1988

Embedding Complete Binary Trees into Butterfly Networks

Ajay K. Gupta

Susanne E. Hambrusch
Purdue University, seh@cs.purdue.edu

Report Number:
88-821

EMBEDDING COMPLETE BINARY TREES INTO BUTTERFLY NETWORKS

Ajay K. Gupta
Susanne E. Hambrusch

CSD-TR-821
November 1988
(Revised October 1989)

Embedding Complete Binary Trees into Butterfly Networks

Ajay K. Gupta [1]
Department of Computer Science
Western Michigan University
Kalamazoo, Michigan.

Susanne E. Hambrusch [2]
Department of Computer Sciences
Purdue University
West Lafayette, Indiana.

November 1988, revised October 1989

[1]Research supported in part by ONR under contract N00014-86-K-0689 and by NSF under Grant MIP-87-15652.

[2]Research supported in part by ONR under contracts N00014-84-K-0502 and N00014-86-K-0689, and by NSF under Grant MIP-87-15652.

Abstract

In this paper we present embeddings of complete binary trees into butterfly networks with or without wrap-around connections. Let m be an even integer and $q = m + \lfloor \log m \rfloor - 1$. We show how to embed a $2^{q+1} - 1$-node complete binary tree $T(q)$ into a $(m+1)2^{m+1}$-node wrap-around butterfly $B_w(m+1)$ with a dilation of 4, and how to embed $T(q)$ into a $(m+2)2^{m+2}$-node wrap-around butterfly $B_w(m+2)$ with an optimal dilation of 2. We also present an embedding of a wrap-around butterfly $B_w(m)$ into a $(m+1)2^m$-node no wrap-around butterfly $B(m)$ with a dilation of 3. Using this embedding we show that $T(q)$ can be embedded into a no wrap-around butterfly $B(m+1)$ (resp. $B(m+2)$) with a dilation of 8 (resp. 5).

Index Terms: Binary tree networks, butterfly networks, dilation, expansion, embeddings.

1 Introduction

Understanding and developing relationships between different interconnection networks is an important issue in parallel processing [2, 3, 4, 5, 6, 7, 8, 10, 12]. In this paper we study the relationship between a complete binary tree network T and a butterfly network B and present embeddings of T into B. Let $T(q)$ be a $2^{q+1}-1$-node complete binary tree, where $q = m + \lfloor \log m \rfloor - 1$ and m is an even integer. Let $B_w(m)$ (resp. $B(m)$) be a $m2^m$-node (resp. $(m+1)2^m$-node) butterfly with (resp. without) wrap-around connections. In [4], Bhatt et. al. showed how to embed $T(q)$ into $B_w(m+3)$ with a dilation of 4. While this embedding has constant expansion, it uses a butterfly that has 8 times[1] as many nodes as necessary. In this paper we first present two improved embeddings for $T(q)$. One embeds $T(q)$ into $B_w(m+1)$ with a dilation of 4. Another one embeds $T(q)$ into $B_w(m+2)$ with an optimal dilation of 2. These two embeddings and the one presented in [4], use the wrap-around connections of a butterfly heavily. We show that a butterfly with wrap-around connections is (up to a constant factor) no more powerful than one without wrap-around connections. More precisely, we show how to embed $B_w(m)$ into a no wrap-around butterfly network $B(m)$ with a dilation of 3. Using this embedding we are able to embed $T(q)$ into $B(m+1)$ with a dilation of 8 and into $B(m+2)$ with a dilation of 5. We next give definitions and notation used throughout this paper.

An embedding $< f, g >$ of T into B is defined by a bijective mapping f from the nodes of T to the nodes of B together with a mapping g that maps every edge $e = (v, w)$ of T onto a path $g(e)$ connecting $f(v)$ and $f(w)$. We refer to f as the assignment. Two commonly and extensively studied cost measures of an embedding are the dilation and the expansion [1, 4, 9, 11]. The *dilation* is defined as the maximum distance in B between two adjacent nodes in T, and the *expansion* is defined as the ratio of the number of nodes in B to the number of nodes in T.

Let $T(q)$ be a complete binary tree having $2^{q+1}-1$ nodes, where $q = m + \lfloor \log m \rfloor - 1$. The $q+1$ levels of $T(q)$ are numbered $0, 1, \ldots, q$. Let $B_w(m)$ be a butterfly network

[1] For simplicity we omit the lower order terms.

with wrap-around connections consisting of m levels, numbered $0, 1, \ldots, m-1$. Every level has 2^m nodes. Every node in $B_w(m)$ is identified by the pair $< l, \beta_0\beta_1 \ldots \beta_{m-1} >$, where l indicates the level the node resides in, and $\beta_0\beta_1 \ldots \beta_{m-1}$ is the PWL (Position Within Level) string, $\beta_i \in \{0, 1\}$ for $0 \leq i \leq m-1$. The nodes in $B_w(m)$ are connected as follows. Node $< l, \beta_0\beta_1 \ldots \beta_l \ldots \beta_{m-1} >$ is connected by a straight edge to node $< (l+1) mod\ m, \beta_0\beta_1 \ldots \beta_l \ldots \beta_{m-1} >$ and by a cross edge to node $< (l+1) mod\ m, \beta_0\beta_1 \ldots \bar{\beta}_l \ldots \beta_{m-1} >$. Note that the nodes on level $m-1$ are connected to nodes on level 0, representing the wrap-around edges. Let $B(m)$ be a butterfly with no wrap-around connections. The difference between $B(m)$ and $B_w(m)$ is that $B(m)$ contains an additional level, level m, and the wrap-around edges from level $m-1$ to level 0 in $B_w(m)$ are replaced by edges from level $m-1$ to level m.

In Section 2 we describe the embedding of tree $T(q)$ into butterfly $B_w(m+1)$ which has a dilation of 4. This embedding achieves the same dilation as the one in [4], but it has an expansion of 2, versus an expansion of 8. We note that it is straight forward to modify the embedding of [4] so that it uses a $B_w(m+2)$ instead of a $B_w(m+3)$. Section 3 contains the embedding of $T(q)$ into $B_w(m+2)$ which has a dilation of 2. With respect to dilation this embedding is optimal since no embedding with $O(1)$ expansion can achieve a dilation less than 2 [4]. Section 4 shows how to embed $B_w(m)$ into $B(m)$ with a dilation of 3 and how to use this result to obtain embeddings of $T(q)$ into butterflies that have no wrap-around connections.

The embeddings of Sections 2 and 3 use the same general approach as described in [4]. More precisely, we also use the notion of a *signatory* 1 and a *serial number* in the PWL strings. These concepts ensure that no two nodes of $T(q)$ are assigned to the same node of the butterfly. For our embeddings, as well as the embedding presented in [4], it is crucial that at some point nodes on the same level of $T(q)$ are assigned to different levels in the butterfly. This is achieved in [4] and by the embedding described in Section 2 by having subtrees grow upwards and then downwards in the butterfly for varying number of levels. Doing so results in a dilation of 4. Our embedding described in Section 3 uses a different technique to achieve assigning nodes on the same level of the tree to different levels of the butterfly and thus achieves a dilation of 2.

2 Embedding $T(q)$ into $B_w(m+1)$

In this section we describe how to embed $T(q)$ into $B_w(m+1)$ with a dilation of 4. The embedding can be viewed as "fine-tuning" the strategy used in [4]. In the embedding presented in [4], tree $T(q)$ is embedded into $B_w(m+3)$ in three stages. The first stage embeds levels 0 to $\log m - 1$ of $T(q)$, the second stage embeds levels $\log m$ to $\frac{m}{2} + \log m - 1$, and the third stage embeds the remaining levels of $T(q)$. In order to avoid collisions among the embeddings of the three stages, each stage is simply embedded in a different copy of $B_w(m+1)$ in $B_w(m+3)$. Our embedding uses bit positions in such a way that the different stages of the embedding can share sub-butterfly networks as much as possible.

Let B_1 be the left half of $B_w(m+1)$ and B_2 be the right half of $B_w(m+1)$, as shown in Figure 2. For clarity reasons, henceforth we refer to the nodes of $T(q)$ as PEs (Processing Elements) and to the nodes of the butterfly as nodes. We embed $T(q)$ into $B_w(m+1)$ in four stages. *Stage* 1 embeds the first $\log m$ levels of $T(q)$ with a dilation of 2 so that the PEs on level $\log m - 1$ of $T(q)$ are assigned to nodes on level $m-1$ of $B_w(m+1)$. Stage 1 uses only the odd-numbered levels in B_1. Let $T_k(m)$ be the k^{th} tree rooted at a PE on level $\log m - 1$ of $T(q)$, $0 \le k \le 2^{\log m - 1} - 1 = m/2 - 1$. See Figure 1. *Stage* 2 embeds the first $m/2 + 1$ levels of $T_k(m)$ with a dilation of 4. The PE on level 0 of $T_k(m)$ remains assigned to the same node as in Stage 1. The PEs on level $m/2$ of $T_k(m)$ are assigned to nodes on level m of B_2. Stage 2 uses the odd-numbered levels in B_1, and in addition, the even level $m-2$ in B_1 and the even level m in B_2. The last $m/2$ levels of $T_k(m)$, are embedded in two stages. In *Stage* 3 we assign the next k levels of $T_k(m)$ with a dilation of 4 so that the PEs on level $m/2 + k$ are assigned to nodes on level $m-1$ of B_2. Stage 3 uses only the odd-numbered levels in B_2. *Stage* 4 assigns the remaining $m/2 - k$ levels of $T_k(m)$ with a dilation of 2. Stage 4 uses even-numbered levels in both B_1 and B_2. The use of the levels is summarized in Table 1. Our embedding assigns no two PEs to the same node of $B_w(m+1)$ and two adjacent PEs are at most distance 4 apart.

