COMPARISON OF TREE AND STRAIGHT-LINE CLOCKING FOR LONG SYSTOLIC ARRAYS

Marios D. Dikaiakos
Kenneth Steiglitz

CS-TR-271-90

June 1990

Comparison of Tree and Straight-Line Clocking for Long Systolic Arrays

Marios D. Dikaiakos and Kenneth Steiglitz

Department of Computer Science
Princeton University

Abstract

A critical problem in building long systolic arrays lies in efficient and reliable synchronization. We address this problem in the context of synchronous systems by introducing probabilistic models for two alternative clock distribution schemes: tree and straight-line clocking. We present analytic bounds for the Probability of Failure and the Mean Time to Failure, and examine the trade-offs between reliability and throughput in both schemes. Our basic conclusion is that as the one-dimensional systolic array gets very long, tree clocking becomes more reliable than straight-line clocking.

1 Introduction

Several problems in scientific computation and signal processing can be solved efficiently by special-purpose one-dimensional systolic architectures [Kun82,KS89,LL85]. Such solutions may have significant practical importance if they perform and scale well for large problem sizes. This implies that ultimately they should comprise many processing elements to achieve a high degree of parallelism. Furthermore, some systolic pipelines have the property of *linear speedup*. For instance, the Lattice-Gas machine reported in [Kug88,KS89] which simulates lattice models for fluid-flow, can achieve throughput proportional to the number of processors in its pipeline. It appears therefore that some future special-purpose machines will be built as very long systolic arrays of fine-grained components.

One of the limiting factors in building long pipelines is the difficulty in achieving proper and reliable synchronization [FK84,KGE82,HC87]. In this paper we investigate clock synchronization failures in such systems, in terms of their length and parameters that characterize clocking circuitry, such as delays in

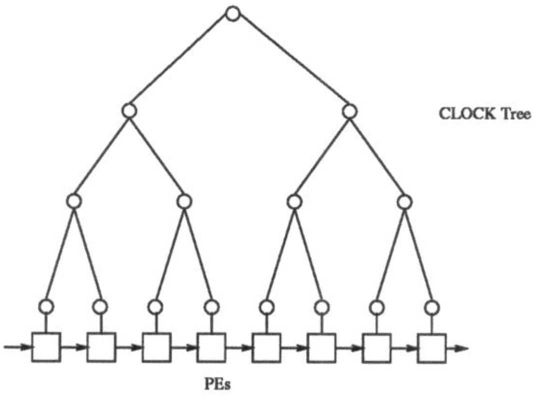

Figure 1: Tree clocking scheme.

buffers and wires, and variance in buffer response time. Since the one-dimensional pipeline is the simplest topology for inter-processor communication, our results also provide some insight into the problem of synchronizing large parallel systems in general.

First, we concentrate on the case where the clock distribution network is implemented as a regular f-ary tree (*tree clocking*) [FP86], and analyze the effect of clock skew on system performance and reliability using a probabilistic model for clock skew. As in [KS] the basic assumption is that the delays added to the clock signal by the elements of the clock distribution tree (buffers, wires), are independent, identically distributed normal random variables. Given this probability distribution and the topology of the clocking network, we analyze parameters such as the *Probability of Clock Synchronization Failure* and the *Mean Time to Failure*, and obtain asymptotic bounds on them.

In addition to tree clocking, *straight-line clocking* is addressed. In this scheme the clock is propagated alongside the pipeline, in parallel with the data-flow. In [FK84] it is suggested that this scheme is effective because skew between adjacent processing elements (PEs) is bounded, and building or extending such a distribution network is fairly easy. In that case, we focus on clock synchronization failures due to the lack of uniformity of clocking buffers in passing rising and falling edges. For instance, if the buffers of the clock distribution network respond more quickly to falling edges than rising edges, the clock pulses will tend to become shorter and some of them may eventually be lost. Clearly, lost pulses create clock synchronization failures. Again, we use a probabilistic approach and derive asymptotic results for the probability of Clock Synchronization Failure and the Mean Time to Failure.

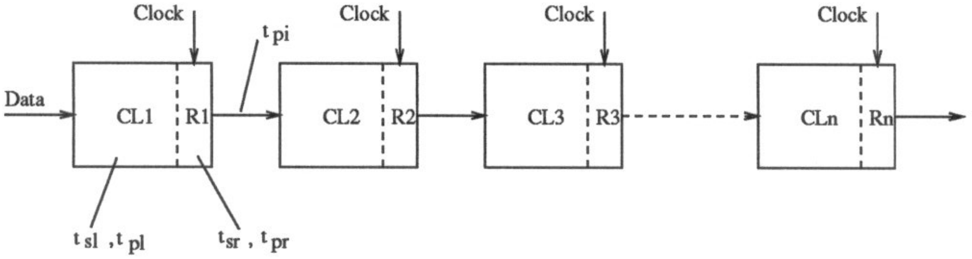

Figure 2: Pipeline Stages.

2 Tree Clocking

2.1 Basic Assumptions

We examine first the clocking of long systolic pipelines where the clock is distributed to pipeline stages (PEs) via a symmetric regular f-ary CLOCK tree. Nodes and edges in the CLOCK tree correspond to buffers and wires in the clock distribution network respectively, and the root of CLOCK corresponds to the clock source. The clock source has the responsibility to drive the entire CLOCK tree and wait for a clock pulse to arrive at all destinations before sending the next pulse (*equipotential* clocking). A one-phase clocking scheme is adopted.

The pipeline stages are attached to the leaves of the CLOCK tree. Their interconnection is serial, i.e., each of them receives data from its predecessor and sends data to its successor, as shown in figure 1. Each PE is formed by two sub-cells (Figure 2): **CL** which is a combinational circuit, and register **R** which is an edge-triggered flip-flop. The following time parameters are associated with these [HC87]: cell computation delay t_{sl}, namely the time needed for CL to complete a computation and settle its output to some valid result; cell propagation delay t_{pl} which corresponds to the minimum time for CL's output to change when its input changes; register settling time t_{sr} (similar definition as for t_{sl}) and register propagation time t_{pr}; and finally, propagation delay time t_{pi} due to the interconnection between communicating cells. T denotes the clock period of the system.

We are interested only in clock skew between adjacent stages of the pipeline because the data transfer occurs only between contiguous PEs. Let the node j in CLOCK be the closest common ancestor of leaves i and $i+1$, and let t_j be the departure time of a clock edge from j. If the arrival times of that clock edge at PEs i and $i+1$ are t_i and t_{i+1} respectively, the clock skew between PEs i and $i+1$ equals $t_{i+1} - t_i$, and is attributed to two causes :

The temporal fluctuations in clock-buffer delays, called *run-time skew* and denoted by δ_i^r;

The variations in delay characteristics of different components (because of different chip characteristics), called *build-time skew* and denoted by δ_i^b.

In other words, the cumulative clock skew can be expressed as follows :

$$\delta_i^r + \delta_i^b = t_{i+1} - t_i \tag{1}$$

where δ_i^r, δ_i^b may be either nonnegative or negative. Built-time skew remains constant after selecting the clock buffers off the shelf, and building the clock distribution network. In real designs, the clock distribution network is tuned so as to minimize the effects of build-time skew. The tuning procedure usually involves the adjustment of delay elements, buffers, and wires and can guarantee a "negligible" build-time clock skew [Wag88,MMS87].

2.2 Model

For the calculation of *run-time clock skew* we assume that CLOCK follows the *metric-free* tree model [Kug88]. In this model, all buffers (nodes) are identical and add to the clock signal a delay modeled by the same probability distribution. Wires (edges) which propagate the clock have equal lengths. Therefore, every wire has the same probability distribution for delay, which can be lumped with the delay of the buffer that follows it. The metric-free tree presents a reasonable abstraction for distribution networks which provide a clock signal to chips on a number of boards in a system.

