Report Number: WUCS-84-7

1984-10-01

Reduction of Clock Delays in VSLI Structures

Sanjay Dhar, Mark A. Franklin, and Donald F. Wan

With the growth in chip size and reduction in line width, delays in driving long lines have become increasingly important in determining overall chip level performance. In synchronous systems the proper distribution of the clock signal is critical in determining system throughput. This paper considers the problem of optimal driving clock lines. A general delay model is developed and applied to a clock tree where the path distances from the root node to each of the leaf nodes are all equal. This strategy reduces clock skew and increases clock rates. A tree delay model is developed and is used... Read complete abstract on page 2.

Reduction of Clock Delays in VSLI Structures

Sanjay Dhar, Mark A. Franklin, and Donald F. Wan

Complete Abstract:

With the growth in chip size and reduction in line width, delays in driving long lines have become increasingly important in determining overall chip level performance. In synchronous systems the proper distribution of the clock signal is critical in determining system throughput. This paper considers the problem of optimal driving clock lines. A general delay model is developed and applied to a clock tree where the path distances from the root node to each of the leaf nodes are all equal. This strategy reduces clock skew and increases clock rates. A tree delay model is developed and is used to determine the optimal number and placement of buffers within the tree so that the clock delay is minimized. AN example of a clock tree driving a synchronous crossbar network is provided, and minimum delay and corresponding number of buffers are indicated as a function of the minimum line width and network size. For a 64*64 network this minimization technique yielded an order of magnitude delay reduction over standard single exponential buffer usage.

REDUCTION OF CLOCK DELAYS IN VSLI STRUCTURES

SANJAY DHAR, MARK A. FRANKLIN
and DONALD F. WANN

WUCS-84-7

Proceedings
IEEE International Conference on
Computer Design
ICCD '84
New York
Oct. 8-11, 1984

DEPARTMENT OF COMPUTER SCIENCE
WASHINGTON UNIVERSITY
SAINT LOUIS, MISSOURI 63130

October 1984

This research has been sponsored in part by funding from
ONR Contract N000014-8D-C-0761 and NSF Grant MCS-78-20731

Proc. IEEE International
Conf. on Computer Design
ICCD '84
New York, Oct. 8-11,1984

REDUCTION OF CLOCK DELAYS IN VLSI STRUCTURES

Sanjay Dhar, Mark A. Franklin and Donald F. Wann

Center for Computer Systems Design
Washington University,
St. Louis, MO. 63130

ABSTRACT

With the growth in chip size and reduction in line width, delays in driving long lines have become increasingly important in determining overall chip level performance. In synchronous systems the proper distribution of the clock signal is critical in determining system throughput. This paper considers the problem of optimally driving clock lines. A general delay model is developed and applied to a clock tree where the path distances from the root node to each of the leaf nodes are all equal. This strategy reduces clock skew and increases clock rates. A tree delay model is developed and is used to determine the optimal number and placement of buffers within the tree so that the clock delay is minimized. An example of a clock tree driving a synchronous crossbar network is provided, and minimum delay and corresponding number of buffers are indicated as a function of the minimum line width and network size. For a 64*64 network this minimization technique yielded an order of magnitude delay reduction over standard single exponential buffer usage.

1.0 Introduction:

Developments in VLSI fabrication technology have rapidly increased the size of chips as well as reduced minimum line width. As this continues, delay in control and data lines is becoming critical in designing high performance systems [SINH82, CART84]. Driving long lines can be especially acute in globally synchronized clocked systems. Two such systems whose clock distribution properties have been analyzed are discussed in FISH83 and WANN83. These studies indicate a close relationship between delays in clock lines and system performance (data rate). To reduce clock skew the clock distribution often takes the form of a H-tree [WANN83]; Kung [KUNG82] has shown that the delay in a full H-tree increases as $O(N^2)$ where N is the network size. This paper presents a technique for minimizing delays in clock trees. The procedure is based on the sectioning and inserting of buffers in the clock line to minimize the propagation delay, and yields the number of buffers, their position and size. In a 64*64 crossbar network example an order of magnitude reduction in the clock distribution delay is obtained over the single optimum exponential buffer case.