	Levels used in B_1	Levels used in B_2
Stage 1	odd-numbered	—
Stage 2	odd-numbered and level $m-2$	level m
Stage 3	—	odd-numbered
Stage 4	even-numbered	even-numbered

Table 1: Use of levels by the four stages

For the description of the embedding given next assume that m is a power of two. Our result holds, however, for any even m and we describe the necessary changes at the end of Section 2.

A *unary-straight* branching is a branching from a node on level l to a node on level $l+1$ using the straight edge. A *unary-cross* branching is a branching from a node on level l to a node on level $l+1$ using the cross edge. Finally, a *binary* branching is a branching from a node on level l to two nodes on level $l+1$ using the straight and the cross edge.

Stage 1: *Embedding the first* $\log m$ *levels of* $T(q)$

Assign the root of $T(q)$ to node $< m - 2\log m + 1, (01)^{m/2-\log m+1}0^{2\log m-1} >$[2]. We then use a unary-straight and a binary branching to assign the two PEs on level 1 of $T(q)$ to the two nodes $< m - 2\log m + 3, (01)^{m/2-\log m+1} * 0^{2\log m-2} >$[3]. Note that the binary branching generates a $*$ in the PWL string and the unary-straight branching keeps a 0 or a 1 as it is. Continuing in this fashion, we assign the 2^i PEs on level i of $T(q)$ to nodes $< m - 2\log m + 2i + 1, (01)^{m/2-\log m+1}(*0)^i 0^{2\log m-2i-1} >$, for $2 \leq i \leq \log m - 1$. Thus, the $m/2$ PEs on level $\log m - 1$ of $T(q)$ are assigned to nodes $< m - 1, (01)^{m/2-\log m+1}(*0)^{\log m-1}0 >$. In Figure 2, we show first few steps of the embedding in this stage. It is easy to see that no two PEs are assigned to the same node, i.e, there are no collisions.

[2] x^α denotes α occurrences of x when $\alpha > 0$ and no occurrence of x when $\alpha \leq 0$.

[3] $*$ denotes a wild card character indicating 0 and 1.

Stage 2: *Embedding levels* $\log m$ *to* $m/2 + \log m - 1$ *of* $T(q)$

As already defined, $T_k(m)$ is the k^{th} tree rooted at a PE on level $\log m - 1$ of $T(q)$, for $0 \leq k \leq 2^{\log m - 1} - 1 = m/2 - 1$. Stage 2 assigns PEs on levels 1 through $m/2$ of $T_k(m)$. The PEs on levels 1 through $m/2 - 2$ are assigned to odd-numbered levels in B_1. The PEs on levels $m/2 - 1$ and $m/2$ are assigned to the even-numbered levels $m - 2$ and m in B_1 and B_2, respectively. Levels $m/2 - 1$ and $m/2$ of $T_k(m)$ are treated differently in order to keep the dilation at 4 when embedding level $m/2 + 1$ of $T_k(m)$.

The goal of Stage 2 is to assign the PEs on level $m/2$ of $T_k(m)$ to nodes $< m, (*0)^{m/2-k-1} * 1(*0)^k 1 >$. The '1' in position $m - 2k - 1$ (i.e., β_{m-2k-1}) in the PWL strings is called the *signatory 1* [4]. Its purpose is to keep the nodes which have PEs of $T_k(m)$ assigned to be different from the nodes that have PEs of $T_j(m)$ assigned, $j \neq k, 0 \leq j, k \leq m/2 - 1$. Let the leftmost $m - 2\log m + 2$ bits (resp. rightmost $2 \log m - 1$ bits) of a PWL string be the *head* (resp. *tail*) of the PWL string. Consider a node that did get a PE assigned to it in Stage 1. Its head consists of alternating 0's and 1's and its tail consists of alternating $*$'s and 0's, with an additional 0 at the rightmost position. For the trees $T_k(m)$ with $k \geq \log m - 1$, the signatory 1 is positioned in the head of the PWL string (observe that the position for the signatory 1 already contains a '1'). For the trees $T_k(m)$ with $k < \log m - 1$ the signatory 1 is positioned in the tail of the PWL strings (at the beginning of Stage 2 this position contains a '0'). We first describe the embedding when the signatory 1 is positioned in the head.

Stage 1 assigned the root of $T_k(m)$ to node $< m - 1, (01)^{m/2 - \log m + 1}(*_k 0)^{\log m - 1} 0 >$[4]. Using two unary-straight branchings and a binary branching, we assign the two PEs on level 1 of $T_k(m)$ to the two nodes $< 1, *1(01)^{m/2 - \log m}(*_k 0)^{\log m - 1} 0 >$. This results in a dilation of 3 between the PEs of level 0 and level 1 of $T_k(m)$. For $2 \leq l \leq m/2 - \log m + 2$, we assign the 2^l PEs on level l of $T_k(m)$ to 2^l nodes on level $2l - 1$ of B_1 using a unary-cross and a binary branching. An exception occurs for the PEs on level $m/2 - k + 1$ of $T_k(m)$ where, in order to keep the signatory 1, we use a unary-straight instead of a

[4] In the string $(*_k 0)^{\log m - 1} = *_k 0 *_k 0 \ldots *_k 0$, the string $*_k *_k \ldots *_k$ denotes the binary representation of k. The $*_k$ in the r^{th} position from left in $*_k *_k \ldots *_k$ is the $*_k$ in position $2r$ of $*_k 0 *_k 0 \ldots *_k 0$, for $0 \leq r \leq \log m - 2$.

Level of $T_k(m)$	Nodes of $B_w(m+1)$
$l = 0$	$< m-1, (01)^{\frac{m}{2}-\log m+1}(*_k 0)^{\log m-1} 0 >$
$l \le \frac{m}{2} - k$	$< 2l-1, (*0)^{l-1} * 1(01)^{\frac{m}{2}-\log m-l+1}(*_k 0)^{\log m-1} 0 >$
$l \le \frac{m}{2} - \log m + 1$	$< 2l-1, (*0)^{\frac{m}{2}-k-1} * 1(*0)^{l-\frac{m}{2}+k-1} * 1(01)^{\frac{m}{2}-\log m-l+1}(*_k 0)^{\log m-1} 0 >$
$l \le \frac{m}{2} - 2$	$< 2l-1, (*0)^{\frac{m}{2}-k-1} * 1(*0)^{k+l-\frac{m}{2}}(*_k 0)^{\frac{m}{2}-l} 0 >$
$l = \frac{m}{2} - 1$	$< m-2, (*0)^{\frac{m}{2}-k-1} * 1(*0)^{k-1} *_k 00 >$
$l = \frac{m}{2}$	$< m, (*0)^{\frac{m}{2}-k-1} * 1(*0)^k 1 >$

Table 2: Assignments of Stage 2 in Section 2 for $k \ge \log m - 1$.

unary-cross branching. Figure 3 shows these first few steps. After having handled the bits in the head of the PWL string, we now use unary-straight (which keeps the 0's) and binary branchings to assign further levels. After having placed the PEs on level $m/2 - 2$ to nodes in level $m - 5$ of B_1, we change the strategy for the remaining two levels as follows. A unary-straight, a binary, and a unary-straight branching assigns the PEs on level $m/2 - 1$ to nodes on level $m - 2$ of B_1. Finally, a binary, a unary-straight, a unary-cross (to level 0 in B_2), and a unary-straight branching (to level m in B_2) assigns the PEs on level $m/2$ of $T_k(m)$ to nodes on level m of B_2. This strategy results in a dilation of 2 between levels l and $l+1$ of $T_k(m)$, for $1 \le l \le m/2 - 3$, and a dilation of 3 (resp. 4) between levels $m/2 - 2$ (resp. $m/2 - 1$) and $m/2 - 1$ (resp. $m/2$).

Table 2 shows the assignment of the PEs of $T_k(m)$ for $k \ge \log m - 1$. The first line of the table states the position of the root. The second line covers the levels up to the point the position of the signatory 1 is encountered. Line 3 covers the remaining position in the head. Line 4 covers the tail, excluding last two levels that are handled differently as given in lines 5 and 6. Obviously, not all lines are meaningful in all situations. When $k = \log m - 1$, the signatory 1 corresponds to the last position in the head and line 3 is thus meaningless. For $m = 4$, lines 2, 3, and 4 are meaningless. In this case the PEs on level 1 of $T_1(4)$ are assigned to nodes on level 2 (using line 5) and the PEs on level 2 of $T_1(4)$ are assigned to nodes on level 4 (using line 6). The branchings done are somewhat different, but the dilation is at most 4.

We next consider the case when the signatory 1 is positioned in the tail. The overall strategy is as before. Table 3 shows the assignments for $1 \leq k < \log m - 1$ and Table 4 the ones for $k = 0$. We briefly discuss Table 3. Line 1 states the position of the root and line 2 covers the assignments of PEs for which bits in the head of the PWL string are changed. Line 3 covers the tail positions up to just before a '0' is changed into a signatory 1. Line 4 covers the remaining assignments up to and including level $m - 5$ of B_1. The assignments for the last two levels are stated in lines 5 and 6. Again, not all lines are meaningful. When $k = 2$, line 3 only covers levels up to level $\frac{m}{2} - 3$. When $k = 1$, line 4 is meaningless (since there are no more bits to be set before the two final levels) and line 3 covers levels up to level $\frac{m}{2} - 2$. Table 4 shows the assignments for tree $T_0(m)$. In order to be consistent with Tables 2 and 3, we show $*_0$'s in Table 4. However, every $*_0$ corresponds to a 0. The situation for tree $T_0(m)$ when $m = 4$ is similar to the situation described for tree $T_1(4)$ whose assignments are given in Table 2. The PEs on level 1 of $T_0(4)$ are assigned to nodes on level 2 (using line 4 of Table 4) and the PEs on level 2 of $T_0(4)$ are assigned to nodes on level 4 (using line 5).