Assumption 2.1 *The delay inserted in the clock signal by buffer k, and the wire leading to it, is considered to be a random deviate τ_k, distributed normally with zero mean and finite variance, i.e., $\tau_k \sim N(0, \sigma^2)$, and independent from clock edge to clock edge.*

Actually, each buffer adds a positive delay to the clock signal, and therefore there is a non-zero mean for the delay distribution. If we consider the distinct paths that route the clock to two adjacent PEs, the difference of their cumulative means is equal to the *build-time skew* between them. Nevertheless, because of CLOCK tree symmetricity, our *run-time skew* analysis is independent of the build-time skew value, and may proceed as if the cumulative means along the two distinct paths cancel out (i.e. as if the build-time skew were zero). Therefore the mean values of τ_k's may be considered zero.

The following two conditions, must be satisfied at each pipeline stage in order to avoid clock synchronization failures :

First, in the case where the clock signal arrives earlier at PE $i+1$ than at PE i (negative clock skew), and the clock period is not large enough, the data computed in PE i may arrive at PE $i+1$ after

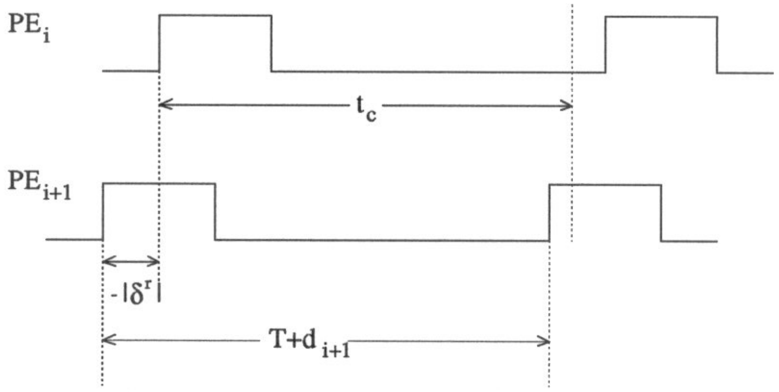

Figure 3: Type-1 Failure: the output of PE i appears too late to be latched by PE i+1.

the arrival of the next pulse's leading edge (see figure 3). In that case, proper synchronization is not guaranteed and data may be lost [HC87]. Therefore, the following condition must be satisfied:

$$T + d_{i+1} > \ t_c - (\delta_i^r \ + \delta_i^b) \iff d_{i+1} + \delta_i^r > \ t_c - \delta_i^b - T \tag{2}$$

where $t_c \ = \ t_{sr} \ + \ t_{pi} \ + \ t_{sl}$ (total computation delay). The random variable d_l is the difference in rising clock edge arrival times at PE l that compensates for the fact that two successive rising clock edges arrive with different delays at the same PE l. Although the two successive pulses have been emitted by the central clock source within time T, they arrive at PE l with a time difference of $T + \sum_{k \in K_{S,l}} (\tau_k' - \tau_k)$ between each other. The random variable τ_k corresponds to the delay inserted in one clock pulse by some buffer in the path from the clock source to PE l, τ_k' corresponds to the delay inserted in the next clock pulse by the same buffer k, and K_{ji} denotes the set of CLOCK nodes (buffers) along the path from node j to node i, not including j, and S corresponds to the clock source (root of CLOCK). Thus, for every PE l, d_l equals :

$$d_l \ = \ \sum_{k \in K_{S,l}} (\tau_k' - \tau_k), \tag{3}$$

In the following sections, we will lump δ_i^b with t_c, and pursue our analysis as if δ_i^r were the only cause of clock failures. As we mentioned previously, δ_i^b remains constant after building the clock distribution network. In fact, for every type of clocking circuitry components there is a known range of delay characteristics [Tex76]. Using those data, we can estimate the worst case value of build-time skew. We denote it as δ_b, where $\delta_b \ \geq \ |\delta_i^b| \ \forall i$ (δ_i^b takes also negative values), and concentrate on the following "worst case" restriction :

$$d_{i+1} + \delta_i^r > \ t_c + \delta^b - T \iff \delta_i > t_c^b - T \tag{4}$$

where $\delta_i \ = \ d_{i+1} + \delta_i^r$, and the factor $t_c^b \ = \ t_c + \delta^b$ absorbs the worst-case effects of build-time skew. At the end, we will estimate the effects of improper tuning, and build-time skew on system performance for

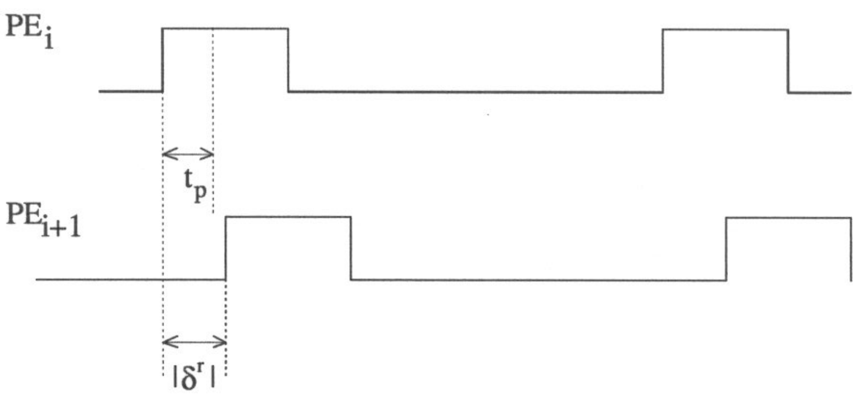

Figure 4: Type-2 Failure: the input to PE i+1 changes before it latches its proper input.

the range of values of T that guarantee very high reliability against "run-time" failures. Failures due to violations of (4), will be referred as *type-1 failures.*

The second type of synchronization failure (*type-2 failure*) occurs whenever a rising clock edge arrives later at PE $i+1$ than at PE i and the propagation time between i and $i+1$ is very small. In that case the data released by PE i at clock cycle k may be stored into register R_{i+1} at the same cycle, and not at the next one, $k+1$. The data to be latched into R_{i+1} on cycle k, is lost (figure 4). The following inequality prevents this type of failure:

$$\delta_i^r \ + \delta_i^b \ < \ t_{pr} + t_{pi} + t_{pl} = \ t_p \tag{5}$$

where t_p is the total propagation delay. It is reasonable to assume that $t_{sl} \ >> \ t_{pl}$ and thus $t_c \ >> \ t_p$. Relation (5) does not involve the clock period T. Consequently, if (5) is not satisfied, clock synchronization failure cannot be prevented, even if the clock frequency is reduced substantially. In the following sections we focus our discussion to type-1 failures, in order to examine the relation between reliability and clock speed in very long pipelines.

2.3 Analysis

We consider the clock skew δ_i^r between PEs i and $i+1$. Let j be their closest common ancestor in the CLOCK tree, t_j be the departure time of some clock edge from node j, and t_i and t_{i+1} be the arrival times of the aforementioned clock edge at nodes i and $i+1$ respectively. As stated in the previous section, δ_i^r equals $t_{i+1} - t_i$, or $(t_{i+1} - t_j) \ - \ (t_i - t_j)$. The differences $t_i - t_j$, $t_{i+1} - t_j$ can be expressed in terms of the individual delays at tree nodes, τ_i:

$$t_i \ - \ t_j \ = \ \sum_{k \in K_{ji}} \tau_k$$

$$t_{i+1} - t_j = \sum_{k \in K_{j,i+1}} \tau_k$$

where K_{ji} is defined as in the previous paragraph. Thus

$$\delta_i^r = \sum_{k \in K_{j,i+1}} \tau_k - \sum_{l \in K_{ji}} \tau_l \tag{6}$$

From (3), (6) we can easily see that :

$$\delta_i = \delta_i^r + d_{i+1} = \sum_{l \in K_{S,i+1}} \tau_l' - \sum_{l \in K_{S,i}} \tau_l \tag{7}$$

By the symmetry of CLOCK ($|K_{ji}| = |K_{j,i+1}|$), assumption (2.1), and the independence of the two terms on right-hand side of (7) we conclude that δ_i is normally distributed with zero mean and $(|K_{S,i+1}| + |K_{S,i}|) \cdot \sigma^2$ variance. But $|K_{S,i}| + |K_{S,i+1}| = 2\log_f N$, and therefore:

$$\delta_i \sim N(0, 2\sigma^2 \log_f N) \tag{8}$$

It is interesting to observe that regardless of where in the CLOCK tree the paths $K_{S,i}$ and $K_{S,i+1}$ are separated, the variance of δ_i remains $2\sigma^2 \log_f N$; and furthermore that the occurrence of type-1 failures depends on the delays added to two *successive* rising clock edges by the paths $K_{S,i}$, and $K_{S,i+1}$ respectively. Note that in [KS] the estimated mean value of clock skew refers to the difference of the delays added to the *same* clock edge by the two paths $K_{S,i}$, and $K_{S,i+1}$.