The propagation delay along a single section of a line is first modeled by a buffer driving a line and a load. The section delay is minimized, and the total line is then divided into a number of sections such that the total line delay is minimized. This technique is then applied to a clock tree, which is first reduced to a single electrically equivalent line by repeatedly folding the tree. The number of buffers, their position and size that minimize the delay are obtained for the folded clock tree. These folded line parameters are mapped back into equivalent parameters for the unfolded clock tree preserving the minimum delay.

2.0 Delay Model of a Section:

The basic circuit configuration is shown in Figure 1a and consists of (a) a source which is an NMOS transistor with pullup to pulldown ratio of k, a gate capacitance of C_{gs}, and a pulldown resistance of R_{gs}; (b) a line of total length L with distributed resistance and capacitance lumped into values R_L and C_L; and (c) an NMOS transistor load with a gate capacitance of C_{gl}. The inductance of the line is neglected and it is assumed that the width and material of the line are uniform and fixed.

In general k is greater than 1 and hence the largest propagation delay occurs when point P_3 is charged to the power supply voltage. In this analysis the worst case pullup delay will be used, although if superbuffers were used the smaller pulldown delay would be appropriate. (It is also possible to employ the pair delay). The line is modeled by a series resistor and shunt capacitor. The load (gate capacitance of the next transistor) is in parallel with the line capacitance.

The complete delay model for a section is shown in Figure 1b. The signal propagation time from point P_1 to point P_3, defined as the product of the equivalent resistance and capacitance of the section, is given by

$$d_s = (k*R_{gs} + R_L)*(C_{gl} + C_L)$$

$$= k*R_{gs}*(C_{gl} + C_L) + R_L*C_L + R_L*C_{gl} \qquad (1)$$

The first term is the delay in driving the line and load transistor capacitances by the source

* This research has been sponsored in part by funding from ONR Contract N00014-8D-C-0761 and NSF Grant MCS-78-20731.

transistor. The second and third terms represent the delays involved in driving the line and load capacitances, (C_L and C_{gl}) by the line resistance (R_L). For a given L, since the width and material of the line are fixed, these two terms cannot be reduced any further. For a long line, C_L will be large and the delay associated with the first term can be minimized by using a technique proposed by [JAEG75]. A series of buffers starting with a minimum size buffer and exponentially increasing in size is used (Figure 2). The delay d_C in driving a capacitance C is then given by

$$d_C = k*tau*e*\ln(C/C_{gs}) \quad (2)$$

where tau is the product of the resistance R_m and capacitance C_m of a minimum size transistor.

This technique is applied to the section in Figure 1a for driving the capacitance ($C_L + C_{gl}$) giving the minimized section delay as

$$d_s = k*tau*e*\ln\{(C_L+C_{gl})/C_{gs}\} + R_L*C_L+R_L*C_{gl} \quad (3)$$

Consider a line of total length L' with x indicating distance along the line, $0 <= x <= L'$. Let R(x) and C(x) describe the variations of the resistance and capacitance. The line is divided into n sections with the position of the end of section i given by L_i, i=1,..,n (Figure 3). Notice that the load transistor of section i is also the source transistor for section (i+1) and $L_0 = 0$. The delay d_i in section i is given by

$$d_i = k*tau*e*\ln[\{C(L_i)-C(L_{i-1})+C_{gs(i+1)}\}/C_{gsi}] + [R(L_i)-R(L_{i-1})]*[C(L_i)-C(L_{i-1})] + [R(L_i)-R(L_{i-1})]*C_{gs(i+1)}, \quad i=1,..,n \quad (4)$$

The total line delay, defined as the sum of the individual delays in all the sections, is given by:

$$d = \sum_{i=1}^{n} (k*tau*e*\ln[\{C(L_i)-C(L_{i-1})+C_{gs(i+1)}\}/C_{gs(i+1)}]) + \sum_{i=1}^{n} (\{R(L_i)-R(L_{i-1})\}*\{C(L_i)-C(L_{i-1})\}) + \sum_{i=1}^{n} (\{R(L_i)-R(L_{i-1})\}*C_{gs(i+1)}) \quad (5)$$

The total delay is seen to be a function of the lengths of the individual sections as well as the number of sections. Assume that the number of sections into which the line is divided is known, then the partial derivative of d with respect to L_i, i=1,...,n can be determined, set equal to zero, and the resulting equations solved to obtain L_i, i=1,...,n. Thus, the lengths of the individual sections that give the minimum delay can be obtained for a given value of n. A global minimum is determined by repeated application of this process with varying n.

