Stack and Queue Layouts for Toruses and Extended Hypercubes

S. Bettayeb  
Department of Computer Science  
University of Houston-Clear Lake  
Houston, TX 77058, U.S.A  
Bettayeb@uchl.edu

L. Morales  
Department of Computer Science  
University of Houston-Clear Lake  
Houston, TX 77058, U.S.A  
MoralesL@uchl.edu

M. H. Heydari  
Department of Computer Science  
James Madison University  
Harrisonburg, VA 22807, U.S.A  
heydarmh@jmu.edu

I. H. Sudborough  
Department of Computer Science  
University of Texas at Dallas  
Dallas, TX 75080, U.S.A  
hal@utdallas.edu

Abstract

Linear layouts play an important role in many applications including networks and VLSI design. Stack and queue layouts are two important types of linear layouts. We consider the stack number, s(G), and queue number, q(G), for multidimensional k-ary hypercubes and toruses. Heath, Leighton, and Rosenberg showed that d-dimensional ternary hypercubes have stack number $\Omega(N^{d/9})$, with $N=3^d$ nodes. Malitz showed that $E$ edges implies stack number $O(\sqrt{E})$. For k-ary d-dimensional hypercubes, with $N = k^d$ vertices, Malitz's bound is $O(k^{d/2})$. We improve this to $2^{d+1}-3$. The $2^{d+1}-3$ bound holds for arbitrary d-dimensional toruses. The queue number of d-dimensional k-ary hypercubes or toruses is bounded by $O(d)$. Hence, Heath, Leighton, and Rosenberg exhibit an exponential tradeoff between $s(G)$ and $q(G)$ for multidimensional ternary hypercubes. Conversely, they conjectured that, for any $G$, $q(G)$ is $O(s(G))$. We present a family $\{H\}$ of modified multidimensional toruses and conjecture that $q(H)$ is not $O(s(H))$.

1. Introduction

Graph embeddings and linear layouts of graphs play an important role in a wide variety of applications. Stack layouts and queue layouts are two specific types of linear layouts that are useful in the study of VLSI design, routing and graph drawing, parallel processing and matrix computation, and permutation sorting [1,2,3,4,5,8,9,10,11,13,14]. In this paper, we give upper bounds on the stack number and queue number of k-dimensional toruses and hypercubes.

A stack layout of a graph, also known as a book embedding, is a linear layout of the vertices of a graph along the spine of a book and an assignment of edges to stacks or pages so that edges assigned to the same stack do not intersect. The minimum number of stacks in which a graph can be embedded is its stack number, denoted by $s(G)$. In the literature, the terms pagenumber and page embedding are sometimes used instead of stack number and stack embedding, respectively.

A queue layout of a graph is another type of linear layout of the vertices of a graph. In this case, the edges of the graph are assigned to queues in such a way that no queue contains a pair of nested edges. The minimum number of queues needed to embed a graph is called its queue number, denoted by $q(G)$.

We show that for any d-dimensional torus or extended hypercube the stack number is bounded by $2^{d+1}-3$. Our upper bound is of interest for multidimensional toruses and extended hypercubes in which some, or all, of the dimension sizes are odd integers. Heath, Leighton, and Rosenberg [6] have shown that a d-dimensional ternary hypercube has stack number $\Omega(3^{d/9})$. Their lower bound shows that the stack number, when each dimension has size 3, must be exponential in the number of dimensions. On the other hand, it is known that when the dimensions are all even integers, the stack number grows linearly with the number of dimensions. Thus, the stack number of d-dimensional toruses and extended hypercubes depends strongly on the parity of its dimensions. We show why this is true and describe a layout technique with sequential “corrections” of the order of vertices that mitigates the problem for the case when dimensions are odd. Basically, the technique modifies the standard layout of alternating left-to-right and right-to-left segments with an amortization of “corrections” that allows the reverse order to be realized without paying the penalty of all edges in the first-to-last “wraparound” connections to be simultaneously pair-wise crossing. So, the advantage of amortization is that the total number of stacks is...
substantially reduced. It is an intriguing open question to determine whether a better upper bound than $2^{d+1}$-3 and better lower bounds than provided by Heath, Leighton, and Rosenberg exist for multidimensional toruses or extended hypercubes with dimension sizes that are odd.

