cube and one NC will be referred to as the basic module or EH$(k, 1)$[5] in the rest of this paper. EH$(k, 1)$ has two levels of hierarchy: the k-cube at the

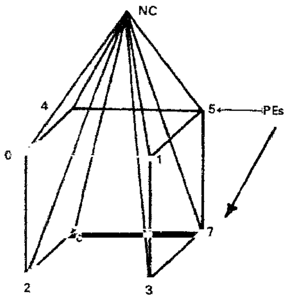

FIG. 1. Extended hypercube EH (3,1). NC: Network controller; PE: Processor element.

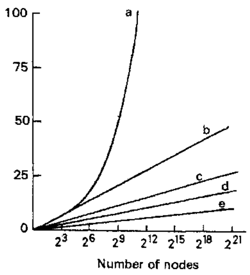

FIG. 2. Diameter, degree of node and cost factor of the hypercube and EH. a. cost factor of the cube, b. cost factor of EH (3,1), c. diameter and degree of hypercube, d. diameter of EH (3,1), e. degree of node of EH, (3,1).

zeroth level and the NC at the first level. The hypercube consisting of PEs is referred to as the HC$(k, 0)$. An EH$(k, 2)$ has 2^k k-cubes of PEs at the zeroth level, a k-cube of NCs at the first level and one NC at the second level. In general, an EH(k, l) l is the degree of the EH), consists of a k-cube of eight NCs/PEs at the $(l-1)$st level and one NC at the lth level. The NCs PEs at the $(l-1)$st level of hierarchy form a k-cube. We will refer to this cube as HC$(k, l-1)$. The NCs at the $(l-2)$nd level of hierarchy form 2^k distinct k-cubes which will be called HC$(k, l-2)$. Thus we have k-cubes at all levels. HC$(k, 0)$s are all computation HCs and HC(k, l)s for $(l > 0)$ are all communication HCs. The basic module of the EH denoted by EH$(k, 1)$ is a constant predefinable building block and the node configuration remains the same regardless of the dimension of the EH. The EH architecture can be easily extended by interconnecting the basic modules. We can interconnect eight 3-cubes (basic modules) to get a 64-node EH—an EH$(3, 2)$. The topology formed by the 3-cube of NCs at the first level of the EH$(3. 2)$ and the controller at the second level is identical to that of the basic module. Thus, we have a hierarchical structure consisting of 64 PEs at the zeroth level (lowest level), eight NCs at the first level and one NC at the second level

The cost factor given by the product, (degree of node)* (diameter) of an EH is given by[5]

$$[\{(k+1) \times 2^{k \times l}\} + (2^k + k + 1) \times \{(2^{k \times l} - 1)/(2^k - 1)\} \times (k + l)]/$$
$$[2^{k \times l} + \{(2^k + k + 1) \times (2^{k \times l} - 1)\}],$$

whereas for the hypercube of dimension n it is given by n^2. The variation of the cost factor for the binary hypercube and the EH of different dimensions is plotted in Fig. 2.

3. Implementation of EH

In a multitransputer system, a transputer is used at the processor element. In this section, we discuss about the implementation aspects of a multitransputer-based EH. The transputer has four communication channels[6] to facilitate communication among the processor elements of a multitransputer system. The degree of node or connectivity is defined as the number of connections at every node of the multitransputer system. The four links of the transputer of a processor node can be used to establish direct communication with only four of its immediate neighbours. In other words, the connectivity of a node of a multitransputer system is limited to four. However, this inadequacy can be overcome by adopting one of the following methods: (i) using more than one transputer per node, (ii) software multiplexing of the physical links to get more number of logical links, and (iii) using the link switch IMS C004 in conjunction with the transputer at every node.

In an EH(k, l) the degree of a node is given by $(k+1)$ for the PE and $(2^k + k + 1)$ for the NC. In an EH$(3, l)$ the degree of node is four for the PE and twelve for the NC. The four links of the transputer can be used to connect to the three neighbouring nodes and the NC. A transputer in conjunction with an IMS C004 switch is used to serve as the NC.

The IMS B008 is a TRAM-module motherboard[7] which plugs into the IBM PC-XT or PC-AT. TRAMs are board-level transputers that integrate processor, memory and peripheral functions and communicate with other transputers (or TRAMS) *via* serial links.

