VERY THIN VLSI-LAYOUTS OF COMPLETE BINARY TREES

J. van Leeuwen and R.B. Tan

RUU-CS-86-7
June 1986

Rijksuniversiteit Utrecht

Vakgroep informatica

Budapestlaan 6 3584 CD Utrecht
Corr. adres: Postbus 80.012 3508 TA Utrecht
Telefoon 030-53 1454
The Netherlands

VERY THIN VLSI-LAYOUTS OF COMPLETE BINARY TREES

J. van Leeuwen and R.B. Tan

Technical Report RUU-CS-86-7
June 1986

Department of Computer Science
University of Utrecht
P.O.Box 80.012, 3508 TA Utrecht
the Netherlands

VERY THIN VLSI-LAYOUTS OF COMPLETE BINARY TREES*

Jan van Leeuwen and Richard B.Tan**

Department of Computer Science, University of Utrecht
P.O.Box 80.012, 3508 TA Utrecht, the Netherlands

Abstract. We give a minimum area VLSI-layout of a complete binary tree T_k with all 2^k leaves on one edge of a rectangular chip. This layout has the following additional properties: (i) there are no wire-crossings, (ii) the root of the tree is accessible and (iii) it has minimum possible width. It is shown that any minimum area VLSI-layout of T_k with all leaves collinear must have width $\lfloor\frac{k}{2}\rfloor+1$ and length

$2^k+2^{\lfloor k/2\rfloor-1}$ for k even and $2^k+2^{\lfloor k/2\rfloor-1}(1+\lfloor k/2\rfloor)$ for k odd.

1. Introduction. In his Ph.D.Thesis, Thompson developed a simple model for VLSI-circuit design based on circuit-drawings (embeddings) in the two-dimensional grid. Ever since, the embedding of a variety of circuits and graphs in a two-dimensional grid has been studied extensively, under a variety of further model-constraints. Typically the problem is to find a VLSI-layout with minimum area, minimum number of edge-crossings and/or minimum average or maximum edge length.

---

*This work was carried out while the second author visited the University of Utrecht in 1982, and completed in 1986 while supported by a grant from the Netherlands Organisation for the Advancement of Pure Research (ZWO).

**Author's address: Department of Mathematics and Computer Science, University of Sciences and Arts of Oklahoma, Chickasha, OK 73018, U.S.A.

Of particular interest is the layout problem for complete binary trees, because of the many applications of complete binary trees as computational structures. For instance, a complete binary tree can serve as a hardware structure for searching [1], databases problems [8] and general purpose multiprocessing [4]. (See [2] for a survey and references to further applications.)

Let T_k be the complete binary tree of depth k and $n=2^k$ leaves. Mead and Rem [7] have shown that one can embed T_k in O(n) area using the well-known "H"-pattern. However, the "H"-layout has only a small number of leaves on the boundary of the chip. There may be good reasons to put the leaves on the boundary, for example, when input/output is done through the leaves. Brent and Kung [3] have shown that it is not possible to design a complete binary tree layout of O(n) area with all the leaves on the boundary. In fact they show that O(nlogn) area is needed just for accomodating all the wires on the chip.

In this paper we study the question of embedding a complete binary tree T_k in optimum area with all leaves on one side of the rectangular boundary. In section 2 we give the exact lower bound for the thinnest possible width of a chip layout of T_k. An optimum layout when all leaves of T_k are packed as close as possible on one edge of the chip is also given. In section 3, we develop an optimum layout for a complete binary tree with all leaves on one edge of the chip with minimum width and no wire-crossing. The width of the layout of T_k is precisely $\lfloor\frac{k}{2}\rfloor+1$, and its area is $(\lfloor\frac{k}{2}\rfloor+1)(2^k+2^{\lfloor\frac{k}{2}\rfloor-1})$ for k even and $(\lfloor\frac{k}{2}\rfloor+1)(2^k+2^{\lfloor\frac{k}{2}\rfloor-1}(1+\lfloor\frac{k}{2}\rfloor))$ for k odd. Section 4 concludes with the optimum area for embedding any T_k with collinear vertices on a rectangular chip.

2. <u>Minimum area embedding of a complete binary tree with all leaves packed tight on one edge</u>. This section contains preliminary results that are needed later on.

Let $s(k)$ be the length of the short side and $l(k)$ the length of the long side of a rectangular chip containing an embedding of a binary tree T_k of depth k. We first give an exact lower bound on $s(k)$.

