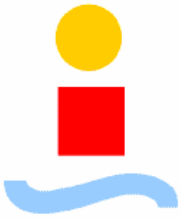


ESCUELA SUPERIOR DE INGENIEROS

UNIVERSIDAD DE SEVILLA

Proyecto Fin de Carrera

Investigación y Simulación del Clock Skew en Circuitos Integrados Modernos

Luis Manuel Santana Gallego

Nota

Este proyecto se ha realizó durante el curso 2005-2006 en la Facultad de Ingeniería Electrónica de la Universidad Técnica de Lodz (Polonia) gracias a una beca Erasmus. El tutor del proyecto fue Mr. Grzegorz Tosik. El estudio del clock skew realizado en este proyecto formará parte del trabajo que está realizando acerca de las redes de distribución de reloj en sistemas digitales VLSI, ya que el clock skew es uno de los parámetros de diseño más importantes en la actualidad dentro de este campo de la electrónica digital. El documento que se presenta a continuación es el que se entregó en la Univeridad Técnica de Lodz y fue redactado en inglés.

POLITECHNIKA LODZKA

Faculty of Electrical, Electronic, Computer and Control Engineering of the Technical University of Lodz

Final Project

Investigation and simulation of the clock skew in modern integrated circuits

Luis Manuel Santana Gallego

Erasmus Student

Supervisor:

Grzegorz Tosik

Lodz, June 2005

POLITECHNIKA LODZ
Department of Electrotechnic and Electronic

Luis Manuel Santana Gallego
Erasmus Student
Final Project

Investigation and simulation of the clock skew in modern integrated circuits

Abstract

Clock skew is one of the main concerns when a clock distribution network is being designed. It imposes important constraints on the system performance and, which the newer technologies, its effects are becoming more significant. There are some existent models to estimate the clock skew within clock distribution networks. The main objective of this work is to present and analyse them in order to decide which is the best model to estimate the clock skew in a generic global clock distribution network. After that, a prediction for future technologies could be done thanks to the conclusions that are going to be drawn.

Contents

1. Introduction

The first integrated circuits appeared in the early 1960s. Called "Small-Scale Integration" (SSI), the circuits contained no more than some tens of transistors and it made possible to develop the first lightweight digital computers. The next step, taken in the late 1960s, introduced devices that contained hundreds of transistors on each chip and was called "Medium-Scale Integration" (MSI). They were attractive because they allowed more complex systems to be produced using smaller circuit boards and less assembly work. Further development led to "Large-Scale Integration" (LSI) in the mid 1970s, with tens of thousands of transistors per chip. LSI circuits began to be produced in large quantities for computer memories and pocket calculators.

The final step in the development process, starting in the 1980s, was "Very Large-Scale Integration", with hundreds of thousands of transistors at the beginning, but with several millions in the latest years (ULSI, "Ultra Large-Scale Integration" is the name when the circuit contains more than 1 million of transistors, but there is no qualitative difference between VLSI and ULSI). For the first time, it became possible to fabricate a CPU on a single integrated circuit, appearing the microprocessors. Also, the first megabit RAM chips were produced.

The introduction of VLSI systems had several important consequences. One of them is that the system operation speed could be considerably increased up to few gigahertz. As semiconductor technologies operate at increasingly higher speeds, system performance has become principally limited by the ability of synchronizing the data flow signals and not only by the delay of the individual logic elements. In high-speed digital systems, clock distribution is a challenging problem that consumes an important fraction of resources and design time.

In a synchronous digital system, clock signals are used to define a time reference for the data flow within the system. This function is essential for the operation of this kind of systems. Therefore, it is fundamental to give much attention to the characteristics of the clock signal and its distribution network. System performance and reliability is directly affected by the design of the clock distribution network.

As technology sizes decrease, the delay of interconnection lines utilized in the clock distribution network increases considerably. The reason is that wire resistance is incremented substantially (smaller wire width and thickness) and, therefore, the RC wire constant. Clock skew, the difference between clock signal arrival times to two sequentially adjacent registers, is directly related to the clock path delay. A higher path delay causes a higher clock skew between these paths.

Clock skew is a key challenge for high-speed circuit designers because it can degrade performance and cause chip failures. There is a designer's rule of thumb that imposes that clock skew must be lower than the 10% of the clock period. As clock frequency goes up, better clock distribution networks are required to keep skew at a constant fraction of the cycle time. The growing die size, clock loads and process variability aggravate the problem. Even when clock skew is completely designed to be zero, environmental and processing variations lead to significant amounts of skew.

At first, the motivation of this work is to give a general overview of the different existent solutions for the implementation of a clock distribution system. Their main parameters and design techniques will be presented. Theoretical background and sources of clock skew are described. The main goal is to analyse the different existent models for estimating the clock skew according to all technology parameters. They are compared for a generic global clock distribution network (an H-tree). Simulations are done with a JAVA program developed to calculate the clock skew according to each model equations. At the end, a prevision for the clock skew value in next years will be given thanks to the use of these models.

This work is organized in the following parts:

- In section 2, a general overview of clock distribution networks is presented. Firstly, synchronous digital systems are described and also the function of a clock distribution network within them. Main strategies of design are presented, starting with the buffered trees and all their possible topologies. Primary parameters of a clock distribution network are introduced because they determine how the clock system should be designed. At the end of this section,

some examples of the clocking issue in commercial digital system are explained.

- In section 3, clock skew is investigated in depth. At first, a theoretical background of the clock skew is presented. Secondly, all possible causes of clock skew (process and circuit parameter variations) are introduced. The third part of this section is one of the most important of the work. Existent clock skew models are described and their equations to calculate clock skew estimations are presented.

- In section 4, the main goal of the work is described. Clock skew models are compared for a specific global clock distribution network (a 256 nodes H-tree, 130 nm technology) that is designed according to some optimization methods (optimal wire and buffer sizes). Different simulations are made with a JAVA program. The results are analysed. At the end, the same simulations are made but with the predicted technology parameters for the future.

- Finally, in section 5, the conclusions of this work are presented.

2. Clock distribution networks

2.1. Synchronous digital systems

In a synchronous digital system, data signals are usually stored in a clocked register. This register is waiting for the arrival of the clock signal. When it happens, data signal leaves the bistable register and propagates through the combinatorial network and, for a properly working system, enters in the next register and is fully latched into that register before the next clock signal appears. In conclusion, the components that make up a general synchronous system are the following three subsystems:

1) Memory storage elements.
2) Logic elements.
3) Clocking circuitry and distribution network.

Clock signals are usually considered as simple control signals. However, they have some characteristics and attributes very special. Clock signals must be provided with a very clean and sharp waveform because they are the time reference of any data signal in a synchronous digital system. Furthermore, technology scaling particularly affects to clock signals. For example, the interconnect line resistivity is hugely increased when the line dimensions are decreased (width and thickness) and, consequently, the delay time grows considerably. This increased line resistance is one of the main reasons of the growing interest in the study of clock distribution network. Finally, differences in the delay of clock signals when they arrive at their destinations, the clock skew, can seriously affect system performance, as well as create catastrophic race conditions in which a data signal may be incorrectly latched in a register. The proper design of the clock distribution network ensures that critical timing requirements are satisfied and no race conditions exist.

Two registers are sequentially adjacent if there is at least a sequence of logic blocks that connects the output of the first register R_i (initial) with the input of the second register R_f (final). Otherwise, any event at the output of R_i does not affect the input of R_f.

We can define a local data path in the case of two sequentially adjacent registers (figure 2.1).

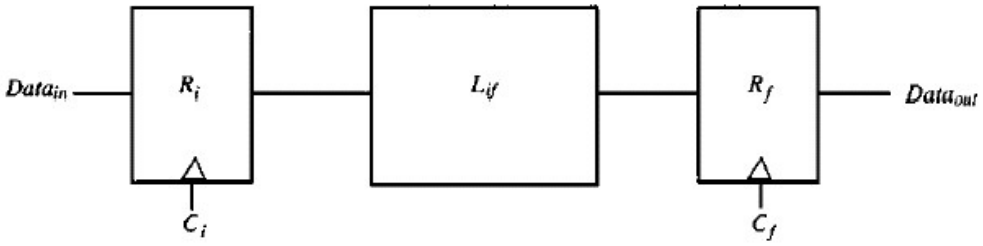


Figure 2.1: Local data path components [FRI-00].

The minimum clock period T_{CP} (min) between any two registers in a sequential data path is given by:

$$T_{CP}(\min) = \frac{1}{f_{CLK}(\max)} = T_{PD}(\min) + T_{skew} \qquad (2.1)$$

where

$$T_{PD} = T_{C-Q} + T_{logic} + T_{int} + T_{setup} \qquad (2.2)$$

The total path delay T_{PD} of a data path is the sum of:

- $T_{C\text{-}Q}$: The maximum time required for the data signal to leave the initial register once clock signal C_i arrives at the initial register.

- T_{logic} and T_{int}: The time necessary for the data signal to propagate through the logic block L_{if} and interconnection lines.

- T_{setup}: The time required for the data signal to successfully propagate and latch within the final register of the data path.

The sum of delays components in (2.2) must satisfy the timing constraints of (2.1) in order to support the minimum clock period T_{CP} (min), which is the inverse of the maximum clock frequency f_{clk} (max). It is important to note that T_{skew} can be positive or

negative depending which clock signal, C_i or C_f, arrives earlier. The waveforms depicted in figure 2.2 show the timing requirement of (2.1) being just satisfied (the data signal arrives at R_f just before the clock signal arrives at R_f).

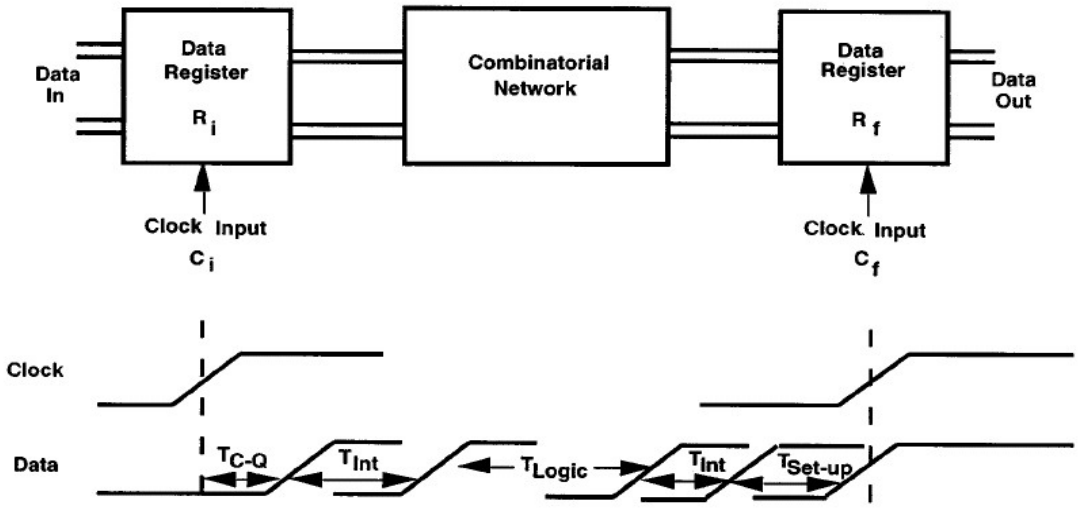


Figure 2.2: Timing diagram [FRI-00].