We next show that there are no collisions within Stage 2. Obviously, if a node gets two PEs assigned, they must come from different $T_k(m)$'s. Let x_k and x_j be two PEs from $T_k(m)$ and $T_j(m)$, respectively, that are assigned to nodes on level r. Let y_k and y_j be the PWL strings of these nodes, respectively. We show that $y_k \neq y_j$. Assume, without loss of generality, that $j < k$. We consider two cases depending on the value of k.

Case 1: $k \geq \log m - 1$.

Case 1.1: $1 \leq r \leq m - 2k - 1$. The PWL strings y_k and y_j differ in the tail. PWL string y_k has $(*_k0)^{\log m - 1}0$ as the rightmost $2\log m - 1$ bits, while y_j has $(*_j0)^{\log m - 1}0$ as the rightmost $2 \log m - 1$ bits.

Case 1.2: $m - 2k + 1 \leq r \leq m$. PWL strings y_k and y_j differ because of the signatory 1. The PWL string y_k has a 1 in position $m - 2k - 1$, while the PWL string y_j has a 0 in that position.

Case 2: $k < \log m - 1$.

Case 2.1: $1 \leq r \leq m - 2\log m + 1$. This case is similar to Case 1.1.

Level of $T_k(m)$	Nodes of $B_w(m+1)$
$l = 0$	$< m-1, (01)^{\frac{m}{2}-\log m+1}(*_k 0)^{\log m-1}0 >$
$l \leq \frac{m}{2} - \log m + 1$	$< 2l-1, (*0)^{l-1} * 1(01)^{\frac{m}{2}-\log m-l+1}(*_k 0)^{\log m-1}0 >$
$l \leq \frac{m}{2} - k$	$< 2l-1, (*0)^{l}(*_k 0)^{\frac{m}{2}-l}0 >$
$l \leq \frac{m}{2} - 2$	$< 2l-1, (*0)^{\frac{m}{2}-k-1} * 1(*0)^{k+l-\frac{m}{2}}(*_k 0)^{\frac{m}{2}-l}0 >$
$l = \frac{m}{2} - 1$	$< m-2, (*0)^{\frac{m}{2}-k-1} * 1(*0)^{k-1} *_k 00 >$
$l = \frac{m}{2}$	$< m, (*0)^{\frac{m}{2}-k-1} * 1(*0)^{k}1 >$

Table 3: Assignments of Stage 2 in Section 2 for $1 \leq k < \log m - 1$.

Level of $T_0(m)$	Nodes of $B_w(m+1)$
$l = 0$	$< m-1, (01)^{\frac{m}{2}-\log m+1}(*_0 0)^{\log m-1}0 >$
$l \leq \frac{m}{2} - \log m + 1$	$< 2l-1, (*0)^{l-1} * 1(01)^{\frac{m}{2}-\log m-l+1}(*_0 0)^{\log m-1}0 >$
$l \leq \frac{m}{2} - 2$	$< 2l-1, (*0)^{l}(*_0 0)^{\frac{m}{2}-l}0 >$
$l = \frac{m}{2} - 1$	$< m-2, (*0)^{\frac{m}{2}-1} *_0 00 >$
$l = \frac{m}{2}$	$< m, (*0)^{\frac{m}{2}-1} * 11 >$

Table 4: Assignments of Stage 2 in Section 2 for $k = 0$.

Case 2.2: $m - 2\log m + 3 \leq r \leq m - 2k - 1$. The PWL strings y_k and y_j are $(*0)^{\frac{r+1}{2}}(*_k0)^{\frac{m}{2}-\frac{r+1}{2}}0$ and $(*0)^{\frac{r+1}{2}}(*_j0)^{\frac{m}{2}-\frac{r+1}{2}}0$, respectively. Since $j < k < \log m - 1$, y_k and y_j differ in at least one of the rightmost $\lfloor \log k \rfloor$ bits.

Case 2.3: $m - 2k + 1 \leq r \leq m$. The PWL string y_k has a 1 in position $m - 2k - 1$, while the same position in the PWL string y_j has a 0.

We complete the discussion of Stage 2 by showing that there are no collisions between Stage 1 and Stage 2. For $m = 4$ there cannot be any collisions since $T_0(4)$ and $T_1(4)$ use only even-numbered levels in Stage 2. Hence, assume $m > 4$. Let r be a level in B_1 that has PEs assigned both from Stage 1 and Stage 2. Since Stage 1 places the root at level $m - 2\log m + 1$ and level $m - 5$ is the largest odd-numbered level used by Stage 2, we have $m - 2\log m + 1 \leq r \leq m - 5$. For $m \geq 8$, the PWL string of any node containing a PE from Stage 1 starts off with 0101; i.e., it contains a 1 in both position 1 and 3. When $m > 8$, any node which is assigned a PE from Stage 2 starts off with either $*0*0$, $*0*1$, or $*1*0$ and thus there cannot be any collisions. For $m = 8$, a collision between Stage 1 and Stage 2 can only occur on level $r = 3$. In this case Stage 2 handles four trees, namely trees $T_0(8), T_1(8), T_2(8)$, and $T_3(8)$. The PWL strings of the nodes on level 3 that get PEs of $T_k(8)$, $k \leq 2$, assigned in Stage 2 are of the form $*0*1*_k0*_k00$ and hence there cannot be any collisions. The PWL strings of the nodes on level 3 that get PEs of $T_3(8)$ assigned in Stage 2 are of the form $*1*1*_30*_300$ with $*_3*_3 = 11$. Stage 1 assigns one PE, namely the root of $T(q)$, to level 3 and it is assigned to the node with PWL string 010100000. Hence, no collision is possible. This completes our description of Stage 2 in which we embedded levels 1 through $m/2$ of $T_k(m)$.

The embedding of the last $m/2$ levels of $T_k(m)$ is similar to Stage 3 in [4]. However, for our purpose, we divide the embedding of the last $m/2$ levels of $T_k(m)$ into two stages, namely Stages 3 and 4 which embed levels $m/2+1$ to $m/2+k$ and levels $m/2+k+1$ to m of $T_k(m)$, respectively. This division shows precisely how we use only the right half and the full $B_w(m+1)$ in Stages 3 and 4, respectively.

Stage 3: *Embedding levels* $m/2+1$ *to* $m/2+k$ *of* $T_k(m)$.

Let $S_i(k)$ be the i^{th} subtree of height $k+1$ rooted at a PE on level $m/2$ of $T_k(m)$, $0 \leq i \leq 2^{m/2}-1$, as shown in Figure 4. Subtrees $S_i(0)$ consists only of one PE and these PEs were already assigned in Stage 2. Hence, Stage 3 embeds the $S_i(k)$'s for $k \geq 1$. In Stage 2 the root of $S_i(k)$ was assigned to node $< m, (*_i 0)^{m/2-k-1} *_i 1(*_i 0)^k 1 >$[5]. As in [4], we refer to positions $\beta_0\beta_2 \ldots \beta_{m-2} = *_i *_i \ldots *_i$ as the *serial number* of $S_i(k)$. We embed $S_i(k)$ in Stage 3 so that the serial number and signatory 1 remain fixed. The serial number helps to keep the subtrees $S_i(k)$ and $S_{i'}(k)$ disjoint, for $i \neq i'$. The Signatory 1 helps to keep $S_i(k)$ and $S_i(k')$ disjoint, for $k \neq k'$.

We embed the first $\lceil k/2 \rceil$ levels of $S_i(k)$ by growing upwards (i.e., towards lower numbered levels) in B_2, the right half of $B_w(m+1)$. We then grow $S_i(k)$ downwards in B_2 to embed the remaining levels of $S_i(k)$. The leaves of $S_i(k)$ are placed at nodes $< m-1, (*_i 0)^{m/2-k-1} *_i 1(*_i *)^k 1 >$. When growing $S_i(k)$ upwards or downwards, we use every fourth level of B_2 and hence the embedding incurs a dilation of 4. The precise process of embedding $S_i(k)$ is described next.

For $1 \leq l \leq \lfloor k/2 \rfloor$, we assign the PEs on level l of $S_i(k)$ to nodes on level $m-4l+1$. The root of $S_i(k)$ was assigned in Stage 2 to node $< m, (*_i 0)^{m/2-k-1} *_i 1(*_i 0)^k 1 >$. By using a binary and two unary-straight branchings, assign the 2 PEs on level 1 of $S_i(k)$ to two nodes $< m-3, (*_i 0)^{m/2-k-1} *_i 1(*_i 0)^{k-1} *_i *1 >$. We now successively use a unary-straight, a binary, and two unary straight branchings to assign the 2^l PEs on level l of $S_i(k)$, $l \geq 2$, to nodes

$$< m-4l+1, (*_i 0)^{m/2-k-1} *_i 1(*_i 0)^{k-2l}(*_i 0 *_i *)^l 1 > .$$

If k is odd, we assign the $2^{\lceil k/2 \rceil}$ PEs on level $\lceil k/2 \rceil$ of $S_i(k)$ by using a unary-straight and a binary branching to nodes

$$< m-4\lfloor k/2 \rfloor - 1, (*_i 0)^{m/2-k-1} *_i 1 *_i *(*_i 0 *_i *)^{\lfloor k/2 \rfloor} 1 > .$$

An exception occurs for $k=1$. The leaves of $S_i(1)$ are at level 1 in $S_i(1)$ and they are assigned to level $m-1$ by making a binary branching from a node on level m.