3.0 Clock Delay in Interconnection Networks:

Figure 4 shows the structure of an N*N synchronously controlled crossbar network built from 2*2 modules [DHAR83]. The analysis presented here focuses on minimizing the delay in the clock distribution. As shown in WANN83, unequal delays in the clock line give rise to clock skew, which reduces the throughput of the network. Using a tree structure ensures that all clock line lengths are equal, thus reducing clock skew and increasing throughput.

In general, large interconnection networks are built from many chips and the clock distribution tree consists of a part that resides on the board containing the chips and distributes the clock to each individual chip, and a part which distributes the clock to submodules within a chip. The part of the tree residing on the board consists of metal lines having negligible resistance; preliminary analysis indicates that the delay in this part of the tree is small compared to the internal chip delay and hence can be neglected. The analysis presented here is restricted to the clock tree internal to a chip and to network chips of sizes N*N, $N = 2^m$, m an integer.

Figure 4 shows the clock distribution tree for an 8*8 network. Due to the horizontal and vertical symmetry, the voltage at all points in the tree equidistant from the root must be equal at all times and all such points can be connected without altering electrical behaviour. Thus, two or more identical parts of the tree can be folded into one single structure. In doing so, the resistance of the folded tree decreases whereas the capacitance increases by the same factor. The tree for an N*N network can be reduced to a single line by repeated folding (Figure 5). The resulting structure has 2m+1 sections, with the resistance per unit length decreasing by a factor of 2 from one section to the next and the capacitance per unit length increasing by a factor of 2. Figure 6 shows the variation of capacitance, resistance and depth of fold with the length of the line. The depth of fold of a section of the folded line is defined as the number of sections of the original tree that were folded to form the equivalent section of the folded line. Notice that the depth of fold is constant over a given section of the folded tree, and increases by a factor of two from one section to the next.

While the equations previously developed for minimizing the delay are easily applied to continuous functions, the minimization procedure becomes complex and computationally intensive if the discontinuous functions of Figure 6 are used. Hence, these functions are approximated by the continous functions given below

$$C_a(x)=a_1(e^{b_1*x}-1)+a_2(e^{b_2*x^2}-1)+\ldots+a_5(e^{b_5*x^5}-1) \quad (8)$$

$$R_a(x)=c_1(1-e^{d_1*x})+c_2(1-e^{d_2*x^2})+\ldots+c_4(1-e^{d_4*x^4}) \quad (9)$$

$$N_a(x)=g_1*e^{h_1*x}+g_2*x*e^{h_2*x^2}+\ldots+g_5*x^4*e^{h_5*x^5} \quad (10)$$

Once the fabrication parameters are known (e.g. resistance and capacitance of the line material), the coefficients (i.e., $a_i, b_i, \ldots, h_i$) above can be found via a computer program.

3.1 Minimization of Clock Line Delay:

In order to apply equation (5), the source capacitances C_{gsi}, i=1,....,n have to be obtained. Consider the effect of increasing the value of C_{gsi} on the delay of the $(i-1)^{th}$ and i^{th} sections. The delay in the $(i-1)^{th}$ section increases because the capacitance in this section increases, while the delay in the i^{th} section decreases since the buffer is larger. Evidently, there is an optimum value that results in minimum delay. To determine this optimum value of C_{gsi} requires the minimization process to treat C_{gsi} as an independent variable. While this could be done in a more general analysis, in this analysis we assume that all buffers in the actual (not folded) clock tree have a minimum size transistor at their input. Let the depth of fold at point x be N(x). Then in the folded clock line the input capacitance of a buffer located at distance x from the root will be $N(x)*C_m$.

