Deadlock Preventing Routing In Hypercycles

N. J. Dimopoulos, R. Sivakumar

Department of Electrical and Computer Engineering
University of Victoria
Victoria, B.C.
CANADA

SUMMARY Hypercycles is a class of multidimensional graphs obtained by allowing each dimension to incorporate more than two elements and a cyclic interconnection. Hypercycles, offer simple routing, and the ability, given a fixed degree, to chose among a number of alternative size graphs. These graphs can be used in the design of interconnection networks for distributed systems tailored specifically to the topology of a particular application. We present and prove a deadlock preventing routing strategy for a subset of hypercycles and a VLSI Hypercycle router component which implements the deadlock preventing routing.

KEYWORDS: Interconnection Networks, Routing, Deadlock Prevention, Routing Engine, Circuit Switching.

1. Introduction

Message passing concurrent computers such as the Hypercube [11], Cosmic Cube [16], MAX [12,13], iPSC [9] J-Machine [6] consist of several processing nodes that interact via messages exchanged over communication channels linking these nodes into one functional entity.

There are many ways of interconnecting the computational nodes, the Hypercube, iPSC and Cosmic Cube, have adopted a regular interconnection pattern corresponding to a binary n-dimensional cube, while MAX adopts an unstructured topology.

Hypercycles [7,13] are products [15] of circulants. The set of circulants used range in complexity from simple rings to fully connected graphs. Hypercycles are generalizations of many popular interconnection networks. Binary n-cubes, k-ary n-cubes [4], generalized hypercubes, rings, toruses, etc. are examples of hypercycles.

Many properties and algorithms used for example in routing and processor allocation can be extended to the entire class of hypercycles making it possible to choose a topology that best suits the system requirements of a specific class of applications.

A number of different routing policies have been introduced for a variety of interconnection networks. Thus in hypercubes e-cube routing [5] prevents deadlocks by ordering the resources (i.e. channels or virtual channels) comprising a path thus guaranteeing that no circular dependencies exist for any paths formed. Similar approaches have been devised for toruses and k-ary n-cubes [5, 10] where virtual channels are introduced in order to break any circular dependencies. A different approach of deadlock avoidance has been introduced with the Hyperswitch [3] and the various backtracking strategies for hypercycles [7]. There, deadlocks are avoided by forcing a forming path to backtrack and try again. These strategies appear to have excellent throughput because they tend to utilize more available paths leading to the destination, but they suffer from thrashing at high offered loads.

In this work, we present a deadlock preventing routing algorithm for certain hypercycles. This routing does not make use of virtual channels and thus it is well suited for circuit switching environments.

This work is divided into the following parts. Section 2. introduces the Hypercycles and discusses their properties. Section 3. presents a deadlock preventing routing strategy for a class of Hypercycles. Section 4 discusses the router component which has been designed and fabricated and finally we conclude with Section 5.

2. Introduction to Hypercycles

2.1 *Mixed Radix Number System*

The mixed radix representation [1], is a positional number representation, and it is a generalization of the standard b-base representation, in that it allows each position to follow its own base independently of the other.

Given a number M factored as $M = m_n \times m_{n-1} \times \cdots \times m_1$, any number $0 \leq X \leq M-1$ can be represented as $(X)_{m_n m_{n-1} \ldots m_1} = x_n x_{n-1} \ldots x_1 \Big|_{m_n m_{n-1} \ldots m_1}$ where $0 \leq x_i \leq (m_i - 1)$; $i = 1,2,\ldots,n$ and the x_i's are chosen so that $X = \sum_{i=1}^{n} x_i w_i$ and $w_i = \dfrac{M}{m_n m_{n-1} \ldots m_i}$

2.1 *Hypercycles.*

An n- dimensional *Hypercycle*, is a regular undirected graph: $\mathcal{G}_m^{\rho} = \{\mathcal{N}_m^{\rho}, \mathcal{E}_m^{\rho}\}$

where $\mathcal{N}^{\rho}_{m}$ is the set of nodes, $\mathcal{E}^{\rho}_{m}$ is the set of edges, $m = m_n, m_{n-1}, \cdots, m_1$ a mixed radix, $\rho = \rho_n, \rho_{n-1}, \cdots, \rho_1$; $\rho_i \leq m_i / 2$ the connectivity vector, determining the connectivity in each dimension ranging from a ring ($\rho_i = 1$) to fully connected ($\rho_i = \lfloor m_i / 2 \rfloor$), and $\mathcal{N}^{\rho}_{m} = \{0, 1, 2, \ldots, M-1\}$. Given $\alpha, \beta \in \mathcal{N}^{\rho}_{m}$ then $(\alpha, \beta) \in \mathcal{E}^{\rho}_{m}$ if and only if there exists $1 \leq j \leq n$ such that $\beta_j = (\alpha_j \pm \xi_j) \bmod m_j$ with $1 \leq \xi_j \leq \rho_j$ and $\alpha_i = \beta_i$; $i \neq j$

Hypercycles degree $d = \sum_{i=1}^{n} f(m_i, \rho_i)$ where $f(m_i, \rho_i) = \begin{cases} 2\rho_i & \text{if } 2\rho_i < m_i \\ m_i - 1 & \text{if } 2\rho_i = m_i \end{cases}$

and diameter $k = \sum_{i=1}^{n} \left\lceil \frac{\lfloor m_i / 2 \rfloor}{\rho_i} \right\rceil$ have been derived in [7].