[5] $*_i$ denotes a 0 or a 1 depending on the binary representation of i.

For $\lceil k/2 \rceil + 1 \leq l \leq k$, we now grow $S_i(k)$ downwards and assign the 2^l PEs on level l to nodes

$$< m - 4(k-l) - 1, (*_i 0)^{m/2-k-1} *_i 1(*_i *)^{2l-k}(*_i 0 *_i *)^{k-l} 1 > .$$

The dilation between the root of $S_i(k)$ and the two PEs on level 1 is at most 3 (it is 1 when $k = 1$ and 3 when $k > 1$). The dilation between the remaining levels of $S_i(k)$ is 4. An exception occurs between levels $\lfloor k/2 \rfloor$ and $\lceil k/2 \rceil$ when k is odd, and between levels $\lfloor k/2 \rfloor$ and $\lfloor k/2 \rfloor + 1$ when k is even: in these cases the dilation is 2. Since signatory 1 is not changed in Stage 3, an argument similar to the one in Stage 2, combined with the observation that serial number is kept fixed in Stage 3, shows that there are no collisions within Stage 3. Stage 3 uses only odd-numbered levels of B_2. Since Stages 1 and 2 use only odd-numbered levels of B_1, even level $m-2$ of B_1, and even level m of B_2, there are no collisions among Stages 1, 2 and 3. This completes the description of Stage 3.

Before describing Stage 4, we briefly explain why the last two levels of $T_k(m)$ embedded in Stage 2 (i.e., levels $m/2-1$ and $m/2$) were handled differently from the other levels. Stage 2 places the PEs on level $m/2-2$ of $T_k(m)$ on level $m-5$ of B_1 and Stage 3 places the PEs on level $m/2+1$ of $T_k(m)$ on level $m-3$ of B_2. In order to keep the dilation at 4, the PEs on level $m/2-1$ are placed on level $m-2$ of B_1 and the PEs on level $m/2$ are placed on level m of B_2, resulting in the different strategy for the last two levels of $T_k(m)$ in Stage 2. In addition, placing the PEs of level $m/2-1$ on level $m-2$ of B_1 allows us to set the signatory 1 correctly for $T_0(m)$.

Stage 4: *Embedding levels $m/2+k+1$ to m of $T_k(m)$.*

Let $T_k(m/2-1)$ denote the subtree of $T_k(m)$ rooted at the root of $T_k(m)$ with height $m/2$. Let R_k be the forest formed by $T_k(m) - T_k(m/2-1) - \bigcup_{i=0}^{2^{m/2}-1}\{S_i(k)\}$ as shown in Figure 4. Stage 4 embeds the trees in R_k with a dilation of 2 using the even levels in both B_1 and B_2. Every tree in R_k has $m/2-k$ levels that need to be embedded. Recall that the roots of these trees (i.e., the leaves of $S_i(k)$) were assigned for $k \geq 1$ in Stage 3 to nodes $< m-1, (*_i 0)^{m/2-k-1} *_i 1(*_i *)^k 1 >$ and for $k = 0$ in Stage 2 to nodes $< m, (*_i 0)^{m/2-1} *_i 11 >$. To make the necessary branchings we need $m/2-k$ bit positions in the PWL strings. Stage 4 uses the $m/2-k-1$ positions containing a 0 (and which are to the left of the signatory 1) and the rightmost position (which is '1' to start with). Stage 4 assigns the PEs on the first level of a tree in R_k to nodes $< 0, (*_i0)^{m/2-k-1} *_i 1(*_i*)^k* >$. This is done by a unary-straight and then a binary branching for $k \geq 1$ and simply a binary branching for the trees in R_0 . We now successively use a unary-straight and a binary branching to embed the remaining levels of R_k. Thus, PEs on level l of R_k are assigned to nodes $< 2l-2, (*_i*)^{l-1}(*_i0)^{m/2-k-l} *_i 1(*_i*)^k* >$, for $2 \leq l \leq m/2-k$.

Since the serial number and signatory 1 remain fixed in this stage, it is easy to see that there are no collisions within Stage 4. Stage 4 uses only even-numbered levels and since Stages 1 and 3 use only odd-numbered levels, there are no collisions between Stage 4 and Stages 1 or 3. See Table 1. Note that while level m is used by Stage 2 in B_2, Stage 4 does not use level m. Hence, the only possible collisions between Stage 2 and Stage 4 can occur on level $m-2$ in B_1. In Stage 4 only leaves of $T_0(m)$ are assigned to level $m-2$ and thus the PWL strings of nodes which have PEs assigned on level $m-2$ are $*^{m-1}1*$. The PWL strings of nodes on level $m-2$ which have PEs assigned in Stage 2 are $(*0)^{m/2-k-1}*1(*0)^{k-1}*_k00$ when $k \geq 1$ and $(*0)^{m/2-1}000$ when $k=0$. Thus, position $m-1$ has a 0 in Stage 2, while the same position has a 1 in Stage 4 and no collision is possible. This completes Stage 4 and hence the embedding of $T(q)$ into $B_w(m+1)$.

We conclude this section by showing how to modify our embedding when m is even and not a power of 2. Recall that $q = m + \lfloor \log m \rfloor - 1$. In stage 1 we embed the first $\lfloor \log m \rfloor$ levels of $T(q)$ by assigning the root of $T(q)$ to node $< m - 2\lfloor \log m \rfloor + 1, (01)^{m/2-\lfloor \log m \rfloor+1}0^{2\lfloor \log m \rfloor-1} >$. Levels 1 to $\lfloor \log m \rfloor - 1$ of $T(q)$ are embedded as before. Every occurrence of $\log m$ is now replaced by $\lfloor \log m \rfloor$. After Stage 1 we don't have $m/2$ PEs embedded on level $m-1$ of B_1, but $2^{\lfloor \log m \rfloor - 1}$. The other three stages embed the remaining m levels of $T(q)$. Stage 2 embeds the next $m/2$ levels (which are levels $\lfloor \log m \rfloor$ to $m/2 + \lfloor \log m \rfloor - 1$ of $T(q)$), and Stages 3 and 4 embed the final $m/2$ levels of $T(q)$. Since the trees $T_k(m)$ are now rooted at PEs on level $\lfloor \log m \rfloor - 1$ of $T(q)$, the range of k is from 0 to $2^{\lfloor \log m \rfloor - 1} - 1$. By reflecting this change throughout Stages 2, 3, and 4 and by using $\lfloor \log m \rfloor$ instead of $\log m$, we obtain the desired embedding of $T(q)$ into $B_w(m+1)$.

The argument of correctness carries over in a straightforward way. The proof that there are no collisions between the different stages must now consider the interaction between Stage 1 and Stage 2 for the case when $m = 6$. However, in this case Stage 1 uses levels 3 and 5, whereas Stage 2 uses levels 1, 4, and 6 and no collisions can occur.

3 Embedding $T(q)$ into $B_w(m+2)$

In Section 2 we described how to embed $T(q)$ into $B_w(m+1)$ with a dilation of 4. Naturally, the question arises whether there exists an embedding that achieves a smaller dilation. We do not yet know how to embed $T(q)$ into $B_w(m+1)$ with a dilation less than 4, but we next show how to embed $T(q)$ into $B_w(m+2)$ with a dilation of 2. This embedding achieves an optimal dilation since no embedding with $O(1)$ expansion can achieve a dilation less than 2 [4]. For the purposes of description, we again assume that m is a power of two. Our results hold, however, for any even integer m and the necessary changes are described at the end of Section 3.

Let B_i be the i^{th} quarter of $B_w(m+2)$ containing $(m+2)2^m$ nodes, $1 \le i \le 4$, as shown in Figure 5. We embed $T(q)$ into $B_w(m+2)$ in three stages. In *Stage* 1 we use B_1 and level $m+1$ of B_2 to embed the first $\log m$ levels of $T(q)$ so that the PEs on level $\log m - 1$ of $T(q)$ are assigned to nodes on level $m+1$ of B_2. Let $T_k(m)$ be again the k^{th} tree rooted at a PE on level $\log m - 1$ of $T(q)$, $0 \le k \le 2^{\log m - 1} - 1 = m/2 - 1$. *Stage* 2 embeds levels 1 to $m/2 + k$ of $T_k(m)$ into B_2 such that the PEs on level $m/2 + k$ of $T_k(m)$ are assigned to nodes on either level $m-1$ or m of B_2. In *Stage* 3 we embed the remaining levels of $T_k(m)$ into B_3 and B_4.

The main difference between this embedding and the one described in the previous section is in Stage 2. Our previous embedding and the one in [4] achieve assigning the PEs on the same level of $T(q)$ to different levels in the butterfly by growing subtrees upwards and downwards for varying number of levels. The embedding of this section achieves the same effect by having different subtrees grow for different number of levels already in Stage 2 right after the signatory 1 has been placed. Avoiding the upwards and downwards growth, together with other modifications, allows us to reduce the dilation from 4 to 2. We now describe each stage in the embedding of this section in more detail.

Stage 1: *Embedding the first* $\log m$ *levels of* $T(q)$

In this stage we could use the embedding of Stage 1 as given in Section 2. However, since we are using a $B_w(m+2)$ instead of a $B_w(m+1)$, we use a simpler embedding (similar to the one in [4]) that allows us to simplify the remaining stages. We can not use this version of Stage 1 in the embedding of Section 2 since it would result in collisions between Stages 1 and 2. For the sake of clarity and completeness, we describe Stage 1 in detail.