Next consider the effect of dividing the folded clock line into n sections in the manner described in section 2.0. The source capacitance of the i^{th} section, C_{gsi}, is then given by

$$C_{gsi} = N(L_1+...+L_{i-1})*C_m, \quad i=1,...,n \qquad (11)$$

This expression for C_{gsi} can now be subsituted into (5); the expressions for capacitance, resistance and depth of fold functions also will be replaced by their approximate functions, $C_a(x)$, $R_a(x)$ and $N_a(x)$ respectively yielding the final approximation to the delay as

$$d = \sum_{i=1}^{n} (k*tau*e*\ln[\{C_a(L_i) - C_a(L_{i-1}) + N_a(L_{i-1})*C_m\}/N_a(L_i)*C_m]) + \sum_{i=1}^{n} (\{R_a(L_i) - R_a(L_{i-1})\}*\{C_a(L_i) - C_a(L_{i-1})\}) + \sum_{i=1}^{n} (\{R_a(L_i) - R_a(L_{i-1})\}*N_a(L_i)*C_m) \qquad (12)$$

where $L' = L_i$ (13)

From (12) and (13) the positions of the individual sections L_i, i=1,...,n that give the minimum delay can be determined using standard continuous function computer optimization schemes.

Let the resistance per square be denoted by R_{sq}, and the capacitance per square micron by C_{sq}. The typical values of R_{sq} and C_{sq} for a 5 micron linewidth NMOS technology are given in Table 1 [MEAD80]. Let lambda denote the minimum resolution of the process (minimum line width is 2*lambda). Let the parameters with a superscript (′) indicate the value of the scaled parameters. Then the scaling of the key parameters as lambda scales from lambda to k_s*lambda ($k_s < 1$) are as follows [MEAD80]:

$$R_{sq}' = R_{sq}/k_s \qquad C_{sq}' = C_{sq}/k_s$$

$$R_m' = R_m \qquad C_m' = k_s*C_m$$

Material	R_{sq} ohms/square	C_{sq} pF/sq. micron
Metal	0.03	0.00003
Polysilicon	15-100	0.00004
Diffusion	10	0.0001

Table 1

Furthermore, assume that the largest square chip that can fabricated be with adequate yield, has a side dimension less than or equal to 1.5 cm. Also assume that this restriction applies as the linewidth decreases. Experience obtained from designing a 2*2 network module [DHAR82] indicates that for a technology with lambda of 2.5 microns, a 32*32 network will occupy a chip area of about 2.25 sq. cm. If L_{ch} is the length of a square chip occupied by an N*N network with a line resolution of lambda, then the length of a square chip L_{ch}' occupied by an N′*N′ network with a line resolution of k_s*lambda is given by $L_{ch}' = L_{ch}*N'*k_s/N$.

Next the clock line parameters will be selected assuming that the fabrication technology provides only one layer of metal. Clock line routing will likely be laid out in part on a material other than metal. In the example presented here, 90% of the clock line is metal and 10% is diffusion.

Consider equation (12) and (13). The functions $C_a(x)$, $R_a(x)$ and $N_a(x)$ can be determined. Also, R_m, C_m and L′ are known for a particular network size and lambda. Hence (12) can be minimized to obtain the minimum delay in the folded clock tree. The actual minimization of equation (12) was performed using a program that finds the minimum of a non-linear continuous function constrained with a set of non-linear but continuous equality and non-equality relations.

3.2 Minimization Results:

Figures 7 show the variation of capacitance with length, for various network sizes with lambda varying from 2.5 to 0.5 microns. The broken lines are the approximate curves and have been fitted with an error of less than 1%. Similar curves can be obtained for the variation of resistance with length. Figures 8 shows the variation of the depth of fold with line length.

Figure 9 shows the variation of the line delay with network size and lambda for the folded clock tree. Both minimized and unminimized (i.e., a single exponential buffer drives the entire folded line) delays are illustrated. In section 4.0 it is shown that these delays hold for the unfolded clock tree also. Comparison of the minimized and unminimized delays show that there is a substantial reduction in the line delay due to the minimization technique. For a network size of 64*64 there is more than an order of magnitude reduction in the delay. Observe that as the network size decreases and lambda increases, the difference in the minimized and unminimized delay decreases. This can be explained as follows: The delay in a line given by (3) is the sum of (a) delay due to driving the line capacitance by a buffer and (b) delay due to driving the line capacitance by the line resistance.