2.2. Clock distribution networks design

In a synchronous digital system, the clock signal is used to define a time reference for the movement of data signals within it. The clock distribution network distributes the clock signal from the source point to all the elements that need it. Since this function is vital to the operation of a synchronous system, much attention has been given to the characteristics of these clock signals and the electrical network used in their distribution. Clock signals are often regarded as simple control signals, however, these signals have some very special characteristics and attributes that must be fulfilled.

Many approaches for designing clock distribution networks exist, from customized design techniques to automatic algorithms. The requirement of distributing a strictly controlled clock signal to each synchronous register, which are located on a large hierarchically structured integrated circuit, within some specific temporal bounds is difficult and problematic. Furthermore, there are important tradeoffs among system

speed, physical die area and power dissipation when the clock distribution network is planned [FRI-00].

The most common strategy when a clock distribution network is designed is to divide it into different hierarchical levels. The difficulty of applying symmetric clock distribution strategies is that they do not easily support the ability of dividing large VLSI systems into hierarchically structured functional blocks (in figure 2.3, the floorplan of a structured custom VLSI circuit is depicted). Preferably, each large functional block would contain its own locally optimized clock distribution network (lower hierarchical levels) to satisfy the local timing of that particular functional block.

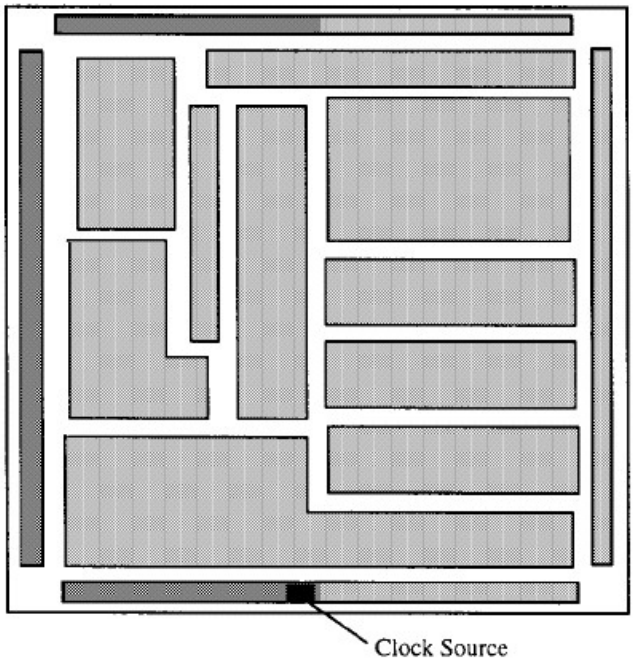


Figure 2.3: Floorplan of a structured custom VLSI circuit [FRI-00].

In conclusion, in a hierarchical clock distribution network, there will be a global clock distribution network in the highest level that distributes the clock signal to each lower level where different clock distribution strategies can be applied in order to optimize the performance of a particular functional sub-block. Each level of the structure can be implemented in a different metal level, preferably on the higher levels where the line sizes are bigger and their resistance lower. This hierarchical structure is depicted in figure 2.4:

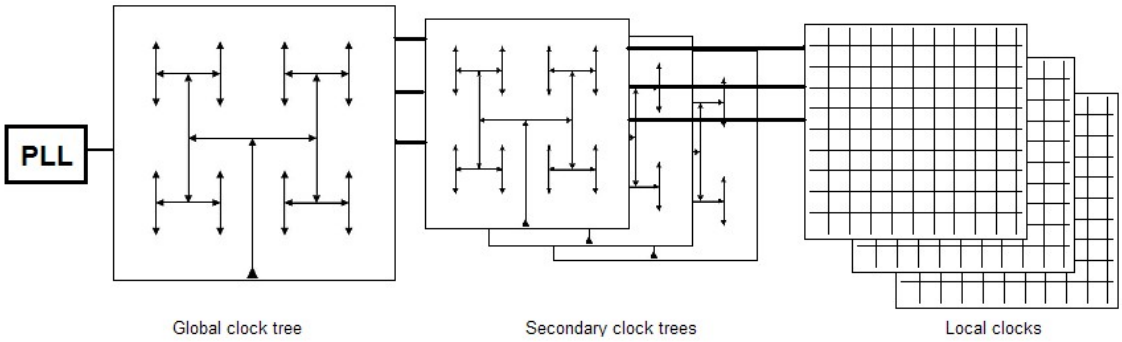


Figure 2.4: Hierarchical clock distribution system.

With regard to the clock generation, there are many issues to consider in the clock generator design. Both power dissipation and jitter impose important constraints when the generation system is chosen, but operating frequency plays the major role, especially in today's systems. A phase locked loop clock generator is nowadays the best solution. Figure 2.5 illustrates the basic block diagram of a PLL-based clock generator.

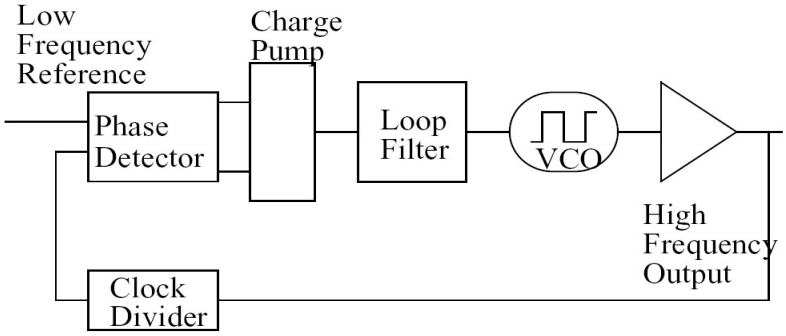


Figure 2.5: Generic PLL block diagram [STE-97].

In a synchronous system, the use of a PLL clock generator can eliminate the clock skew caused by active device variations. Since the output of the PLL is in-phase with the input, if clock distribution buffers of the distribution network are included within the loop, their delay is essentially removed from the system. This does not eliminate the chip-to-chip or on-chip clock skew due to passive device variations and interconnect delay [STE-97].

Various clock distribution strategies have been developed. The most common and general approach to equipotential clock distribution is to use buffered trees (section

2.2.1), especially for custom VLSI circuits and in the final levels of a hierarchical clock distribution network. Furthermore, there are different topologies for these buffered trees that can be applied in some cases to satisfy specific conditions (section 2.2.2). A mesh version exits. Also, there is a grid-like structure. Other very important strategy is the symmetric H-tree, which is generally used in the global part of clock distribution networks, distributing the clock signal to each different functional block. Compensation techniques to control clock signal properties within the clock distribution network exist and are presented in section 2.2.3.

2.2.1. Buffered clock trees

The most common strategy to distribute the clock signal in VLSI systems is to insert buffers, either at the clock source and/or along clock paths, forming a tree structure as shown in figure 2.6. The single clock source is called root. The trunk is the initial tree portion. Branches are the individual paths driving each register. These registers are called leaves.

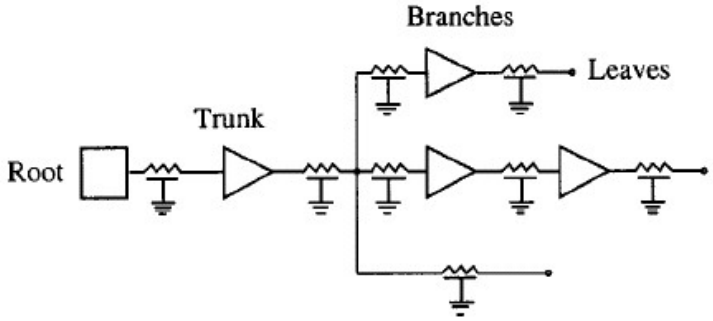


Figure 2.6: Buffered clock tree [FRI-00].

If the buffer interconnect resistance at the clock source is small compared to the output buffer resistance, a single buffer is often used to drive the whole clock distribution network. The primary requirement is that the buffer must provide enough current to drive the entire network capacitance, both interconnection and fanout, while maintaining high-quality waveform (short transition times). Also, it is necessary to miniminize the interconnection resistance effects by ensuring that the output buffer resistance is much higher than the interconnection section resistance that it drives. This

strategy may be appropriated if the clock completely distributed on metal, making load balance of the network less critical [FRI-00].

An alternative to use a single buffer at the clock source is to distribute buffers throughout the clock distribution network. This approach requires additional area, but improves greatly the precision and control of the clock signal waveform and it is necessary if the interconnection line resistance is not insignificant. Furthermore, distributed buffers serve the double function of amplifying the clock signal degraded by the distributed interconnection impedances and isolating local clock distributing networks (lower levels in the hierarchal structure) from upstream load impedances. The number of buffer stages between the clock source and each clocked registers depend upon the total capacitance load and the maximum permissible clock skew. It is worth noting that buffers are the main source of clock skew within a well-balanced clock distribution network since active devices characteristics vary much more greatly than passive device characteristics.

2.2.2. Topologies

- Binary tree

Some early clock routing algorithms [RAM-89], [JAC-90], [KAH-91] defined the delay as a measure of the total wire length along a path. These algorithms attempt to equalize the lengths of each net from the root of the clock tree to each of the leaf nodes. The strategy used is to construct binary tree-like structures with the clock pins at the leaf nodes. Minimal skew clock distribution networks are created using a recursive bottom-up approach. The point where two zero-skew clock sub-nets connect is chosen such that the effective delay from that point to each clocked register is identical. This process continues up the clock distribution tree. The connection point of each new branch is chosen to satisfy the zero-skew design goal. The layout process terminates when the root (or source) of the clock tree is reached. The schematic diagram of this geometric matching process is illustrated in figure 2.7.

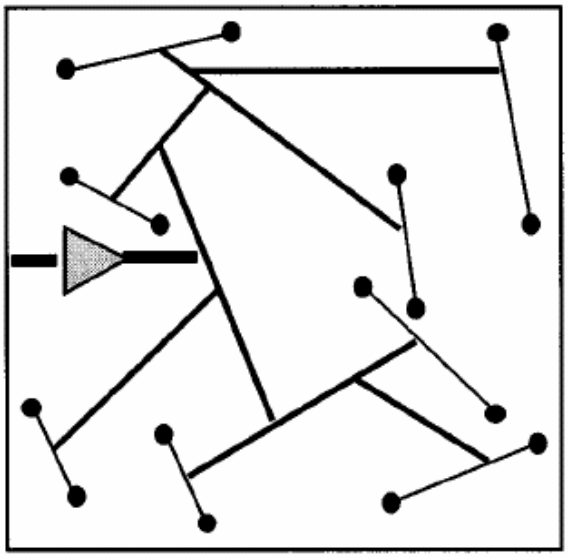


Figure 2.7: Geometric matching for zero clock skew [FRI-00].

Elmore delay instead of the path length can be other design rule in order to determine the binary tree. Here, the delay along a path is the summation of the products of the branch resistance and the downstream capacitance of every branch on the path from the root to the clock pin of the register:

$$T_{Di} = \sum_{k} R_{ki} C_k \qquad (2.3)$$

where C_k is the capacitance at node k and R_{ik} is the resistance of the portion of the unique path between the input and the output node i, that is common with the unique path between the input and node k [FRI-00].

Automated algorithms present many problems. For example, no intermediate buffers can be placed and a single clock buffer must drive the whole network. Also, the possibility of impossible wiring schemes can arise (two wires crossing in a point).