A lower bound of $s(k)\geq\lceil k/2\rceil$ is stated by Leiserson [6]. A careful analysis yields the bound of $s(k)\geq\lfloor k/2\rfloor+1$, and we show that this bound is achievable. We prove the result here in detail, as the technique employed will be used repeatedly.

Theorem 2.1. $s(k)\geq\lfloor k/2\rfloor+1$.

Proof.

We follow the proof by van Leeuwen [5], with a slight modification. Let the columns of the long side be numbered 1,2,3, ... Now define inductively a finite sequence of pairs of columns (d_j,e_j) with $j=0,1,2,\ldots$, such that (i) $d_0\leq d_1\leq\ldots$, (ii) $e_0\geq e_1\geq\ldots$, (iii) $d_j\leq e_j$, (iv) every vertical line positioned between columns d_j and e_j cuts through at least j independent paths in T_k and (v) there is a subtree T_{k-2j} which does not contain any of these paths and whose leaves are all located in columns d_j through e_j.

Taking $d_0=1$ and $e_0=l(k)$ gives a correct start. Assuming that d_j and e_j have been defined, the next pair is obtained as follows.

Consider the T_{k-2j} subtree that is associated with the j^{th} pair (in clause (v)) and split it into four T_{k-2j-2}'s. Let d_{j+1} be the first column $\geq d_j$ which contains a leaf of one of these subtrees and let e_{j+1} be the last column $\leq e_j$ which contains a leaf from a different one of these subtrees. The path connecting the two leaves through the root of T_{k-2j} (see figure 1) will be a $(j+1)^{st}$ path independent of the previous j and any one of the two T_{k-2j-2} subtrees that do not contain one of these leaves will satisfy clause (v).

A next pair can always be obtained as long as the subtree T_{k-2j} can be splitted into another four T_{k-2j-2}'s. Let J be the largest index for which a pair is obtained. Then $k-2J=0$ or $k-2J=1$ depending on whether k is even or odd, respectively. We thus have $J=\lfloor k/2\rfloor$ independent paths. Furthermore, by construction, there is at least a leftover subtree T_0 (if k is even) or T_1 (if k is odd) which does not contain any of these J independent paths. The vertical line that cuts through a node of the leftover T_0 or T_1 must pass through at least $\lfloor k/2\rfloor+1$ squares of edges and vertices. □

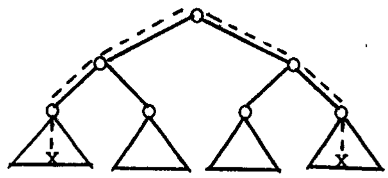


Figure 1.

Note that the above proof holds whether the leaves of the subtrees are interleaved or not.

<u>Theorem</u> 2.2. The bound of $s(k)=\lfloor k/2 \rfloor+1$ is exact.

<u>Proof</u>.

The rather complicated algorithm for generating the actual embedding of T_k with $s(k)=\lfloor k/2 \rfloor+1$ will be given in section 3. Figure 2 shows the embeddings of the first few k. □

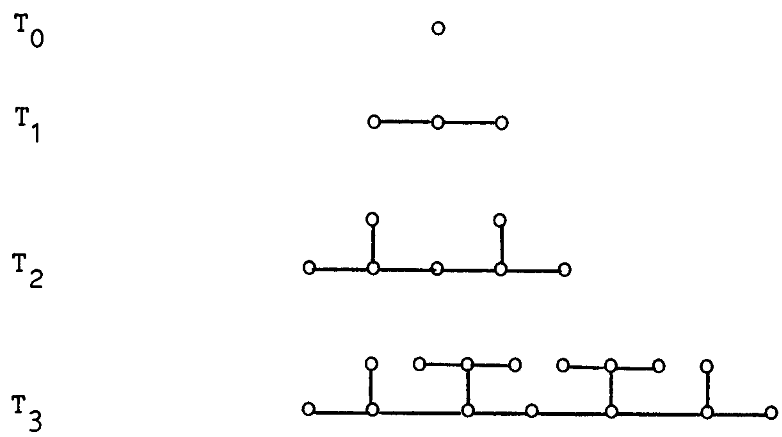


Figure 2.