The first part of the delay is minimum when the total line capacitance is driven by one buffer whereas the second part is minimum when the line is divided into numerous small segments and each one is driven separately. When the network size decreases and lambda increases, the contribution to the unminimized delay due to the second part is small and hence there is little or no reduction in the delay due to minimization. Hence "small" lines (definition of a small line depends on the length of the line as well as the resistance and capacitance per unit length of the line) can be driven as a whole without incurring any significant delay penalty.

Figure 10 shows the variation of the number of buffers required to obtain the minimum delay with lambda and network size. As the size of the network increases and lambda decreases, more buffers are needed to obtain the minimum delay. This is because the contribution of the product of the resistance and capacitance to the total unminimized line delay increases as the network size increases and lambda decreases. Since this delay term decreases as the number of buffers increases, the variations shown in Figure 9 are produced. For the current state of VLSI technology, only a few buffers would be required to drive the longest lines in minimum time. Of course, more buffers would be needed for long diffusion or polysilicon lines whose resistance per square is much larger than metal.

4.0 Placement of Buffers:

Starting from the root of the tree of Figure 4, traverse the tree to each leaf. In doing so, mark off lengths of L_1, L_2, ... and L_n from the root. These are the positions of the buffers. Number the buffers so that all buffers in a path from the root to a leaf of the tree that are at a distance of L_i from the root are numbered i. Notice that the number of buffers as predicted for the folded clock tree has increased due to the process of unfolding the clock tree. The i^{th} buffer in the folded clock tree has now been distributed in the actual clock tree into all buffers that are numbered i. This distribution of a buffer in the folded tree to that in the unfolded tree takes place according to the depth of fold function. The input capacitance of each of these distributed buffers is N times less than the input capacitance of the buffer in the unfolded tree, where N is the depth of fold at that point. Consider the i^{th} buffer in the folded tree being distributed in the above manner in the unfolded tree. The delay in the i^{th} section of the folded tree is given by equation 4. Consider the effect of unfolding the tree on each term of (4). The capacitance being driven by each buffer in the unfolded tree decreases by a factor of $N(L_{i-1})$ while the resistance increases by the same factor. The first term of (4) consists of a ratio of capacitances and since the numerator and denominator decrease by the same factor, it remains unchanged in the unfolded tree. The second and third terms are products of capacitance and resistance which also remain the same with unfolding. Hence, (4) also gives the delay in the i^{th} section of the unfolded tree. Since no delay term is affected by the unfolding, the delay in the unfolded tree is equal to the delay in the folded tree and is also the minimum. Figure 11 shows a folded clock tree and the positions of the buffers in the actual clock tree.

5.0 Conclusion:

The delay minimization equations developed in this paper were applied to the case of a binary tree distributing clock to all sub-modules of an N*N interconnection network chip. For N = 64, the minimization resulted in more than an order of magnitude reduction in the delay. In order to apply the analysis of a line to the clock tree structure, the clock tree was first collapsed to the form of a single line by making use of its symmetric structure. The minimization of delay is achieved by recognizing that the contribution of the product of the resistance and capacitance to the total line delay increases faster than the line length. Since the resistance and capacitance of a line are typically proportional to the line length, less delay is encountered if a number of smaller length lines are driven rather than the whole line. For long lines, the reduction in delay is appreciable. Such reductions are possible only if the the contribution of the resistance-capacitance product to the delay is significant. The minimization technique gives the number, position and size of the buffers required to obtain the calculated delay.

REFERENCES

CART84 Carter, D.L. and Guise, D.F.: "Effects of Interconnections on Submicron Chip Performance", VLSI Design, Jan. 1984.

DHAR83 Dhar, S., Franklin, M.A. and Wann, D.F.: "Timing Control in VLSI Based NlogN and Crossbar Networks", Proc. 1983 Inter. Conf. on Parallel Processing, Aug. 1983.

DHAR82 Dhar, S. and Bhatia, P.: "Design and VLSI Implementation of a Synchronous Crosspoint Switch", CCSD Tech. Rpt. 101, Dept. of Elec. Engr., Washington Univ., St. Louis, MO., Sept. 1982.

FISH83 Fisher, A.L. and Kung, H.T.: "Synchronizing Large Processor Networks", 1983 Symp. on Comp. Arch., Jan. 1983.

