NEW BOUNDS FOR THE LONGEST EDGE OF A TREE IN A VLSI LAYOUT

by

Michael Kaufmann

A 84/14

FB 10
Universität des Saarlandes
6600 Saarbrücken
West-Germany

April 1984

Abstract:

In the last three years many results were published about graph layout in VLSI. One aspect of graph layout is the minimization of the longest edge; for this problem Bhatt and Leiserson (1982) recently demonstrated a new technique to shorten the longest edge, and they thus achieved an upper bound of $O(\sqrt{N}/\log N)$ for trees. Unfortunately, no good universal lower bounds exist. This paper presents a general technique for proving lower bounds for trees. A second technique to embed trees is presented, which provides really good upper bounds for the maximal edge length in relation to the disposable area.

1. Introduction

For VLSI theory it is interesting to consider how to lay out a graph in the plane. The reason is that a network in VLSI can be interpreted (with some simplifications) as a graph, with transistors corresponding to nodes and wires to the edges of the graph. Thus we can solve many problems in VLSI by solving the related graph theoretic problems.

One important aspect here is the longest wire in a network layout. This wire can determine the performance of the circuit, if the time to propagate information through a wire grows with the lenght of the wire. The capacitive model by Mead and Conway (1980) supports this view. In systolic systems thus, the period of the system clock must be longer than the longest propagation delay.

For general graphs, there exist some results developed by Bhatt and Leiserson and by Leighton (1981,1982). But if we restrict ourselves to the class of trees, these results are rather weak; therefore this problem is mentioned as an open problem in all of these papers.

We divide this paper into 4 sections. Section 2 reviews the standard model, as well as some definitions and lemmata. Then we present a nearly trivial lower bound for the maximal edge length. In section 3, we develop the main result. We prove lower and upper bounds. We give a simple example which explains the technique of which we make use subsequently to obtain the lower bound. The upper bound is an application of the method of Bhatt/Leiserson. Thus we achieve nearly tight bounds for the considered classes of trees. Then, using a method of Paterson/Ruzzo/Snyder, we choose the edge length with respect to the disposable area and improve the upper bounds.

2. Background

The first model for graph layout was suggested by Thompson (1979). He considered the layout as an embedding in a two-dimensional grid. This grid consists of horizontal and vertical lines space apart at unit intervals.

A layout of a graph is an assignment of nodes to crossings of horizontal and vertical lines. The edges follow the lines, they

may cross each other but not overlap for any distance, and they may not cross nodes which they are not connecting. The area of the layout equals the product of the numbers of vertical and horizontal lines that contain a node or edge segment of the graph.
For this layout model, we restrict the graphs to those having nodes with degree at most four, and we only consider binary trees; extension to other classes of graphs is easy. Most layout techniques base on the divide-and-conquer paradigm and use graph separators to recursively partition a given circuit. For trees, this technique yields a linear upper obound on the area.

Definition 1:

A class of graphs closed under the subgraph relation has an f(x)-separator theorem, if there are constants a,b with $0<a\leq 1/2$ and $b>0$, such that by removing at most $b\times f(n)$ edges, every N-node graph of the class can be partitioned into disjoint subgraphs having $c\times N$ and $(1-c)\times N$ nodes, where $a\leq c\leq 1-a$.

Definition 2:

An f(x)-separator is said to be perfect if $c\times N = \lceil N/2\rceil$ or $\lfloor N/2\rfloor$.

The lemmata 1 and 2 below can be proved by familiar techniques.

Lemma 1:

Any tree has an O(log N)-perfect separator.

A complete binary tree can be laid out as an H-tree (Figure 1) in linear area as shown by the recurrence equation for the side length s of the layout:

$$s(N) = 2s(N/4) + 1$$

If we extend the H-tree edges to channels with width $(N/2^i)^\alpha$ for level i, $0\leq i\leq \log N$, $0<\alpha<1/2$, then the '1' in the recurrence equation is replaced by N^α (Figure 2).

Lemma 2:

The linear upper bound for the area still holds for an extended H-tree (used by Bhatt/Leiserson) with channel width $(N/2^i)^\alpha$ for all levels i, $0 \leq i \leq \log N$.

For different channelwidth, we get the following upper bounds for the area:

channelwidth of level i	area
$(N/2^i)^\alpha, \alpha < 1/2$	$O(N)$
$\sqrt{N/2^i}$	$O(N \times \log^2 N)$
$(N/2^i)^\alpha, \alpha > 1/2$	$O(N^{2\alpha})$

The following trivial lower bound for the length of the longest edge shows that the upper bound of Bhatt and Leiserson is optimal for complete binary trees.