The probability that no type-1 failure occurs when a clock pulse is sent to the PEs through CLOCK, is equal to the probability that at each stage of the pipeline, relation 4 holds. Let

$$G(1, N-1) = Pr\Big[t_c^b - T < \delta_1, \ldots, t_c^b - T < \delta_{N-1}\Big]$$

where N is the length of the pipeline. Introducing the notation $\mathcal{R}(m,n)$ to denote the set of restrictions $\{t_c^b - T < \delta_m, \ldots, t_c^b - T < \delta_n\}$, we can express $G(1, N-1)$ as follows :

$$G(1, N-1) = Pr\Big[\mathcal{R}(1, \frac{N}{f}), \mathcal{R}(\frac{N}{f}+1, N-1)\Big]$$

Clearly:

$$G(1, N-1) = Pr\Big[t_c^b - T < \delta_{\frac{N}{f}}, \mathcal{R}(1, \frac{N}{f}-1), \mathcal{R}(\frac{N}{f}+1, N-1)\Big] \Rightarrow$$

$$G(1, N-1) = Pr\Big[t_c^b - T < \delta_{\frac{N}{f}} \mid \mathcal{R}(1, \frac{N}{f}-1), \mathcal{R}(\frac{N}{f}+1, N-1)\Big] \cdot Pr\Big[\mathcal{R}(1, \frac{N}{f}-1), \mathcal{R}(\frac{N}{f}+1, N-1)\Big]$$

The random variables δ_i involved in the set of restrictions $\mathcal{R}(1, \frac{N}{f}-1)$ are independent of the ones involved in $\mathcal{R}(\frac{N}{f}+1, N-1)$. Therefore :

$$Pr\Big[\mathcal{R}(1, \frac{N}{f}-1), \mathcal{R}(\frac{N}{f}+1, N-1)\Big] = Pr\Big[\mathcal{R}(1, \frac{N}{f}-1)\Big] \cdot Pr\Big[\mathcal{R}(\frac{N}{f}+1, N-1)\Big]$$

and :

$$G(1,N-1) \;=\; Pr\Big[t_c^b - T < \delta_{\frac{N}{f}} \mid \mathcal{R}(1,\frac{N}{f}-1), N-1)\Big] \cdot G(1,\frac{N}{f}-1) \cdot G(\frac{N}{f}+1, N-1) \qquad (9)$$

The first term of the product in Equation (9) is difficult to specify analytically, so we will find a lower bound for it. To do this, we need the following lemmas (rigorous proofs given in the appendix) :

Lemma 2.1 [Kug88] For any random variable y, any α, β, and any random event C, it is true that:

$$Pr[\alpha < y \mid C,\ \beta < y] \;\geq\; Pr[\alpha < y \mid C] \qquad (10)$$

Lemma 2.2 [Kug88] Let $y_i,\ i = 1, 2, \ldots, N$ be independent identically distributed (iid) random variables, let $\tau_j,\ j = 1, 2, \ldots, n$ be sets of y_i's (not necessarily disjoint), and let

$$t_j \;=\; \sum_{l \in \tau_j} y_l \ , j = 1, 2, \ldots, n$$

Then for each $j \in \{1, 2, \ldots, n\}$ it is true that:

$$Pr[\alpha \;<\; t_j \mid \alpha < t_1 \ , \ldots, \alpha < t_{j-1}, \alpha < t_{j+1}, \ldots, \alpha < t_n] \geq Pr[\alpha < t_j] \qquad (11)$$

From the definition (7), and assumption (2.1) it is clear that δ_i is distributed as the sum of the *iid* random variables $\tau_l,\ \tau_l'$: $\sum_{l \in K_{S,i+1}} \tau_l' \ + \sum_{l \in K_{S,i}} \tau_l$. Therefore, lemma 2.2 can be applied to yield the following bound:

$$Pr\Big[t_c^b - T < \delta_{\frac{N}{f}} \mid \mathcal{R}(1,\frac{N}{f}-1), \mathcal{R}(\frac{N}{f}+1, N-1)\Big] \geq Pr\Big[t_c^b - T < \delta_{\frac{N}{f}}\Big] \qquad (12)$$

Denoting the probability $Pr[t_c^b - T < \delta_i]$ as g_i we get: [1]

$$(9),\ (12) \Rightarrow G(1,N-1) \geq g_{\frac{N}{f}} \cdot G(1,\frac{N}{f}-1) \cdot G(\frac{N}{f}+1, N-1) \qquad (13)$$

Similarly :

$$G(\frac{N}{f}+1, N-1) \geq g_{\frac{2N}{f}} \cdot G(\frac{N}{f}+1, \frac{2N}{f}-1) \cdot G(\frac{2N}{f}+1, N-1) \qquad (14)$$

From the symmetry of CLOCK tree, we can easily see that :

$$G(1,\frac{N}{f}-1) \;=\; G(\frac{N}{f}+1, \frac{2N}{f}-1) \quad \text{and} \quad g_{\frac{N}{f}} \;=\; g_{\frac{2N}{f}}$$

Therefore,

$$(13),\ (14) \Rightarrow G(1,N-1) \geq g_{\frac{N}{f}}^2 \cdot G(1,\frac{N}{f}-1)^2 \cdot G(\frac{2N}{f}+1, N-1) \qquad (15)$$

[1] We assume that N is an integer power of f

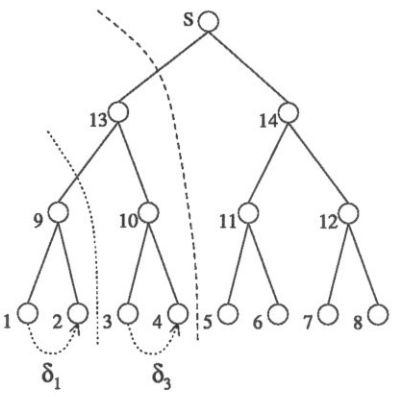

Figure 5: Clock Skew in a Binary Tree.