Assign the root of $T(q)$ to node $< m - \log m + 1, 0^{m+2} >$. Using binary branchings we assign, as done in [4], the 2^l PEs on level l of $T(q)$ to nodes $< m - \log m + l + 1, 0^{m-\log m+1} *^l 0^{\log m - l+1} >$, for $1 \leq l \leq \log m - 2$. Finally, by using a binary and then a unary-cross branching we assign the $m/2$ PEs on level $\log m - 1$ of $T(q)$ to nodes $< m+1, 0^{m-\log m+1} *^{\log m-1} 10 >$. In Figure 5 we show a few steps of the embedding in this stage. Observe that in this stage we use only B_1 and level $m+1$ of B_2. The dilation between levels l and $l+1$ of $T(q)$ is 1 for $0 \leq l \leq \log m - 3$ and it is 2 between levels $\log m - 2$ and $\log m - 1$. It is straightforward to see that there are no collisions in this stage.

Stage 2: *Embedding levels* 1 *to* $m/2+k$ *of* $T_k(m)$

As done in Section 2, we partition a PWL string into head and tail. The head now consists of leftmost $m - \log m + 1$ bits and the tail consists of rightmost $\log m + 1$ bits of the PWL string. While in Section 2 the head was of even and the tail of odd length, we now have both either even or odd. This is the reason for using $\lceil$'s in the following description. Observe that a node with a PE assigned to it from Stage 1 has a head consisting of 0's and a tail consisting of $*$'s with 10 as the two rightmost positions.

The goal of this stage is to assign, with a dilation of 2, the PEs on level $m/2+k$ of $T_k(m)$ to nodes $< m, (*0)^{m/2-k-1} * 1 *^{2k} 10 >$ for $k \geq 1$ and the PEs on level $m/2$ of $T_0(m)$ to nodes $< m-1, (*0)^{m/2} 10 >$. For tree $T_k(m)$, $k \geq 1$, the PWL strings get a signatory

Level of $T_k(m)$	Nodes of $B_w(m+2)$
$l = 0$	$< m+1, 0^{m-\log m+1} {*_k}^{\log m-1} 10 >$
$l \leq \frac{m}{2} - k$	$< 2l-1, (*0)^l 0^{m-\log m-2l+1} *_k^{\log m-1} 10 >$
$l \leq \frac{m}{2} + k - \log m$	$< \frac{m}{2} - k + l, (*0)^{\frac{m}{2}-k-1} * 1 *^{k+l-\frac{m}{2}} 0^{\frac{m}{2}+k-\log m-l+1} *_k^{\log m-1} 10 >$
$l \leq \frac{m}{2} + k$	$< \frac{m}{2} - k + l, (*0)^{\frac{m}{2}-k-1} * 1 *^{k+l-\frac{m}{2}} *_k^{\frac{m}{2}+k-l} 10 >$

Table 5: Assignments of Stage 2 in Section 3 for $k \geq \lceil \frac{\log m-1}{2} \rceil$.

1 set in position $m-2k-1$. For $k=0$, no signatory 1 will be set. We first describe the embedding when signatory 1 is placed in the head (i.e., we have $k \geq \lceil \frac{\log m-1}{2} \rceil$).

Stage 1 assigned the root of $T_k(m)$ to node $< m+1, 0^{m-\log m+1} *_k^{\log m-1} 10 >$. For $1 \leq l \leq m/2 - k$, we assign the 2^l PEs on level l of $T_k(m)$ to nodes on level $2l-1$ of B_2 by using a unary-straight and a binary branching. We then place the signatory 1 by making a unary-cross branching and then use a binary branching to assign the PEs on level $m/2-k+1$ of $T_k(m)$ to nodes on level $m-2k+1$ of B_2. Once the signatory 1 has been placed, we use only binary branchings to assign the PEs on level l of $T_k(m)$ to nodes on level $m/2-k+l$, for $m/2-k+2 \leq l \leq m/2+k$. Table 5 shows the precise assignment for $T_k(m)$ with $k \geq \lceil \frac{\log m-1}{2} \rceil$. The first line of the table shows assignment of the root and the second line covers the assignments up to the point the position of the signatory 1 is encountered. The third line covers the remaining positions in the head and the fourth line covers the tail. As in Section 2, not all lines are meaningful for all situations. For $k = \lceil \frac{\log m-1}{2} \rceil$, line 3 is meaningless. In this case the signatory 1 corresponds to the rightmost position of the head in case $\log m$ is odd and to the second rightmost position in case $\log m$ is even.

The embedding for the case when the signatory 1 is positioned in the tail is similar. In order to place 0's in the odd-numbered positions before the placement of the signatory 1, we now make a unary-cross or unary-straight branching depending on the value of $*$. We show the assignments for $1 \leq k < \lceil \frac{\log m-1}{2} \rceil$ in Table 6. The assignments for tree $T_0(m)$ are shown in Table 7. For $T_0(m)$ we stop at level $m-1$ in B_2 by assigning the leaves of $T_0(m)$ to nodes $< m-1, (*0)^{\frac{m}{2}} 10 >$. As already stated, no signatory 1 is set

Level of $T_k(m)$	Nodes of $B_w(m+2)$
$l = 0$	$< m+1, 0^{m-\log m+1} {*_k}^{\log m-1} 10 >$
$l \leq \lfloor \frac{m-\log m+1}{2} \rfloor$	$< 2l-1, (*0)^l 0^{m-\log m-2l+1} *_k^{\log m-1} 10 >$
$l \leq \frac{m}{2} - k$	$< 2l-1, (*0)^{l-1} * *_k^{m-2l+1} 10 >$
$l \leq \frac{m}{2} + k$	$< \frac{m}{2} - k + l, (*0)^{\frac{m}{2}-k-1} * 1 *^{k+l-\frac{m}{2}} *_k^{\frac{m}{2}+k-l} 10 >$

Table 6: Assignments of Stage 2 in Section 3 for $1 \leq k < \lceil \frac{\log m-1}{2} \rceil$.

Level of $T_0(m)$	Nodes of $B_w(m+2)$
$l = 0$	$< m+1, 0^{m-\log m+1} {*_0}^{\log m-1} 10 >$
$l \leq \lfloor \frac{m-\log m+1}{2} \rfloor$	$< 2l-1, (*0)^l 0^{m-\log m-2l+1} *_0^{\log m-1} 10 >$
$l \leq \frac{m}{2}$	$< 2l-1, (*0)^{l-1} * *_0^{m-2l+1} 10 >$

Table 7: Assignments of Stage 2 in Section 3 for $k = 0$.

for $T_0(m)$. The position for the signatory 1 for $T_0(m)$ would be position $m-1$ in the PWL strings. Setting this 1 would require the use of the unary-cross edges from level $m-1$ to m in B_2 and thus place the PEs on level $m/2$ of $T_0(m)$ on level m. This would result in a dilation of 3 since the PEs on level $m/2-1$ of $T_0(m)$ are assigned to level $m-3$ in B_2. Not having the signatory 1 set for $T_0(m)$ requires Stage 3 to handle $T_0(m)$ somewhat differently.

It is easy to see that the dilation is 2 in Stage 2. The argument to show that there are no collisions within this stage is similar to the one given for Stage 2 in Section 2 and it is omitted. Stage 1 uses only B_1 and level $m+1$ of B_2 and Stage 2 uses B_2 only up to level m. Hence, there are no collisions between Stage 1 and Stage 2.

Stage 3: *Embedding levels $m/2+k+1$ to m of $T_k(m)$.*

This stage is similar to Stage 4 in Section 2. However, since the PEs on level $m/2$

of $T_0(m)$ are assigned to nodes on level $m-1$ instead of m in B_2, we handle $T_0(m)$ differently. Let $R_i(m/2-k)$ be the i^{th} subtree rooted at a PE on level $m/2+k$ of $T_k(m)$, for $0 \leq i \leq 2^{m/2+k}-1$ and $0 \leq k \leq 2^{\log m-1}-1 = m/2-1$. We first describe the embedding of the last $m/2$ levels of $T_0(m)$ (i.e., embedding $R_i(m/2)$ for $0 \leq i \leq 2^{m/2}-1$), and then describe the embeddings of $R_i(m/2-k)$ for $k \geq 1$.

k = 0:

Recall that in Stage 2 we assigned the root of $R_i(m/2)$ to node $< m-1, (*_i0)^{m/2}10 >$. By using a binary branching we next assign the two PEs on level 1 of $R_i(m/2)$ to two nodes $< m, (*_i0)^{m/2-1} *_i *10 >$. We then assign the PEs on level 2 of $R_i(m/2)$ to nodes $< 0, (*_i0)^{m/2-1} *_i *^2 1 >$ by using a binary and a unary-cross branching, i.e., the PEs on level 2 are placed on level 0 of B_3 or B_4. We now use a unary-straight and a binary branching successively to assign the remaining levels of $R_i(m/2)$. In general, the 2^l PEs on level l of $R_i(m/2)$ are assigned to nodes $< 2l-4, (*_i*)^{l-2}(*_i0)^{m/2-l+1} *_i *^2 1 >$, for $3 \leq l \leq m/2$. The leaves of $R_i(m/2)$ are thus assigned to level $m-4$.

k ≥ 1:

Stage 2 assigned the root of $R_i(m/2-k)$ to node $< m, (*_i0)^{m/2-k-1} *_i 1 *_i^{2k} 10 >$. Assign the PEs on level 1 of $R_i(m/2-k)$ to nodes $< 0, (*_i0)^{m/2-k-1} *_i 1 *_i^{2k} *1 >$ by using a binary and a unary-cross branching. Then use a unary-straight and a binary branching to assign the 2^l PEs on level l of $R_i(m/2-k)$ to nodes $< 2l-2, (*_i*)^{l-1}(*_i0)^{m/2-k-l} *_i 1 *_i^{2k} *1 >$, for $2 \leq l \leq m/2-k$. The leaves of $R_i(m/2-k)$ are thus assigned to level $m-2k-2$.