Note that all the leaves in the minimum width embedding shown are located on the border as is the root, thus making the root very accessible. Most of the leaves are on one edge of the long side and can be arranged so that all the leaves are collinear. Furthermore, the layout contains no wire-crossing.

For the remainder of the paper we assume embeddings of T_k have all the leaves on the border of the chip and in fact collinear. Also the root must be easily accessible and located on the border.

We now ask the question: If the leaves of T_k are packed as tight as possible on one edge of the chip, what is the optimum layout?

As there are $n=2^k$ leaves in T_k, the long side of the chip must satisfy $l(k) \geq 2^k$. It turns out that the short side s(k) can no longer be the minimum given in section 2, but it need not be much larger.

<u>Theorem</u> 2.3. If T_k is embedded with collinear leaves in a rectangular chip of dimensions 2^kx s(k) then $s(k) \geq \lfloor k/2 \rfloor + 2$, $k \geq 1$.

<u>Proof</u>.

By applying the technique in theorem 2.1 of successively splitting T_k into four T_{k-2} subtrees with $k \geq 2$, we arrive at the following. There are at least $\lfloor k/2 \rfloor$ independent paths with the last remaining subtree being T_1 if k is odd or $\lfloor k/2 \rfloor - 1$ independent paths with the last subtree being T_2 if k is even. Now if the leaves are to be packed tight, and collinearly on one edge of the chip, then T_1 requires at least 2 more rows and T_2 at least 3 more rows (see figure 3). Thus altogether we must have at least $\lfloor k/2 \rfloor + 2$ rows. □

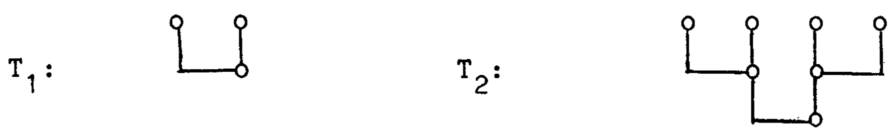
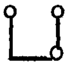


Figure 3.

The bound on s(k) for this case turns out to be again exact.

<u>Theorem</u> 2.4. T_k can be embedded with collinear leaves in a rectangular chip of dimensions 2^kx s(k), where $s(k) = \lfloor k/2 \rfloor + 2$, $k \geq 1$. Furthermore, the root of T_k is located on the edge and there are no wire-crossings.

<u>Proof</u>.

It it easily verified that T_1:  satisfies the condition. We proceed by induction on k. Assume we can construct the required embedding with $s(k)=\lfloor k/2\rfloor+2$ for T_k, $k\geq 1$. If k is odd then $s(k+1)=s(k)+1$, so the short side increases, and we can construct T_{k+1} as in figure 4. Note that the root does not lie on the last column of the chip. If k is even then $s(k+1)=s(k)$ and we cannot increase the short side. T_{k+1} can be constructed as in figure 5 by joining two T_k's constructed for the case k odd as in figure 4. □

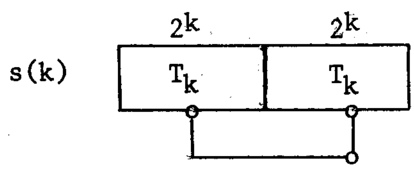


Figure 4.

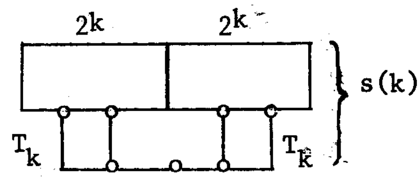


Figure 5.

<u>Corollary</u> 2.5. If T_k is embedded in a rectangular chip with all the leaves packed tight on one edge of the chip, then the minimum area is $(\lfloor k/2\rfloor+2)\cdot 2^k$.

3. <u>Minimum area embedding of T_k with minimum width and all leaves on one edge of the chip</u>. We now concentrate on the construction of embeddings of T_k with the thinnest possible width, i.e., with $s(k)=\lfloor k/2\rfloor+1$. The leaves of T_k are assumed to lie on the border of the chip and can be made collinear, but are no longer required to be packed "tightly". First we shall compute the minimum length $l(k)$ and then provide an optimum layout.

It is convenient to distinguish two cases and to settle the question for even k first.

<u>Case I</u>. k is even.

<u>Theorem</u> 3.1. If k is even and $s(k)$ is minimum then $l(k)\geq 2^k+2^{\lfloor k/2\rfloor-1}$, $k\geq 2$.