The n-cube is a Hypercycle, with $M = 2 \times 2 \times \cdots \times 2 = 2^n$ and $\rho = 1, 1, 1, \ldots, 1$.

2.2 *Routing*

Hypercycles, have routing properties that are similar to those of the n-cube. Given nodes $(\alpha)_{m_n m_{n-1} \ldots m_1} = \alpha_n \alpha_{n-1} \ldots \alpha_i \ldots \alpha_1$ and $(\alpha^*)_{m_n m_{n-1} \ldots m_1} = \alpha_n \alpha_{n-1} \ldots \xi \ldots \alpha_1$ a walk, from α to α^*, is formed as follows: $\alpha_n \alpha_{n-1} \ldots \alpha_i \ldots \alpha_1, \alpha_n \alpha_{n-1} \ldots \xi_1 \ldots \alpha_1, \alpha_n \alpha_{n-1} \ldots \xi_2 \ldots \alpha_1, \ldots, \alpha_n \alpha_{n-1} \ldots \xi \ldots \alpha_1$.
such that

$$\xi_{j_i+1} = \begin{cases} (\xi_{j_i} + \rho_i) \bmod m_i & \text{if } (\xi - \xi_{j_i}) \bmod m_i = |\xi_{j_i}, \xi| > \rho_i \quad [a] \\ (\xi_{j_i} + |\xi_{j_i}, \xi| \bmod \rho_i) \bmod m_i & \text{if } (\xi - \xi_{j_i}) \bmod m_i = |\xi_{j_i}, \xi| > \rho_i \quad [b] \\ & \text{and } |\xi_{j_i}, \xi| \bmod \rho_i \neq 0 \\ (\xi_{j_i} - \rho_i) \bmod m_i & \text{if } (\xi_{j_i} - \xi) \bmod m_i = |\xi_{j_i}, \xi| > \rho_i \quad [c] \\ (\xi_{j_i} - |\xi_{j_i}, \xi| \bmod \rho_i) \bmod m_i & \text{if } (\xi_{j_i} - \xi) \bmod m_i = |\xi_{j_i}, \xi| > \rho_i \quad [d] \\ & \text{and } |\xi_{j_i}, \xi| \bmod \rho_i \neq 0 \\ \xi & \text{if } |\xi_{j_i}, \xi| \leq \rho_i \quad [e] \end{cases}$$

$$\xi_o = \alpha_i \qquad \xi_{max} = \xi \qquad [1]$$

Equation 1 defines all the minimum-length paths from a source to a destination in a single dimension. Parts (a), and (c) constitute a greedy strategy where the maximum step towards the destination is taken. Parts (b) and (d) form alternate paths by allowing the step described in part (e) to be taken earlier. Observe that there is only one step of length smaller than the maximum, and when it is taken it is guaranteed that the remaining steps will be maximal.

Given an origin $(\alpha)_{m_n m_{n-1} \ldots m_1} = \alpha_n \alpha_{n-1} \ldots \alpha_i \ldots \alpha_1$ and a destination $(\beta)_{m_n m_{n-1} \ldots m_1} = \beta_n \beta_{n-1} \ldots \beta_i \ldots \beta_1$ then distinct walks of minimum length that connect them are constructed by sequentially modifying the source address, each time substituting a source digit by an intermediate walk digit determined according to equation 1, until the destination is formed. The following walk connects **source** to **destination.**

$$\textbf{source} = \alpha_n \alpha_{n-1} \ldots \alpha_3 \alpha_2 \alpha_1,\ \alpha_n \alpha_{n-1} \ldots \alpha_3 \xi_1 \alpha_1,\ \alpha_n \alpha_{n-1} \ldots \psi_1 \xi_1 \alpha_1$$
$$\alpha_n \alpha_{n-1} \ldots \psi_2 \xi_1 \alpha_1,\ \ldots \alpha_n \alpha_{n-1} \ldots \beta_3 \xi_1 \alpha_1,\ \ldots \beta_n \beta_{n-1} \ldots \beta_3 \beta_2 \beta_1 = \textbf{destination}$$

If only the greedy strategy is followed, it results to a total of $l = \binom{q}{q_n, q_{n-1}, \ldots q_1} = \frac{q!}{q_n! q_{n-1}! \ldots q_1!}$ paths of minimum length that connect them with q_i being the distance along dimension i. The total distance between source and destination is given as $dis(a, b) = q = \sum_{i=1}^{n} q_i$. We call such paths *greedy paths.*

As an example, inFig. 1a both walks *01 11 10 20* and *01 00 10 20*.have the same minimum length and connect source *01* to destination *20.*

3. Deadlock-Preventing Routing in Hypercycles

In section 2.2 above, we presented a routing function that establishes at least one path of minimum length from a source to a destination node. In this part, we are concerned with optimally choosing one of the paths. Routing must be efficient and deadlock free.Deadlocks must be prevented, avoided or detected and broken. Deadlocks occur when resources (in this case node to node communication segments) are allocated so that the completion of a partial path requires a segment already allocated to a different partial path which in turn waits for a segment in the first partial path. It is obvious that no messages can propagate over the deadlocked paths, and the only remedy is to break the already established and deadlocked partial paths and try again.