We now show that there are no collisions within Stage 3. Since the serial number (denoted by the $*_i$'s in the PWL strings) is kept fixed, there are no collisions between $R_i(m/2-k)$ and $R_{i'}(m/2-k)$, for $i \neq i'$. The signatory 1 is not changed in Stage 3 and hence there are no collisions between $R_i(m/2-k)$ and $R_i(m/2-k')$, for $k \neq k'$ and $1 \leq k, k' \leq 2^{\log m-1}-1 = m/2-1$. The signatory 1 in position $m-2k-1$, for $k \geq 1$, also avoids the possibility of collision between $R_i(m/2)$ and $R_i(m/2-k)$. This is seen as follows. Assume level r, $r \leq m-2k-2$, has PEs of both $R_i(m/2)$ and $R_i(m/2-k)$ assigned. Then, a node containing a PE of $R_i(m/2-k)$ has its signatory 1 in the position

$m-2k-1$ of its PWL string and the leaves of $R_i(m/2-k)$ are placed on level $m-2k-2$. A node containing a PE of $R_i(m/2)$ starts off with a 0 in position $m-2k-1$ which is changed into a 1 (by a binary branching) when going from level $m-2k-1$ to level $m-2k$. (The 0 in position $m-3$ is never changed.) Hence, there cannot exist a node that is assigned both a PE from $R_i(m/2)$ and $R_i(m/2-k)$.

Finally, it is easy to see that there are no collisions between Stages 1 or 2 and Stage 3. Stages 1 and 2 use only B_1 and B_2, while Stage 3 uses B_3 and B_4 except for $T_0(m)$ which also uses level m of B_2 in Stage 3. However, the signatory 1's keep the nodes on level m that have PEs of $T_k(m)$ assigned in Stage 2, for $k \geq 1$, distinct from the nodes that have PEs of $T_0(m)$ assigned in Stage 3. This completes Stage 3 and hence our second embedding.

We conclude this section by showing how to modify our embedding of this section when m is an even integer and not a power of two. In Stage 1, we embed PEs on levels 0 to $\lfloor \log m \rfloor - 2$ of $T(q)$ to nodes on levels $m - \lfloor \log m \rfloor + 1$ to $m-1$ of B_1. The PEs on level $\lfloor \log m \rfloor - 1$ of $T(q)$ are assigned to nodes on level $m+1$ of B_2. Every occurrence of $\log m$ is now replaced by $\lfloor \log m \rfloor$. After Stage 1 we have $2^{\lfloor \log m - 1 \rfloor}$ PEs embedded on level $m+1$ of B_2. Hence, we have $2^{\lfloor \log m - 1 \rfloor}$ trees $T_k(m)$'s and the range of k is from 0 to $2^{\lfloor \log m - 1 \rfloor} - 1$. By reflecting this change throughout Stages 2 and 3 and by using $\lfloor \log m \rfloor$ instead of $\log m$, we obtain the desired embedding of $T(q)$ into $B_w(m+2)$.

4 Embeddings using no wrap-around

In the previous two sections we described two embeddings of complete binary trees into a butterflies with wrap-around connections. Both embeddings make heavy use of the wrap-around connections. In this section we first show that, while wrap-around connections are convenient, they do not make the butterfly more powerful (in an asymptotic sense). More precisely, we next describe an embedding of $B_w(m)$ into $B(m)$ which has a dilation of 3. To the best of our knowledge this useful result has not been documented in the literature. By making use of this result, we then show that $T(q)$ can be embedded into

$B(m+1)$ (resp. $B(m+2)$) with a dilation of 8 (resp. 5).

For clarity, henceforth we refer to the nodes of $B_w(m)$ as PEs and to the nodes of $B(m)$ as nodes. The embedding of $B_w(m)$ into $B(m)$ achieves an expansion of $\frac{(m+1)2^m}{m2^m} < 2$. This is optimal since $B(m)$ is the smallest butterfly with no wrap-around consisting of at least $m2^m$ nodes. Any embedding of $B_w(m)$ into $B(m)$ must achieve a dilation of at least 2. To show this, assume, for the sake of contradiction, that there exists an embedding ρ with a dilation of 1. In ρ every node of $B(m)$ is assigned at most one PE of $B_w(m)$. Furthermore, an edge in $B(m)$ is assigned an edge in $B_w(m)$. Since the number of nodes in $B(m)$ is greater than the number of PEs in $B_w(m)$, there exists a node v in $B(m)$ that has no PE of $B_w(m)$ assigned to it. No edge incident to v has an edge of $B_w(m)$ assigned to it (otherwise the dilation would be more than 1). Now, since v has at least 2 incident edges, at most $m2^{m+1}-2$ edges of $B(m)$ have edges of $B_w(m)$ assigned to them in ρ. Since $B_w(m)$ consists of $m2^{m+1}$ edges and every edge of $B(m)$ is assigned at most 1 edge of $B_w(m)$, ρ can not have a dilation of 1.

We assign the PEs on level 0 of $B_w(m)$ to nodes on level 0 of $B(m)$. The remaining PEs are assigned so that two PEs on adjacent levels in $B_w(m)$ are at most 2 levels apart in $B(m)$ and the PEs on level $m-1$ are assigned to nodes on level 3. Furthermore, nodes on level 1 of $B(m)$ get no PE assigned. Formally, let $p = < l, \beta_0\beta_1\ldots\beta_{m-1} >$ be a PE of $B_w(m)$, $0 \le l \le m-1$. Then p is assigned to node $< f(l), \alpha_0\alpha_1\ldots\alpha_{m-1} >$, where

$$f(l) = \begin{cases} 2l & \text{for } 0 \le l \le \lfloor \frac{m}{2} \rfloor \\ m & \text{if } m \text{ is odd and } l = \lceil \frac{m}{2} \rceil \\ 2m-2l+1 & \text{for } \lceil \frac{m}{2} \rceil + 1 \le l \le m-1, \end{cases} \quad \text{and}$$

$$\alpha_i \;=\; \begin{cases} \beta_{i/2} & \text{for } i = 0,2,4,\ldots,2\lfloor \frac{m}{2} \rfloor - 2 \\ \beta_{m-\frac{i+1}{2}} & \text{for } i = 1,3,5,\ldots,2\lfloor \frac{m}{2} \rfloor - 1 \\ \beta_{\lfloor m/2 \rfloor} & \text{if } m \text{ is odd and } i = m-1. \end{cases}$$

For example, when $m = 8$ a PE with PWL string $\beta_0\beta_1\beta_2\beta_3\beta_4\beta_5\beta_6\beta_7$ is assigned to the node that has PWL string $\alpha_0\alpha_1\alpha_2\alpha_3\alpha_4\alpha_5\alpha_6\alpha_7 = \beta_0\beta_7\beta_1\beta_6\beta_2\beta_5\beta_3\beta_4$. Furthermore, PEs on levels $0, 1, 2, 3, 4$ are assigned to nodes on levels $0, 2, 4, 6, 8$, respectively, and PEs on levels $5, 6, 7$ are assigned to nodes on levels $7, 5, 3$, respectively.

It is easy to see that no two PEs are assigned to the same node and hence we have an embedding of $B_w(m)$ into $B(m)$. We next show that the dilation is 3. Assume that m is an even integer (the argument for odd m's is analogous). Let (p_1, p_2) be a cross edge of $B_w(m)$ with $p_1 =< l, \beta_0 \ldots \beta_l \ldots \beta_{m-1} >$ and $p_2 =< (l+1) \bmod m, \beta_0 \ldots \bar{\beta}_l \ldots \beta_{m-1} >$, $0 \le l \le m-1$. Depending on the value of l we distinguish four cases.

Case 1: $0 \le l \le \frac{m}{2} - 1$.

PEs p_1 and p_2 are assigned to nodes $v_1 =< 2l, \beta_0\beta_{m-1} \ldots \beta_l\beta_{m-l-1} \ldots \beta_{m/2-1}\beta_{m/2} >$ and $v_2 =< 2l+2, \beta_0\beta_{m-1} \ldots \bar{\beta}_l\beta_{m-l-1} \ldots \beta_{m/2-1}\beta_{m/2} >$, respectively. Node v_1 is connected to v_2 by a path consisting of a cross and a straight edge resulting in a dilation of 2.

Case 2: $l = m/2$.

PE p_1 is assigned to node $v_1 =< m, \beta_0\beta_{m-1} \ldots \beta_{m/2-1}\beta_{m/2} >$ and PE p_2 is assigned to node $v_2 =< m-1, \beta_0\beta_{m-1} \ldots \beta_{m/2-1}\bar{\beta}_{m/2} >$. Hence v_1 and v_2 are connected by a cross edge and the dilation between p_1 and p_2 is 1.

Case 3: $\frac{m}{2} + 1 \le l \le m-2$.