For $d$-dimensional toruses and extended hypercubes whose dimension sizes are both odd and even, the queue number is known to grow linearly with the number of dimensions. The parity of dimension sizes is seemingly not important for queue number. Heath, Leighton, and Rosenberg [6] considered stack number and queue number tradeoffs. As they indicate, the stack number of $d$-dimensional ternary hypercubes grows exponentially with $d$, while the queue number grows linearly with $d$. They conjecture that, conversely, families of graphs with large queue number and small stack number do not exist. Nonetheless, we will describe a family of $d$-dimensional modified hypercubes and we conjecture that the queue number does not grow linearly with the number of dimensions. As the stack number grows linearly with $d$, a proof of our conjecture represents an intriguing challenge, since it refutes the Heath, Leighton, and Rosenberg conjecture. The $d$-dimensional modified hypercube family represents the first family of graphs, at least that we know of, for which queue number is conceivably much larger than stack number.

In Section 2 we develop terminology useful for describing stack layouts and queue layouts of multidimensional toruses and $k$-ary hypercubes. In Sections 3 and 4, we present improved upper bounds on the stack number of these classes of graphs. In Section 5 we define a family $d$-dimensional modified $k$-ary hypercubes and argue that this family has stack number that grows linearly in the dimension, $d$, and queue number that grows exponentially in $d$. In Section 6 we summarize our results and discuss open problems.

2. Definitions and terminology

A linear layout, $L$, of a graph $G=(V,E)$ is a total order of the vertices of $G$. Let $i$, $j$, $x$, and $y$ be vertices in $G$ such that $L(i)<L(j)$ and $L(x)<L(y)$. Let $(i,j)$ and $(x,y)$ be edges in $G$. If either $L(x)<L(i)<L(j)<L(y)$ or $L(i)<L(x)<L(j)<L(y)$, then the edges $(i,j)$ and $(x,y)$ are said to intersect with respect to $L$. By contrast, if either $L(i)<L(x)<L(y)<L(j)$ or $L(x)<L(i)<L(j)<L(y)$, then the edges $(i,j)$ and $(x,y)$ are said to nest with respect to $L.

A stack embedding is a linear order of the vertices and an assignment of the edges to stacks or pages such that no page contains a pair of edges that intersect. A stack embedding is also called a book embedding. The stack number of a layout is the minimum number of pages required to embed the edges. The stack number of a graph, denoted by $s(G)$, is the minimum stack number over all layouts of the graph.

A queue embedding is a linear layout of the vertices of a graph and an assignment of the edges to queues such that no queue contains a pair of edges that nest. The queue number of a layout is the minimum number of queues required to embed the edges. The queue number of a graph, denoted by $q(G)$, is the minimum queue number over all layouts of the graph.

A $d$-dimensional torus is obtained from a $d$-dimensional mesh by adding wraparound edges between the first and last node in each row, and between the first and last node in each column. The notation $[n_1 \times n_2 \times \cdots \times n_d]$ denotes the $d$-dimensional torus with dimensions of size $n_1, n_2, \ldots, n_d$. For example, $[3 \times 2 \times 2]$ denotes the 3-dimensional torus in which the first dimension has size 3, and the second and third dimensions have size 2. The $d$-dimensional torus has nodes labeled with $d$-tuples from the set $Z_{n_1} \times Z_{n_2} \times \cdots \times Z_{n_d}$. We use the notation $<u_1,u_2,\ldots,u_d>$ to denote the $d$-tuples that label the nodes. When the context is clear, we omit the angle brackets $<$ and > from the node labels for the sake of readability. There is an edge between nodes $<u_1,u_2,\ldots,u_d>$ and $<v_1,v_2,\ldots,v_d>$ if there is an $i$ ($0 \leq i < d$) such that $u_i \equiv v_i \pm 1 \pmod{d}$, and for all $j$ ($0 \leq j < d$) such that $j \neq i$, $u_j \equiv v_j$.

For example, the three-dimensional torus $[3 \times 2 \times 2]$ has 12 nodes labeled 000, 001, 010, 011, 100, 101, 110, 111, 200, 201, 210, and 211. Nodes 110 and 210 are adjacent because the first coordinates differ by 1 and the other coordinates are the same. Figures 1 and 2 depict the $[3 \times 2 \times 2]$ and the $[4 \times 2]$ torus, respectively. (Due to formatting constraints, all figures appear at the end of this paper).