Deadlock may occur easily in cases where the segments that form the paths are chosen at random. Certain routing algorithms prevent deadlocks by ordering the re-

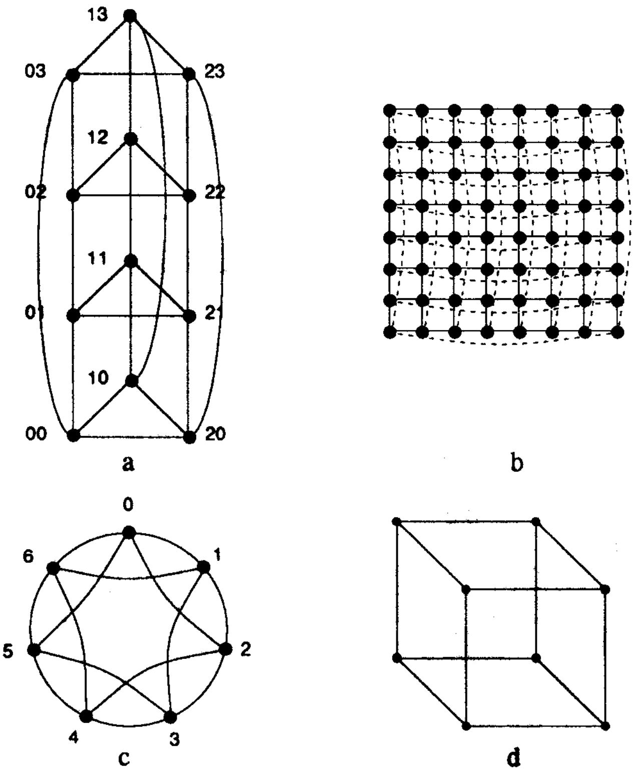

Figure 1. *Examples of Hypercycles*

a: Hypercycle $\mathcal{G}_{43}^{11}$ *b: ILIAC IV* $\mathcal{G}_{88}^{11}$

c: Circulant $\mathcal{G}_{7}^{2}$ *d: Hypercube* $\mathcal{G}_{222}^{111}$

sources (channels) to be allocated. Thus a lower order resource cannot be committed if a needed higher order resource cannot be obtained. It has been proven that the e-cube routing [5] prevents deadlocks in the case of the hypercube.

In this section, we introduce a deadlock-preventing routing for certain Hypercycles and prove that it indeed is deadlock-preventing.

Definition 1. Given a graph $\mathcal{G}$ on which a circuit switching routing is used, we denote by P_i the partially completed path between a source S_i and a destination D_i, by I_i the last node of the partial path P_i, and by L_i the set of all legal outgoing links from node I_i which can be used in order to complete the partial path to the destination D_i.

Observe that if $L_i = \emptyset$, then $I_i = D_i$ i.e. the path is complete, while if $P_i = \emptyset$, then $I_i = S_i$ and the path has not commenced yet.

Definition 2. With reference to a graph $\mathcal{G}$ on which circuit switching routing is used, we define the corresponding dependence graph $\mathcal{P} = \{\mathcal{N},\mathcal{E}\}$, where $\mathcal{N}$ is the set of partially completed paths, and $(P_i, P_j) \in \mathcal{E}$ *iff* $\exists\, l \in L_i$ such that $l \in P_j$.

Definition 3. We say that a set of source-destination pairs (S_i, D_i);i=1,2,...μ is deadlocked if $\forall l \in L_i \quad \exists j$ such that $l \in P_j; i = 1, 2, \ldots, \mu$.

Lemma 1. If a set of source-destination pairs (S_i, D_i);i=1,2,...μ is deadlocked, then the corresponding dependence graph contains at least one cycle.

Proof. If there are no cycles in the dependence graph, then it possesses at least one terminal node P. If P is a terminal node, there does not exist a succeeding partial path connected to P, i.e. $\forall l \in L \ \nexists j$ such that $l \in P$. But this contradicts definition 3.

Q.E.D.

In order to achieve deadlock preventing routing, we introduce asymmetry in the way each node routes messages. Take as an example the 4-node hypercycle $\mathcal{G}_4^1$ as depicted in Fig. 1. Node 2 can be reached from node 0 travelling either clockwise or counter clockwise. Similarly, node 3 can be reached from node 1 traversing the hypercycle in two directions. If the directionality is chosen at random, or identically for all nodes, deadlock can occur. If on the other hand, the directionality alternates, as we shall prove, deadlocks are prevented.