PE p_1 is assigned to node $v_1 =< 2m - 2l + 1, \beta_0\beta_{m-1} \ldots \beta_{m-l-1}\beta_l \ldots \beta_{m/2-1}\beta_{m/2} >$ and PE p_2 is assigned to node $v_2 =< 2m - 2l - 1, \beta_0\beta_{m-1} \ldots \beta_{m-l-1}\bar{\beta}_l \ldots \beta_{m/2-1}\beta_{m/2} >$. Hence v_1 and v_2 are connected by a path that consists of a straight and a cross edge, and the dilation is 2.

Case 4: $l = m-1$.

PE p_1 is assigned to node $v_1 =< 3, \beta_0\beta_{m-1} \ldots \beta_{m/2-1}\beta_{m/2} >$ and PE p_2 is assigned to node $v_2 =< 0, \beta_0\bar{\beta}_{m-1} \ldots \beta_{m/2-1}\beta_{m/2} >$. Node v_1 is connected to v_2 by a path consisting of a straight, a cross, and a straight edge, and hence the dilation between p_1 and p_2 is 3.

While in our embedding the wrap-around edges are the ones having a dilation of 3, any level l of $B_w(m)$ can be chosen so that edges from level l to $l+1$ have a dilation of

3. This completes the embedding of $B_w(m)$ into $B(m)$ with a dilation of 3, and we next briefly describe the embedding of $T(q)$ into a no wrap-around butterfly.

By first embedding $T(q)$ into $B_w(m+1)$ with a dilation of 4 (using the embedding of Section 2) and then embedding $B_w(m+1)$ into $B(m+1)$ with a dilation of 3, it immediately follows that $T(q)$ can be embedded into $B(m+1)$ with a dilation of at most $4 * 3 = 12$. Similarly, $T(q)$ can be embedded into $B(m+2)$ with a dilation of at most $2 * 3 = 6$. However, since only the wrap-around edges incur a dilation of 3 and the remaining edges incur a dilation of at most 2, we can do better. Every edge of $T(q)$ maps onto a path of length at most 4 in $B_w(m+1)$ and every such path contains at most one wrap-around edge. Hence, the dilation of every edge of $T(q)$ is at most $3+2+2+2=9$. Similar arguments can be used to embed $T(q)$ into $B(m+2)$ with a dilation of only $3+2=5$.

In order to obtain the claimed dilation of 8 for the embedding of $T(q)$ into $B(m+1)$, we change the embedding of $B_w(m+1)$ into $B(m+1)$ so that the edges from level 1 to 2 of $B_w(m+1)$ have a dilation of 3 in $B(m+1)$ and the remaining edges have a dilation of at most 2. Edges of $T(q)$ using an edge from level 1 to 2 of $B_w(m+1)$ have a dilation of only 2 and thus have a dilation of $3+2=5$ in $B(m+1)$. Edges of $T(q)$ having a dilation of 4 in $B_w(m+1)$ now incur a dilation of $4 * 2 = 8$.

5 Conclusions

In this paper we described two embeddings of a complete binary tree T into a wrap-around butterfly B_w, and two embeddings of T into a no wrap-around butterfly B. The first embedding embeds $T(q)$ into $B_w(m+1)$ with a dilation of 4, where $q = m + \lfloor \log m \rfloor - 1$. The second one embeds $T(q)$ into $B_w(m+2)$ with a dilation of 2. The first embedding achieves an expansion of 2, while the second one achieves an expansion of 4 and an optimal dilation. An embedding optimizing both the dilation and the expansion would embed $T(q)$ into $B_w(m+1)$ with a dilation of 2 (for even m) [4]. We feel that a technique different from the one using signatory 1's is needed to achieve this bound.

Our third (resp. fourth) embedding embeds $T(q)$ into $B(m+1)$ (resp. $B(m+2)$) with a dilation of 8 (resp. 5).

6 Acknowledgements

We would like to thank an anonymous referee for helpful suggestions and comments.

References

[1] R. Aleliunas and A. Rosenberg. On embedding rectangular grids into square grids. *IEEE Trans. on Computers*, V-31:907–913, 1982.

[2] M. J. Atallah. On multidimensional arrays of processors. *IEEE Transactions on Computers*, 37(10):1306–1309, October 1988.

[3] S. Bettayeb, I. H. Sudborough, and Z. Miller. Embedding Grids into Hypercubes. In *the Proc. of* 3^{rd} *Aegean Workshop on Computing*, 1988.

[4] S. Bhatt, F. Chung, J. -W. Hong, F. T. Leighton, and A. L. Rosenberg. Optimal simulations by Butterfly networks. In *the Proceedings of* 20^{th} *Symposium on Theory of Computing*, pages 192–204, May 1988.

[5] S. Bhatt, F. Chung, F. T. Leighton, and A. Rosenberg. Optimal simulations of tree machines. In *the Proceedings of* 27^{th} *Symposium on Foundation of Computer Science*, pages 274–282, 1986.

[6] M. Y. Chan and F. Y. L. Chin. On embedding rectangular grids in hypercubes. *IEEE Transactions on Computers*, 37(10):1285–1288, October 1988.

[7] J. P. Fishburn and R. A. Finkel. Quotient networks. *IEEE Transactions on Computers*, C-31(4):288–295, April 1982.

[8] A. K. Gupta and S. E. Hambrusch. Embedding large tree machines into small ones. In *the Proceedings of* 5^{th} *MIT Conference on Advanced Research in VLSI*, pages 179–199, March 1988.

[9] J. -W. Hong, K. Mehlhorn, and A. Rosenberg. Cost trade-offs in graph embeddings, with applications. *JACM*, 709–728, 1983.

[10] S. R. Kosaraju and M. Atallah. Optimal simulations between mesh connected array of processors. *Journal of the Association of Computing Machinery*, 35(3):635–650, July 1988.

[11] A. L. Rosenberg. Preserving proximity in arrays. *SIAM Journal on Computing*, 443–460, 1979.

[12] Y. Saad and M. H. Schultz. Topological properties of hypercubes. *IEEE Transactions on Computers*, 37(7):867–872, July 1988.

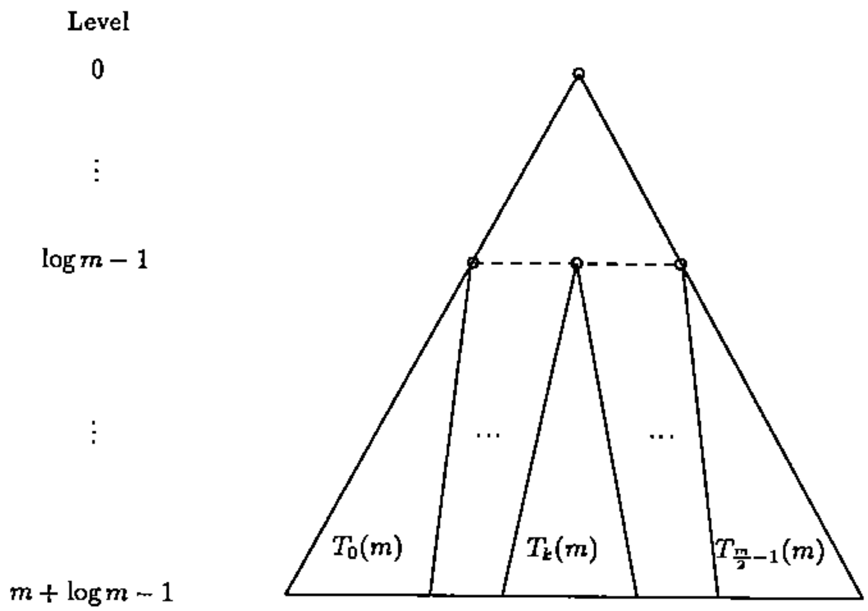

Figure 1: Tree T(q) showing trees T_k(m), for $0 \leq k \leq m/2\text{-}1$.

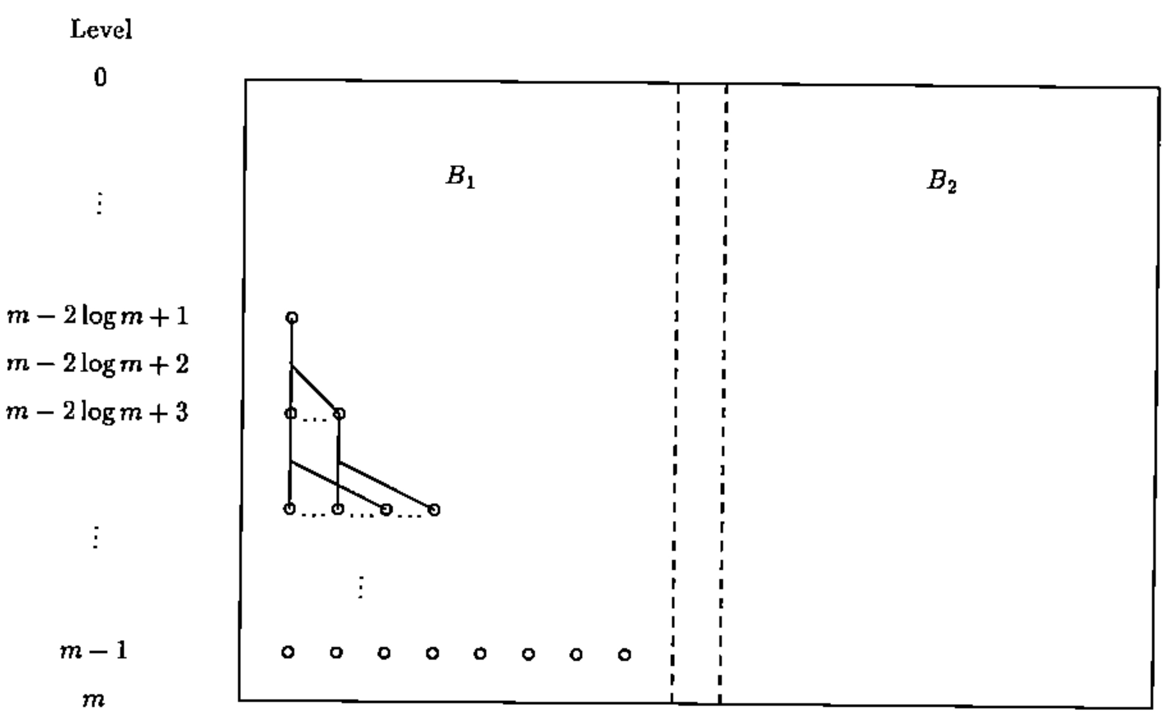

Figure 2: B_w(m+1) showing first 3 steps of the embedding in Stage 1 of Section 2.