Lemma 3:

Let L be the maximal edge length for a layout of a tree, and let $v_1,\ldots,v_N$ be its nodes. Let $MD = \sum_{i,j} d(v_i,v_j)/N^2$ be the average path length, where $d(v_i,v_j)$ is the number of nodes on the path from v_i to v_j. Then $L \geq \Omega(\sqrt{N}/MD)$.

Proof: Partition any layout into three stripes V_1,V_2,V_3, such that V_1 and V_3 each contain N/10 nodes. (Figure 3)
The paths between the nodes of V_1 and V_3 have at least length $\sqrt{N}$ (side length of linear layout area). Clearly, at least one path from a node in V_1 to a node in V_3 contains at most O(MD) nodes. Thus we can conclude that the longest edge must have a length of at least $\Omega(\sqrt{N}/MD)$.

The average number of nodes on a path between two nodes in a complete binary tree is $O(\log N)$. This implies a lower bound for the longest edge of a complete binary tree of size $\Omega(\sqrt{N}/\log N)$.
On the other hand, for the tree that consists of just one path we find a linear area layout with edge length 1.
Intuitively, we can conclude:
The denser the tree, the longer the longest edge.

3. New Bounds

3.1 Lower bounds

To explain the technique, we look at the following simple example.

Consider a complete binary tree of depth r, extended by chaines of length m hanging down from the leaves. (Figure 4)

It is clear that $N = 2^r \times m + 2^r - 1 \leq 2^{r+1} \times m$.

Let m be $\sqrt{N}$, hence $r = \Theta(\log N)$.

chaines $= 2^r \times m \geq N/2 / \sqrt{N} = \sqrt{N}/2$

In any layout of the tree, let d be the radius of the smallest circle around the root which contains all nodes of the r upper levels.

$L \geq d/r$

Case 1: $d > \sqrt{N/2\pi}$ $\qquad L \geq \Omega(\sqrt{N}/\log N)$

Case 2: $d < \sqrt{N/2\pi}$

at least N/2 nodes lie outside the circle and so more than half of the chaines of length m lie outside the circle.

The circumference $2\pi d$ must be $> 1/2 \cdot \sqrt{N}/2$.

$d > \sqrt{N}/8\pi$.

$L \geq \Omega(\sqrt{N}/\log N)$

In contrast, note that lemma 3 yields only a lower bound of $\Omega(1)$.

Now we sketch the general method for lower bounds:

Let range (v,d) be the number of nodes which have distance d from node v.

Let $k = \min_v \{d \mid \text{range}(v,d) = c\sqrt{N}\}$. c is a constant, and let v_0 be the node determined by the minimization.

We look only at trees with the following property:

If we consider less than $c/2 \times \sqrt{N}$ edges which have a greater distance from v_0 than k, then the number of the nodes in all subtrees hanging on these edges is less then $c' \times N$. c' is a constant > 1.

Now, we show again, that $d \geq \Omega(\sqrt{N})$; d is the radius of the smallest circle around v_0, where the $c \times \sqrt{N}$ nodes on the first k levels must lie inside the circle.

We assume that $d < c\sqrt{N}/4\pi$.

circumference $< c\sqrt{N}/2$.

We see that the circle 'cuts' some edges which connect some parts of the tree which lie in- and outside the circle. The circumference of the circle is an upper bound of the size of the cut, and we conclude after assumption that there are at least $(1 - c')\times N$ nodes inside the circle.

$$\pi d^2 \geq (1-c')\times N$$
$$d \geq \sqrt{(1-c')\times N/\pi}$$
$$\underline{L \geq \Omega(\sqrt{N}/k).}$$

3.2 Upper bounds

We improve the embedding technique of Bhatt/Leiserson using a new concept of Leighton (1982), but it remains nearly the same.

Definition 3:

A graph G has an $(F_1,F_2,\ldots F_r)$-decomposition tree, if G can be partitioned in two disjoint subgraphs G_0 and G_1 by removing at most F_1 edges, each of these subgraphs can be partitioned again in two subgraphs G_{00} and G_{01} resp. G_{10} and G_{11} by removing at most F_2 edges. Repeat this until step r, where the subgraphs are empty or they consist of only one node.