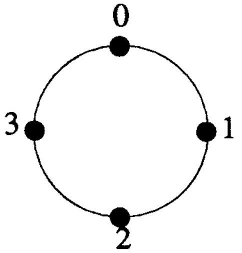

Figure 2. *4-node Hypercycle* $\mathcal{G}_4^1$

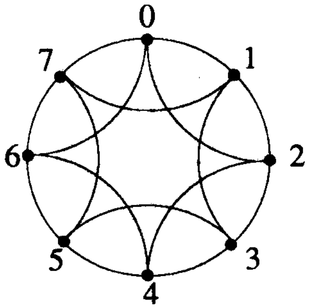

Figure 3. *8-node Hypercycle* $\mathcal{G}_8^2$

In a similar fashion, one can impose asymmetry in the routing for larger graphs such as the $\mathcal{G}_8^2$ as depicted in Fig. 2, where if nodes 0, 1, 4, and 5, route on a clockwise orientation and nodes 2,3,6 and 7 on a counterclockwise orientation, deadlocks do not occur.

We shall name this type of asymmetric routing the odd/even preference routing.

Definition 4. In one dimensional hypercycles $\mathcal{G}_m^{\rho}$ where a destination can be reached through two alternate routes i.e. when $(\alpha-\beta)\,\mathbf{mod}\,m = (\beta-\alpha)\,\mathbf{mod}\,m = |\alpha, \beta|$, where α is the source and β is the destination, we define the odd/even preference routing so as in the greedy mode, routes originating at source nodes with an even ρ-quotient proceed in the clockwise (counterclockwise) direction, while routes originating at nodes with an odd ρ-quotient, proceed in the opposite direction. If $\alpha*$ denotes the next intermediate node, then $\alpha^* = \begin{cases} (\alpha+\rho)\,\mathbf{mod}\,m \text{ if } \lfloor \alpha/\rho \rfloor \text{ is even} \\ (\alpha-\rho)\,\mathbf{mod}\,m \text{ if } \lfloor \alpha/\rho \rfloor \text{ is odd} \end{cases}$

Example 1. To illustrate the definitions stated above, we present the dependence graph for the odd/even routing in $\mathcal{G}_4^1$. To make the representation of the dependence graph legible, we have omitted the nodes corresponding to paths 012, 103 230 and 321. These nodes are terminal nodes and as such do not contribute to any possible loops. As it can be verified, this dependence graph is devoid of loops and thus the odd/even routing in $\mathcal{G}_4^1$ is deadlock free. If arbitrary routing was permitted, then edges such as the one depicted by the dashed line in the diagram would be permitted, giving rise to loops and therefore possible deadlocks during routing.

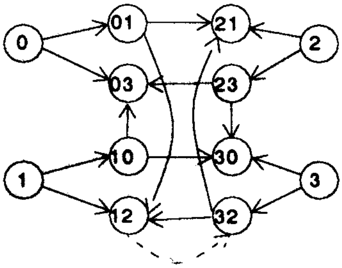

Figure 4. *Dependence graph for* $\mathcal{G}_4^1$

Lemma 2. In $\mathcal{G}_4^1$ with the odd/even preference routing, the corresponding dependence graph includes at most two distinct partial paths P_1 and P_2 with $P_1, P_2 \neq \emptyset$, and $L_1, L_2 \neq \emptyset$

Proof. Assume that there was a third node P_3, $P_3 \neq \emptyset$, $L_3 \neq \emptyset$, in the dependence graph. Observe that, because the hypercycle in question is the $\mathcal{G}_4^1$, each of the $P_i;i=1,2,3$, contains exactly one link, and all are different from each other. Let

$l_i = (\alpha_i, \beta_i) \in P_i$;i=1,2,3 .

Since there are only two even nodes in the hypercycle, assume without loss of generality that α_1 and α_2 are even while α_3 is odd. Then, because of the odd/even routing,

$$\beta_1 = (\alpha_1 + 1) \mathbf{mod}\, 4 \quad [2]$$

$$\beta_2 = (\alpha_2 + 1) \mathbf{mod}\, 4 \quad [3]$$

$$\beta_3 = (\alpha_3 - 1) \mathbf{mod}\, 4 \quad [4]$$

Since α_3 is assumed to be odd 4 above implies that β_3 is even. Thus, β_3 can only be either α_1 or α_2 . Assume, without loss of generality, that

$$\beta_3 = \alpha_1 \quad [5]$$

Then 4 becomes $\alpha_1 = (\alpha_3 - 1) \mathbf{mod}\, 4$ and substituting in 2 we obtain

$$\beta_1 = ((\alpha_3 - 1) \mathbf{mod}\, 4 + 1) \mathbf{mod}\, 4 = \alpha_3 \quad [6]$$

Thus 5, 6 imply that $l_3 \equiv l_1$ a contradiction, since the paths are assumed to be distinct. In a similar manner, one can prove that there cannot exist four distinct paths.

Q.E.D.

Theorem 1. In $\mathcal{G}_4^1$ the odd/even preference routing is deadlock preventing.

Proof By lemma 2 there are at most two partial paths P_i and P_j with $P_i, P_j \neq \emptyset$, and $L_i, L_j \neq \emptyset$ in the dependence graph. Then the only possible cycle comprises P_i and P_j.