The binary hypercube of dimension $d$, denoted by $Q_2(d)$, has $2^d$ nodes, labeled by binary strings of $d$ bits. Two nodes, $i$ and $j$, are adjacent if their labels differ in exactly one bit position. Extending this definition, the $k$-ary hypercube of dimension $d$, denoted by $Q_k(d)$, has $k^d$ nodes, labeled with $d$-tuples from the set $Z_k \times Z_k \times \cdots \times Z_k$ ($d$ times). There is an edge between nodes $<u_1, u_2, \ldots, u_d>$ and $<v_1, v_2, \ldots, v_d>$ if there is an $i$ ($0 \leq i < d$) such that $u_i \equiv v_i \pm 1 \pmod{k}$, and for all $j$ ($0 \leq j < d$) such that $j \neq i$, $u_j \equiv v_j$, that is, the node labels differ by $\pm 1 \pmod{k}$ in exactly one position. Observe that the $k$-ary hypercube of dimension $d$ is isomorphic to the $d$-dimensional torus $[k \times k \times \cdots \times k]$ ($d$ times). $Q_3(3)$, the binary hypercube of dimension 3, and $Q_2(3)$, the ternary hypercube of dimension 3 are shown in Figure 3 and 4, respectively.
3. Stack layouts for multi-dimensional toruses and k-ary hypercubes where the size of each dimension is even.

We are interested in embeddings that minimize the stack number of multi-dimensional toruses and k-ary hypercubes. The design of such a stack embedding for a d-dimensional torus \( T = [n_1 \times n_2 \times \ldots \times n_d] \) is intrinsically dependent on the parity of the sizes of the dimensions \( n_1, n_2, \ldots, n_d \). In particular, if \( n_i \) is even, for all \( 1 \leq i \leq d \), then there is a book embedding of \( T \) with stack number \( 2d-1 \). For example, if \( T \) is the two-dimensional torus \([4 \times 6]\), then the book embedding with the nodes laid out by rows in alternating left-to-right and right-to-left order yields stack number 3. Specifically, one lays out the nodes \( a_{ij} \) of \( T \), for \( 0 \leq i \leq 3 \) and \( 0 \leq j \leq 5 \), in the order illustrated in Figure 5 at the end of this paper.

The purpose of alternating left-to-right order with right-to-left order for successive rows is to enable edges connecting corresponding vertices in adjacent rows to nest one within the other and, hence, to be placed within the same stack. Firstly, in this example, the edges connecting vertices \( (i,j) \) and \( (i+1,j) \), for any \( i \) \( (0 \leq i \leq 3) \) and any \( j \) \( (0 \leq j \leq 5) \) can all go into stack 1, because they form a cycle among contiguous groups of six vertices. Secondly, the edges connecting \( (i,j) \) and \( (i,j+1) \), for any even \( i \) \( (0 \leq i \leq 3) \) and any \( j \) \( (0 \leq j \leq 5) \) can all go into stack 2. This is due to the nesting of edges, which is a direct consequence of the alternation of left-to-right order and right-to-left order of rows of \( T \)’s vertices, as well as the disjointedness of the sets of such edges for different even integers. For example, the edge connecting \( (i,j) \) and \( (i,j) \) is inside the edge connecting \( (i,j) \) and \( (i,j) \) and the edge connecting \( (i,j) \) and \( (i,j) \) is inside the edge connecting \( (i,j) \) and \( (i,j) \). Thirdly, the edges connecting \( (i,j) \) and \( (i+1,j) \) \( (0 \leq i \leq 3) \) and any \( j \) \( (0 \leq j \leq 5) \) can all go into stack 3. This is again due to nesting. For example, the edge connecting \( (i,j) \) and \( (i,j) \) is nested inside the edge connecting \( (i,j) \) and \( (i,j) \).

To extend this to a torus of higher dimension, say \( T = [n_1 \times n_2 \times \ldots \times n_d] \), where \( n_i \) is even, for all \( 1 \leq i \leq d \), one can build a layout inductively. Start with a basic row-by-row layout \( L \) of the 2-dimensional torus \([n_1 \times n_2]\) similar to that described in Figure 5. For the inductive step, assume that \( L \) is a layout of the \((i-1)\)-dimensional \([n_1 \times n_2 \times \ldots \times n_{i-1}]\) torus and create a layout \( L' \) of the \(i\)-dimensional \([n_1 \times n_2 \times \ldots \times n_{i-1} \times n_i]\) torus, by creating \( n_i \) copies of \( L \) in the alternating order \( L^0 L \) \( \ldots \) \( L^R L \), where the \(j\)th copy of the layout \( L \) in \( L' \) (either in the form \( L \) or \( L^R \)) denotes the layout of all vertices with \(i\)th coordinate equal to \( j \). This is illustrated in Figure 6 with a layout of the 3-dimensional torus \([4 \times 4 \times 2]\).