<u>Proof</u>.

By applying the technique in theorem 2.1 and succesively splitting T_k into four T_{k-2} subtrees we eventually obtain $\lfloor k/2 \rfloor -1$ independent paths with the last remaining subtrees being T_2. Each time we split a T_k, we obtain two more T_{k-2} subtrees that are left untouched by the previous independent paths. We can apply the splitting technique recursively to both of the remaining subtrees. Thus at the end of $\lfloor k/2 \rfloor -1$ splittings, we have $2^{\lfloor k/2 \rfloor -1}$ T_2 subtrees that are obtained in this recursive process. Now there are two more rows left to accomodate these T_2's. The thinnest embedding of T_2 that uses two rows requires an extra column to accomodate the root of T_2 (see figure 2).

Thus altogether we have 2^k leaves plus $2^{\lfloor k/2 \rfloor -1}$ extra columns for the roots of the remaining T_2's. □

Theorem 3.2. For $k \geq 2$ and even, there is an embedding of T_k with colinear leaves in a rectangular chip of dimensions $s(k) \times l(k)$, where $s(k)$ is the minimum width $\lfloor k/2 \rfloor +1$ and $l(k)$ is the minimum length $2^k + 2^{\lfloor k/2 \rfloor -1}$. Furthermore, this embedding has no wire-crossings and has the root on an edge of the chip.

Proof.

We construct the layout inductively on k. For k=2, l(2)=5 and figure 2 provides a correct layout. Assume we can embed T_k for even $k \geq 2$. Then T_{k+2} can be constructed as in figure 6.

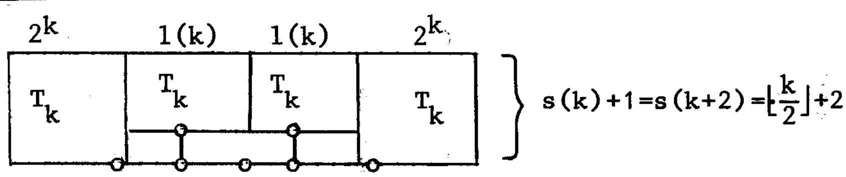


Figure 6.

The two middle T_k's are obtained from the induction hypothesis, with no wire-crossings and the root located on the edge, and the two outer T_k's are the subtrees derived from theorem 2.4 where the leaves are packed tight. Note that in the construction of the embedding for T_k in theorem

2.4 (cf. figure 4) the root node is accessible to the left or right. Finally, observe that $l(k+2)=2\cdot 2^k+2\cdot(2^{\lfloor k/2\rfloor -1}+2^k) = 2^{k+2} = 2^{\lfloor (k+2)/2\rfloor -1}$, as claimed. □

<u>Corollary</u> 3.3. For $k\geq 2$ and even, an embedding of T_k with collinear leaves in a rectangular chip with minimum width has minimum area $(\lfloor k/2\rfloor +1)\times(2^k+2^{\lfloor k/2\rfloor -1})$.

<u>Case II</u>. k is odd.

We now settle the case for odd k. Unfortunately, a proof and construction similar to the preceding case will not work for odd k. This is because the root of T_k for odd k is not accessible to the left or right as we will show below (compare the construction in theorem 2.4, cf. figure 5). Thus we cannot apply a similar construction as in theorem 3.2 (cf. figure 6), and apparently the outer two T_k subtrees can afford to have all their leaves packed tight. We formalize this notion precisely in the following proposition.

<u>Proposition</u> 3.4. For $k\geq 3$ and odd, in any embedding of T_k with all the leaves packed tight on one edge of the chip, i.e., $l(k)=2^k$ and $s(k)=\lfloor k/2\rfloor +2$, the root of T_k is not accessible to the left or right.

<u>Proof</u>.

Note that the dimensions stated are minimum for embeddings with tightly packed leaves on one edge, as given in corollary 2.5. Suppose by way of contradiction that there is an embedding of T_k for some $k\geq 3$ and odd which has the root accessible to the left or right, say as in figure 7.

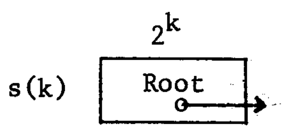


Figure 7.

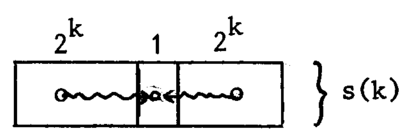


Figure 8.