Since $\mathcal{G}_4^1$ has diameter 2, the two partial paths P_i and P_j include one link each. Name these links λ_i and λ_j respectively. Obviously, $\lambda_i \neq \lambda_j$. Name by l_i and l_j the requested links for the continuation of the partial paths P_i and P_j, and suppose that P_i and P_j constitute a cycle. Then

$$l_j = \lambda_i \text{ and } l_i = \lambda_j. \quad [7]$$

Denote now the links explicitly as pairs of source and destination nodes as:

$\lambda_i = (\alpha_i, \beta_i)$, $\lambda_j = (\alpha_j, \beta_j)$ and

$l_i = (\beta_i, \gamma_i)$, $l_j = (\beta_j, \gamma_j)$

Then because of 7, $\beta_i = \beta_j = \beta$. Therefore,

$\lambda_i = (\alpha_i, \beta)$ $\lambda_j = (\alpha_j, \beta)$

Assume, without loss of generality, that β is odd. Then, one distinguishes two cases. (1) Both α_i and α_j are even. Then from the definition of the odd/even routing,

$\beta = (\alpha_i + 1)\mathbf{mod}\ 4$ and

$\beta = (\alpha_j + 1)\mathbf{mod}\ 4$.

Thus $(\alpha_i + 1)\mathbf{mod}\ 4 = (\alpha_j + 1)\mathbf{mod}\ 4. \Rightarrow (\alpha_i)\mathbf{mod}\ 4 = (\alpha_j)\mathbf{mod}\ 4. \Rightarrow \alpha_i = \alpha_j$, a contradiction.

(2) One of the α_i, α_j is odd, the other is even. Assume, without loss of generality, that α_i is odd and α_j is even. Then from the definition of the routing, we have

$\beta = (\alpha_i - 1)\mathbf{mod}\ 4$ and

$\beta = (\alpha_j + 1)\mathbf{mod}\ 4$

Thus $(\alpha_i - 1)\mathbf{mod}\ 4 = (\alpha_j + 1)\mathbf{mod}\ 4$. This implies that either

$\alpha_i - 1 = \alpha_j + 1 \Rightarrow \alpha_i = \alpha_j + 2$, a contradiction since α_i is odd and α_j is even, or

$$(\alpha_j + 1 - \alpha_i + 1) = 4k \quad \Rightarrow (\alpha_j - \alpha_i + 2) = 4k \qquad [8]$$

Since in $\overset{1}{\mathcal{G}_4}$, $|\alpha_j - \alpha_i| < 3$ then from 8 above, we have $|\alpha_j - \alpha_i| = 2 \Rightarrow \alpha_i = \alpha_j \pm 2$ a contradiction since α_i is odd and α_j is even.

Q.E.D.

In the subsequent treatment, we make use of the terms *maximum links* and *link length.* One–dimensional hypercycles can be visualized as rings with chords. Thus, any node will have links connecting it to its two immediate neighbors lying to its left and right (i.e. to nodes $(\alpha \pm 1)\mathbf{mod} m$) but also to more distant nodes through the chords (i.e. to nodes $(\alpha \pm x)\mathbf{mod} m$, $x \leq \rho$). Links therefore can be differentiated as to whether they correspond to the chords or not, as well as to the length of the chord.

Definition 5. The length of a link connecting nodes α and $(\alpha \pm x)\mathbf{mod} m$ in a one-dimensional hypercycle $\overset{\rho}{\mathcal{G}_m}$ is defined to be x. If $x = \rho$ then it is called a maximum link.

Theorem 2. One-dimensional hypercycles $\overset{\rho}{\mathcal{G}_m}$ of diameter two and $\lfloor m/2 \rfloor < 2\rho$ have greedy routing as outlined in section 3.1, which is deadlock preventing.

Proof Since $\lfloor m/2 \rfloor < 2\rho$ it means that any two-link path from a source to a destination, consists of a maximum link of length ρ followed by a link of length less than ρ. Thus, in the dependence graph, all the partially completed paths P_i such that $P_i \neq \emptyset$, and $L_i \neq \emptyset$ consist of maximum links while all the requested links in the sets L_i are not maximum links. Thus $\forall l_j \in L_j \quad \nexists \sigma$ such that $l_j \in P_\sigma$. Therefore, a cycle cannot exist in such a dependency graph, and the routing is deadlock preventing.

Q.E.D.

Theorem 3. One-dimensional hypercycles $\mathcal{G}_m^{\rho}$ with $m = 4\rho - k$; $k = 1,2,3$ are deadlock free under the greedy routing.

Proof Since $\lfloor m/2 \rfloor = \lfloor (4\rho - k)/2 \rfloor < 2\rho$ then theorem 2 applies.

Q.E.D.

Theorem 4. One dimensional Hypercycles $\mathcal{G}_m^{\rho}$ with $m = 4\rho$ have a deadlock preventing odd/even preference routing.

Proof Since $m = 4\rho$, one can number the nodes of this hypercycle as $\{0, 1,...,\rho - 1, \rho , \rho + 1,..., 2\rho -1, 2\rho , 2\rho +1,..., 4\rho -1\}$.