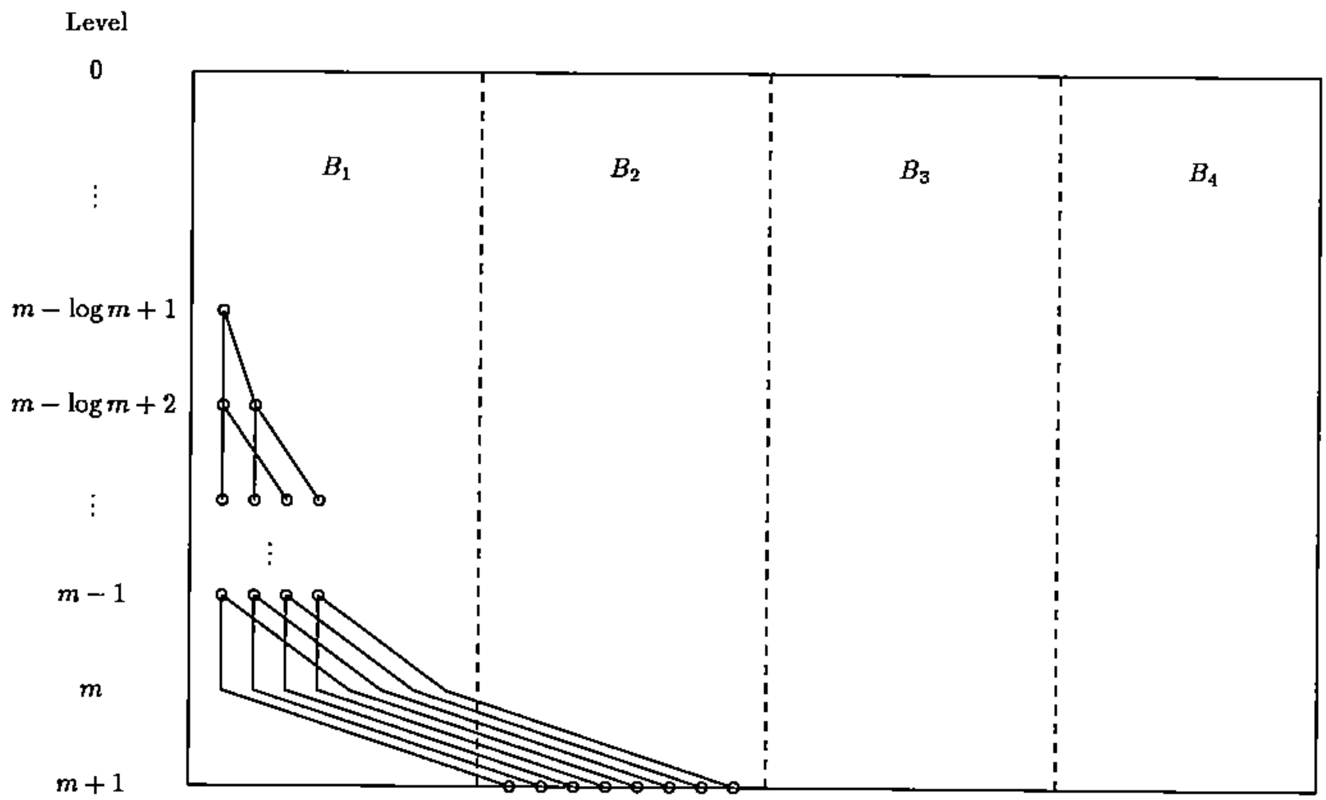

Figure 5: B_w(m+2) showing assignments in Stage 1 of Section 3.

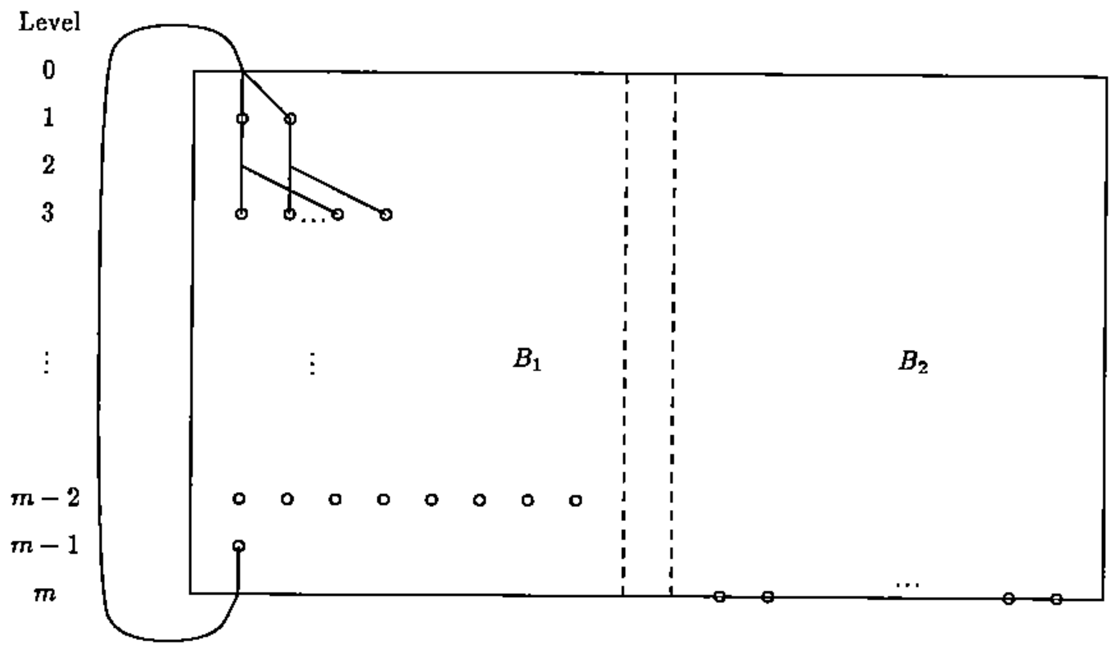

Figure 3: First few steps in the embedding of Stage 2 in Section 2.

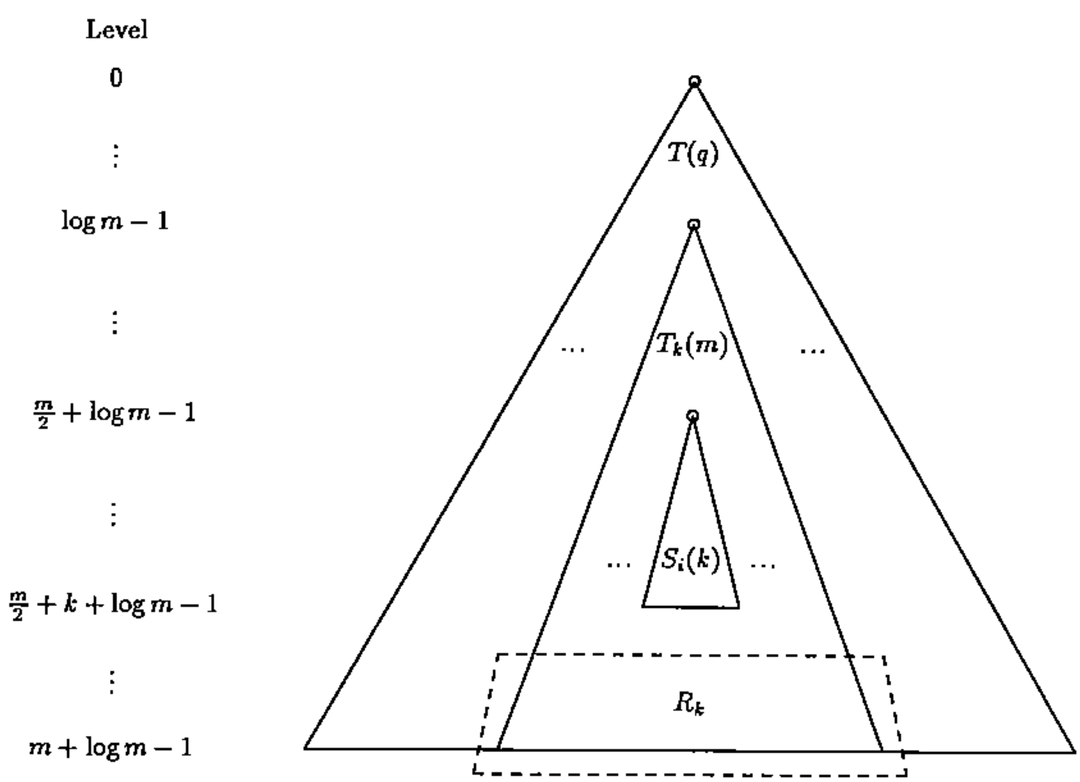

Figure 4: Tree T(q) showing tree T_k(m), subtree S_i(k), and forest R_k.