We can now construct an embedding of T_{k+1} by joining two of these odd T_k's together and adding an extra column for the root of T_{k+1}, as in figure 8. Note that $s(k)=\lfloor k/2\rfloor+2=\lfloor (k+1)/2\rfloor+1$, as k is odd. This is the minimum short side required for any embedding of T_{k+1}. But theorem 3.1 claims that $l(k+1)\geq 2^{k+1}+2^{\lfloor (k+1)/2\rfloor-1}$, whereas the constructed embedding has $l(k+1)=2^k+1+2^k=2^{k+1}+1$. Contradiction. □

The proposition implies that we definitely have to follow a different construction. As the width s(k) is minimum and fixed, we must increase the length l(k).

<u>Theorem</u> 3.5. If $k\geq 3$ and odd, and s(k) is minimum, i.e., $s(k)=\lfloor k/2\rfloor+1$, then $l(k)\geq 2^k+2^{\lfloor k/2\rfloor-1}(1+\lfloor k/2\rfloor)$ for any embedding of T_k with all leaves on one edge.

<u>Proof</u>.

By successively splitting T_k into four T_{k-2}'s we obtain a sequence of T_{k-2j}'s for j from 1 to $\lfloor k/2\rfloor$, where the inner pair of the T_{k-2j}'s occupy space of minimum width and the outside pair have one extra row space.

We first count the number of extra columns (columns that contain no leaves) needed in the embedding.

<u>Claim 1</u>. There are at least $2^{\lfloor k/2\rfloor}$ extra columns needed in the embedding.

We get this by successively splitting the two inner T_{k-2j}'s for j from 1 to $\lfloor k/2\rfloor$ until we have only T_1's left. This requires $\lfloor k/2\rfloor$ steps and thus we have at the end $2^{\lfloor k/2\rfloor}$ T_1's obtained from the inner subtrees. Now there is only one row space left to accomodate these T_1's and thus T_1 must be o—o—o. In other words, one of the column spaces must be taken up by the root of T_1. Hence the claim.

Next, we proceed to count the extra columns needed in the outer T_{k-2j}'s for j from 1 to $\lfloor k/2\rfloor-1$, where the last one is T_3.

<u>Claim 2</u>. There are at least $2^{\lfloor k/2\rfloor-1}\cdot(\lfloor k/2\rfloor-1)$ extra columns needed to accomodate the outer subtrees. Note that the above extra columns are in addition to the ones in claim 1.

According to proposition 3.4, even though each outer T_{k-2j} for j from 1 to $\lfloor k/2 \rfloor -1$ has an extra row space, the root of the subtree is not accessible to the left or right. Split each of the outer T_{k-2j}'s for j from 1 to $\lfloor k/2 \rfloor -1$ into two T_{k-2j-1} subtrees. At least one of the T_{k-2j-1}'s for j from 1 to $\lfloor k/2 \rfloor -1$ must have one less row space on account of the accessibility of the root to the left or right. Thus the short side available for this T_{k-2j-1} is the minimum width. As k-2j-1 is even, the extra columns needed to accomodate this T_{k-2j-1} is at least $2^{\lfloor (k-2j-1)/2 \rfloor -1} = 2^{\lfloor k/2 \rfloor -j-1}$, by theorem 3.1. At each splitting of the T_{k-2j} subtrees for j from 1 to $\lfloor k/2 \rfloor -1$ there are 2^j outer T_{k-2j}'s, thus we have $2^{\lfloor k/2 \rfloor -1}$ extra columns for each splitting and the claim follows.

Summing the extra columns in claim 1 and claim 2, we obtain $2^{\lfloor k/2 \rfloor} + 2^{\lfloor k/2 \rfloor -1}(\lfloor k/2 \rfloor -1) = 2^{\lfloor k/2 \rfloor -1}\ (1+\lfloor k/2 \rfloor)$. Adding 2^k columns for the leaves we have the theorem. □

<u>Theorem</u> 3.6. For k odd, there is a minimum embedding of T_k with collinear leaves in a rectangular chip of dimensions s(k)x l(k), where s(k) is the minimum width $\lfloor k/2 \rfloor +1$ and l(k) is the minimum length. Furthermore, this embedding has no wire-crossings and has the root on an edge of the chip.

<u>Proof</u>.