JAEG75 Jaeger, R.C.: "Comments on "An optimized Output Stage for MOS Integrated Circuits"", IEEE J. Solid-State Circuits, June 1975.

KUNG82 Kung, S.Y. and Gal-Ezer, R.J.: "Synchronous vs Asynchronous Computation in VLSI Array Processor", Proc. SPIE, Vol. 341, May 1982.

MEAD80 Mead, C. and Conway, L.: "Introduction to VLSI Systems", Addison-Wesley Pub. Co., 1980.

PENF81 Penfield, P. and Rubinstein, J.: "Signal Delay in RC Tree Networks", Proc. 18th Design Auto. Conf., June 1981.

SINH82 Sinha, A.K., Cooper, J.A. and Levinstein, H.J.: "Speed Limitations Due to Interconnect Time Constants in VLSI Integrated Circuits", IEEE Electron Device Letters, EDL-3, No. 4, April 1982.

WANN83 Wann, D.F. and Franklin, M.A.: "Asynchronous and Clocked Control Structures for VLSI Based Interconnection Networks", IEEE Trans. on Comp., March 1983.

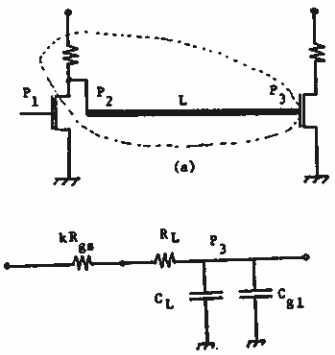

Figure 1: Delay model of a section.

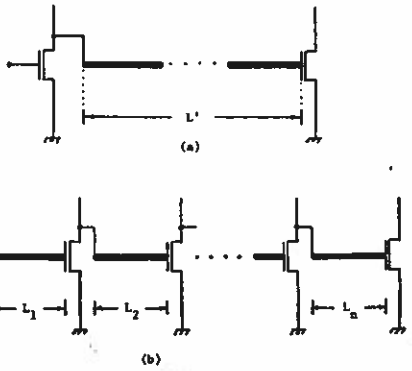

Figure 3: Partitioning a line into n sections.

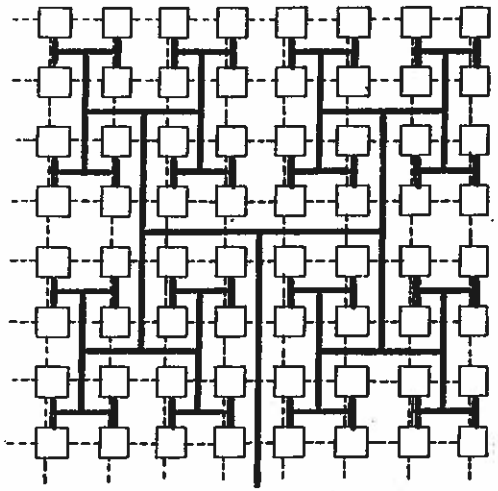

Figure 4: Clock distribution for an 8*8 crossbar using 2*2 modules (solid lines=clock tree, dotted lines=data paths).

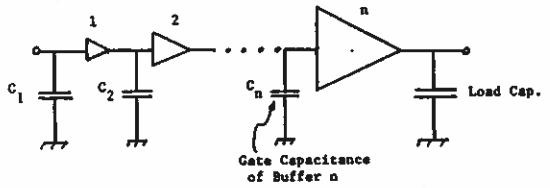

Figure 2: Exponetial buffer driving capacitive load.

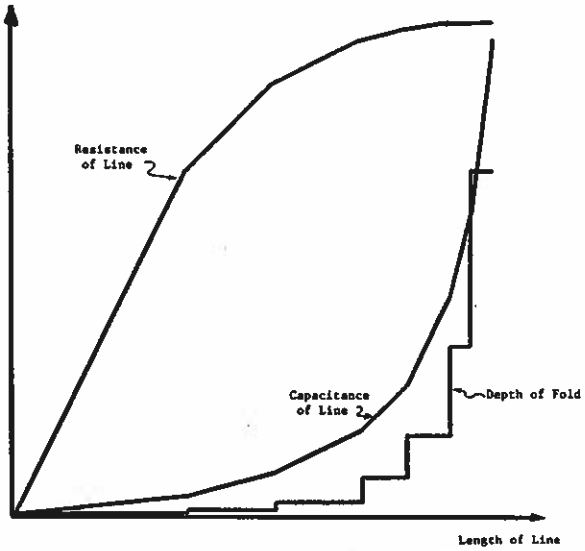

Figure 6: Variation of resistance, capacitance and depth of fold with line length.