Definition 4:

An $(F,F/2^{\alpha},F/2^{2\alpha},\ldots)$-decomposition tree is called an 2^{α}-befurcator of size F or shorter $(F,2^{\alpha})$-bifurcator.

α will be chosen as $0<\alpha\leq 1/2$. Leighton chooses it as $\alpha = 1/2$. We know that each tree has an $(N^{\varepsilon},2^{\varepsilon})$-bifurcator, $\varepsilon>0$, since we can partition trees in equal parts by removing at most $O(\log N)$ edges.

Definition 5:

The decomposition tree for a graph with N nodes is called fully balanced, if

a) each subgraph G_{ω} in the tree is the ancestor of two equalsized subgraphs $G_{\omega 0}$ and $G_{\omega 1}$ ($\omega \in \{0,1\}^*$)
b) the number of edges of G incident to precisely one node in $G_{\omega 0}$ differs by at most an additive constant from the number of edges of C incident to precisely one node in $G_{\omega 1}$.

In the following, we repeat some theorems (1982), which show us, how to fully balance decomposition trees without big expense.

The basic lemma is the following:

Lemma (Goldberg/West)

Given any ordering of n balls in a line, n_i of which are colored i for $1 \leq i \leq k$, it is possible to break the line in at most k places so that the union of the balls contained in every other segment contains precisely $\lfloor n_i/2 \rfloor$ or $\lceil n_i/2 \rceil$ balls which are colored i for $1 \leq i \leq k$.

Leighton uses this lemma coloring the nodes of a subgraph on the i-th stage of the decomposition tree by color j, if on stages 1 until i-1 j incident edges were cut.

By this technique he can prove the following theorem:

Theorem 1

Let G be any N-node graph with an $(F_1, F_2, \ldots, F_r)$-decomposition tree T. It is possible to construct a fully balanced $(F_1, F_2', \ldots, F'_{\log N})$-decomposition tree T' for G where $F_i' = 6 \sum_{s=i}^{r} F_s$.

For trees, this means that we will be able to construct fully balanced $(N^{\varepsilon}, 2^{\varepsilon})$-bifurcators, where $\varepsilon > 0$ is a constant as small as we like. (for details see (1982))

Now, we will embed the tree in an extended H-tree.

Sketch of the layout procedure: (Figure 5)

Place the edges of the fully balanced decomposition at the first stage and the corresponding nodes parallel to each other with length L on the top channel and place the edges incident to the embedded nodes in such way that the distance between their nodes is L.

This step is repeated until some nodes are placed at the end of the top channel.

Then place the next edges using Leighton's coloring technique such that about half of them goes to the right and half of them to the left of the second channel. In the middle of this channel place the edges of the second decomposition stage, too.

Repeat this procedure until you reach the level in the H-tree where you can trivially lay out the remaining subtrees of size L^2 with maximal edge length L.

We will show that to determine the length of the longest edge, we only need consider the subtree at the top channel of the H-tree.
More precisely we show now:

> If we can embed the part of the tree with the highest density in the top channel of the H-tree with maximal edge length L, hence we can embed the whole tree with max. edge length $c \times L$ in the H-tree with linear area, where c is a rather small constant.

Proof:

We know that if we embed the most dense part of the tree by our technique, there are $f(\sqrt{N}/L)$ nodes at the end of the top channel, where L is minimal so that $f(\sqrt{N}/L) \leq N^{\alpha}$, $\alpha < 1/2$.
f is monotone and f(x) describes the maximal number of nodes you can reach by paths of length x, starting from a node v.
At a channel on level i in the H-tree, the number of nodes

$$\leq \sum_{j=0}^{i} \frac{N^{\varepsilon}}{2^{\varepsilon j}} \times f\left(\frac{4\sqrt{N}/2^{j/2}}{c \times L}\right)/2^{i-j}$$

, if we use the length $c \times L$ and where $4\sqrt{N}/2^{j/2}$ is an upper bound for the distance in the H-tree from level j to level i; $1 \leq j \leq i \leq \log N$. $N^{\varepsilon}/2^{\varepsilon j}$ is an upper bound for the size of the fully balanced decomposition tree on stage j.
We have to show that there is a constant c, such that

$$\text{for all } i: 0 \leq i \leq \log N \quad \sum_{j=0}^{i} \frac{N^{\varepsilon}}{2^{\varepsilon j}} \cdot f\left(\frac{4\sqrt{N}}{c \times L \times 2^{j/2}}\right)/2^{i-j} \leq \underbrace{N^{\alpha}/2^{\alpha i}}_{\text{width of the channel}}$$