Assume, by the inductive hypothesis, that \( L \) is a layout with stack number \( 2(d-1)-1 = 2d-3 \). Then, \( L' \) uses two more stacks than \( L \), so the stack number of \( L \) is \( 2d-1 \). That is, for even values of \( j \), a new stack is needed for all nested edges connecting the \(i\)th vertex of copy \(j\) and the \(i\)th vertex of copy \(j+1\) \((mod n_i)\), for each \( i \) \((0 \leq i \leq N-1)\), where \( N \) is the number of vertices of \([n_1 \times n_2 \times \ldots \times n_{d-1}]\), namely \( N = n_1 \cdot n_2 \cdot \ldots \cdot n_{d-1} \). And, for odd values of \( j \), a new stack is needed for all nested edges connecting the \(i\)th vertex of copy \(j\) and the \(i\)th vertex of copy \(j+1\) \((mod n_i)\), for each \( i \) \((0 \leq i \leq N-1)\). Since the number of additional stacks increases by 2 for each dimension, the total number of stacks for a \(d\)-dimensional torus \( T = [n_1 \times n_2 \times \ldots \times n_d] \), where every \( n_i \) is even, is \( 2d-1 \). For example, using this technique, the torus \([4 \times 4]\), which is isomorphic to \( Q(4) \), requires 3 stacks. Figure 5 gives the explicit layout, but not the edges, and Figure 12 shows the 3 stacks as 3 disjoint sets of colored edges. In Figure 12, the node labels have been abbreviated for readability.

4. Stack layouts for multi-dimensional toruses and k-ary hypercubes where the size of each dimension is odd.

The minimum stack number for a \(d\)-dimensional \( T = [n_1 \times n_2 \times \ldots \times n_d] \), where \( n_i \) is odd, for all \( i \) \((1 \leq i \leq d)\), is not exactly known. However, it is known that the \(d\)-dimensional torus, where every dimension is of size 3, called a ternary hypercube, has stack number \( \Omega(N^{1/9}) \) \([6]\), where \( N = 3^d \) is the number of nodes. Thus, there is an exponential lower bound \( \Omega(3^{d/9}) \) on the stack number for a \(d\)-dimensional ternary hypercube. An upper bound \([6]\) is based on the general result \([7]\) that a graph with \(E\) edges has stack number no larger than \(O(\sqrt{E})\). Using this we get, for example, an upper bound for the stack number of \(O(\sqrt{3})\) for the \(d\)-dimensional ternary hypercube where \( N = 3^d \) vertices and \( E = 3^{d+1} \) edges. For a \(k\)-ary \(d\)-dimensional hypercube, with \( N = k^d \) vertices and \( E = k^{d+1} \) edges, the corresponding upper bound on the stack number is \( O(k^{d+2}) \). In what follows we improve the upper bound to \( O(k^{d+1} - 3) \). That is, our upper bound is not dependent on \(k\), but only on the dimension, \(d\).

To illustrate the difficulty of making a book embedding with small stack number for a torus with dimensions that are all odd, consider the arrangement of the vertices of the two-dimensional 5-ary hypercube, or equivalently the \([5 \times 5]\) torus, shown in Figures 7 and 8 at the end of the paper.
This layout is similar to the one illustrated in Figures 5 and 12 for dimensions of even sizes. It lays out the torus by rows using alternating left-to-right and right-to-left orders. That is, if \( L \) is a list of the vertices in a row in left-to-right order, then the entire layout can be loosely expressed as \( L L^k L L^k L \). However, with an odd number of rows, the first and last rows are arranged in the same order. As a result, every distinct pair of wraparound edges, namely those connecting \((0)\) and \((k)\), for all \(i\) \((0 \leq i \leq 4)\), intersect and, hence, no two can be in the same stack. Thus the stack number of this layout is at least 5. Similarly, if one were to use a corresponding layout for the 2-dimensional torus \([m \times m]\), for any odd integer \(m\), the stack number would be at least \(m\). Moreover, for a \(d\)-dimensional \([k \times k \times k \times \ldots \times k]\), where \(k\) is odd, this type of layout would give stack number \(\Omega(k^d)\). In fact, as mentioned earlier, the \(d\)-dimensional ternary hypercube has stack number \(\Omega(3^{d/3})\) \([6]\), so there is an exponential jump in stack number going from \(d\)-dimensional \(k\)-ary hypercubes, when \(k\) is even, to \((k+1)\)-ary hypercubes, where the dimension sizes are all odd.

We give a book embedding strategy for a \(d\)-dimensional \(k\)-ary hypercube or a \(d\)-dimensional torus \([m \times m]\), where the sizes of the dimensions are odd, that gives stack number \(2^d - 3\). That is, the stack number is dependent on the dimension of the hypercube or torus, but not on the magnitudes of the dimensions. The basic row-by-row layout, which alternates between laying out a row in left-to-right order and laying out a row in right-to-left order, as illustrated in Figures 7 and 8, results in the problem that the first and last rows are laid out in the same order and hence all pairs of edges connecting corresponding vertices in the first and last rows intersect. The extension of this layout to multidimensional meshes was previously described and illustrated in Figure 16, using alternating left-to-right and right-to-left orders of copies of the same layout for the sub-torus of one less dimension.