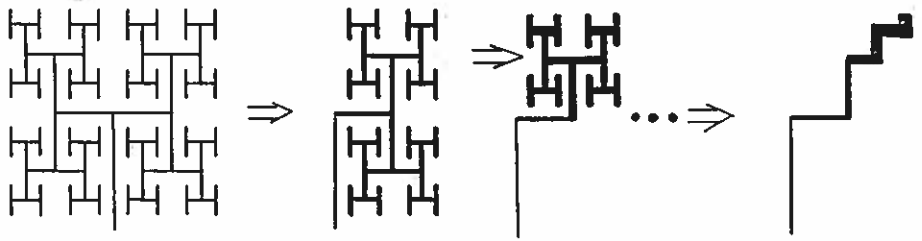

Figure 5: Folding the clock tree into a single line.

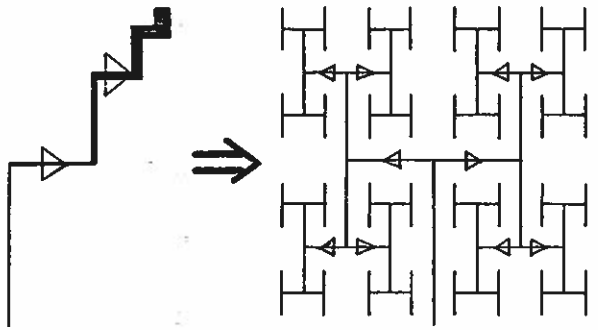

Figure 11: Placement of buffers in folded and unfolded clock tree

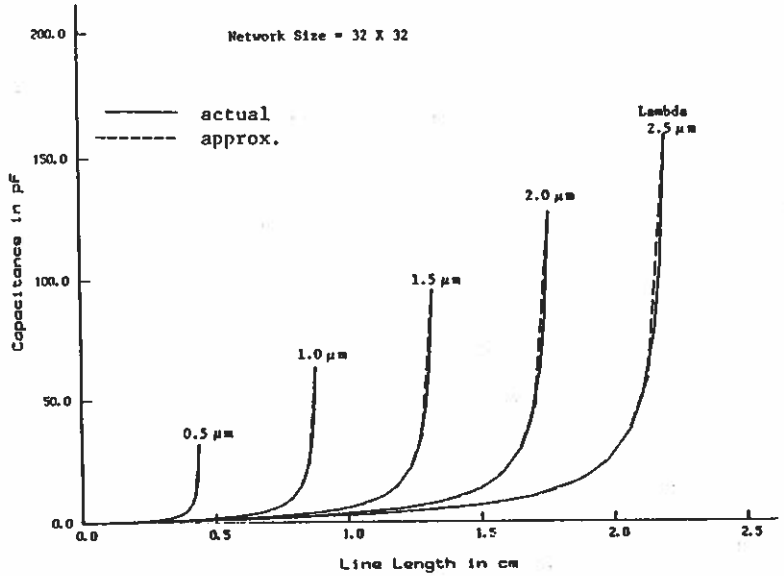

Figure 7: Variation of capacitance with line length as a function of lambda.