For simplicity, let the sum be 'A'.
We claim that

$$A \leq N^{\varepsilon} \times (i+1) \times f\left(\frac{4\sqrt{N}}{c \times L}\right)/2^{j}.$$

This inequation follows from the following fact:
A is the number of nodes at the end of the i-th stage in the H-tree as the sum over j where there are i parts of subtrees, each begins at stage j and is partitioned (i-j)-times. If we assume that all these subtrees start at the top level and are partitioned i-times, it is clear, that from each of these sub-

trees, there would be more nodes at level i than above.

$$
\begin{aligned}
(i+1)\times N^{\varepsilon}\times f(4\sqrt{N}/c\times L)/2^{i} &\lessapprox N^{\varepsilon}\times \log N\times f(4\sqrt{N}/c\times L)/2^{i} \\
&\leqq N^{\varepsilon'}\times f(4\sqrt{N}/c\times L)/2^{i}, \qquad \varepsilon'>\varepsilon \\
&\leqq N^{\varepsilon'+\alpha}/2^{i} \qquad c = 4 \\
\text{with } \alpha+\varepsilon' = \beta < 1/2 \qquad &\leqq N^{\beta}/2^{\beta i}
\end{aligned}
$$

We can conclude that we only need consider the top channel of the H-tree. The maximal edge length L,we get here, must be increased only by a multiplicative constant factor to be valid for the whole tree.
We only need consider even the biggest subtree T on the top channel, since there are O(logN) subtrees and for all $\alpha < 1/2$ there is a $\beta < 1/2$: $N^{\beta}\times \log N \leqq N^{\beta}$.

Let range(v,d) be the number of nodes, which have distance d from node v. Let $k = \min_{v} \{d \mid \text{range}(v,d) = c\times\sqrt{N}/d \text{ and } \text{range}(v,d) \leqq m\times\sqrt{N}/d^{2},\ m<d\}$

Then we can embed the first k levels of the subtree T on the top channel of the tree. We neglect the other subtrees on the topchannel, they can increase the area at most by a factor log N.
The depth of T is k, and hence the length L_1 of the edges is $\sqrt{N}/k$. There are at most $\sqrt{N}/k$ nodes at the end of the channel. We know that the length L_2 of the longest edge of T is not as long as the number of nodes at the channel end (Figure 6).
Thus

$$\underline{L = L_1 = L_2 \leqq O(\sqrt{N}/k).}$$

The supposition for the vertical part L_2 of the longest edge is not very fine; it would be optimal, if we can say, that this length is as big as width/number of steps; now we could choose $c\times\sqrt{N}$ as the width of the channel. Hence k is decreased and we reach our lower bound asymptotically. Our goal is to use edges of length $O(\sqrt{N}/k)$, but to increase the number of nodes which can be reached in k steps. For this, we use a technique of Paterson/Ruzzo/Snyder, who have shown an area/max. edge length tradeoff for complete binary trees whose leaves lie on the perimeteer of the smallest convex region enclosing all points of the embedding.

If $k \geq \Omega(N^{1/4})$ then the number of nodes increases at most by 1 from level i to the next level i+1 in the tree for all $1 \leq i$. Hence we can embed the tree with edge length O(1), since we can draw this tree having a horizontal and vertical distance of 1 between two adjacent nodes using the trivial embedding technique. In this case, we reach the lower bound. Now, we assume that $k < O(N^{1/4})$.

By the technique of Paterson/Ruzzo/Snyder, we embed the tree T in the channel partitioned into L-shapes (Fig. 7). At the end of the channel there are $\sqrt{N}/f(k)$ nodes, where f(k) is an expression depending on k.
We collect the leaves into groupes of $\sqrt{N}/k$, which lie at the end of each 'L'. So we see that the number of L's is $\sqrt{N}/f(k) \big/ \sqrt{N}/k = k/f(k)$.
Now we make the construction as shown in figure 8:
In k_2 steps we reach k/f(k) nodes and each of these nodes is the root of a subtree embedded in an L. To embed this first part of the subtree T we use edge lengths of O(k/f(k)) $=: L_0$.
To reach the nodes at the end of the channel we need k_2 steps and so we need an edge length of $\sqrt{N}/k_2 = L_1$.
Furthermore we know that the length of the edge which lies on the right channel and joins the smallest L and the root of the corresponding subtree, must not be greater than $O(\sqrt{N}/k)$ $=: L_2$. So we have to embed the tree that the number of nodes lying at the end of the horizontal legs of the L's is not greater than $\sqrt{N}/k$.
Now we have three sizes for the length of the longest edge: $O(\sqrt{N}/k)$, $O(\sqrt{N}/k_2)$ and $O(k/f(k))$.
A fourth size is the length of the edges in the horizontal part of the biggest L: $\sqrt{N}/f(k) \big/ (k_1-k_2)$.
So we have partitioned the tree in the following way and got the following upper bound:

$$O(\sqrt{N}/k_2 + \sqrt{N}/(f(k) \times (k_1-k_2)) + k/f(k))$$

(see figure 9).

If $f(k) > k_2 \times k/\sqrt{N}$, we have $O(\sqrt{N}/k_2)$, for the linear case $k \leqq N^{1/4} \times \sqrt{f(k)}$. We Know: $k = k_1 + k_2$, $k_1 > k_2$.

Now we ask what these bounds give us for some special cases:
For all cases we assume that $k_1 = c \times k$, $1/2<c<1$ constant and that $\sqrt{N}/k \geqq k/f(k)$. Hence $k_2 = (1-c) \times k$.

a) If $f(k) = N^{\varepsilon}$, $\varepsilon>0$, we have a linear area and the following theorem:

If we reach at most $N^{1/2-\varepsilon}$ nodes in k steps, we can embed this tree with a maximal edge length of $O(\sqrt{N}/k)$.

b) If $f(k) = c$, c constant, we have an area of $N \times \log^2 N$ and vertical length of the greatest L of $\sqrt{N} \times \log N$, and so

we yield an upper bound of $\sqrt{N} \times \log N/k$ for the edge length, if we reach $\sqrt{N}$ nodes in k steps.

c) If we have an area A, $A \gg N \times \log^2 N$, at our disposal, f.e. $A = N^{\alpha}$, $\alpha>1/2$, we can choose f(k) as $\sqrt{N} \times \log N/\sqrt{A}$, where $\sqrt{A}$ is the width of the top channel and $\sqrt{N}/f(k) = \sqrt{A}/\log N$ is the width of the biggest subtree T at the end of the channel. The length of the longest L is $\sqrt{A}$, and so

we have an upper bound of $O(\sqrt{A}/k)$ for the edge length, if we reach $\sqrt{A}/\log N$ nodes in k steps.

This can be better than the lower bound for the edge length, where we have a linear area; if we reach in k' steps at most N nodes, we have a lower bound of $O(\sqrt{N}/k')$ for the edge lenght and linear area.
Now it is possible that we have a class of trees where we reach in k' steps at most $\sqrt{N}$ nodes, but in k steps $\sqrt{A}/\log N$ nodes, and we can embed these trees with a maximal edge length $O(\sqrt{A}/(\log N \times k))$, which is less than $O(\sqrt{N}/k')$.

To see that this case is realistic, look at the following example (Figure 10):
$k' = N^{0,1}$, in $k = N^{0,25}$ steps we reach $N^{0,6} \log N$.
If we have linear area, we have a lower bound of $\Omega(N^{0,4})$ for the length of the longest edge, but if we have an area of $N^{1,2}$, we yield an upper bound of $O(N^{0,35})$, if we use

$O(\sqrt{A}/(\log N\times k))$ as upper bound.
For this case $f(k) = \log N/N^{0,1}$ and $k/f(k) = N^{0,35}\times\log N$, what is our upper bound for the edge length.

Conclusion

We have shown how to embed a tree with short edges, that we can nearly reach the lower bound for linear area and that there are some classes of trees, for that we can minimize the length of the longest edge by increasing the embedding area.