We construct the embedding inductively on k. For k=1, T_1: o–o–o is minimum. An optimal embedding of T_3 is formed as in figure 9.

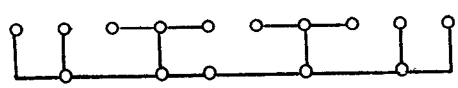


Figure 9.

Observe that the length l(3)=10, which is the minimum required by theorem 3.5.

Assume we can construct an embedding of T_k as desired for some odd k≥3. Then T_{k+2} can be constructed as shown in figure 10.

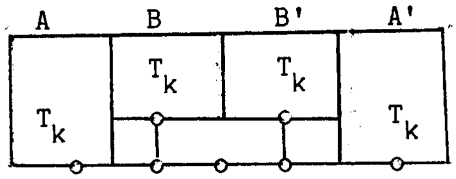


Figure 10.

The B and B' parts are embeddings of T_k obtained by induction. The A and A' parts are embeddings of T_k formed by using the following figure 11.

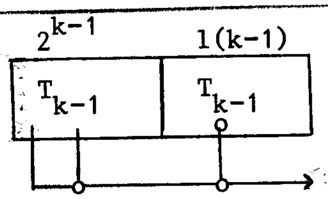


Figure 11.

In figure 11, the left T_{k-1} subtree is constructed by using the construction of theorem 2.4 (cf. figure 4) and has all the leaves packed tight with length 2^{k-1}. The right T_{k-1} subtree is constructed by using theorem 3.2 (cf.figure 6) and has length $l(k-1)=2^{\lfloor(k-1)/2\rfloor-1}+2^{k-1}$. Thus the total length of the A part as given in figure 11 is $2^k+2^{\lfloor k/2\rfloor-1}$.

Observe that the length of T_{k+2} is twice that of part A and B, so that

$$\begin{aligned} l(k+2) &= 2\cdot(2^k+2^{\lfloor k/2\rfloor-1})+2\cdot(2^k+2^{\lfloor k/2\rfloor-1}(1+\lfloor k/2\rfloor)) \\ &= 2^{k+2}+2^{\lfloor k/2\rfloor+1}+2^{\lfloor k/2\rfloor}\lfloor k/2\rfloor \\ &= 2^{k+2}+2^{\lfloor (k+2)/2\rfloor-1}(1+\lfloor (k+2)/2\rfloor), \end{aligned}$$

which is minimum by theorem 3.5. □

Corollary 3.7. For $k\geq 3$ and odd, an embedding of T_k with all leaves along one edge of the rectangular chip and with minimum width has minimum area $(\lfloor k/2\rfloor+1)\times(2^k+2^{\lfloor k/2\rfloor-1}(1+\lfloor k/2\rfloor))$.

4. Minimum area embedding of T_k with minimum width and all leaves collinear.

Let $A(k)$ be the minimum area of a rectangular embedding of T_k with all leaves collinear. Brent and Kung [3] have shown that the area needed to accomodate the wires of T_k is $\geq \frac{1}{6}k2^k$. Thus $A(k) \geq \frac{1}{6}k2^k$. In this section we give the exact minimum area required for such a layout.

To ease our subsequent analysis, we let $A_1(k)=s_1(k) x l_1(k)$ be the area of the rectangular embedding given in corollaries 3.3 and 3.7 with minimum width $s_1(k)=\lfloor k/2 \rfloor+1$ and all leaves on one edge of the chip. Also let $A_2(k)=s_2(k) x l_2(k)$ be the area of an embedding with all the leaves packed tight, where $l_2(k)=2^k$ and $s_2(k)=\lfloor k/2 \rfloor+2$, as given in corollary 2.5.

<u>Lemma</u> 4.1. $A_1(k)<A_2(k)$, for all $k \geq 1$.

<u>Proof</u>.

We distinguish between k even and k odd.

<u>Case I</u>: k is even.

Then
$$\begin{aligned} \Delta &= A_2(k)-A_1(k)= \\ &= (\lfloor k/2 \rfloor+2)\cdot 2^k-(\lfloor k/2 \rfloor+1)\cdot(2^k+2^{\lfloor k/2 \rfloor-1})= \\ &= 2^{k+1}-2^k-2^{k/2-1}(k/2+1)= \\ &= 2^{k/2-1}(2^{k/2+1}-(k/2+1)), \end{aligned}$$
and $\Delta>o$ as $2^m>m$ for all $m \geq o$.