We transform the alternating layout into a so-called \(M\)-layout by making a limited number of corrections per copy (where a “copy” refers to a row in a \(2\)-dimensional torus or a layout of a sub-torus of one less dimension in higher dimensions). The purpose of the corrections is to obtain a layout of the last copy that is the reversal of the order of the first copy, which will allow the connections between corresponding vertices in these copies to be nested and, hence, placed on one stack. This is illustrated in the \(M\)-layout of a \(2\)-dimensional \([5 \times 5]\) torus in Figures 9 and 10. Figure 9 gives the explicit layout, but not the edges, and Figure 10 shows the 4 stacks as 4 disjoint sets of colored edges. In Figure 10, the node labels have been abbreviated for readability.

Observe that in the \(M\)-layout of the 2-dimensional \([5 \times 5]\) torus in Figures 9 and 10 there is one correction per copy. That is, the order of the layout of the second row \(almost\) corresponds to the reversal of the order given in the layout of the first row, except that node \((0)\) has been moved in front of \((1)\) \((2)\) \((3)\). This is a single correction. Similarly, the layout of the third row \(almost\) corresponds to the reversal of the order given in the layout of the second row, except that node \((2)\) has been moved to a position between \((1)\) and \((3)\). This is another correction. Similarly, \((3)\) is corrected in the order of the fourth row and \((4)\) is corrected in the order of the fifth row, as illustrated. Note that 4 edges are nested, and, as there is one correction per copy, only one edge crosses the nested edges, namely, the edge connecting the corrected node to its corresponding node in the previous copy. Furthermore, as illustrated, the last copy (with the cumulative corrections) is in the reverse order of that given for the first copy, so all of the edges connecting corresponding nodes can nest and one stack can be used for all such edges.

Let us now give a general description of the \(M\)-layout. Let \(k\) be an arbitrary odd integer and suppose each row of a \(2\)-dimensional torus has \(k\) nodes, which we denote with the integers \(0, 1, \ldots, k-1\). We consider the \(M\)-layouts given in Figure 11.

Note that the order of nodes in layout \(L_i\) is the reverse of the order of the nodes in the layout \(L_0\), except that node 0 has been moved to the first position. In general, for \(i \geq 2\), the order of nodes in layout \(L_i\) is the reverse of the order of the nodes in layout \(L_{i-1}\), except that node \(i-1\) has been moved to a position next to node \(i-2\). That is, there is a reversal of the order except for one correction.

For all \(i\) \((0 \leq i \leq k-1)\), define \(\text{Next}(L_i) = L_{i+1 \text{ (mod } k)}\). Let \(L\) be any sequence of \((p-1)\)-tuples (i.e., node labels), for some positive integer \(p\). Then \(i \text{* } L\) denotes the sequence of \(p\)-tuples obtained by changing the \(j^{th}\) \((p-1)\)-tuple in \(L\), say \(<x_1, x_2, \ldots, x_p>\), into the \(j^{th}\) \(p\)-tuple \(<i, x_1, x_2, \ldots, x_p>\) of \(i \text{* } L\) by adding the integer \(i\) as a first coordinate. For any \(2\)-dimensional torus \(T\) with \(k\) rows, where \(k\) is odd, the \(M\)-layout is \(L(0) = 0 \text{* } L_0, 1 \text{* } L_1, 2 \text{* } L_2, \ldots, (k-1) \text{* } L_{k-1}\). For example, this is the layout given in Figures 9 and 10, for the \(2\)-dimensional torus, when \(k=5\). Note that \(L(0)\) is a layout (or ordering) of \(2\)-tuples (or pairs) which are arranged in \(k\) blocks, where \(2\)-tuples within a block have the same first coordinate. The block order in this case is given by one of the sequences of Figure 11, say \(L_i\), and the \(j^{th}\) block, for all \(j\) \((0 \leq j \leq k-1)\), is a sequence of \(2\)-tuples whose second coordinates are in one of the sequences of Figure 11, say \(L_j\).