For any tree, we derived the following lower and upper bounds:

$$\max_{ST}\left\{ \frac{\sqrt{N_{ST}}}{k} \;\middle|\; \begin{array}{l} N_{ST} \text{ is the number of nodes in subtree ST and} \\ k \text{ is the number of steps reaching } \sqrt{N_{ST}} \text{ nodes} \end{array} \right\}$$

$$\min_{\text{decomp.}} \max_{ST}\left\{ \frac{\sqrt{N/2^i}/f(k)}{k} \;\middle|\; \begin{array}{l} \text{in } k \text{ steps we reach } \sqrt{N/2^i}/f(k) \text{ nodes} \\ \text{of ST, on level } i \text{ in the H-tree lie} \\ \text{its top edges, the used area determines} \\ \text{the factor } f(k) \end{array} \right\}$$

References

Bhatt, S.N. & Leiserson, C.E. (1982). Minimizing the longest edge in a VLSI Layout M.I.T. VLSI Memo

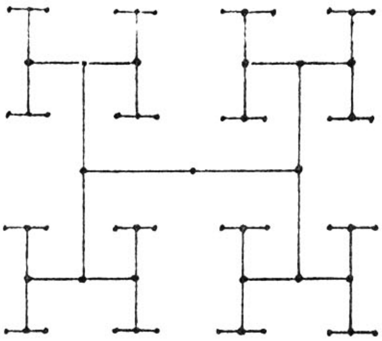
Leigthon, F.T. (1981). New lower bound tecchniques for VLSI. 22. Annual Symposium on Foundation of Computer Science IEEE

Leighton, F.T. (1982). A Layout Strategy for VLSI which is provably good. 14. Annual ACM Symosium on Theory of Computing

Mead, C. & Conway, L. (1980). Introduction to VLSI Systems. Addison-Wesley

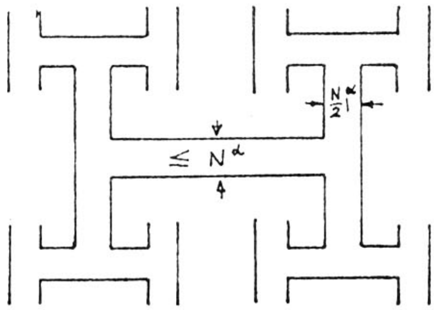
Paterson, M. , Ruzzo, W. & Snyder, L. (1981). Bounds on minimax edge length for complete binary trees. 13. Annual ACM Symposium on Theory of Computing

Thompson, C.D. (1979). Area-time complexity for VLSI. 11. Annual ACM Symposium on Theory of Computing