Partition now these nodes into ρ groups of four nodes each as follows.

$g_a = \{k\rho + a; k = 0, 1, 2, 3\}\ a = 0, 1, 2, \ldots, \rho - 1$

Observe that $\{k\rho + a + \rho\}\ \mathbf{mod}\, 4\rho = \{(k+1)\rho + a\}\ \mathbf{mod}\, 4\rho \in g_a$.

Thus, every node in each group g_a can be reached from any other node in the same group, with a path that consists entirely of nodes in g_a.

Therefore, for routing purposes, $\mathcal{G}_m^{\rho}$ can be partitioned in ρ groups, each of which is closed under the hypercycle routing, and each can be mapped onto $\mathcal{G}_4^1$ $(k\rho + a \leftrightarrow k)$, for which, it has been proven (Thm. 1) that the odd/even routing is deadlock preventing.

Q.E.D.

Theorem 5. Fully connected graphs are deadlock free.

Proof. Since the graph is fully connected, all partial paths between any source-destination pair have lengths of at most one. Therefore, the corresponding dependence graph is devoid of cycles since if (P_i, P_j) is an edge in the dependence graph, $P_i = \emptyset$, and thus, according to definition 2, there cannot be another edge of the form (P_x, P_i).

Q.E.D.

Theorem 6. Generalized e-cube routing is deadlock preventing on a graph that is the product of fully connected graphs.

Proof. Assume that there is a cycle present in the dependence graph. Name the sequence of partial paths which form the cycle as $P_i; i = 0,1,2,...k$, such that $P_i, L_i \neq \emptyset$. Since this sequence of partial paths forms a cycle, then

$\forall i \;\; \exists l_i \in L_i$ such that $l_i \in P_{(i+1) \bmod k}$. Observe now that because of the generalized e-cube routing, a link of a higher dimension cannot be allocated unless all the lower dimensioned links required have been allocated. Thus,$l_i \prec l_{(i+1) \bmod k}$;$i = 0, 1, \ldots, k$. But this is a contradiction because of the transitivity and strict ordering of the relation $\prec$.

Q.E.D.

We are ready now to define a deadlock preventing routing for multidimensional product graphs. In the generalized e-cube routing, a link cannot be reserved unless all the necessary links at higher dimensions have been allocated to the path forming.

Theorem 7. Generalized e-cube routing is deadlock preventing on a graph that is the product of graphs each of which possesses deadlock preventing routing.

Proof. In a similar manner as in Thm. 5,assume that there is a cycle in the corresponding dependence graph. Then, there will be a sequence of partial paths P_i;$i = 0,1,2,...k$, such that P_i,$L_i \neq \emptyset$ and $\forall i \;\; \exists l_i \in L_i$ such that $l_i \in P_{(i+1) \bmod k}$

Observe now that because of the generalized e-cube routing, one cannot allocate a link to a partial path unless all the required links at a lower or equal dimension have been allocated. Thus,.$l_i \preceq l_{(i+1) \bmod k}$;$i = 0, 1, \ldots, k$ This implies that all the requested links must lie at the same dimension. Thus, one can form a cycle in the dependence graph, consisting of portions of the partial paths relevant to this dimension. But this is not possible, since we assumed that each of the component graphs in the product graph possesses a deadlock preventing routing.

Q.E.D.

4. Router Implementation Status

A VLSI component implementing the deadlock preventing routing, as discussed above, has been designed and fabricated. Fig. 5 gives a block diagram of an n-dimensional Hypercycle router. It comprises two distinct parts. The Decoder and Next Port Generators and the Port Selector and Validator. The Decoders and Next Port Generators establish whether the deadlock preventing routing can be used on the Hypercycle configured. The next port generator implements the greedy and odd-even preference routing. The Port Selector and Validator selects the highest dimension if it is free, otherwise a *No_Ports_Available* is generated which will cause the routing to stop and wait until the required port is freed.

The router is programmable in that the Hypercycle Network on which the routing is performed is described through its mixed radix $\boldsymbol{m}$ and connectivity ρ vectors.