- Symmetric H-trees and X-trees

One of the most important approaches for distributing clock signals utilizes a hierarchy of planar symmetric X-tree or H-tree (figure 2.8) to ensure zero clock skew. It

is achieved by maintaining identical the distributed buffers and interconnections from the clock source to each clocked register.

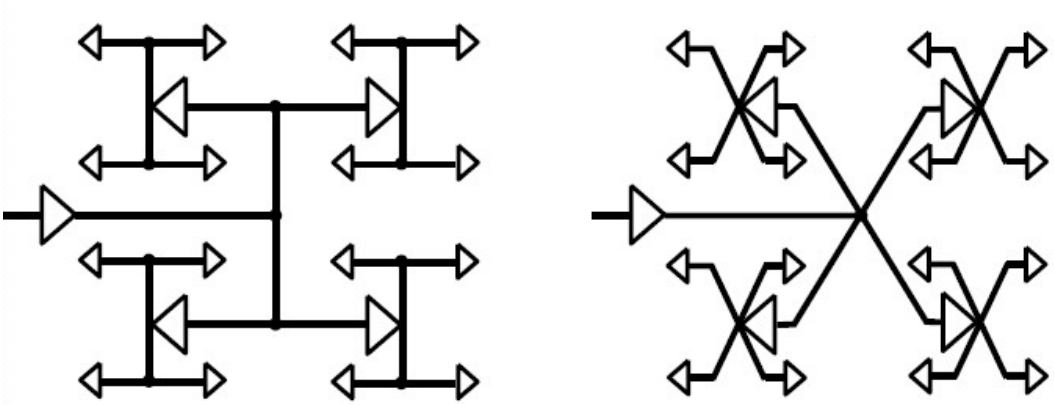


Figure 2.8: H-tree and X-tree.

In this approach, the clock driver is connected to the centre of the main “H” (or “X”). The clock signal is transmitted to the four corners of the main “H”. These four almost identical clock signals are provide to next level “H” structures. The process continues through several levels progressively. The H-tree end points are used to drive local registers or to provide the clock signal to the next hierarchical clock network level.

Here, each clock path, from the clock source to the local registers, has practically the same delay. The primary cause of different delays at the clock paths is the process parameter variations that affect interconnection impedances and, in particularly, any distributed buffer. The amount of clock skew in a H-tree clock distribution network is strongly dependent upon the physical size, semiconductor process control and the number of distributed buffers utilized in the network [FRI-00].

Planar H-tree structures place constraints on the physical layout of clock distribution networks as well in the methodology design. Furthermore, the interconnect capacitance and the power dissipated is much greater as compared with standard buffered clock trees since the wires total length tend to be greater. This capacitance growth exemplifies an important tradeoff between clock delay and clock skew when high-speed clock distribution networks are being designed [FRI-00].

Symmetric structures are utilized to minimize the clock skew. However, an increase in clock signal delay is incurred. Therefore, the increased clock delay must be considered when choosing between buffered tree and H-tree clock distribution networks. Also, since clock skew only affects sequentially adjacent registers, the obvious advantages of using highly symmetric structures to distribute clock signals are significantly degraded. There may, however, be certain sequentially adjacent registers distributed across the integrated circuit. For this situation, a symmetric H-tree structure is particularly appropriate to distribute the global portion of the clock network [FRI-00].

- Mesh

Occasionally, a mesh version of a clock tree structure is used. Nodes are connected in other metallization level by a mesh to minimize the interconnect resistance within the clock tree. A mesh structure places the branch resistances in parallel, minimizing the clock skew. In the figure 2.10, a 3D view of a mesh structure is illustrated.

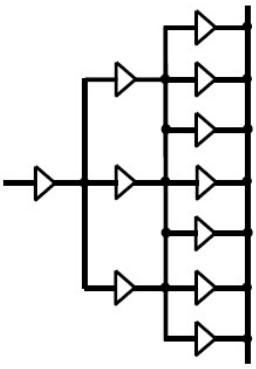


Figure 2.9: Mesh structure.

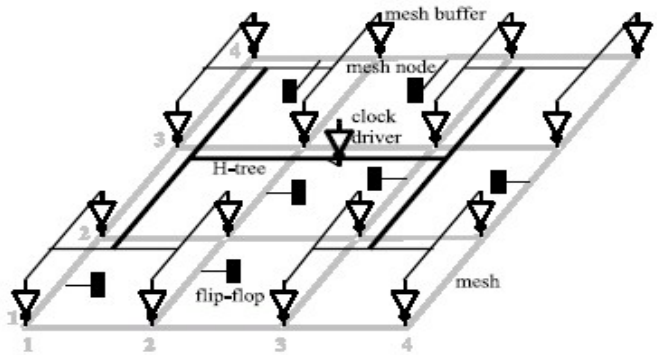


Figure 2.10: 3D view of a mesh structure [YEH-06].

This is the traditional way to distribute clocks in high-performance microprocessors. Under any H-tree, there is usually placed a mesh. It achieves very low skew and jitter (i.e., robustness to variations), but sacrifices power and is difficult to analyze.

- Grid

A clock grid is probably the simplest clock distribution structure and is usually utilized for distributing the clock signal in the final stages of a hierarchical network. Essentially, it consists in a grid of wires driven by one or more buffers. In figure 2.11, this structure is depicted. The main advantage of a clock grid is that the distribution of the clock signal to each destination point is simple. The reason is that the regular structure arrives near to each place where clock signal is needed. Furthermore, the clock grid ensures good process variation tolerances and an easy design methodology.

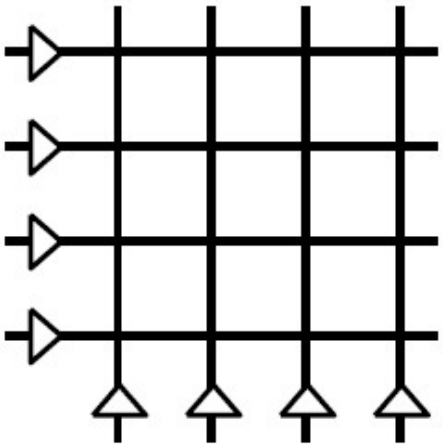


Figure 2.11: Grid structure.

There are some disadvantages with this structure. The main of them is that the grid wires increase the total capacitive load of the clock network, increasing the power consumption. Furthermore, the clock grid is inefficient with regard to used area.

2.2.3. Compensation techniques

It is usual to utilize compensation techniques when clock distribution networks for customized VLSI circuit are designed. They are necessary to minimize the variation of interconnect impedances and capacitive loads between different clock paths. Figure 2.12 shows this situation. Nodes i, j and k represent different loads. The paths to these nodes also have different buffering levels. Buffer locations are often chosen so that the active buffer output impedance is comparable to or greater than the interconnect resistance seen at the buffer output. This characteristic ensures that the locally distributed interconnect section can be accurately modelled as being mostly capacitive. However, in general, the interconnect impedance should be modelled as a distributed RC section of interconnect.

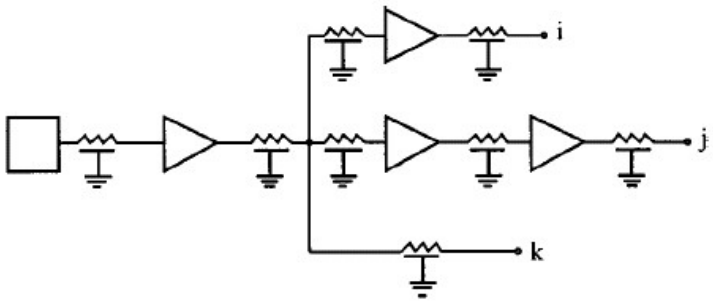


Figure 2.12: Different interconnect impedances and capacitive loads [FRI-00].

There are two important techniques to control and compensate the delay of each clock signal path and minimize the skew between them. Above all, they are effective in buffered tree clock distribution networks. First of them is to insert passive RC elements [IBM-85]. The second is to size geometrically transistor widths [FRI-86].

Clock buffers are located along the clock path such that the highly resistive long lines drive loads with low capacitance and vice versa. Placing a centralized module of parameterized clock buffers within the buffered tree (figure 2.13) it is possible to control the clock skew. Parameterizing the current of clock buffers in the module compensates the variation of the clock signal delay between each one of the functional elements.

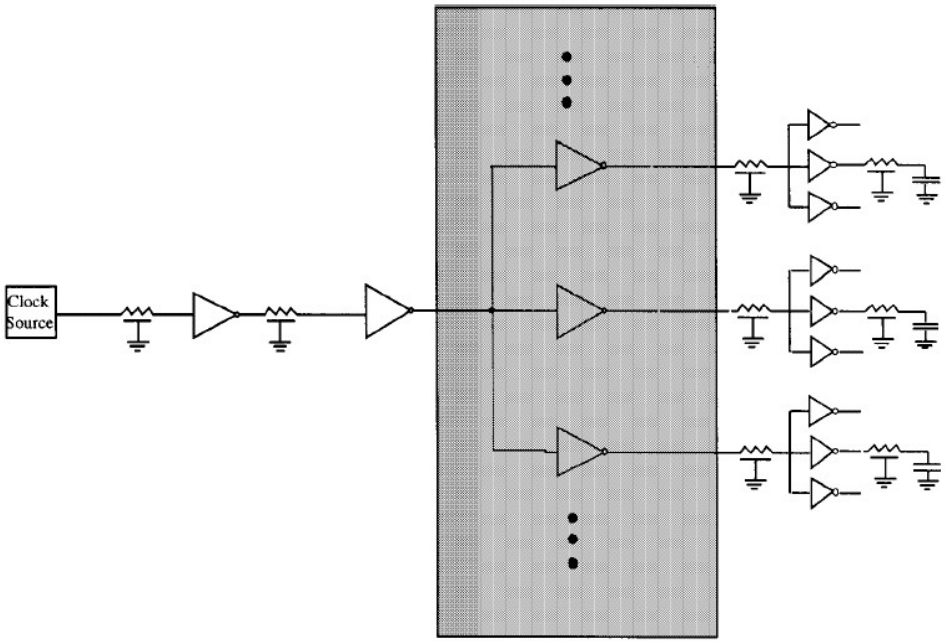


Figure 2.13: Module of parameterized buffers [FRI-00].

- Compensation advantages

The most important advantage of using compensation techniques is controlling and reducing the on-chip skew. Furthermore, the clock signal delay from the clock source to the clocked register is reduced. The reason is the improved partition of the RC loads. Since inverters located within each functional block often drive large capacitive loads, the interconnection impedance (resistance in particular) driven by any specific clock buffer is small compared to output buffer resistance. The fairly long distances of the clock signal paths are fairly resistive. However, these paths are isolated of the highly capacitive loads. In this way, RC time constants are decreased, reducing the total clock delay (RC delay was analysed in section 2.3.1).

Other advantage is that the partitioning of clock system becomes easier with the buffer insertion. The VLSI circuit design can be partitioned hierarchically in many different functional sub-blocks in which different clock distributions strategies can be implemented according different requirements. The utility of compensation techniques is very dependent upon the ability of characterize the interconnect devices and impedances within the clock distribution network. With accurate impedance

estimations, parameterized buffers can be designed to satisfy a specific clock skew schedule.

- Compensation disadvantages

Is important to note an important disadvantage of this compensation technique. Unlike interconnection impedances, transistor conductance tends to be highly sensitive to variations in supply voltage and process and environmental conditions (radiations and temperature for example). The delay of a clock signal path dominated by device impedances is more susceptible to the clock skew than a path dominated by interconnection impedances.