By expanding the last term of the previous product, we get :

$$G(1, N-1) \geq g_{\frac{N}{f}}^{f-1} \cdot G(1, \frac{N}{f} - 1)^f \tag{16}$$

Furthermore:

$$G(1, \frac{N}{f} - 1) \;\geq\; g_{\frac{N}{f^2}} \cdot Pr\Big[\mathcal{R}(1, \frac{N}{f^2} - 1), \mathcal{R}(\frac{N}{f^2} + 1, \frac{N}{f} - 1)\Big]$$

The random variables δ_i that appear in restrictions $\mathcal{R}(1, \frac{N}{f^2} - 1)$, and $\mathcal{R}(\frac{N}{f^2} + 1, \frac{N}{f} - 1)$ have terms in common. Nevertheless, it is true that:

$$Pr\Big[\mathcal{R}(1, \frac{N}{f^2} - 1), \mathcal{R}(\frac{N}{f^2} + 1, \frac{N}{f} - 1)\Big] \;=\; Pr\Big[\mathcal{R}(1, \frac{N}{f^2} - 1)\Big] \;\cdot\; Pr\Big[\mathcal{R}(\frac{N}{f^2} + 1, \frac{N}{f} - 1)\Big]$$

We are going to show this fact, using a simple example with $f = 2$, and $N = 8$ (see figure 5). Clearly :

$$\delta_1 \;=\; (\tau'_{13} + \tau'_9 + \tau'_2) - (\tau_{13} + \tau_9 + \tau_1) \;=\; (\tau'_{13} - \tau_{13}) + (\tau'_9 + \tau'_2 - \tau_9 - \tau_1) \;=\; r + r'$$

and :

$$\delta_3 \;=\; (\tau'_{13} + \tau'_{10} + \tau'_4) - (\tau_{13} + \tau_{10} + \tau_3) \;=\; (\tau'_{13} - \tau_{13}) + (\tau'_{10} + \tau'_4 - \tau_{10} - \tau_3) \;=\; r + r''$$

The terms r', r'' are independent random variables; r is the random variable which represents the common term of δ_1 and δ_3. We can easily see that:

$$Pr[t_c^b - T \;<\; r \;+\; r' \;, t_c^b - T \;<\; r \;+\; r''] \;=\; Pr[t_c^b - T \;<\; r \;+\; r'] \;\cdot\; Pr[t_c^b - T \;<\; r \;+\; r'']$$

Using the previous remark, we get:

$$G(1, \frac{N}{f} - 1)^f \geq g_{\frac{N}{f^2}}^{(f-1)f} \cdot G(1, \frac{N}{f^2} - 1)^{f^2}$$

$$\cdots$$

$$G(1, f^2 - 1)^{\frac{N}{f^2}} \geq g_f^{(f-1)\frac{N}{f^2}} \cdot G(1, f-1)^{\frac{N}{f}} \geq g_f^{(f-1)\frac{N}{f^2}} \cdot g_1^{\frac{(f-1)N}{f}}$$

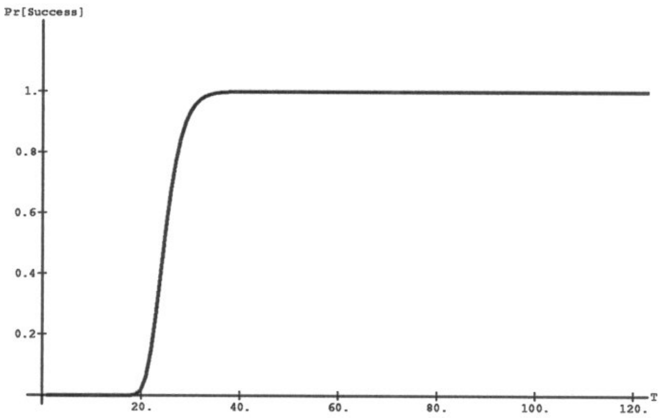

Figure 6: Lower Bound for the Probability of Type-1 Success (N=2000,$t_c^b = 20, \sigma = 1, f = 2.$)

Thus :

$$G(1, N-1) \;\geq\; \left(g_{\frac{N}{f}} \cdot g^{f}_{\frac{N}{f^2}} \cdot g^{f^2}_{\frac{N}{f^3}} \cdots g_f^{\frac{N}{f^2}} \cdot g_1^{\frac{N}{f}}\right)^{f-1} \tag{17}$$

where

$$g_i \;=\; Pr[t_c^b - T < \delta_i] \;=\; 1 - Pr[\delta_i \leq t_c^b - T] \Rightarrow$$
$$g_i \;=\; 1 - \Phi\left(\frac{t_c^b - T}{\sqrt{2\sigma^2 \log_f N}}\right) \;=\; \Phi\left(\frac{T - t_c^b}{\sqrt{2\sigma^2 \log_f N}}\right)$$

$\Phi(x)$ equals $\frac{1}{\sqrt{2\pi}} \int\limits_{-\infty}^{x} e^{-x^2/2} dx$. The value of g_i is not influenced by i, so for the sake of simplicity we get rid of the indices, and use instead:

$$g \;=\; \Phi\left((T - t_c^b)/\sqrt{2\sigma^2 \log_f N}\right) \tag{18}$$

Relations (17), (18) give:

$$G(1, N-1) \;>\; \left(g^{1+f+f^2+\ldots+\frac{N}{f}}\right)^{f-1} \tag{19}$$

or equivalently :

$$G(1, N-1) \;>\; g^{N-1} \tag{20}$$

Therefore, we can state the following theorem:

Theorem 2.1 *Consider a very long one-dimensional systolic array clocked by a central clock source with fixed frequency $1/T$, via a clock distribution network compatible with the metric-free tree model. The probability that clock skew does not cause synchronization failure by violating condition (4) within a time period between two successive pulses, satisfies :*

$$G(1, N-1) \;\geq\; \left[\Phi\left(\frac{T - t_c^b}{\sqrt{2\sigma^2 \log_f N}}\right)\right]^{N-1} \tag{21}$$

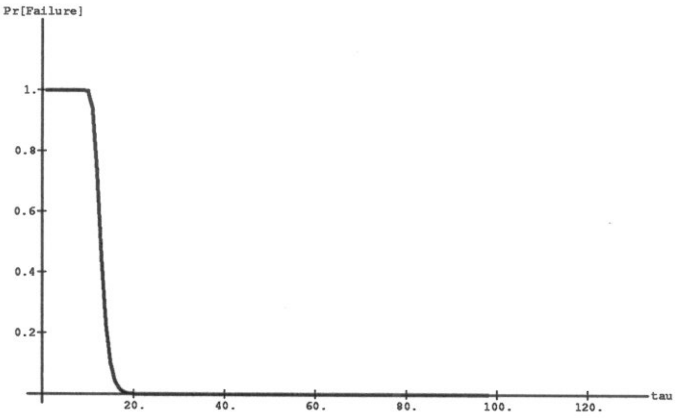

Figure 7: Upper Bound for the Probability of Type-1 Failure (N=1000,f=2).

We define the *normalized margin* τ :

$$\tau \;=\; \frac{T - t_c^b}{\sigma} \tag{22}$$

and use the following property to approximate the value of the right-hand side of relation (21), [Fel68, AS64]:

$$\forall\; u \geq 0,\; 1 - \Phi(u) \;<\; \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{u} \cdot e^{-u^2/2} \tag{23}$$

The combination of (22), (21) and (23) gives:

$$G(1, N-1) \;\geq\; \left(1 - \frac{1}{\sqrt{\pi}} \cdot \frac{\sqrt{\log_f N}}{\tau} \cdot e^{-\tau^2/4\log_f N}\right)^{N-1} \tag{24}$$

This relation provides some insight into the tradeoff between clock period and reliability. For known values of t_c^b and σ which depend on the circuit implementation, we can plot the lower bound for the probability of type-1 "success", or the upper bound for the probability of type-1 "failure" when two clock pulses are sent to the pipeline. Figure 6 shows the lower bound for the probability of success, $G(1, N-1)$, as a function of the clock period in a 2000-stage pipeline with $\sigma = 1$, and $t_c^b = 20$. The upper bound for the probability of failure when $N = 1000$, and $f = 2$ is plotted in figure 7 as a function of τ. Figure 8 focuses on the more interesting area of figure's 7 plot, namely on the range of values of τ where the probability of failure gets very close to zero. It contains three curves in base-10 logarithmic scale: one for $N = 10^3$ (lower curve), one for $N = 10^5$, and one for $N = 10^6$. From the shape of these plots, it is obvious that there is a tight range of clock period values where the guaranteed reliability (i.e., the lower bound of probability of success) changes sharply from low to very high. The design challenge is to satisfy two conflicting goals: first, the achievement of an acceptable reliability level which requires the increase of T. Second, the maximization of pipeline throughput and PE utilization, attained by making T as close as possible to t_c^b.