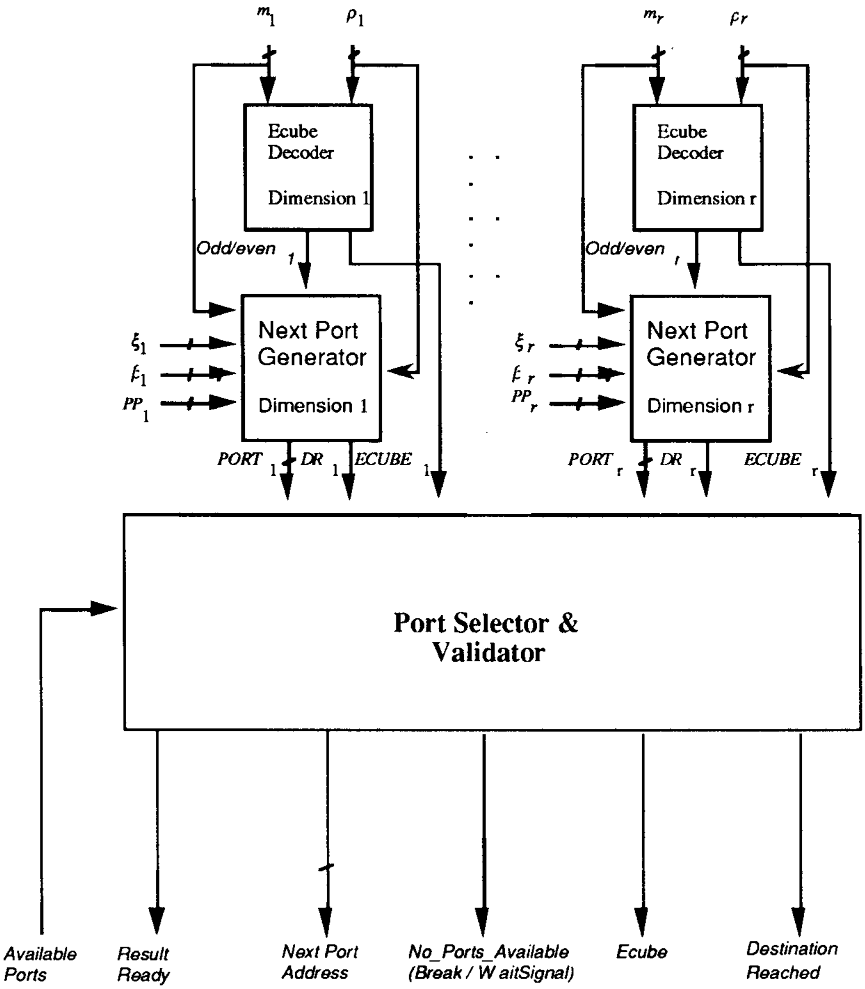

Figure 5. *Block Diagram of the Deadlock Preventing Router.*

These vectors together with the node address, available ports and destination address are stored in registers. The designed router is configurable for hypercycles of up to dimension 4, and degree 16. It incorporates in excess of 20000 transistors and it is designed to perform in excess of 16 million routing decisions a second. This router is part of a programmable routing engine for Hypercycles which will incorporate in addition to deadlock preventing routing deadlock avoiding routing and be capable of adopting the most suitable routing.

A micrograph of the designed deadlock preventing router is given in Fig. 6, while a deadlock avoiding router is discussed in [14]

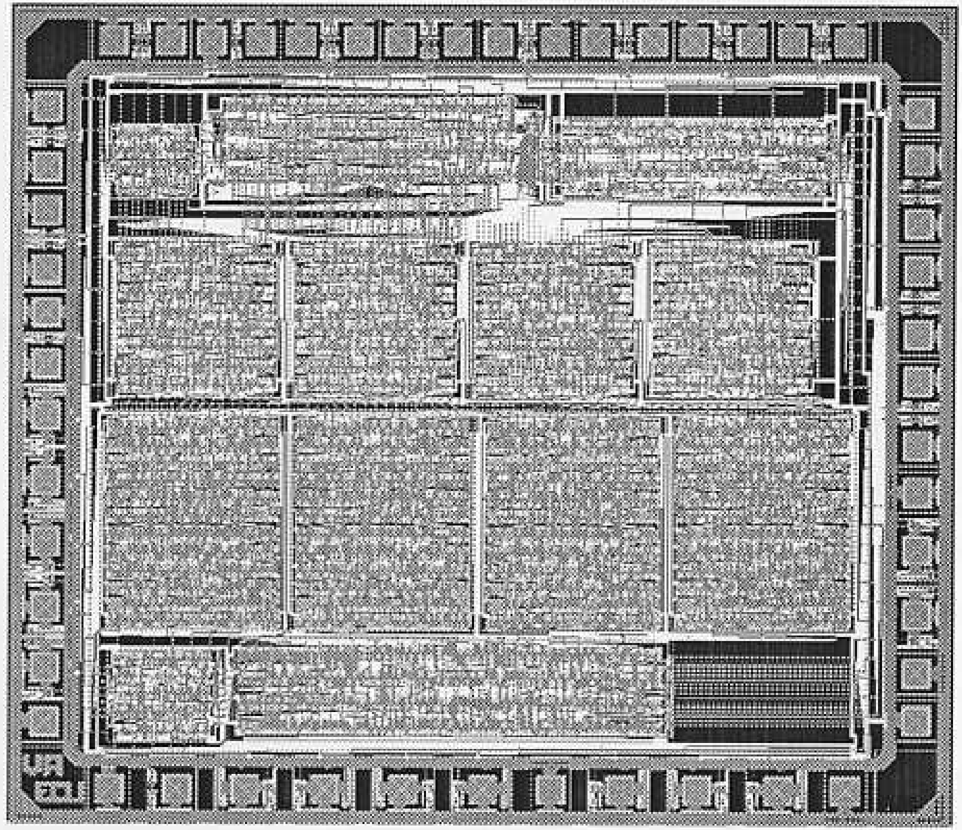

Figure 6. *Layout of the Deadlock Preventing Router in 1.2 μ NTCMOS4S technology*

5. Conclusions and Discussion

In this work, we presented the Hypercycles, a class of multidimensional graphs, which are essentially generalizations of several well known graphs including the n-cube, toruses, k-ary n-cubes, rings etc.