2.3. Clock distribution network parameters

The function of a clock distribution network is to provide a global time reference to all the clocked registers of a synchronous digital system. This way, it regulates the flow of information within the chip. In this section, main parameters involved in a clock distribution network are described. They are essential when the clock system is designed.

2.3.1. Propagation delay

One of the targets when a clock distribution system is designed is to minimize the propagation delay time of the clock signal from the source to the registers. It is required because the clock period time must be larger than the time for the clock signal to propagate from the clock source to the end of the longest clock path. The clock signal in the source cannot change before it arrives to every destination.

A clock signal path within a clock distribution network has a single input, the clock generator, and a single output, the register clock input. Branches are typically composed of distributed buffers and interconnect sections. In order to calculate the clock path

delay and skew, a simple model of a CMOS inverter driving another inverter with line resistance and capacitance between the two inverters is often used (figure 2.14).

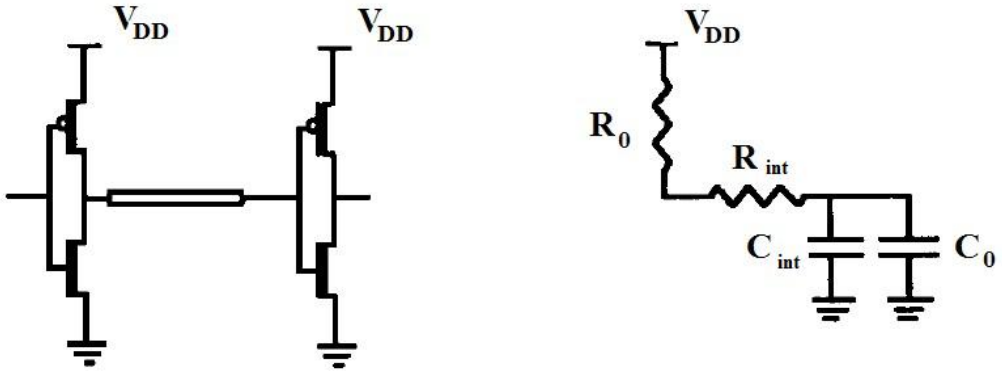


Figure 2.14: Interconnection delay model [AFG-89].

A well-known empirical estimate of the rise or fall time (90 %) of a single CMOS inverter driving an interconnect section with a capacitive load (representing the following CMOS inverter) is [SAK-93]:

$$T_{R/F} = 1.02 R_{int} C_{int} + 2.30\left(R_0 C_{int} + R_{int} C_0 + R_0 C_0\right) \qquad (2.4)$$

where R_{int} and C_{int} is the resistance and capacitance of the interconnect section respectively, and R_0 and C_0 is the output on-resistance of the driving buffer and the input load capacitance of the following buffer respectively. These four parameters are given by:

$$\begin{aligned}
&R_0 = \frac{1}{K \cdot \left(V_{DD} - V_T\right)}, \quad K = \frac{\mu \cdot C_{ox} \cdot W}{L_{eff}}, \quad C_{ox} = \frac{\varepsilon_{ox}}{t_{ox}} \\
&C_0 = C_{ox} \cdot W \cdot L_{eff} \\
&R_{\text{int}} = \frac{\rho}{W_{\text{int}} \cdot t_{\text{int}}} \cdot L_{\text{int}} \\
&C_{\text{int}} = \varepsilon_{ILD} \cdot L_{\text{int}} \cdot \left[\frac{W_{\text{int}}}{T_{ILD}} + 0.77 + 1.06 \cdot \left(\frac{W_{\text{int}}}{T_{ILD}} \right)^{0.25} + 1.06 \cdot \left(\frac{t_{\text{int}}}{T_{ILD}} \right)^{0.5} \right]
\end{aligned} \qquad (2.5)$$

where:

- W and L_{eff}: width and effective length of the transistor.
- C_{ox}: gate unit area capacitance.
- t_{ox}: gate oxide thickness.
- μ: charge carrier mobility.
- V_T: threshold voltage.
- ρ: metal resistivity.
- ε_{ox}: oxide dielectric constant.
- ε_{ILD}: interlevel dielectric constant.
- W_{int}, L_{int} and t_{int}: width, length and thickness of the interconnection line.
- T_{ILD}: Interlevel dielectric thickness.

C_{int} is derived from the empirical formula given in [WES-94] that includes the contribution of fringing fields. Occasionally, they are not taken into account.

The physical delay model represented by (2.3) is a fairly simple approximation of the delay of a CMOS inverter driving a distributed *RC* impedance. More complex and accurate delay models exist.

2.3.2. Clock skew

Clock skew is another important parameter to take into account when a clock distribution network is designed. It imposes timing constraint as propagation delay does. Clock skew can be defined as the difference in arrival times of clock signals to two sequentially adjacent clocked registers. In figure 2.15, clock skew is illustrated. There are two possibilities depending on in which register the clock signal arrives before.

Clock distribution systems are generally designed to achieve that clock signals arrive just at the same moment to every destination (zero skew design), but there are several factors that can cause differences in the propagation delay time between different clock paths (i.e. different line lengths, differences in delays of any active buffer, process and environmental parameter variations). In this case, the clock skew

can seriously affect to the system performance, reducing the maximum clock system frequency and causing catastrophic race conditions.

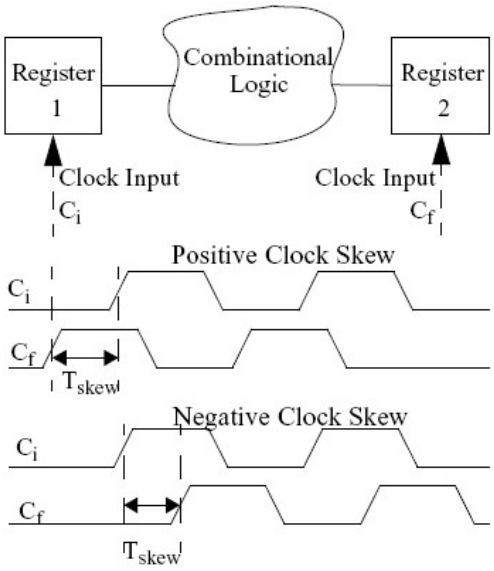


Figure 2.15: Clock Skew [STE-97].

A complete analysis of the clock skew is presented in section 3.1.

2.3.3. Clock jitter

Jitter represents the time varying behaviour of the clock signal. The main effect of jitter is that clock edge locations are shifted with regard to the ideal edge location. Therefore, the duty cycle of a clock signal could be different to the ideal value.

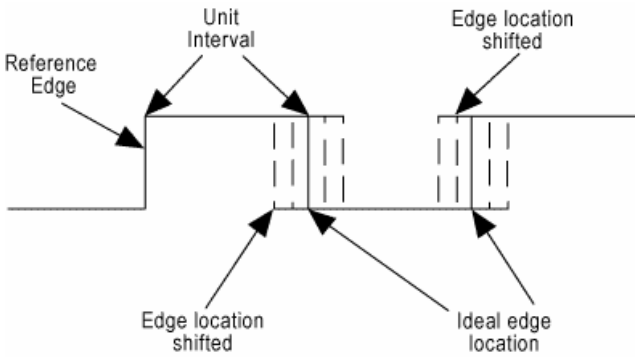


Figure 2.16: Clock jitter.

Clock source has always a specific clock jitter, but also clock distribution network introduces jitter in the clock signal. The noise sources that contribute to jitter most significantly are the following [STE-97]:

- Noise coupled through the circuits' power and ground connections.
- Noise coupled through adjacent or intersecting wires.
- Noise inherent to the circuits' transistors themselves.

2.3.4. Clock power dissipation

Another essential parameter in the design of a clock distribution network is the power dissipated by it. In a VLSI modern system, clock distribution networks have to drive thousands of registers, creating a great capacitive load that should be efficiently sourced. Therefore, each clock signal transition changes the state of each capacitive node within the network. Both high capacitive loads and the continuous demand of higher frequencies have led to an increasingly larger proportion of the total power of a system dissipated within the clock distribution network, in some applications much greater than 25% of the total power.

Total power consumption can be divided in three main sources:

$$P_{total} = P_{dyn} + P_{SC} + P_{lkg} \qquad (2.6)$$

where:

- P_{dyn}: It is the average dynamic power dissipation caused by the continuous charging and discharging of the capacitive nodes within the clock distribution network (parasitic interconnection capacitances and buffer input capacitances).

$$P_{dyn} = fC_{total}V_{DD}^2 \qquad (2.7)$$

- P_{SC}: It is the short-circuit power dissipation. The cause is that in a CMOS inverter, there is a small period in which both transistors, NMOS and PMOS, are simultaneously conducting. The power dissipated in this period is dependent on short-circuit current I_{SC}, input and output signal slopes, device parameters and supply voltages.

$$P_{SC} = I_{SC} V_{DD} \qquad (2.8)$$

- P_{lkg}: It is the power dissipated when there is no activity on the circuit. It is due to the leakage current I_{lkg}: reverse-bias diode leakage and the sub-threshold leakage. The first occurs when the drain-to-bulk voltage of either the PMOS or NMOS transistor is reverse biased. The second occurs when the gate-to-source voltage of the inverter is below threshold voltage, but it is enough to put the inverter into weak inversion.

$$P_{lkg} = I_{lkg} V_{DD} \qquad (2.9)$$

For CMOS VLSI circuits, the primary component of power dissipation is the dynamic power. The goal is to minimize the expression (2.6) without decrease system clock frequency. Therefore, for a given circuit implementation low dynamic power dissipation is best achieved by employing certain design techniques that minimize the power supply and/or the capacitive load.

2.4. Clock distribution networks examples

There are a lot of different examples to show different approaches for clock distribution networks in synchronous digital VLSI systems, from highly specialized customized circuits to commercial processors. These examples are going to be shown in chronological order to present a historical perspective of the general clock distribution design problem and related tradeoffs over the past two decades.

2.4.1. The DEC/Compaq 64-Bit Alpha Microprocessor Family

The DEC/Compaq 64-bit Alpha microprocessor family represent perfectly the evolution of VLSI circuits during the last decades and specifically their clock distribution networks. In the following table, some parameters of three DEC/Compaq microprocessors are shown:

Generation	**21064**	**21164**	**21264**
Technology (μm)	0.75	0.5	0.35
Transistors (million)	1.68	9.3	15.2
Frequency (MHz)	200	300	600
Die size (mm)	16.8 x 13.9	18.1 x 16.5	16.7 x 18.8
Capacitive clock load (nF)	3.25	3.75	----
Power supply (V)	3.3	3.3	2.2
Supply current (A)	< 10	~ 15	> 30
Total power (W)	~ 30	~ 50	~ 72
Power percentage dissipated within the clock distribution network	40 %	40%	44 %

Table 2.1: Characteristics of three Alpha Microprocessor generations [FRI-00].

In the Alpha 21064 microprocessor (year 1992), a single-phase clock signal is distributed globally on the higher level of the metal process (M3). The reason is that the resistivity per unit length of the third layer and the metal to substrate capacitance are smaller compared to the other layers.

The distribution of the loads is asymmetric, so it is necessary a specialized strategy for the clock network. The single 200-MHz clock signal is distributed through five levels of buffering, as shown in figure 2.17, where the total network consists of 145 separate elements. Each of the elements contains four levels of buffering with a final output stage locally driving the clocked registers. Vertical straps are placed on the second level of metal (M2), configuring a clock mesh structure, to minimize the skew that could appear within the initial four-stage portion of the buffer tree. This strategy is accomplished by centrally locating the clock generation circuitry within the integrated circuit [HOR-92].