We notice that if we demand that the lower bound in (24) be equal to $1 - \epsilon$, where ϵ is a very small

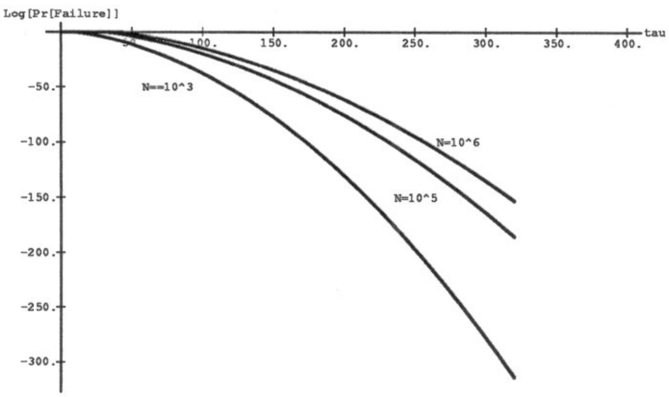

Figure 8: Logarithm of the Upper Bound for the Probability of Type-1 Failure ($f = 2$).

positive real number, the probability of success would be very close to one. Consequently, by letting :

$$\left(1 - \frac{1}{\sqrt{\pi}} \cdot \frac{\sqrt{\log_f N}}{\tau} \cdot e^{-\tau^2/4\log_f N}\right)^{N-1} \; = \; 1 - \epsilon$$

we can estimate an asymptotic behavior for τ as $N \to \infty$, guaranteeing the very high reliability of $1-\epsilon$ from the standpoint of synchronization failures. For large N, the previous equality becomes :

$$\epsilon \; = \; \frac{N\sqrt{\log_f N}}{\sqrt{\pi}\cdot\tau} \cdot e^{-\tau^2/4\log_f N}$$

Assuming that :

$$\tau \; = \; \tau(N) \; = \; \frac{2}{\sqrt{\log_f e}} \cdot \log_f N \tag{25}$$

we get: $\epsilon \; = \; 1/(2\sqrt{\pi} \cdot \sqrt{\log_f N})$ which tends to zero as $N \to \infty$. Therefore, the asymptotic growth of $\tau(N)$ described in (25), is sufficient to guarantee high reliability against clock synchronization failures in very long systolic pipelines. Combining the definition of τ in (22), and equation (25) we get:

$$T \; = \; t_c^b \; + \; \frac{2}{\sqrt{\log_f e}} \cdot \log_f N \tag{26}$$

From this, we conclude that as the length of the one-dimensional systolic array increases, an increase in the clock period proportional to $\log_f N$ is sufficient to guarantee negligible failure probability.

In the case where CLOCK has not been tuned, we assume that for two random off-the-shelf buffers of the same type, the difference in their delay times ranges between $-A_b$ and A_b. The maximum value that δ^b can take, equals $2A_b \log_f N$. Substituting t_c^b by $\delta^b + t_c$ in (26), we get:

$$T \; = \; 2 \cdot A_b \log_f N + t_c + \frac{2}{\sqrt{\log_f e}} \cdot \log_f N \Rightarrow T \; = \; t_c + (2 \cdot A_b + \frac{2}{\sqrt{\log_f e}}) \cdot \log_f N \tag{27}$$

Therefore, we conclude that build-time skew does not affect the pipeline throughput asymptotically. In practice however, certain values of the constant A_b in (27) might require substantially higher values of T to guarantee highly reliable functioning of the systolic array.

2.4 Mean Time to Failure

The previous discussion about reliability does not address the temporal behavior of the pipeline. Instead, it deals with the problem of potential synchronization failures when one clock pulse is sent through CLOCK towards the PEs, and attempts to satisfy the inequalities (4) and (5). A temporal approach would try to estimate how many cycles a pipeline would run without failures (*Mean Time to Failure*), under the presence of clock skew. Let F be the random variable corresponding to the time when type-1 failure occurs, and $F(t)$ be the probability that clock synchronization failure does not occur before the tth clock pulse, i.e. :

$$F(t) \ = \ Pr[F > t] \ = \ Pr\Big[\mathcal{R}_1(1, N-1), \mathcal{R}_2(1, N-1), \ldots, \mathcal{R}_t(1, N-1)\Big]$$

where $\mathcal{R}_l(m, n)$ is the set of inequalities $\{t_c^b - T < \delta_m, \ldots t_c^b - T < \delta_n\}$, for the l-th clock pulse. Using Bayes' rule, and lemma (2.2) we can easily see that:

$$Pr\Big[\mathcal{R}_1(1, N-1), \mathcal{R}_2(1, N-1), \ldots, \mathcal{R}_t(1, N-1)\Big] \ \geq \ (g^{N-1})^t$$

Consequently :

$$F(t) \ > \ g^{t\cdot(N-1)} \tag{28}$$

Using the property ([Ros85]) that for any nonnegative random variable X,

$$E[X] \ = \ \int_0^\infty Pr[X \geq x] \cdot dx$$

we can easily estimate a lower bound for $F(t)$'s mean value, namely for the mean time to failure (MTF) :

$$\begin{aligned} MTF &= \int_0^\infty F(t) \cdot dt \Rightarrow \\ MTF &> \int_0^\infty g^{t(N-1)} \cdot dt \Rightarrow \\ MTF \ &> \ \frac{-1}{(N-1) \cdot \ln g} \end{aligned} \tag{29}$$

The lower bound in Equation (29) is positive because g is positive and less than 1. Substitution of Equation (24) into (29), yields the following theorem :

Theorem 2.2 *In a very long one-dimensional systolic array of length N, clocked by a central clock source with frequency $1/T$, via a clock distribution network compatible with the metric-free tree model, the Mean Time to Failure satisfies :*

$$MTF \ > \ \frac{-1}{(N-1) \cdot \ln\left(1 - \sqrt{\log_f N} \cdot e^{-\tau^2/4\log_f N} / \sqrt{\pi} \cdot \tau \ \right)} \tag{30}$$

Figure 9 contains a plot of the base-10 logarithm of the MTF, as a function of τ for three pipelines with 10^3, 10^5, 10^6 PEs (left, middle, and right curves respectively). For a pipeline with 10^6 PEs, the mean time to failure is greater than 10^{30} for $\tau \geq 80$.

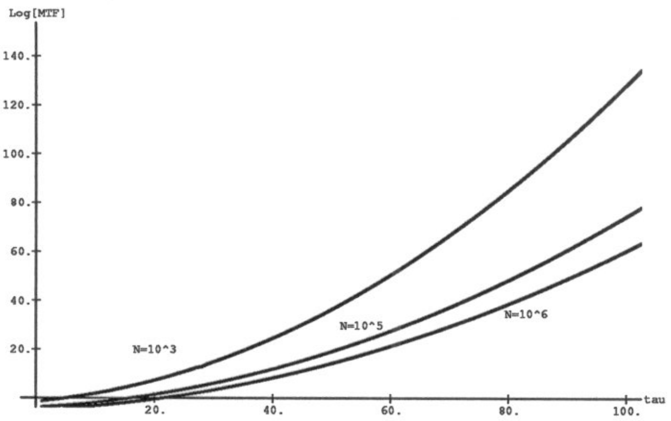

Figure 9: Logarithm of the Lower Bound for the Mean Time to Failure Type-1 ($f = 2$).

3 Straight-Line Clocking

3.1 Basic Assumptions

Straight-line (pipelined) clocking represents an alternative to equipotential clocking for synchronous systems. Under the straight-line scheme, the clock distribution network is composed of a series of buffers (repeaters) which carry successive pulses from the global clock source, so that several clock pulses are simultaneously active in the system (figure 10). Straight-line clocking represents a simple, and easily expandable architectural design where clock and data are transferred along the pipeline in parallel. As Fisher and Kung point out [FK84], straight-line clocking is most applicable in cases where PE speeds are very high, and interconnect is long and has high impedance. In that case, equipotential clocking would impose a slow clock, and result in PE under-utilization. In contrast, a pipelined clocking scheme with short interconnection paths would run at speeds independent of the pipeline's length and close to the PE switching speed. Tuning of interconnection delays in the straight-line clock distribution network is crucial so as to guarantee the arrival of data in one processor before the arrival of the corresponding clock pulse.