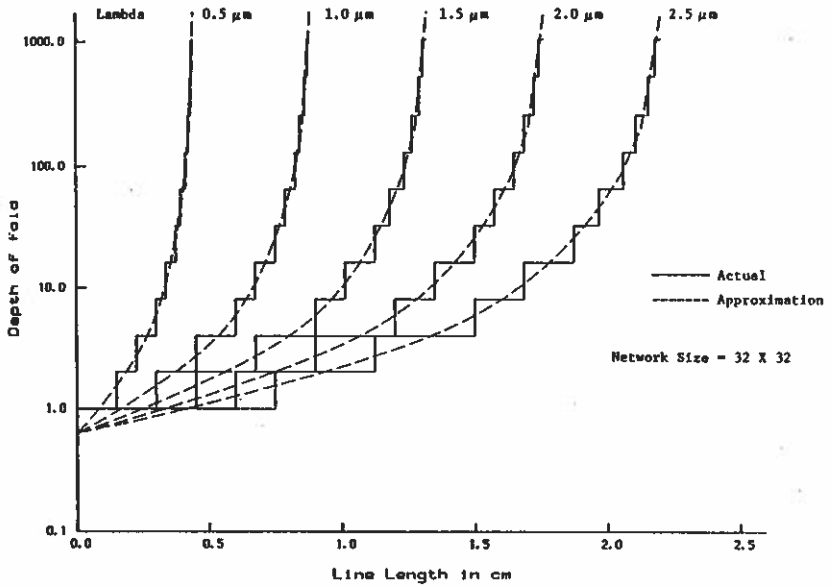

Figure 8: Variation of depth of fold with line length as a function of lambda.

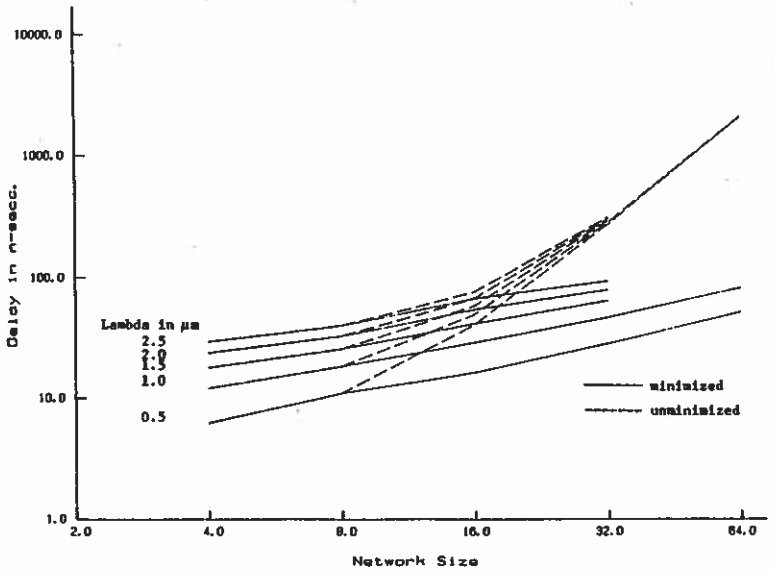

Figure 9: Variation of clock line delay with network size.

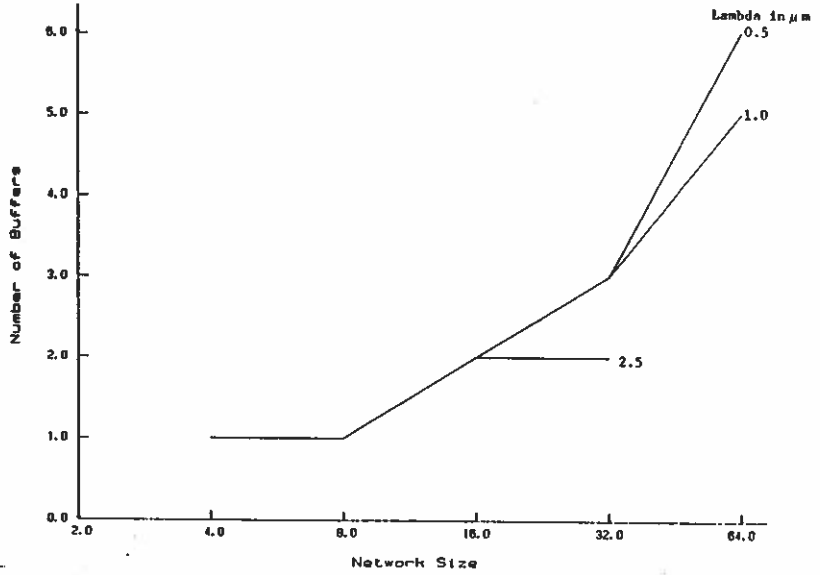

Figure 10: Variation of number of buffers in folded clock tree with line length.