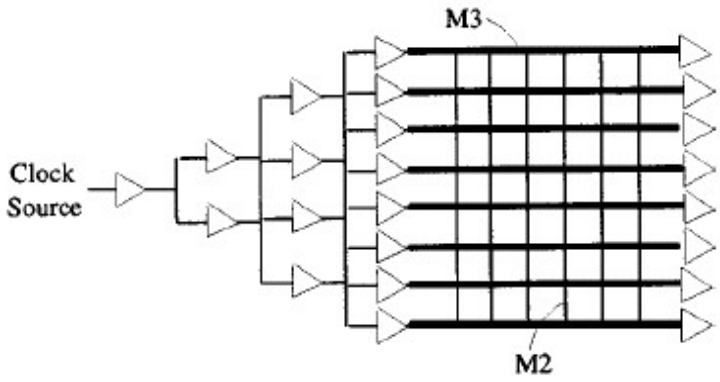


Figure 2.17: Clock distribution network of Alpha 21064 microprocessor [HOR-92].

The Alpha 21264 microprocessor (year 1998) is the first version of the Alpha family that utilizes a hierarchy of clocks. That hierarchy includes a gridded global clock, six gridded major clocks and many local and conditioned local clocks [BAI-98], [GRO-98].

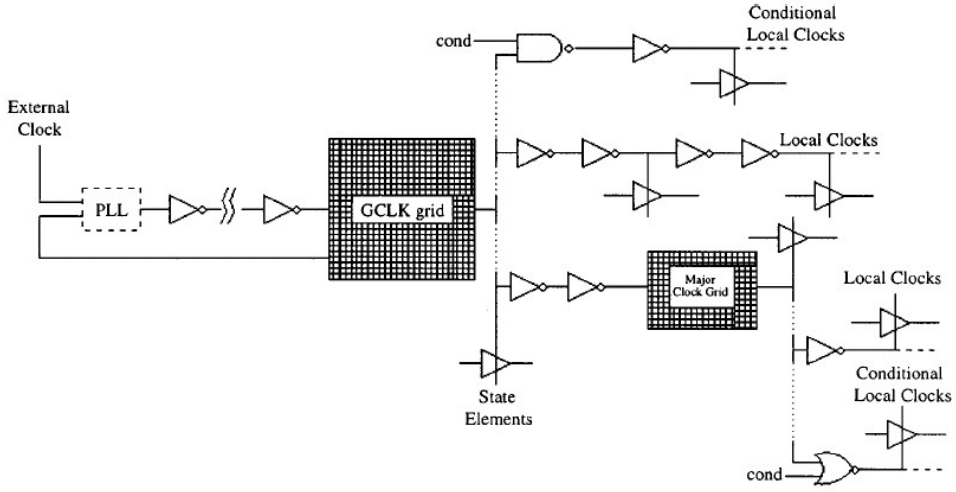


Figure 2.18: Hierarchy in Alpha 21264 microprocessor clock network [GRO-98].

An on-chip PLL is used to generate the clock signal. As shown in figure 2.19, the clock signal is routed from the center of the die and distributed by X and H-trees to 16 distributed clock drivers. The final portion of the clock distribution network uses a tree configuration as compared to an H-tree to save power and area. All interconnections within the GCLK grid are shielded (both laterally and vertically by power/ground lines).

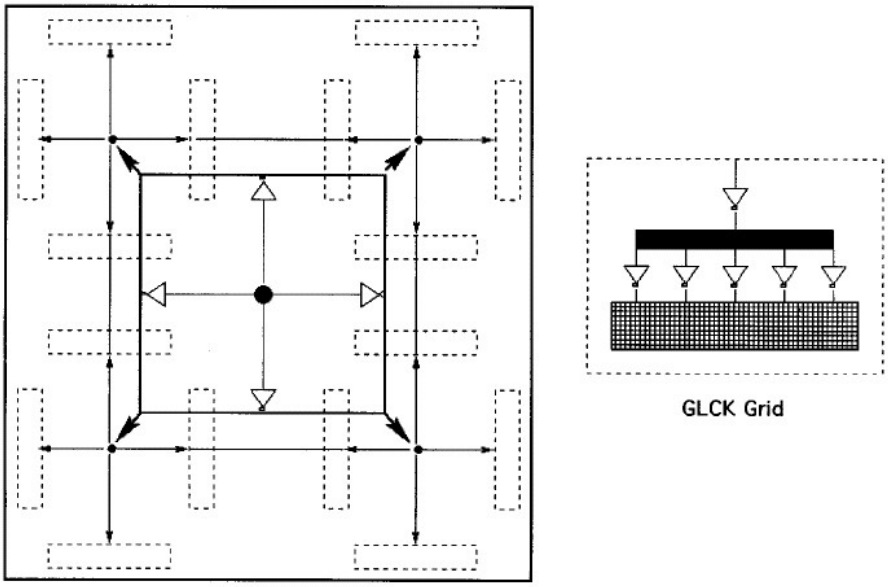


Figure 2.19: Global clock distribution network in the Alpha 21264 [BAI-98].

2.4.2. The Intel IA-64 Microprocessor

The Intel IA-64 is a recent example of a microprocessor designed to operate beyond the gigahertz frequency level [RUS-00]. The general clock distribution design strategy applied in this microprocessor is to minimize the clock skew through the use of distributed programmable deskew circuits while supporting local optimization of the clock distribution network.

The architecture of the IA-64 clock distribution network consists of three components: a balanced tree structure to globally distribute the clock signal, multiple deskew circuits distributed regionally with balanced clock trees, and multiple local clock buffers that drive the individual circuits and registers. In addition, a separate reference clock signal is distributed along with the global clock signal for use within the deskew circuit.

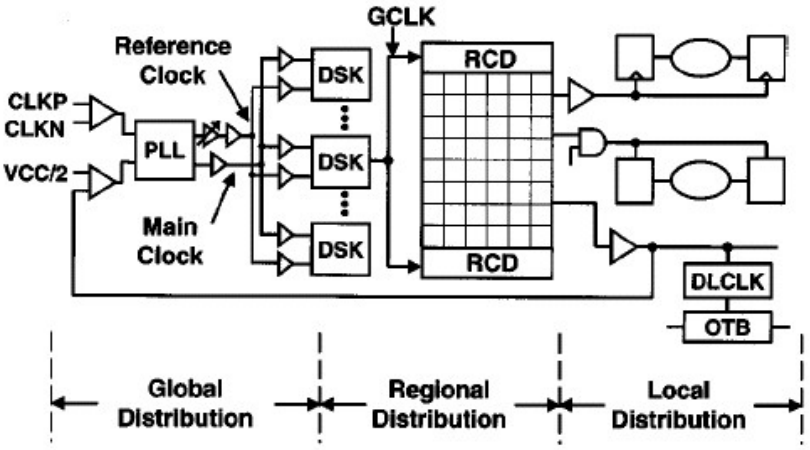


Figure 2.20: IA-64 clock distribution topology [RUS-00].

The global portion of the clock distribution network is originated from the clock source (on-chip PLL), and distributes the clock signal through a network structured as an H-tree until it reaches one of eight deskew clusters, each containing up to four deskew circuits. Clock lines are shielded by ground and power lines in order to minimize the effects of any capacitive and inductive coupling between the clock lines and any adjacent signal lines.

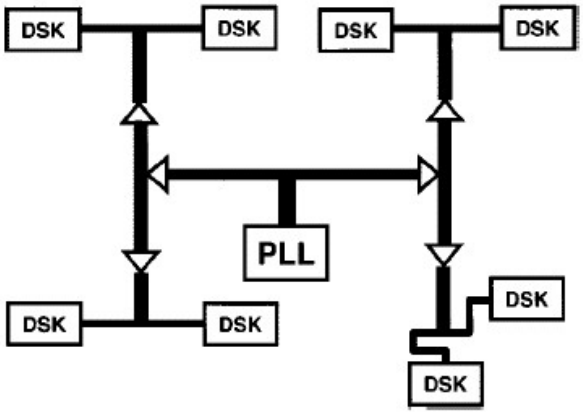


Figure 2.21: IA-64 global clock distribution [RUS-00].

The deskew buffer consists of a phase detector and a digitally controlled analog delay line. The buffer provides local delay compensation by comparing the reference clock signal to a local feedback signal, permitting the delay of the clock signal to be adjusted through the analog delay line and the phase detector.

The output signal of each deskew circuit is distributed through a balanced tree network to a set of local buffers. These buffers drive a regional grid of clock ports. Thus, by applying this intermediate compensation technique, the global clock distribution design problem can be treated as a local design problem, making the entire chip integration process significantly easier and more manageable. Localized clock skew scheduling and clock gating for power management are also supported at this level of the clock distribution network.

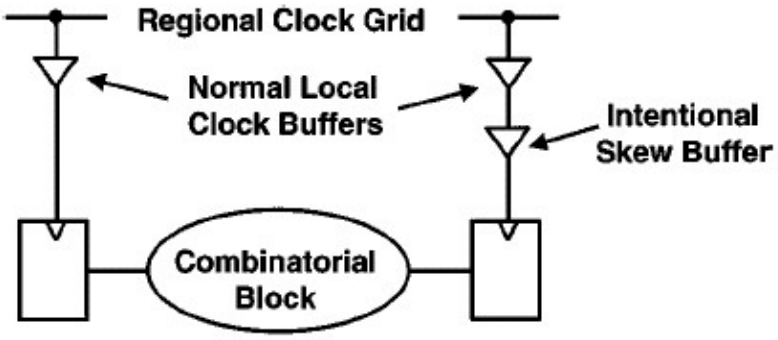


Figure 2.22: IA-64 local clock distribution [RUS-00].

2.4.3. The Intel Itanium 2 Microprocessor

The Itanium 2 microprocessor [TAM-04] is one of the newest developed by Intel in the last years. It is fabricated on the 130-nm CMOS process with six layers of copper interconnects. The processor has a total of 410 million transistors and operates at 1.5 GHz at 1.3 V. The clock distribution network design employs a tree-based differential global clock network. To address the clock skew issue and to increase the frequency of operation, a fuse-based de-skew circuit is implemented, reducing the clock skew caused by on-die process variations and clock network design mismatches.

The clock distribution network is implemented using a multilevel tree architecture. Figure 2.23 shows the clock distribution hierarchy.