A potential cause of clock synchronization failure is the lack of uniformity in repeaters when passing falling and rising clock edges. Worst case analysis has shown that *differences in the delay between leading and falling edges may cause the disappearance of clock pulses and thus synchronization failure* [GS88,GS]. Thus, the failure mode of straight-line clocking is different from that of tree clocking.

Our principal assumption is that whenever a pulse of width w goes through a buffer, its width is changed and becomes $w + \epsilon$, where ϵ is a normal random variable with zero mean and finite variance. We model the pulse width as a random process, where the state of the process corresponds to pulse width, and the discrete time corresponds to the buffer stage where the pulse is currently in. If w ever reaches

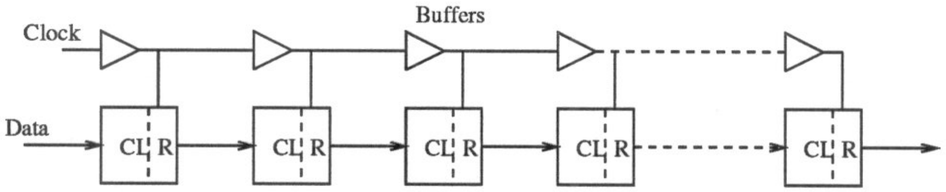

Figure 10: Straight-Line clocking scheme.

zero, the pulse disappears and the process is *absorbed.* The assumption about the random changes (increments) ϵ of the pulse width and the process absorption at zero leads us to a thoroughly analyzed form of random process, *Brownian Motion with Absorbing Barrier at zero* [KT75].

Our main assumptions and definitions concerning pipelined clocking, are :

(1) Synchronization failure happens when the width of a pulse becomes equal to zero.

(2) w is the width of the pulse emitted by the clock source and n is the number of buffers in the clock distribution network (which can be considered equal to the pipeline length N).

(3) $W_i, i = 1, \ldots, n$ is the clock pulse width at the output of the ith buffer, and $X_i, i = 1, \ldots, n$ is the *random process* which models W_i (definition of X_i follows).

(4) $Y_i, i = 0, \ldots, n$ is a Brownian Motion random process, such that:

$Y_0 = Z_0 = w$ (constant)

$\forall i, 0 \leq i \leq n : Y_{i+1} - Y_i = Z_{i+1}$, where Z_i's are deviates following the Normal Distribution $N(0, \sigma^2)$

For each pair (i, j) of buffers with $i \neq j$, the increments $Z_i = Y_i - Y_{i-1}$ and $Z_j = Y_j - Y_{j-1}$ are independent random variables

(5) $X_i,\ i = 0, \ldots, n$ is a Brownian Motion random process corresponding to Y_i, *restricted* by an absorbing barrier at zero. Thus :

$$X_i = \begin{cases} Z_0 + Z_1 + \ldots + Z_i = Y_i, & \text{iff } Y_j > 0, \forall j,\ 0 \leq j \leq i \\ 0, & \text{if } \exists j, \text{ with } j < i \text{ such that } Y_j = 0 \end{cases}$$

(6) D is a deviate corresponding to the number of the first repeater after which a clock pulse disappears.

Note that we have chosen the mean value of the pulse's random increments to be zero, which means that we implicitly assume that repeaters are designed to respond uniformly to rising and falling edges. Failures occur because of the variance in the response time.

3.2 Analysis

We will first estimate $Pr[D > j]$, i.e., the probability that a clock pulse sent to the pipeline disappears for the first time after the jth stage. By definition of $X_i,\ i = 0, \ldots, n$:

$$Pr[D > j] \; = \; Pr[\min_{i=0,\ldots,j} X_i \; > \; 0 \mid X_0 = w] \; = \; Pr[D > j] \; = \; Pr[\min_{i=0,\ldots,j} Y_i \; > \; 0 \mid Y_0 = w] \tag{31}$$

Equation (31) is true because the Brownian Motion Y_i is identical to the Absorbing Brownian Motion X_i up to the point where the latter "hits the barrier". After that point, X_i becomes identically zero, whereas Y_i may continue the random walk to positive or negative values. The following lemma presents known basic properties of Brownian Motion, and will be used subsequently.

Lemma 3.1 [KT75] *If Y_i is a Brownian Motion random process, the following equations are true:*

$$Pr[\min_{i=0,\ldots,t} Y_i > 0 \mid Y_0 = w] \; = \; Pr[\max_{i=0,\ldots,t} Y_i < 2 \cdot w \mid Y_0 = w] \tag{32}$$

$$Pr[\max_{i=0,\ldots,t} Y_i < 2 \cdot w \mid Y_0 = w] \; = \; Pr[\max_{i=0,\ldots,t} Y_i < w \mid Y_0 = 0] \tag{33}$$

$$Pr[\max_{i=0,\ldots,t} Y_i \geq w \mid Y_0 = w] \; = \; 2 \cdot Pr[Y_t > \; w \mid Y_0 = 0] \tag{34}$$

Using (32,33,34, 31), and the definition of Y_i's we get :

$$\begin{aligned} Pr[D > j] \; &= \; Pr[\max_{i=0,\ldots,j} Y_i < 2 \cdot w | Y_0 = w] \; = \\ 1 \; &- \; Pr[\max_{i=0,\ldots,j} Y_i \geq w | Y_0 = 0] \; = \\ 1 \; &- \; 2 \cdot Pr[Y_j \geq w | Y_0 = 0] \Rightarrow \\ Pr[D > j] \; &= \; 1 \; - \; \frac{2}{\sigma\sqrt{2\pi j}} \cdot \int_w^\infty e^{-u^2/(2j\sigma^2)} du \end{aligned} \tag{35}$$

Alternatively, $Pr[D > j]$ may be written as :

$$Pr[D > j] \; = \; 1 \; - \; 2 \cdot \Phi(\frac{-w}{\sigma\sqrt{j}}) \Rightarrow Pr[D > j] \; = \; 2 \cdot \Phi(\frac{w}{\sigma\sqrt{j}}) \; - \; 1 \tag{36}$$

The following inequality then gives an analytic bound for probability $Pr[D > j]$ [Gal60]:

$$\forall u \; \geq \; 0, \frac{u}{\sqrt{2\pi}} \geq \Phi(u) - \Phi(0) \geq \frac{u}{\sqrt{2\pi}} - \frac{u^3}{6\sqrt{2\pi}} \tag{37}$$

and :

$$(23), (36) \Rightarrow \frac{2w}{\sigma\sqrt{2\pi j}} \; \geq Pr[D > j] \; \geq \; \frac{2w}{\sigma\sqrt{2\pi j}} \; - \; \frac{w^3}{3\sigma^3\sqrt{2\pi j^3}} \tag{38}$$

3.3 Reliability and Mean Time to Failure

In order to insert the notion of clock speed in (38), we note that the pulse width w is related to the clock period T. Assuming that the central clock source generates a waveform with a *duty cycle* $\alpha = w/T$, (38) becomes:

$$\alpha_1 \cdot \frac{T}{\sigma\sqrt{j}} \geq Pr[D > j] \geq \alpha_1 \cdot \frac{T}{\sigma\sqrt{j}} - \alpha_2 \cdot \frac{T^3}{\sigma^3\sqrt{j^3}} \tag{39}$$

where :

$$\alpha_1 = \frac{2 \cdot \alpha}{\sqrt{2\pi}}, \quad \alpha_2 = \frac{\alpha^3}{3\sqrt{2\pi}},$$

When there is one clock buffer corresponding to every PE, i.e. $n = N$, the probability that no synchronization failure occurs along the pipeline equals $Pr[D > N]$, and is bounded as:

$$\alpha_1 \cdot \frac{T}{\sigma\sqrt{N}} \geq Pr[D > N] \geq \alpha_1 \cdot \frac{T}{\sigma\sqrt{N}} - \alpha_2 \cdot \frac{T^3}{\sigma^3\sqrt{N^3}} \tag{40}$$