<u>Case II</u>: k is odd.

If $k=1$ then $A_2(k)=4$ and $A_1(k)=3$ (cf. figure 2 and figure 4), so $A_1(k)<A_2(k)$. Assume $k \geq 3$.

Then
$$\begin{aligned} \Delta &= A_2(k)-A_1(k)= \\ &= (\lfloor k/2 \rfloor+2\cdot 2^k-(\lfloor k/2 \rfloor+1)\cdot(2^k+2^{\lfloor k/2 \rfloor-1}(1+\lfloor k/2 \rfloor))= \\ &= 2^{k+1}-2^k-(\lfloor k/2 \rfloor+1)^2\cdot 2^{\lfloor k/2 \rfloor-1}= \\ &= 2^{\lfloor k/2 \rfloor-1}(2^{\lfloor k/2 \rfloor+2}-(\lfloor k/2 \rfloor+1)^2), \end{aligned}$$
as k odd, $k = \lfloor k/2 \rfloor-1+\lfloor k/2 \rfloor+2$.

Again $\Delta>o$ as $2^{m+1}>m^2$ for all $m \geq o$. □

We can now give the exact value for the minimum area $A(k)$.

<u>Theorem</u> 4.2. The minimum area of any rectangular embedding of T_k with all leaves collinear is $A(k)=A_1(k)$.

Proof.

Suppose $A(k)=s(k)\times l(k)$ is the minimum area. Then by theorem 2.1, the width $s(k)\geq\lfloor k/2\rfloor+1=s_1(k)$, so that $s(k)=s_1(k)+x$ where $x\geq o$. Also, the length $l(k)\geq 2^k=l_2(k)$, thus $l(k)=l_2(k)+y$, for some $y\geq o$.

$$\begin{aligned} \text{If } x>o \text{ then } A(k) &\geq (s_1(k)+x)\cdot l_2(k) \\ &\geq A_2(k), \text{ since } s_2(k)=s_1(k)+x \\ &> A_1(k), \text{ by lemma 4.1.} \end{aligned}$$

As $A(k)$ is minimum, this is a contradiction. So $x=o$ and $s(k)=s_1(k)$, the minimum width. But then corollaries 3.3 and 3.7 state that $l(k)=l_1(k)$. □

Thus the minimum rectangular area embedding of T_k with collinear leaves is actually the one with the minimum width. Furthermore, theorems 3.2 and 3.6 give recursive layouts for such an embedding and the layouts can be made with no wire-crossings. In fact the layouts given can be rearranged such that there are leaves on all three edges of the rectangle with the root accessible on the edge of the remaining side.

References.

[1] Bentley, J.L. and H.T. Kung, A Tree Machine for Searching problems, Proceedings IEEE International Conference on Parallel Processing, 1979, pp. 257-266.

[2] Bhatt, S.N. and C.E. Leiserson, How to Assemble Tree Machines, Proceedings 14^{th} Annual ACM Symposium on Theory of Computing, 1982, pp. 77-84.

[3] Brent, R.P. and H.T. Kung, On the Area of Binary Tree Layouts, Information Processing Letters, No.11, 1980, pp. 44-46.

[4] Browning, S., The Tree Machine: A Highly Concurrent Computing Environment, Ph.D.Thesis, California Institute of Technology,

Pasadena, CA, 1980.

[5] Leeuwen, J. van, The Bounded Aspect Ratio Problem for VLSI-layouts of Perfect Binary Trees, Technical Report RUU-CS-81-16, Department of Computer Science, University of Utrecht, 1981.

[6] Leiserson, C.E., Area-Efficient Graph Layouts (for VLSI), Proceedings 21st Annual IEEE Symposium on Foundations of Computer Science, 1980, pp. 270-281.

[7] Mead, C.A. and M.Rem, Cost and Performance of VLSI Computing Structures, IEEE J.Solid State Circuits, Volume SC-14, No.2 (1979), pp. 455-462.

[8] Song, S.W., A Highly Concurrent Tree Machine for Database Applications, Proceedings IEEE International Conference on Parallel Processing, 1980, pp. 259-268.

[9] Thompson, C.D., A Complexity Theory for VLSI, Ph.D. dissertation, Department of Computer Science, Carnegie-Mellon University, 1980.