For any such \(L\), define \(\text{Next}(L)\) to be the layout whose block order is changed from \(L_i\) to \(\text{Next}(L_i)\) and
each such block is changed so that its second coordinates, say in the order $L_j$, are modified to be in the order Next$(L_i)$. For example, for the layout given for the 2-dimensional torus $[5 \times 5]$ in Figures 9 and 10, where $L(0) = 0*L_0, 1*L_1, 2*L_2, 3*L_3, 4*L_4$, we have Next$(L(0)) = 0*L_1, 1*L_2, 2*L_3, 1*L_4$, and Next$^2$(L(0)) = Next(Next(L(0))) is $2*L_4, 3*L_0, 4*L_1, 1*L_3$, where $L_0$ is the M-layout of the (d-1)-dimensional k-ary hypercube. Therefore, if $2^{d-3}$ stacks suffice for the edges in the M-layout of the (d-1)-dimensional k-ary hypercube, $2^d - 3 = 2^{d+1} - 3$ stacks suffice for the edges in the M-layout of the d-dimensional k-ary hypercube.

For example, given the M-layout of the 2-dimensional torus $[3 \times 3]$ denoted by $L = 0*(<0,0>,<0,2>,<1,0>,<1,2>,<2,0>,<2,2>), 1*(<0,0>,<2,0>,<2,2>,<0,1>,<1,1>,<1,3>), 2*(<0,0>,<0,2>,<2,0>,<2,2>,<0,1>,<1,1>,<1,3>)$, the M-layout of the 3-dimensional torus $[3 \times 3 \times 3]$ is given by $L' = 0*L_0, 1*Next(L), 2*Next^2(L)$. For example, in this case, Next(L) = $(<0,0>,<0,2>,<2,0>,<0,1>,<1,1>,<1,3>)$ and Next$^2$(L) = $(<1,0>,<1,2>,<1,1>,<1,3>,<0,1>,<0,2>,<0,3>,<2,0>,<2,2>,<2,1>,<2,3>)$.

We now compute an upper bound on the number of stacks needed for our M-layout of a d-dimensional k-ary hypercube, where $k$ is odd. Firstly, observe that two stacks are sufficient for edges connecting a block of vertices to a block of vertices in the order given by $L_0$ and each block arranged in an M-layout of (p-2)-tuples. The M-layout $L'$ of $p$-tuples can be used for the edges connecting vertices in $1*Next(L)$ with vertices in $0*L$ and the edges connecting vertices in $1*Next(L)$ with vertices in $2*Next^2(L)$ intersect, one needs $2(2^{d-1}) = 2^d$ stacks. So, by repeating the use of stacks for subsequent connections, $2^d$ stacks suffice for all edges connecting blocks of the M-layout of the d-dimensional k-ary hypercube. Therefore, if $2^{d-3}$ stacks suffice for the edges in the M-layout of the (d-1)-dimensional k-ary hypercube, $2^d - 3 = 2^{d+1} - 3$ stacks suffice for the edges in the M-layout of the d-dimensional k-ary hypercube.

Theorem 1. For any positive integer $d$ and odd positive integer $k$, the stack number of the d-dimensional k-ary hypercube is at most $2^{d+1}$.

Corollary. For any positive integers $d$ and odd positive integer $k$, the d-dimensional $[k \times k \times k \ldots \times k]$ torus has stack number at most $2^{d+1}$.

The corollary follows immediately, as the d-dimensional $[k \times k \times k \ldots \times k]$ torus is isomorphic to the d-dimensional k-ary hypercube. In the following we extend the results based on M-layouts to include a torus with dimensions of different sizes.