Using property (23), we can also get the following lower bound for $Pr[D > N]$:

$$Pr[D > N] \geq 1 - \frac{\sigma\sqrt{N}}{\sqrt{2\pi}\alpha T} \cdot exp[-\frac{\alpha^2 \cdot T^2}{2\sigma^2 N}] \tag{41}$$

which is useful for smaller values of N, when the right-hand side of (40) becomes negative. By combining (40), and (41) we can state the following theorem:

Theorem 3.1 *Consider a very long one-dimensional systolic array of length N, clocked by a central clock source with frequency $1/T$, via a straight-line clock distribution network. The probability that a clock pulse emitted at one end of the array will eventually reach the other end satisfies the following relation:*

$$\alpha_1 \cdot \frac{T}{\sigma\sqrt{N}} \geq Pr[D > N] \geq max\left\{\alpha_1 \cdot \frac{T}{\sigma\sqrt{N}} - \alpha_2 \cdot \frac{T^3}{\sigma^3\sqrt{N^3}}\ ,\ 1 - \frac{\sigma\sqrt{N}}{\sqrt{2\pi}\alpha T} \cdot exp[-\frac{\alpha^2 \cdot T^2}{2\sigma^2 N}]\right\}$$

When N is very large (long pipelines), the lower bound of $Pr[D > N]$ in relation (40) is dominated by its first term, and therefore the probability of success has the following asymptotic behavior:

$$Pr[D > N] \approx \frac{\alpha_1 \cdot T}{\sigma\sqrt{N}} \tag{42}$$

If $Pr[D > N] = \beta$, where β is a given desired level of reliability, we can use the left-hand side of relation (40) to obtain:

$$\frac{\alpha_1 \cdot T}{\sigma\sqrt{N}} > \beta$$

Since $\beta \approx 1$:

$$T > \frac{\sigma \cdot \sqrt{N}}{\alpha_1} \tag{43}$$

In addition to the previous inequality, the clock period T should always be greater than the computation time t_c. Therefore :

$$T \; > \; max\{t_c, \frac{\sigma \cdot \sqrt{N}}{\alpha_1}\} \tag{44}$$

Whenever straight-line clocking is adopted, this presents a *necessary* but not *sufficient* condition for the systolic array to function with a very high reliability. We therefore conclude that as N gets larger, the clock period should grow faster than the square root of the pipeline length. Otherwise, a frequent occurrence of clock failures is expected. It is interesting to note that this conclusion agrees with the heuristic argument presented in [FK84].

In order to estimate the *Mean Time to Failure*, we use the terms F and $F(t)$, both defined in the discussion of tree clocking. The Mean Time to Failure is defined here as the mean number of clock *pulses* emitted by the central clock source, before the occurrence of some failure, i.e., the disappearence of some clock pulse. We make the assumption that the passage of different clock pulses from the pipelined clock distribution network constitute independent random events. Consequently:

$$F(t) = Pr[F > t] = (Pr[D > N])^t$$

and

$$MTF \; = \; \int_0^{\infty} F(t)dt \; = \; \int_0^{\infty} (Pr[D > N])^t dt \; \Rightarrow \; MTF \; = \; \frac{-1}{\ln Pr[D > N]} \tag{45}$$

An asymptotic bound for MTF can be readily derived using (42):

$$MTF \; \approx \; \frac{-1}{\ln\left(\alpha_1 T/\sigma\sqrt{N}\right)}$$

which completes the proof of the following theorem:

Theorem 3.2 *Consider a very long one-dimensional systolic array of length N, clocked by a central clock source with frequency $1/T$, via a straight-line clock distribution network. The Mean Time to Failure of that pipeline, has the following asymptotic expression:*

$$MTF \; \approx \; \frac{1}{\frac{1}{2} \cdot \ln N - \ln T + \ln \sigma - \ln \alpha_1}$$

4 Conclusions

Our first conclusion refers to the tree clocking scheme: we proved that for very long one-dimensional systolic arrays of length N, a growth in the clock period proportional to $\log_f N$ is sufficient to guarantee very high reliability with regard to synchronization failure.

The second conclusion relates to straight-line clocking: in that case we showed that a *necessary* condition for a systolic pipeline to function with very high reliability, is that its clock period grow

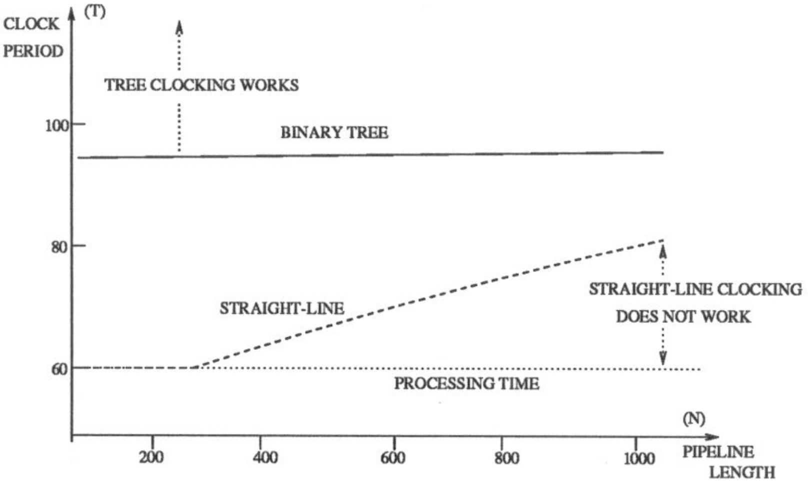

Figure 11: Straight-Line vs Tree Clocking Schemes: $N = 100$ to 1000.

proportionally to $\sqrt{N}$. In both cases, the acceptable reliability levels are determined by corresponding estimates of the Mean Time to Failure.

Given these conclusions, we can see that as the systolic array gets very long, tree clocking is preferable to straight-line clocking.

As a concrete example, figures 11 and 12 show plots of the clock period T for values of pipeline length $N = 100$ to $N = 1000$, and $N = 1000$ to $N = 50000$ respectively, when $t_c = 60$, $\sigma = 1$, and $\alpha = 0.5$. The probability of failure in both schemes is at most 10^{-30}. In figure 11, where the pipeline length is less than 1000, we cannot draw any conclusions about which clocking scheme is better. We do know that the tree clocking scheme will "work" with the required reliability if T is around or above 100, but we don't know the failure-rate of straight-line clocking for that range of T's. In contrast, for longer pipelines ($N >> 1000$), tree clocking is clearly better because it guarantees at least the required reliability for values of T and N ranging in the area between the solid and the dotted curves in figure 12, where straight-line clocking does not work, i.e., presents an unacceptable high rate of synchronization failures.

Although tree-shaped clock distribution networks are substantially more reliable for synchronizing very long one-dimensional systolic arrays than straight-lines, the straight-lines still have some desirable architectural features: they are simple and expandable in their design and implementation. The failure mode of the straight-line clocking scheme is in the disappearence of clock pulses. One way to alleviate this problem might be to replace the buffers by one-shots [GS88,GS], which have the property of emitting a high pulse of standard width. However, one-shots require more hardware, and furthermore, a worst case argument presented in [GS88] has shown that even in this scheme clock pulses may disappear, resulting

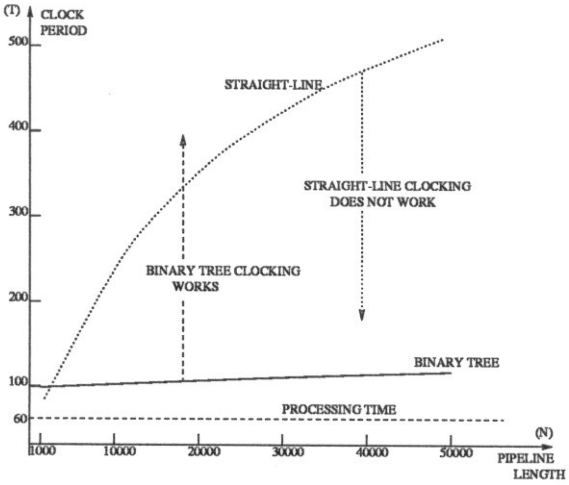

Figure 12: Straight-Line vs Tree Clocking Schemes:N = 1000 to 50000.

in synchronization failure. The analysis of straight-line clocking with one-shots will be investigated in future work.