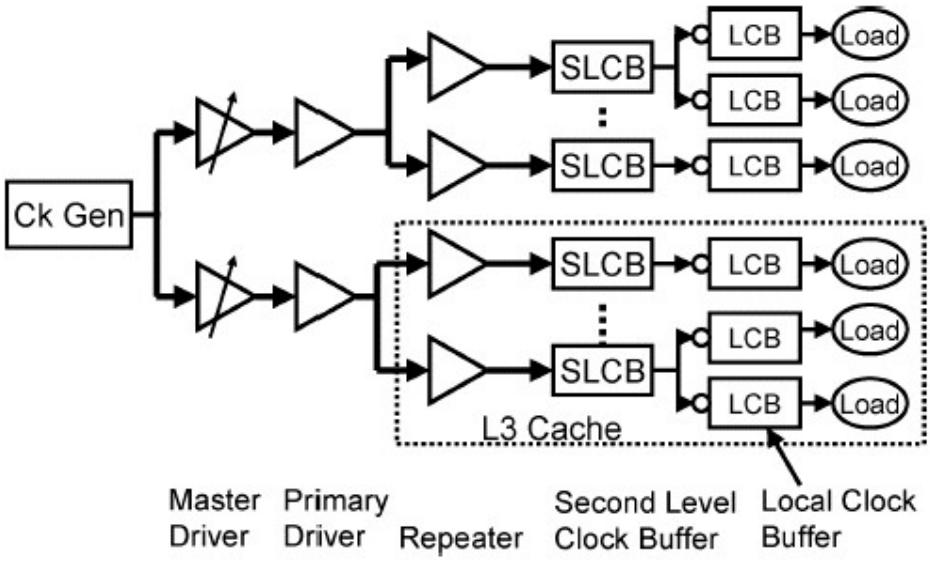


Figure 2.23: Clock distribution hierarchy [TAM-04].

The global clock distribution is a differential two-level tree structure that originates at the clock generator (Ck Gen) and terminates at the second-level clock buffer (SLCB). The master drivers, the primary drivers, and the repeaters are used along the distribution to increase the signal strength.

The zonal clock distribution consists of the SLCB and the local clock buffer (LCB). The SLCB converts the differential global clock to a single-ended zonal clock, SLCBO. There are 23 zonal clocks in the core, strategically partitioned to support the various functional units. Deskew circuits are implemented in this part of the clock distribution network to minimize clock skew effects.

To minimize the inductance effects in the clock networks, the clock lines are shielded in layers to provide low-impedance current return paths. To minimize skin effects, wires wider than about 4 µm are not used.

3. Clock skew

3.1. Definitions

For two sequentially adjacent registers, as shown in figure 2.1, C_i and C_f are the clock signals that drive the local data path. Both clock signals are generated in the same clock source. The propagation delay of the clock signals from the source to the registers R_i and R_f, is T_{Ci} and T_{Cj} respectively. They define the timing reference of when the data signals leave each register. There is a clock distribution network designed to generate a specific signal waveform. Ideally, clock events occur at all registers simultaneously. Given this strategy of global clocking, the clock signal arrival time to each register is defined with respect to a universal time reference.

The difference in clock signals arrival time between two register sequentially adjacent is the clock skew T_{skew}. We can define the clock skew mathematical expression as: $T_{skew} = T_{Ci} - T_{Cf}$. If the signals C_i and C_f are in complete synchronism, it means they arrive at the exact same moment, the clock skew is zero. It is important to note that the clock skew between is only relevant to sequentially adjacent registers that make up a local data path. Thus, the clock skew, at system or chip level, between two registers non-sequentially adjacent has no effects on the performance and reliability of a synchronous digital system from an analysis viewpoint.

Different clock signal paths can have different delays due to several reasons. We can summarize them in the following three reasons:

1) Differences in the wire lengths from the clock source to the clocked registers.
2) Differences in the delays of any active buffer in the clock distribution network.
3) Differences in the interconnection passive parameters.

We can note that, for a well-designed balanced clock distribution network, distributed buffers are the primary clock skew source.

Clock skew magnitude and polarity have two different effects on system performance and reliability. Depending on which signal, C_i or C_f, arrive earlier and the magnitude of T_{skew} with respect to data path time delay T_{PD}, system reliability and performance can be degraded or improved. Both cases are discussed below.

1) Maximum data path/clock skew constraint relationship

If the clock signal arrival time to the final register, T_{Cf}, is previous to the arrival time to the initial register, T_{Ci}, the clock skew is positive ($T_{Ci} > T_{Cf}$). Under this condition the maximum operation reachable frequency is decreased. A positive clock skew is the additional time amount that must be added to the minimum clock period to apply a new clock signal edge to the final register without any problem.

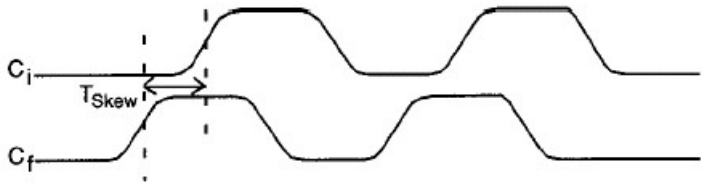


Figure 3.1: Positive clock skew.

For a specific design, the greatest propagation delay T_{PD} (max) of any local data path between two sequentially adjacent registers must be less than the minimum clock period T_{CP} (min).

$$T_{skew} \le T_{CP} - T_{PD}(\max) = T_{CP} - \left(T_{C-Q} + T_{\log ic}(\max) + T_{\text{int}} + T_{setup}\right), \text{ where } T_{Ci} > T_{Cf} \qquad (3.1)$$

This situation is the typical analysis of the critical data path in a synchronous system. If this constraint is not satisfied, the system will not operate correctly with this specific clock period. Therefore, T_{CP} must be increased if we want the circuit operates without any problem. In a circuit where the clock skew tolerance is small, data and clock signals should run in the same direction, thereby forcing that C_i leads C_f and making the clock skew negative.

2) Minimum data path/clock skew constraint relationship

If the clock signal arrival time to the final register, T_{Cf}, is later than arrival time to the initial register, T_{Ci}, the clock skew is negative ($T_{Ci} < T_{Cf}$). It can be used to improve the maximum performance of a synchronous system by the reduction of the critical data path. However, there is a minimum constraint to avoid race conditions.

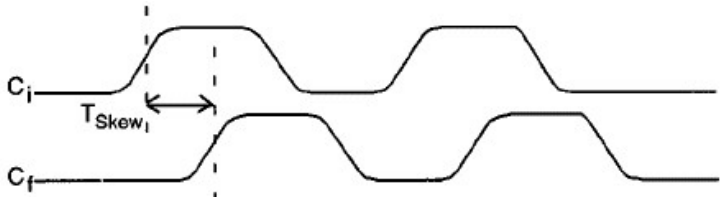


Figure 3.2: Negative clock skew.

When C_f follows to C_i, clock skew must be less than the required time for the data signal to leave the initial register, propagate through the combinatorial logic and interconnections and setup in the final register input. If this condition is not met, the data stored in the final register is overwritten with the data that was stored in the initial register because it arrives to the R_f input earlier than the clock signal (race condition). Furthermore, a circuit operating close to this restriction could not work correctly at unpredictable times due to environmental temperature or power supply voltage fluctuations:

$$\left|Tskew\right| \le T_{PD}(\min) = T_{C-Q} + T_{\log ic}(\min) + T_{\text{int}} + T_{setup}, \text{ where } T_{Ci} < T_{Cf} \qquad (3.2)$$

where T_{PD} (min) is the minimum data path delay between two sequentially adjacent registers.

3.2. Clock skew sources

Clock skew appears due to differences in clock paths from the source to each destination register. These differences can be unequal wire lengths or different resistive and/or capacitive wire parameters. In a balanced clock tree, the nominal value for clock

skew is zero, since clock paths are designed to be equal. However, clock skew appearance is still possible due to variations in the clock paths caused by process and circuit parameter tolerances. We can classify them in the following way:

- Transistor parameter variations

In the integrated circuit fabrication process, all transistor parameters are subject to deviations from their nominal values. Statistical models have been developed for transistor parameters such as threshold voltage (V_T), gate oxide thickness (t_{ox}), charge carrier mobility (μ), transistor width (W) and effective channel length (ΔL_{eff}).

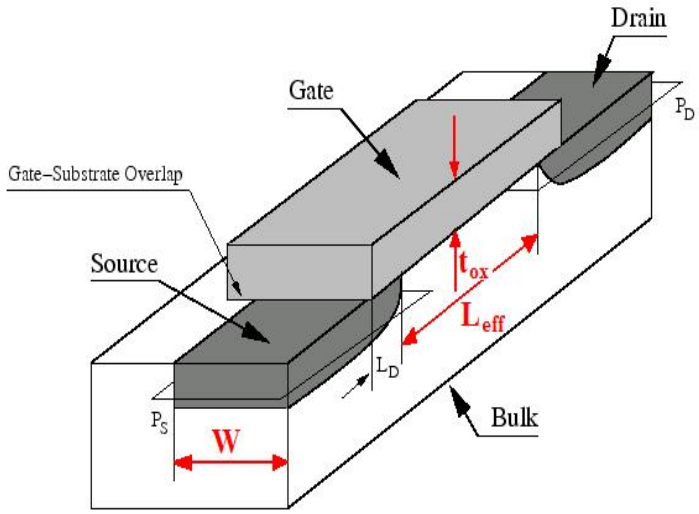


Figure 3.3: Vertical section of a MOS transistor.

In table 3.1, typical values for this parameter are presented according to different technologies. In table 3.2, standard deviations are shown.

Parameter	130 nm	100 nm	70 nm	45 nm
V_T	0.19 V	0.15 V	0.06 V	0.021 V
T_{ox}	3.3 nm	2.5 nm	1.6 nm	1.4 nm
L_{eff}	130 nm	100 nm	70 nm	45 nm
W (min)	130 nm	100 nm	70 nm	45 nm

Table 3.1: Typical values for different technologies [ITRS].

Parameter	Description	Standard Deviation
σ_{VT}	Threshold voltage	4.2 %
σ_{μ}	Charge carrier mobility	2 %
σ_{tox}	Gate oxide thickness	1.3 %
σ_{W}	Transistor width	5%
σ_{Leff}	Transistor effective channel length	5 %

Table 3.2: Typical values for standard deviations [KAH-01].

- Interconnect Parameter Variations

Interconnect width (W_{int}) and thickness (t_{int}) and interlevel dielectric thickness (T_{ILD}) variations are the main parameters of interest. As technology advances, the number of interconnect layers increases, and the interconnect lines become more non-uniform. This non-uniformity, which is caused by manufacturing processes, produces large variations of interconnect parameter values.

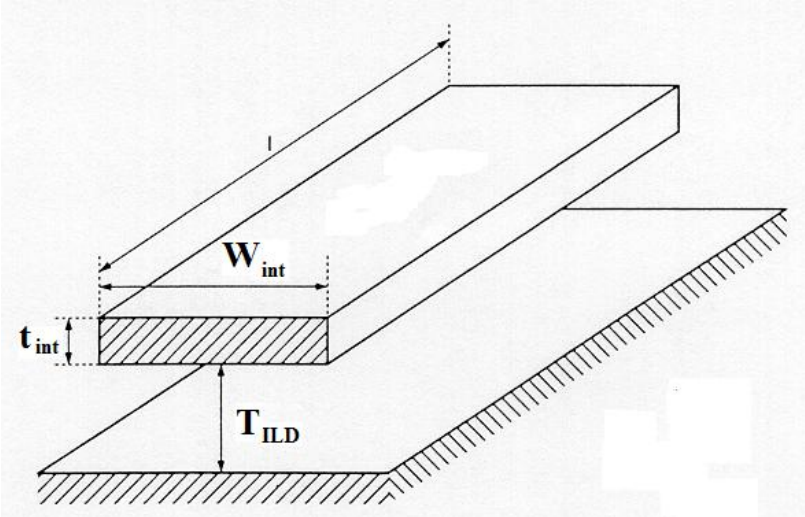


Figure 3.4: Interconnect segment main parameters.

In table 3.3, some typical values for this parameter are presented according to different technologies. In table 3.4, standard deviations are shown.