Although these graphs are not the densest possible, they are attractive, because of their simple routing. Since the node addresses are represented in a mixed radix as a sequence of n-digits, each one of these digits is processed independently and in parallel with the remaining digits. Thus the hardware involved in the routing can be made fast (because of the parallelism) and simple (since each module need only handle arithmetic **mod**m_i, as compared to arithmetic **mod**$m_1m_2...m_r$ needed when all the address digits are necessary as is the case with such networks as the chordal rings [8], or the cube connected cycles [2]).

We have established a deadlock preventing routing strategy for a subclass of the hypercycles and presented a deadlock preventing router which has been fabricated. We are currently developing a programmable routing engine for hypercycles which will incorporate a variety of routing strategies and be configured for a large class of hypercycle topologies.

6. Acknowledgement

This work has been supported by the Natural Sciences and Engineering Research Council Canada, under grant #OGP0001337, by the Institute for Robotics and Intelligent Systems under the National Networks of Centers of Excellence Program, and by the Canadian Microelectronics Corporation

7. Bibliography

1. Bhuyan, L. N., and D. P. Agrawal, "Design and Performance of Generalized Interconnection Networks" *IEEE Trans. Comput.* Vol. C-32, pp. 1081-1090, Dec. 1983.
2. Carlsson, G. E., J. E. Cruthirds, H. B. Sexton, and C. G. Wright "Interconnection Networks Based on a Generalization of Cube-Connected Cycles" *IEEE Trans. Comput.*, Vol. C-34, No. 8, pp. 769-772, Aug. 1985.
3. E. Chow, H. Madan, J. Peterson "A Real-Time Adaptive Message Routing Network for the Hypercube Computer" *Proceedings of the Real-Time Systems Symposium*, pp. 88-96, San Jose CA., 1987.
4. W. J. Dally "Performance Analysis of k-ary n-cube Interconnection Networks" *IEEE Trans. Comput.*, Vol. 39, no. 6, pp. 775-784, June 1990.
5. Dally, W.J., and C. L. Seitz "Deadlock-Free Message Routing in Multiprocessor Interconnection Networks" *IEEE Trans. Comput.* Vol. C-36, pp. 547-553, May 1987.

6. W. J. Dally, J. A. Stuart Fiske, J. S. Keen, R. A. Lethin, M. D. Noakes, P. R. Nuth, R. E. Davison, G. Fyler "The Message-Driven Processor: A Multicomputer Processing Node with Efficient Mechanisms" *IEEE Micro* pp. 23-40, April 1992.

7. Dimopoulos, N. J., D. Radvan, K.F. Li "Performance Evaluation of the Backtrack to the Origin and Retry Routing for Hypercycle based Interconnection Networks" *Proceedings of the Tenth International Conference on Distributed Systems*, Paris, pp. 278-284, 1990.

8. Imase, M., T. Soneoka, and K. Okada, "Connectivity of Regular Directed Graphs with Small Diameters" *IEEE Trans. Comput.*, Vol. C-34, pp. 267-273, Mar. 1985.

9. iPSC User's Guide, No. 17455-3, Intel Corp., Portland, Ore., 1985.

10. D. H. Linder and J. C. Harden "An Adaptive and Fault Tolerant Wormhole outing Strategy for k-ary n-cubes" *IEEE trans. Comput.* vol. 40, No. 1, pp. 2-12, Jan. 1991.

11. Peterson, J.C., J. O. Tuazon, D. Lieberman, M. Pniel "The MARK III Hypercube -Ensemble Concurrent Computer" *Proceedings of the 1985 International Conference on Parallel Processing* pp. 71-73, 1985.

12. Rasmussen, R. D., G. S. Bolotin, N. J. Dimopoulos, B. F. Lewis, and R. M. Manning "Advanced General Purpose Multicomputer for Space Applications" *Proceedings of the 1987 International Conference on Parallel Processing* pp. 54-57, 1987.

13. Rasmussen, R. D., N. J. Dimopoulos, G. S. Bolotin, B. F. Lewis, and R. M. Manning "MAX: Advanced General Purpose Real-Time Multicomputer for Space Applications" *Proceedings of the IEEE Real Time Systems Symposium* pp. 70-78, San Jose, CA.,1987.

14. R. Sivakumar, N. J. Dimopoulos, V. Dimakopoulos, M. Chowdhury, D. Radvan "Implementation of the Routing Engine for Hypercycle Based Interconnection Networks" *Proceedings of the 1991 Canadian Conference on Very Large Scale Integration* pp. 6.4.1-6.4.7, Kingston, ON. 1991.

15. G. Sabidussi "Graph Multiplication" *Math. Zeitschr.* Vol. 72, pp. 446-457, 1960.

16. Seitz, C. L. "The cosmic cube" *CACM* vol. 28, pp.22-33, Jan. 1985