A modification of our M-layout can be used for the case of a d-dimensional torus $T = [n_1 \times n_2 \times \ldots \times n_d]$, where each dimension size is odd and where not all dimension sizes are identical. The more general case, where dimension sizes are both even and odd, will also be discussed. Without loss of generality, assume the dimensions are listed in sorted order, so $n_1 \leq n_2 \leq \ldots \leq n_d$. We now give an inductive description of the layout. We start with the two dimensional torus $[n_1 \times n_2]$. This can be laid out row by row, where each row has $n_1$ nodes, and where the first $n_1$ rows forms the sub-torus $[n_1 \times n_1]$. Order the nodes of the first $n_1$ rows by the M-layout of the torus $[n_1 \times n_1]$. As we have seen the M-layout makes one correction per row and thereby creates an order for row 1 which is the reversal of row 1’s order. As both $n_1$ and $n_2$ are odd, $(n_2-n_1)$ is even and, therefore the remaining $(n_2-n_1)$ rows can be laid out in alternating left-to-right and right-to-left orders so that the order of row $n_2$ is also the reverse of the order of row 1. This allows the wraparound edges of the torus $[n_1 \times n_2]$, i.e., those that connect corresponding nodes of row 1 and row $n_2$, to nest and hence be placed into the same stack. The resulting layout of the $[n_1 \times n_2]$ torus has stack number $5 = 2^3 - 3$ just the same as our M-layout of the $[n_1 \times n_1]$ torus.
An extension of this idea works for the inductive step. That is, let L be a layout of the \([n_1 \times n_2 \times \cdots \times n_d]\) torus with stack number \(2^{k+1}-3\). We give a layout of the \([n_1 \times n_2 \times \cdots \times n_{k+1}]\) torus with stack number \(2^{k+2}-3\). Assume that \(2^k\) stacks are sufficient for edges connecting blocks in a layout L of the k-dimensional torus \([n_1 \times n_2 \times \cdots \times n_k]\). A layout \(L'\) of the \((k+1)\)-dimensional torus \([n_1 \times n_2 \times \cdots \times n_{k+1}]\) is \(0*L,1*Next(L), 2*Next^2(L), \ldots, (n_{k+1})*Next^t(L)\), \((n_{k+1})*Next^t(L)\), \((n_{k+1})*Next^t(L)\), \ldots, \((n_{k+1})*Next^t(L)\), \((n_{k+1})*Next^t(L)\), \((n_{k+1})*Next^t(L)\), where \(t=n_{k+1}-1\) and \(L_t=Next^t(L)\). As the edges connecting vertices in \(0*L\) with vertices in \(L'\) is \(0*L, 1*Next(L), 2*L, \ldots, n_{i}*(LR)\). Thus, the number of extra stacks needed for each even dimension is \(2^k\) where \(d\) is based on the dimensionality of the sub-torus. It follows that, if \(k\) of the dimensions \(n_1, n_2, \ldots, n_d\) are odd and \(k\) is odd, then the resulting stack number is \(2^{k+1}+2(d-k)\).

Theorem 3. The d-dimensional torus \([n_1 \times n_2 \times \cdots \times n_d]\) in which \(k\) of the dimensions are odd and \((d-k)\) are even has stack number \(2^{k+1}-3+2(d-k)\).

5. A conjectured tradeoff between stack number and queue number.

As mentioned earlier, Heath, Leighton, and Rosenberg [6] showed that the d-dimensional ternary hypercube has stack number \(O(3^d)\), but it has queue number \(O(d)\). So, there is an exponential tradeoff between stack number and queue number for some graphs. They also conjectured that there is no significant tradeoff in the other direction. That is, they conjecture that, for any graph G, \(q(G) = O(s(G))\), where \(q(G)\) and \(s(G)\).

Observe that the d-dimensional k-ary hypercube has queue number \(2d-1\) and that a layout with this queue number is obtained by keeping the ordering the nodes in each subcube the same. Such a layout is illustrated for the two dimensional 5-ary hypercube in Figure 13 at the end of the paper.

At least 3 queues are necessary for the layout shown in Figure 13, because, for example, the edge connecting \((2^1)\) and \((2^2)\) nests within the edge connecting \((2^1)\) and \((2^2)\) and this edge in turn nests within the edge connecting \((2^1)\) and \((2^2)\). Three queues are sufficient for the layout in Figure 13, as all of the edges connecting \((2^1)\) and \((2^2)\), for all \(i (0 \leq i \leq 4)\), can be placed in queue 1, of all of the edges connecting \((2^1)\) and \((2^2)\), for all \(i (0 \leq i \leq 4)\), and \((2^1)\) and \((2^2)\), can be placed in queue 2, and all of the edges connecting \((2^1)\) and \((2^2)\), for all \(i (0 \leq i \leq 3)\) and \((2^1)\) and \((2^2)\), can be placed in queue 3. Inductively, two additional queues are needed for each extra dimension, as all edges making a connection between successive blocks can be put into one new queue, and the edges making wraparound connections between the first and last blocks can be put into a second new queue.

Define the \(d\)-dimensional \(k\)-ary modified hypercube to be the graph with vertices in \(Z_k \times Z_k \times \cdots \times Z_k\) and edges as follows: (1) there is an edge between the vertex \((a_1, a_2, \ldots, a_k)\) and the vertex \((b_1, b_2, \ldots, b_k)\), for all \(a_i\) and \(b_i\) in \(\{0, 1, \ldots, k-2\}\), when there is exactly one coordinate index, say i, such that \(b_i=a_i+1\) and, such that, for all \(j (j \neq i)\), \(b_j=a_j\), and (2) there is an edge between the vertex \((a_1, a_2, \ldots, a_k)\), when at least one of the coordinates \(a_i\) is 0, and the vertex \((a_1, a_2, \ldots, a_k)\), where \(a_k\) denotes \(k-a_k\). An illustration of the 2-dimensional 5-ary modified hypercube is shown in Figure 14. Note that the 2-dimensional 5-ary modified hypercube is isomorphic to the \([5 \times 5]\) modified torus.