Parameter	130 nm	100 nm	70 nm	45 nm
W_{int} (min)	335 nm	237 nm	160 nm	103 nm
t_{int} (min)	670 nm	500 nm	352 nm	235 nm

Table 3.3: Typical values for different technologies [ITRS].

Parameter	Description	Standard Deviation
σ_{Wint}	Wire width	3 %
σ_{tint}	Wire thickness	3 %
σ_{TILD}	ILD thickness	3 %

Table 3.4: Typical values for standard deviations [FAN-98].

- System Parameter Variations

Besides process parameter variations, which are mainly the tolerances of device and interconnect physical parameters, system level fluctuations may create clock skew. Power supply voltage fluctuation (V_{DD}) and temperature variations (T) are considered as system level parameter variations.

In table 3.5, some typical values V_{DD} are presented according to different technologies. In table 3.6, standard deviations are shown.

Parameter	130 nm	100 nm	70 nm	45 nm
V_{DD}	1.2 V	1 V	0.9 V	0.6 V

Table 3.5: Typical values for different technologies [ITRS].

Parameter	Description	Standard Deviation
σ_{VDD}	Power supply voltage	3.3 % [KAH-01]
σ_{T}	Temperature	8 % [GRO-98]

Table 3.6: Typical values for standard deviations.

The thermal image of the Alpha 21064 microprocessor, presented in section 2.4.1, shows a 30º C temperature gradient over the entire chip that gives a temperature variation of about 8 % [GRO-98]. In figure 3.6, this image is depicted.

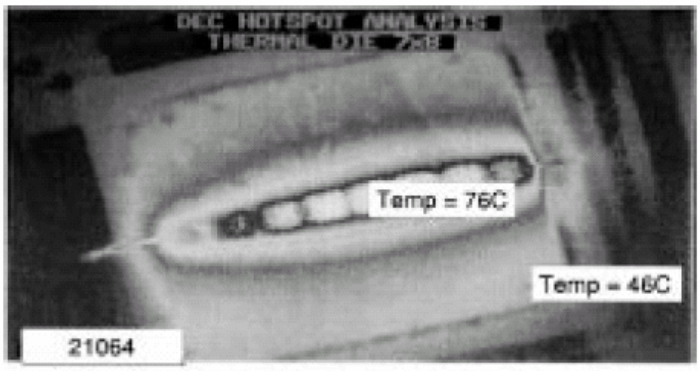


Figure 3.6: 20164 thermal image [GRO-98].

3.3. Clock skew models

An important research area in VLSI circuits is timing analysis, where simplified models are used to estimate the delay through a CMOS circuit according to process and circuit parameter variations. At first, a probabilistic model for the accumulation of clock skew in synchronous systems is described. Using this model, upper bounds for expected skew and its variance in tree distribution systems are derived. Thereafter, a model for tapered H-Tree is described, where no buffers are placed at branching points and the wires are widened to avoid reflections. The clock skew is calculated as function of device, system and interconnect parameter variations. The first statistical model (upper bounds model) is too conservative for estimating the clock skew of a well-balanced H-tree clock distribution network because correlation between overlapped parts of paths are not considered. Finally, a new approach to estimate the mean value and variance of clock skew is described taking into account this correlation.

3.3.1. MODEL 1: Statistical model to estimate upper bounds of clock skew

This model is described in depth in Appendix 1.

Kugelmass and Steiglitz [KUG-88] present a probabilistic model for the accumulation of clock skew in synchronous systems. Using this model, it's possible to

estimate upper bounds for expected clock skew between processing elements (and its variance) in symmetric tree distribution systems with *N* synchronously clocked processing elements.

The first assumption in this model is the topology of the clock distribution network. It must be a symmetric tree-like structure with a single source and *N* end points called processing elements (PE). There must be only one path from the source to the PEs.

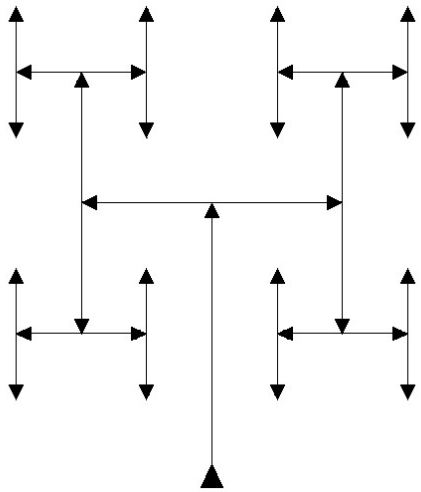


Figure 3.7: Model 1 tree structure.

Each clock path is composed of delay elements: buffers and interconnection wires. It is possible to associate a random variable to each element that gives the delay contribution of it. The total delay from the clock source to each PE is the sum of all the random variables along the path. By the Central Limit Theorem, the sum converges to a normal distribution.

According to these authors, it is possible to define a random variable that characterizes the clock skew of the clock distribution network. It is $R = A_{max} - A_{min}$, where A_{max} and A_{min} are the maximal and the minimal arrival time to any of the *N* PEs.

Random variables that compose *R* are dependent in a clock tree because they are sums of overlapping variables. However, thanks to a demonstrated theorem, the

expected mean value of *R* is smaller than the same case but with independent random variables.

Another theorem says that if *R* is composed of *N* independent identically distributed random variables (it is the case for a symmetric clock distribution network), then, the asymptotically expected value of *R* is:

$$E[R] = \sigma\left[\frac{4\ln N - \ln\ln N - \ln 4\pi + 2C}{(2\ln N)^{1/2}} + O\left(\frac{1}{\log N}\right)\right] \qquad (3.3)$$

where C ≈ 0.5772 is Euler constant and σ is the standard deviation of the path delay. The variance of *R* is given by:

$$Var[R] = \frac{\sigma^2}{\ln N}\frac{\pi^2}{6} + O\left[\frac{1}{\log^2 N}\right] \qquad (3.4)$$

Equation (3.3) is therefore asymptotic upper bound on the expected skew in a clock distribution tree with *N* leaves.

To apply these model equations to the proposed H-tree depicted in figure 3.1, it is necessary to know the value of clock path standard deviation, σ. It has two different components (and independent according to the model assumption):

$$\sigma = \sqrt{\sigma_b^2 \log_2^2 N + \sigma_w^2\left(2\left(\sqrt{N} - 1\right)\right)^2} \qquad (3.5)$$

where σ_b is the standard deviation of buffer delays and σ_w is standard deviation of the wire in the lowest level.

The next step to calculate *E[R]* it is necessary to determine the delay variance σ_b^2 through a buffer of the clock distribution tree and delay variance σ_w^2 through a wire of the clock distribution tree.

Using Sakurai's model for interconnection delay, described in section 2.3.1, and the possible clock skew sources considered by the authors (V_T, t_{ox}, L_{eff}, V_{DD}, T_{ILD}, W_{int}, t_{int}), the value of σ_b^2 and σ_w^2 is determined, in terms of variances of the independent random variables that compose them, by the following expressions:

$$\sigma_b^2 = \left(\frac{\partial T_{Delay}}{\partial V_T}\right)^2 \sigma_{V_T}^2 + \left(\frac{\partial T_{Delay}}{\partial t_{ox}}\right)^2 \sigma_{t_{ox}}^2 + \left(\frac{\partial T_{Delay}}{\partial L_{eff}}\right)^2 \sigma_{L_{eff}}^2 + \left(\frac{\partial T_{Delay}}{\partial V_{DD}}\right)^2 \sigma_{V_{DD}}^2 \qquad (3.6)$$

$$\sigma_w^2 = \left(\frac{\partial T_{Delay}}{\partial T_{ILD}}\right)^2 \sigma_{T_{ILD}}^2 + \left(\frac{\partial T_{Delay}}{\partial W_{\text{int}}}\right)^2 \sigma_{W_{\text{int}}}^2 + \left(\frac{\partial T_{Delay}}{\partial t_{\text{int}}}\right)^2 \sigma_{t_{\text{int}}}^2 \qquad (3.7)$$

where:

$$\begin{aligned}
\frac{\partial T_{Delay}}{\partial V_T} &= \frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial V_T} = 2.30\left(C_0 + C_{\text{int}}\right)\frac{R_0}{V_{DD} - V_T} \\
\frac{\partial T_{Delay}}{\partial t_{ox}} &= \frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial t_{ox}} + \frac{\partial T_{Delay}}{\partial C_0}\frac{\partial C_0}{\partial t_{ox}} = 2.30\left(C_0 + C_{\text{int}}\right)\frac{R_0}{t_{ox}} + 2.30\left(R_0 + R_{\text{int}}\right)\frac{C_0}{t_{ox}} \\
\frac{\partial T_{Delay}}{\partial L_{eff}} &= \frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial L_{eff}} + \frac{\partial T_{Delay}}{\partial C_0}\frac{\partial C_0}{\partial L_{eff}} = 2.30\left(C_0 + C_{\text{int}}\right)\frac{R_0}{L_{eff}} + 2.30\left(R_0 + R_{\text{int}}\right)\frac{C_0}{L_{eff}} \\
\frac{\partial T_{Delay}}{\partial V_{DD}} &= \frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial V_{DD}} = 2.30\left(C_0 + C_{\text{int}}\right)\frac{R_0}{V_{DD} - V_T} \\
\frac{\partial T_{Delay}}{\partial T_{ILD}} &= \frac{\partial T_{Delay}}{\partial C_{\text{int}}}\frac{\partial C_{\text{int}}}{\partial T_{ILD}} = \left(1.02R_{\text{int}} + 2.30R_0\right)\frac{C_{\text{int}}}{T_{ILD}} \\
\frac{\partial T_{Delay}}{\partial W_{\text{int}}} &= \frac{\partial T_{Delay}}{\partial R_{\text{int}}}\frac{\partial R_{\text{int}}}{\partial W_{\text{int}}} + \frac{\partial T_{Delay}}{\partial C_{\text{int}}}\frac{\partial C_{\text{int}}}{\partial W_{\text{int}}} = \left(1.02C_{\text{int}} + 2.30C_0\right)\frac{R_{\text{int}}}{W_{\text{int}}} + \left(1.02R_{\text{int}} + 2.30R_0\right)\frac{C_{\text{int}}}{W_{\text{int}}} \\
\frac{\partial T_{Delay}}{\partial t_{\text{int}}} &= \frac{\partial T_{Delay}}{\partial R_{\text{int}}}\frac{\partial R_{\text{int}}}{\partial t_{\text{int}}} = \left(1.02C_{\text{int}} + 2.30C_0\right)\frac{R_{\text{int}}}{t_{\text{int}}}
\end{aligned} \qquad (3.8)$$

3.3.2. MODEL 2: Statistical model for clock skew in tapered H-trees

This model is described in depth in Appendix 2.

Zarkesh-Ha, Mule' and Meindl [ZAR-98] described a compact model to enable first-order estimation for clock skew in tapered H-trees. In this kind of structure, there

are not intermediate buffers at the split points and the wires must be widened to avoid reflections in those points.

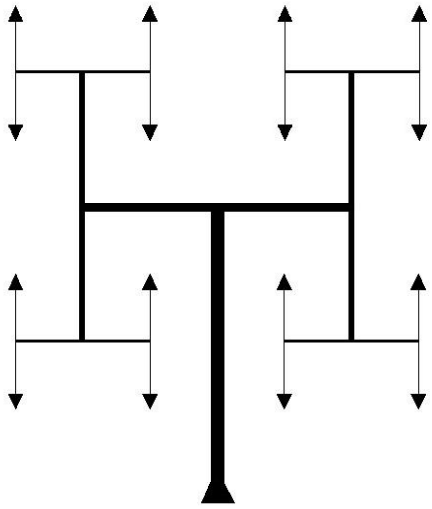


Figure 3.8: Model 2 tree structure [ZAR-98].