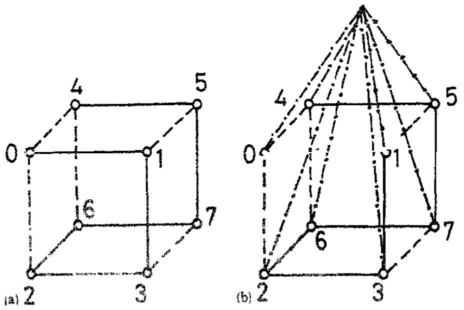

FIG. 3. Implementation of (a) hypercube (3-cube) and (b) EH(3, 1). ——— Direct link; ——— Indirect link *via* link switches; —·—·— Indirect link to NC.

The module motherboard has ten slots, *viz.*, slots 0–9, for inserting TRAM cards. Links 1 and 2 from each of the TRAM slots are hardwired to form a pipeline of connections among the TRAMS in slots 0 to 9. Link 3 of the TRAM in slot0 and links 1 and 3 of all other TRAMs are connected to the IMS C004 programmable link switch and these links can be softwired by appropriately programming the link switch. The link switch is controlled by the TRAM in slot0 *via* an IMS T212 16-bit transputer by sending the configuration data. A program written in Occam, Parallel Pascal or Parallel C developed under MS-DOS environment can control the TRAMs on the module motherboard by means of an interface with the IBM PC bus. The interface consists of a link adapter device IMS C012, which converts serial data into parallel form and *vice versa* to facilitate interface between the serial links of the transputer and the parallel data bus of the IBM PC bus. Link 0 of slot0 is connected to the link adapter and hence all communications from the PC are handled by the TRAM in slot0. Several IMS B008 boards can be cascaded to increase the capacity of the multitransputer system.

The link switch can be programmed to configure the TRAMs in various topologies. Links 0 and 3 of the TRAMs are softwired to arrange the nodes as a 3-cube or as an EH (3, 1), as indicated in Fig. 3. The TRAM of slot0 in conjunction with the link switch is used as the NC and the TRAMs in other slots are used as the PEs of the EH(3, 1). Link 3 of the NC is connected to the link switch which can be programmed to interconnect one of the PEs with the NC in a round-robin fashion.

4. Performance studies of the transputer-based EH

Transputer-based multiprocessor systems have been built in recent years. BBN butterfly[8], and Computing Surface[9] have been used for a variety of applications. Performance studies of transputer-based systems have been carried out in the past by several researchers. Dussa *et al*[10] have explored the feasibility of dynamic partitioning in multiprogrammed, multitransputer environments. Das and Fay[11] have carried out performance studies of multitransputer architectures with static and dynamic links. Efficient implementation of scientific programs is discussed by Hey and Pritchard[12]. In this section, we discuss the implementation of communication-intensive problems on a multitransputer systems. The performance of the EH and the hypercube in implementing the multinode broadcast and the total exchange[13] algorithms is compared. In the following, we assume that the PEs of the multitransputer system perform tasks which involve several iterations and to execute each iteration, the PEs have to carry out a multinode broadcast or total exchange. A computation state (R) is the time taken for completing one iteration involving a set of basic arithmetic/logic operations and a communication state (C) is the time taken to transmit a message from one PE to another. The multinode broadcast and total exchange problems are executed on the IMS B008 board configured as: (i) binary hypercube of dimension three, and as (ii) EH(3, 1). In executing the above problems the computation performed by each PE during every iteration involves simple arithmetic/logic instructions and the value of R is in the range 20–200 microseconds. The time taken to complete one message transfer operation on the INMOS links is C = 10–15 microseconds (includes all the overheads such as creating processes, filling link buffers and signal transfer rate on the links). We compare the performance of an n-cube with $2^n = \mathrm{N}$ nodes with that of an EH(k, l) with $2^{k \times l} = \mathrm{N}$ nodes.

4.1. *Multinode broadcast*

In a multinode broadcast problem every node in the system broadcasts a message to all other nodes. The EH efficiently makes use of the NCs in executing the mutltinode broadcast problem as discussed by Mohan Kumar *et al*[2]. The time spent by each node in executing a multinode broadcast is given as,

$$\mathrm{T}_{mb}(\text{hypercube}) = \mathrm{R} + \mathrm{C} \times (\mathrm{N} - 1);$$

$$\mathrm{T}_{mb}(\text{EH}) = \mathrm{R} + \mathrm{C} \times (k + 1).$$

Assuming that all PEs work in parallel, T_{mb} is also the time taken to complete the multinode broadcast in a hypercube, whereas in an EH the time taken to complete the multinode broadcast is given by $\mathrm{R} + \mathrm{C} \times (2^k)$. The time spent by each PE in executing communication and computation tasks and the utilization factor is given in Table I.