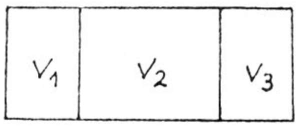
<u>Figure 1</u>

H-tree with 63 nodes



<u>Figure 2</u>

extended H-tree



<u>Figure 3</u>

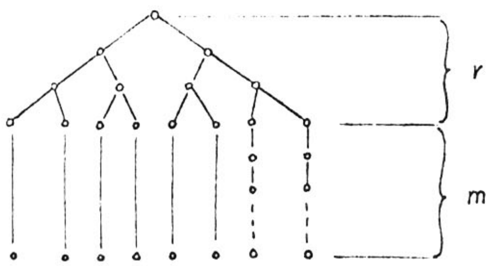


Figure 4

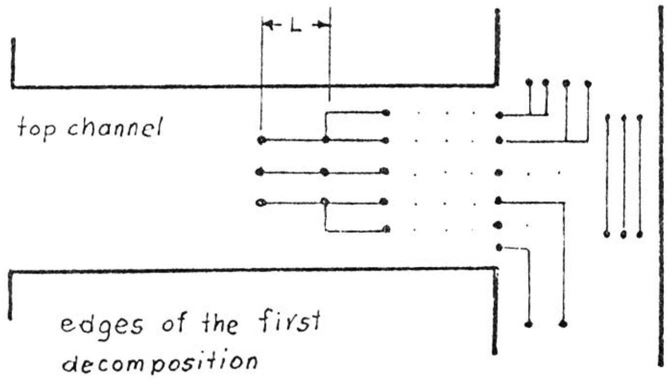


Figure 5

Embedding the tree in an H-tree

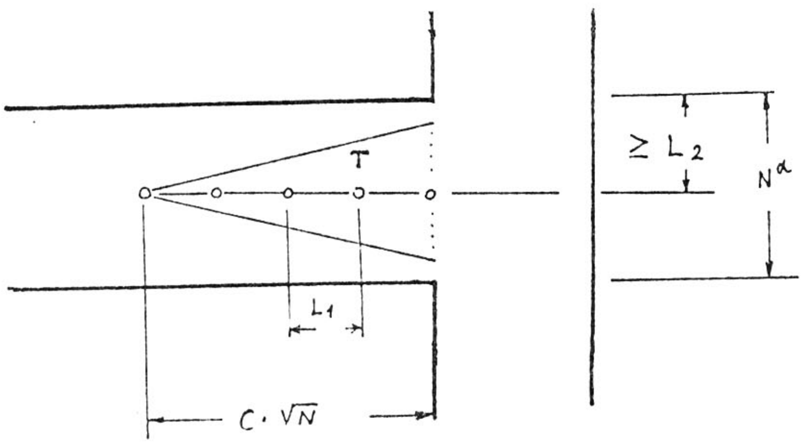


Figure 6

Embedding of subtree T

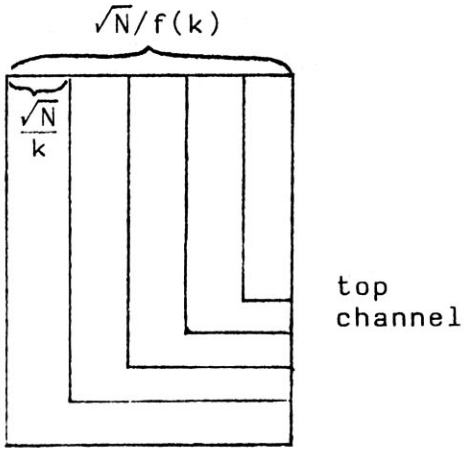


Figure 7

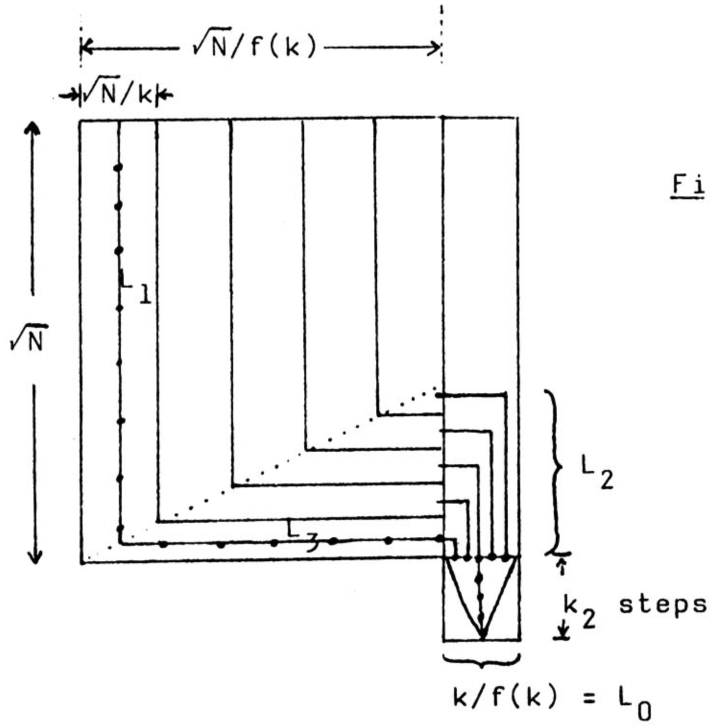


Figure 8

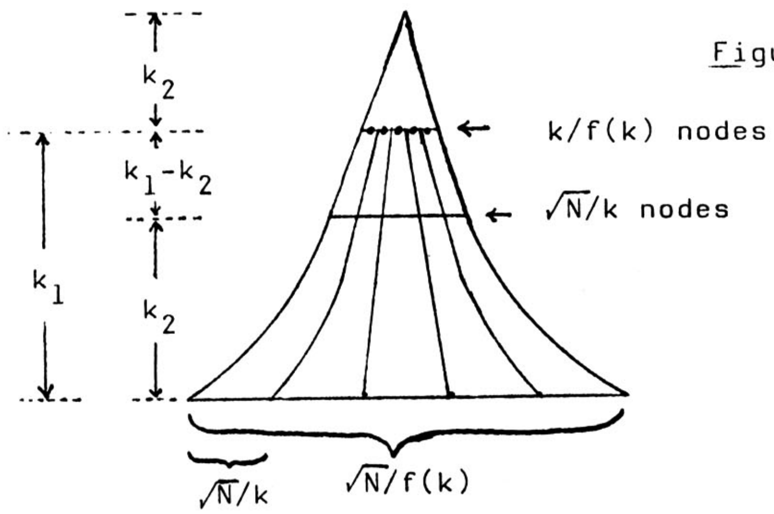


Figure 9

Figure 10