Acknowledgments

We would like to thank D. Doukas, M. Greenstreet, and R. Lipton for their helpful comments and suggestions. This work was supported in part by NSF Grant MIP-8912100, and U.S. Army Research Office-Durham Contract DAAL03-89-K-0074.

References

[AS64] M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables, Applied Mathematics Series 55.* National Bureau of Standards, 1964.

[Fel68] W. Feller. *An Introduction to Probability Theory and Its Applications*, volume I. John Wiley, New York, 1968.

[FK84] A.L. Fisher and H.T. Kung. Synchronizing Large VLSI Processor Arrays. *IEEE Transactions on Computers*, c-34:734–740, August 1984.

[FP86] E. Friedman and S. Powell. Design and Implementation of a Hierarchical Clock Distribution System for Synchronous Standard Cell/Macrocell VLSI. *IEEE Journal of Solid-State Circuits*, vol. SC-21, No. 2:240–246, April 1986.

[Gal60] Robert Gallager. *Information Theory and Reliable Communication.* Wiley, 1960.

[GS] M. Greenstreet and K. Steiglitz. Bubbles Can Make Self-Timed Pipelines Fast. *Journal of VLSI Signal Processing*, To Appear.

[GS88] M.R. Greenstreet and K. Steiglitz. Throughput of Long Self-Timed Pipelines. Technical report, CS-TR-190-88, Dept. of Computer Science, Princeton University, November 1988.

[HC87] Mehdi Hatamian and Glenn L. Cash. Parallel Bit-Level Pipelined VLSI Design for High-Speed Signal Processing. In *Proceedings of the IEEE*, volume 75, No. 9, pages 1192–1202, September 1987.

[KGE82] S.Y. Kung and R.J. Gal-Ezer. Synchronous versus Asynchronous Computation in Very Large Scale Integrated Array Processors. In *Proc. SPIE, Real Time Signal Processing V*, volume vol. 341, pages 53–64, 1982.

[KS] S. Kugelmass and K. Steiglitz. An Upper Bound on Expected Clock Skew in Synchronous Systems. *IEEE Transactions on Computers*, To Appear.

[KS89] S. Kugelmass and K. Steiglitz. A Scalable Architecture for Lattice-Gas Simulations. *Journal of Computational Physics*, vol. 84:311–325, October 1989.

[KT75] S. Karlin and H. Taylor. *A First Course in Stochastic Processes.* Academic Press, 1975.

[Kug88] Steven D. Kugelmass. *Architectures for Two-Dimentional Lattice Computations with Linear Speedup.* PhD thesis, Princeton University, 1988.

[Kun82] H.T. Kung. Why Systolic Architectures? *IEEE Computer Magazine*, vol. 15, January 1982.

[LL85] R.J. Lipton and D. Lopresti. A Systolic Array for Rapid String Comparison. In *Proceedings of the 1985 Conference on VLSI*, pages 363–376, 1985.

[MMS87] R. Maini, J. McDonald, and L. Spangler. A Clock Distribution Circuit with a 100ps Skew Window. In *Proceedings of Bipolar Circuits and Technology Meeting*, pages 41–43, 1987.

[Ros85] Sheldon M. Ross. *Introduction to Probability Models.* Academic Press, 1985.

[Tex76] Texas Instruments. *The TTL Data Book for Design Engineers*, 1976.

[Wag88] Kenneth Wagner. Clock System Design. *IEEE Design and Test of Computers*, pages 9–21, October 1988.

A Appendix

Lemma A.1 *For any random variable y, any α, β, and any random event C, it is true that:*

$$Pr[\alpha < y \mid C,\ \beta < y] \ \geq\ Pr[\alpha < y \mid C] \tag{46}$$

Proof: By the definition of conditional probability, we can readily see that:

$$Pr[\alpha < y \mid C,\ \beta < y] \ = \ \frac{Pr[\alpha < y, \beta < y|C]}{Pr[\beta < y|C]}$$

under the assumption that $Pr[\beta < y|C] \neq 0$.
When $\beta \geq \alpha$, $Pr[\alpha < y, \beta < y|C]$ becomes $Pr[\beta < y|C]$ and thus

$$Pr[\alpha < y \mid C,\ \beta < y] = 1$$

which means that the inequality (46) holds.
When $\alpha > \beta$, $Pr[\alpha < y, \beta < y \mid C] = Pr[\alpha < y \mid C]$. Since $(1/Pr[\beta < y \mid C]) \geq 1$, again we can easily see that (46) is true. □

Lemma A.2 *Let y_i, $i = 1, 2, \ldots, N$ be independent identically distributed (iid) random variables, let τ_j, $j = 1, 2, \ldots, n$ be sets of y_i's (not necessarily disjoint), and let*

$$t_j \ = \ \sum_{l \in \tau_j} y_l \ , j = 1, 2, \ldots, n$$

Then for each $j \in \{1, 2, \ldots, n\}$ it is true that:

$$Pr[\alpha \ < \ t_j \ |\alpha < t_1 \ , \ldots, \alpha < t_{j-1}, \ \alpha < t_{j+1}, \ldots, \alpha < t_n] \geq Pr[\alpha < t_j] \tag{47}$$

Proof: Let L be the intersection of events $[\alpha < t_i]$ participating in the conditioning of the probability at the left side of (47) :

$$L \ = \ [\alpha < t_1, \ldots, \alpha < t_{j-1}, \alpha < t_{j+1}, \ldots, \alpha < t_n].$$

For any k_1 we denote by L_{k_1} the portion of L which includes only events $[\alpha < t_i]$ such that the set τ_i contains the random variable y_{k_1}. These events can be expressed also as $[\alpha - t_i^{(k_1)} < y_{k_1}]$, where $t_i^{(k_1)} = t_i - y_{k_1}$. Let L'_{k_1} be the remaining portion of L, i.e., $L = L_{k_1} \cap L'_{k_1}$.
Clearly, the probability in (47) can be written as follows:

$$Pr[\alpha \ < \ t_j \mid L\,] = \ Pr[\alpha - t_j^{(k_1)} < y_{k_1} \mid L_{k_1} \cap L'_{k_1}]$$

This reminds us of relation (46), since all events of L_{k_1} can take the form $[\alpha - t_i^{(k_1)} < y_{k_1}]$. Consequently, using Lemma 2.1, we get :

$$Pr[\alpha \ < \ t_j \mid L\ ] \ \geq \ Pr[\alpha - t_j^{(k_1)} < y_{k_1} | L'_{k_1}]$$

Repeating this procedure for appropriate k_2 such that y_{k_2} is included in some of the events defining L'_{k_1}, we write $L'_{k_1} = L_{k_2} \cap L'_{k_2}$, and get:

$$Pr[\alpha - t_j^{(k_2)} < y_{k_2} | L_{k_2} \cap L'_{k_2}] \;\geq\; Pr[\alpha - t_j^{(k_2)} < y_{k_2} | L'_{k_2}].$$

Following the same technique we can go on, decreasing at each step the number $|L'_{k_2}|$ of events that participate in the conditioning of $Pr[\alpha < t_j | L'_{k_2}]$. Finally, we get for some q either $L_{k_{q-1}} = L'_{k_q}$, or $L_{k_{q-1}} = L_{k_q}$. In the first case, obviously

$$Pr[\alpha < t_j | L'_{k_q}] \;=\; Pr[\alpha < t_j],$$

whereas in the second case :

$$Pr[\alpha < t_j | L_{k_q}] \;\geq\; Pr[\alpha < t_j]$$

by Lemma A.1. This concludes the proof of (47). □