The stack number of the \(d\)-dimensional \(k\)-ary modified hypercube, when \(k\) is odd, is \(O(d)\), as the standard alternating left-to-right and left to right order of blocks yields nested edges from each block i to the subsequent block i+1 and, in addition, with an odd
number of blocks the last block is in the same order as the first block, which allows the edges described in part (2) of the definition to be nested. Such a layout is illustrated in Figure 15.

We conjecture that the queue number of the d-dimensional k-ary modified hypercube grows more rapidly than any linear function of the dimension d. Although we do not have a proof, one can observe that the standard queue layout of block after block in the same order creates the problem that all of the edge described in part (2) of the definition are nested and hence need to be placed on separate queues. One can, of course, modify the standard queue layout so that, say, one correction is made per block, in a manner similar to that described in Section 4 for stack layouts. However, for multidimensional k-ary modified hypercubes, the queue number of such a layout would still grow exponentially with the number of dimensions.

Conjecture: Let G be the d-dimensional k-ary modified hypercube. Then q(G) is not O(d).

6. Conclusions and Open Problems

Our primary result is that for any d-dimensional torus or extended hypercube the stack number is bounded by $2^{d+1} \cdot 3$. This upper bound is of interest for multidimensional toruses and extended hypercubes in which some, or all, of the dimension sizes are odd integers. Heath, Leighton, and Rosenberg [6] have shown that a d-dimensional ternary hypercube has stack number $\Omega(3^{n/3})$. This lower bound shows that the stack number, when the dimensions are all odd, must be exponential in the number of dimensions (at least for dimensions all of size 3). On the other hand, it is known that when the dimensions are all even integers, the stack number grows linearly with the number of dimensions. It is an intriguing open question to determine whether better upper and lower bounds exist for multidimensional toruses or extended hypercubes with dimension sizes that are odd.

For d-dimensional toruses and extended hypercubes of both odd and even dimension sizes, the queue number is known to grow linearly with the number of dimensions. The parity of dimension sizes is seemingly not important for queue number. Heath, Leighton, and Rosenberg [6] considered stack number and queue number tradeoffs. As indicated, the stack number of d-dimensional ternary hypercubes grows exponentially with d, while the queue number grows linearly with d. They conjectured that, conversely, families of graphs with large queue number and small stack number do not exist. Nonetheless, we describe a family of d-dimensional modified hypercubes and conjecture that the queue number of graphs in this family does not grow linearly with the number of dimensions. As the stack number grows linearly with d, a proof of our conjecture represents an intriguing open question. The d-dimensional modified hypercube family represents the first family of graphs that we know of, for which queue number is conceivably much larger than stack number.

7. References


8. Figures

Figure 1. The [3x2x2] torus

Figure 2. The [4x4] torus

Figure 3. The binary hypercube of dimension 3 (isomorphic to the [2x2x2] torus)

Figure 4. The ternary hypercube of dimension 3 (isomorphic to the [3x3x3] torus)

Figure 5. Layout of the torus [4x6] by rows with alternating left-to-right and right-to-left order.
Figure 6. A layout of the 3-dimensional [4×4×2] torus created by alternating left-to-right ordered and right-to-left ordered copies of the same layout of the 2-dimensional [4×4] torus.

Figure 7. A layout of the torus [5×5] by rows with alternating left-to-right and right-to-left order.

Figure 8. A stack layout of the [5x5] torus which permits 5 stacks

Figure 9. An M-layout for the 2-dimensional [5×5] torus which permits 4 stacks.

Figure 10. A stack layout of the [5x5] torus which permits 4 stacks

L₀:  0, 1, 2, 3, … , k-3, k-2, k-1
L₁:  0, k-1, k-2, k-3, … , 3, 2, 1
L₂:  2, 3, … , k-3, k-2, k-1, 1, 0
L₃:  0, 1, 2, k-1, k-2, k-3, … , 3
L₄:  0, 1, 2, 3, … , k-3, k-1, k-2
…
Lₖ₋₂: 0, 1, 2, 3, … , k-3, k-1, k-2
Lₖ₋₁: k-1, k-2, k-3, … , 3, 2, 1, 0

Figure 11. M-layouts L₀, L₁, L₂, L₃, … , Lₖ₋₂, Lₖ₋₁ of k vertices.
Figure 12. A stack layout of the [4x4] torus which permits 3 stacks

Figure 13. A layout of the 2-dimensional 5-ary hypercube with queue number 3.

Figure 14. The modified [5x5] torus

Figure 15. A stack layout of the modified [5x5] torus which permits 3 stacks

Figure 16. An M-layout of the 3-dimensional ternary hypercube.