4.2. *Total exchange*

In a total exchange problem each node transmits a separate packet to every other node. The time spent by each node in executing the total exchange problem is given as,

$$\mathrm{T}_{te}(\text{hypercube}) = \mathrm{R} + \mathrm{C} \times (\mathrm{N} - 1) + \mathrm{C} \sum_{i=1}^{n} (i - 1) \times {}^{n}\mathrm{C}_i$$

$$\mathrm{T}_{te}(\text{EH}) = \mathrm{R} + \mathrm{C} \times (\mathrm{N} - 1) + \mathrm{Max}[\mathrm{C} \sum_{i=1}^{k} (i - 1) \times {}^{k}\mathrm{C}_i, \delta],$$

where δ is the switching delay.

Table I
Multinode broadcast and total exchange

Topology of the system	R = 64 ms; C = 10 ms		R = 128 ms; C = 15 ms		R = 64 ms; C = 5 ms	
	Communication (time in ms)	Utilization factor	Communication (time in ms)	Utilization factor	Communication (time in ms)	Utilization factor
Multinode broadcast						
Hypercube						
3-cube	23.33	2.74	35	3.65	35	1.82
6-cube	210	0.30	315	0.41	315	0.20
9-cube	1703.3	0.03	2555	0.05	2555	0.025
EH						
EH(3, 1)	10	6.4	15	8.5	15	4.25
EH(3, 2)	10	6.4	15	8.5	15	4.25
EH(3, 3)	10	6.4	15	8.5	15	4.25
Total exchange						
Hypercube						
3-cube	53.33	1.20	80	1.6	80	0.8
6-cube	690	0.09	1035	0.12	1035	0.06
9-cube	7293	0.008	10940	0.011	1940	0.005
EH						
EH(3, 1)	17.5	3.65	26.25	4.81	26.25	2.40
EH(3, 2)	210	0.30	315	0.41	315	0.20
EH(3, 3)	1703.3	0.03	2555	0.05	2555	0.025

The performance of the binary hypercube and the EH in executing the total exchange problem is given in Table I.

5. Conclusions

Implementation and performance of a transputer-based extended hypercube are discussed. The performance of the extended hypercube and the binary hypercube in executing communication-intensive problems is compared. The utilization factor of the EH is more than that of the hypercube in executing problems discussed in Section 4. A transputer-based EH is a cost-effective, communication-efficient multiprocessor system.

References

1. Bhuyan, L. N. and Agrawal, D. P. Generalized hypercube and hyperbus structures for a computer network, *IEEE Trans.*, 1984, **C-33**, 323–333.
2. Mohan Kumar, J., Patnaik, L. M. and Prasad Divya, K. A transputer-based extended hypercube, *Microprocessing Microprogramming*, 1990, **29**, 225–236.

3. JAGADISH, N., MOHAN KUMAR, J. AND PATNAIK, L. M. — An efficient scheme for interprocessor communication using dual ported RAMS, *IEEE Micro*, Oct. 1989, 10–19.

4. SAAD, Y. AND SCHULTZ, M. H. — *Topological properties of hypercube*, TR RR-462, Yale University, 1988.

5. MOHAN KUMAR, J. AND PATNAIK, L. M. — Extended hypercube: a hierarchical network of hypercubes, to be published in *IEEE Trans. Parallel Distributed Systems*.

6. *The transputer databook*, 1989, INMOS.

7. *IMS B008 User guide reference manual*, 1989, INMOS.

8. FELDMAN, J. A., FANTY, M. A., GODDARD, N. H. AND LYNNE, K. J. — Computing with structured connectionist networks, *Commun. ACM*, 1988, 31, 170–187.

9. FORREST, B. M., ROWETH, D., STROUD, N., WALLACE, D. AND WILSON, G. V. — Implementing neural network models on parallel computers, *Computer J.*, 1987, **30**, 413–419.

10. DUSSA, K., CARLSON, B., DOWDY, L. AND PARK, K-H. — Dynamic partitioning in a transputer environment, *Proc. ACM SIGMETRICS Conf. on Measurement and Modeling of Computer systems*, 1990, pp 203–213.

11. DAS, P. K. AND FAY, D. Q. M. — Performance studies of multi-transputer architecture with static and dynamic links, *Proc. Conf. Euromicro*, 1988.

12. HEY, A. J. G. AND PRITCHARD, D. J. — *Parallelism in scientific programming and its efficient implementation on transputer arrays*, University of Southampton, 1989.

13. BERTSEKAS, D. P. AND TSITSIKLIS, J. N. — *Parallel and distributed computation*, 1989, Prentice-Hall.