Authors propose that any H-tree circuit can be simplified in the following equivalent circuit shown in figure 3.9.

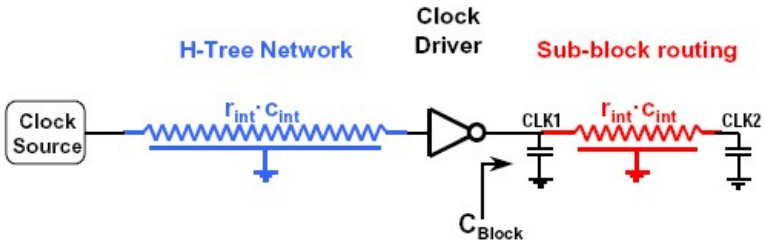


Figure 3.9: Equivalent circuit of clock H-tree network [ZAR-98].

Using the equivalent circuit, the delay of the entire clock network of figure 3.9 is divided into two parts:

- Interconnect delay from the clock source to clock the driver: If the H-tree network is driven by a single driver, then the delay expression for a distributed RC line using Sakurai's model (*50%* of time delay) is:

$$T_{H-tree} = 0.4 \cdot \left(\frac{\rho \cdot \varepsilon_r}{t_{\text{int}} \cdot T_{ILD}} \right) \cdot D^2 \cdot \left(1 - \frac{1}{2^{n/2}} \right)^2 + \frac{\sqrt{\varepsilon_r}}{c_0} \cdot D \cdot \left(1 - \frac{1}{2^{n/2}} \right) \quad (3.9)$$

where ε_r is the relative dielectric constant of the ILD material, ρ the line resistivity, c_o the speed of light in free space, D the die size, and n the number of H-tree levels.

- Transistor delay of the sub-block clock driver: the delay expression is according to Sakurai's Model (50% of time delay):

$$T_{driver} = 0.7 \cdot \left(\frac{L_{eff} / W}{\mu \cdot C_{ox} \cdot (V_{DD} - V_T)} \right) \cdot C_L \qquad (3.10)$$

where C_L is the capacitive load of the sub-block clock driver.

The overall delay of the entire clock distribution network is the sum of the previous components: $T_{Delay} = T_{H\text{-}tree} + T_{Driver}$. This model gives first order estimation of the clock skew:

$$T_{CSK}(x) = \Delta T_{Delay} \approx \left| \frac{\partial T_{Delay}}{\partial x} \right| \Delta x \qquad (3.11)$$

where x is any variation of clock skew components such as ΔV_T, Δt_{ox}, ΔL_{eff} , ΔH_{int}, ΔT_{ILD}, ΔV_{DD}, ΔT and ΔC_L. Table 3.1 shows the closed form equations for each individual clock skew component by using (3.11):

Physical parameter and derivation used	**Clock skew component**
Threshold voltage fluctuation	$T_{CSK}(V_T) = 0.7 \cdot R_0 \cdot C_L \cdot \left(\frac{V_T}{V_{DD} - V_T} \right) \cdot \frac{\Delta V_T}{V_T}$
Gate oxide thickness tolerance	$T_{CSK}(t_{ox}) = 0.7 \cdot R_0 \cdot C_L \cdot \frac{\Delta t_{ox}}{t_{ox}}$
Transistor channel length tolerance	$T_{CSK}(L_{eff}) = 0.7 \cdot R_0 \cdot C_L \cdot \frac{\Delta L_{eff}}{L_{eff}}$
Wire thickness variation	$T_{CSK}(t_{int}) = 0.4 \cdot (r_{int} \cdot c_{int}) \cdot D^2 \cdot \left(1 - \frac{1}{2^{n/2}} \right)^2 \cdot \frac{\Delta t_{int}}{t_{int}}$

ILD thickness variation	$T_{CSK}(T_{ILD}) = 0.4 \cdot (r_{\text{int}} \cdot c_{\text{int}}) \cdot D^2 \cdot \left(1 - \frac{1}{2^{n/2}}\right)^2 \cdot \frac{\Delta T_{ILD}}{T_{ILD}}$
IR drop	$T_{CSK}(V_{DD}) = 0.7 \cdot R_0 \cdot C_L \cdot \left(\frac{V_{DD}}{V_{DD} - V_T}\right) \cdot \frac{\Delta V_{DD}}{V_{DD}}$
Non uniform register distribution	$T_{CSK}(C_L) = 0.7 \cdot R_0 \cdot C_L \cdot \frac{\Delta C_L}{C_L}$
Temperature gradient	$T_{CSK}(\mathrm{T}) = 0.7 \cdot R_0 \cdot C_L \cdot \left(\frac{E_g / q + V_T}{V_{DD} - V_T}\right) \cdot \frac{\Delta \mathrm{T}}{\mathrm{T}}$

Table 3.1: Clock skew components.

It is important to note that the model equations can be easily modified to be more similar to other models, where the Sakurai's expressions are used with 90% of time delay. It only supposes to change the coefficients de $T_{H\text{-}tree}$ and T_{Driver}.

- T_{H-tree}: $0.4 \rightarrow 1.02 \Rightarrow T_{H-tree} = 1.02 \cdot (r_{\text{int}} c_{\text{int}}) \cdot l^2 + \frac{\sqrt{\varepsilon_r}}{c_0} \cdot l$
- T_{driver}: $0.7 \rightarrow 2.3 \Rightarrow T_{driver} = 2.3 \cdot R_0 \cdot C_L$

3.3.3. MODEL 3: Statistical model for clock skew considering path correlations

This model is described in depth in Appendix 3.

Jiang and Horiguchi [JIA-01] propose a new approach to estimate the mean value and variance of clock skew for general clock distribution networks. The novelty is that clock paths can be not identical and path delay correlation caused by the overlapped parts of path lengths is considered. In this way, clock skew mean and variance is accurately estimated for general clock distribution networks.

Clock paths of a clock distribution network usually have some common branches over their length. These common branches cause correlation among the delays of these paths. Only the overlapped parts of two paths determine the correlation between them.

If ξ is the maximum value of the propagation delay and η the minimum value, then the mean value and the variance of the clock skew, χ, are given by:

$$E(\chi) = E(\xi) - E(\eta) \tag{3.12}$$

$$D(\chi) = D(\xi) + D(\eta) - 2\rho\sqrt{D(\xi) \cdot D(\eta)} \tag{3.13}$$

Here, $E(\cdot)$ and $D(\cdot)$ represent the mean value and the variance of a random variable, respectively, and ρ is the correlation coefficient of ξ and η. Author propose a recursive approach for evaluating the parameters $E(\xi)$, $E(\eta)$, $D(\xi)$, $D(\eta)$ and ρ. Based on this algorithm, an expression can be derived for the clock skew of a well-balanced *H*-tree clock distribution networks.

- Clock skew estimation for H-tree clock distribution networks

Before developing the models, the H-tree itself must first be defined. The H-tree presents intermediate buffers at each branching point and, without loss of generality, it has *n* hierarchical levels, where *n* denotes the tree depth. The level *0* branch corresponds to the root branch, and level *n* branches to the branches that support sinks. A level *i* branch begins in a level split *i* point and ends in a level *i+1* split point. The H-tree illustrated in figure 3.10 is drawn for *n=4* and it is used to distribute the clock signals to *16* processors.

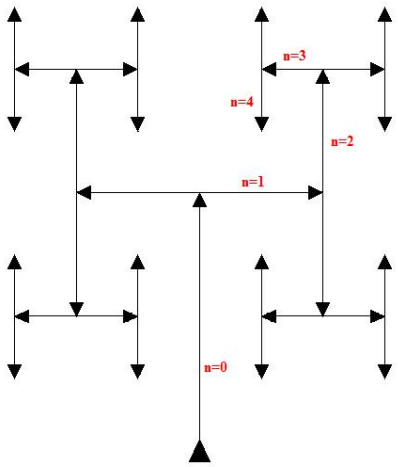


Figure 3.10: A well-balanced H-tree clock distribution network for 16 processors.

For a n level well-balanced H-tree, let d_i, $i=0, \ldots, n$ be the actual delay of branch i of a clock path. The clock skew $E(\chi)$ and skew variance $D(\chi)$ of the n level well-balanced H-tree are given by:

$$E(\chi) = \frac{2}{\sqrt{\pi}} \sum_{i=1}^{n} \sqrt{\sum_{k=1}^{i} \left(\frac{\pi - 1}{\pi} \right)^{k-1} \cdot D\left(d_{n-i+k}\right)} \tag{3.14}$$

$$D(\chi) = 2 \cdot (1 - \rho) \cdot \sum_{i=0}^{n} \left(\frac{\pi - 1}{\pi} \right)^{i} \cdot D\left(d_i\right) \tag{3.15}$$

The closed-form expressions (3.14)–(3.15) indicate clearly how the clock skew is accumulated along the clock paths and with the increase of H-tree size.

- Clock skew calculation in function of its components

The delay of a branch may then be obtained by averaging the rise and fall times using Sakurai's model for interconnection delays (90 % of time delay), described before in section 2.3.1.

One approach to calculating the delay variance of a branch due to the variations of process parameters is express the parameter relations in terms of independent variables. Authors consider the following variables to calculate the variance of the delay of any branch $D(d_{n-i+k})$: V_T, μ, t_{ox}, L_{eff}, W, T_{ILD}, W_{int}, t_{int}. The variance of the delay in a branch is the following:

$$\begin{aligned} \sigma^2_{T_{Delay}} = & \left(\frac{\partial T_{Delay}}{\partial V_T} \right)^2 \sigma^2_{V_T} + \left(\frac{\partial T_{Delay}}{\partial \mu} \right)^2 \sigma^2_{\mu} + \left(\frac{\partial T_{Delay}}{\partial t_{ox}} \right)^2 \sigma^2_{t_{ox}} + \left(\frac{\partial T_{Delay}}{\partial L_{eff}} \right)^2 \sigma^2_{L_{eff}} + \left(\frac{\partial T_{Delay}}{\partial W} \right)^2 \sigma^2_{W} \\ & + \left(\frac{\partial T_{Delay}}{\partial T_{ILD}} \right)^2 \sigma^2_{T_{ILD}} + \left(\frac{\partial T_{Delay}}{\partial W_{\text{int}}} \right)^2 \sigma^2_{W_{\text{int}}} + \left(\frac{\partial T_{Delay}}{\partial t_{\text{int}}} \right)^2 \sigma^2_{t_{\text{int}}} \end{aligned} \tag{3.16}$$

where:

$$\frac{\partial T_{Delay}}{\partial V_T} = \frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial V_T} = 2.30\left(C_0 + C_{\text{int}}\right)\frac{R_0}{V_{DD} - V_T}$$

$$\frac{\partial T_{Delay}}{\partial V_T} = \frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial \mu} = 2.30\left(C_0 + C_{\text{int}}\right)\frac{R_0}{\mu}$$

$$\frac{\partial T_{Delay}}{\partial t_{ox}} = \frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial t_{ox}} + \frac{\partial T_{Delay}}{\partial C_0}\frac{\partial C_0}{\partial t_{ox}} = 2.30\left(C_0 + C_{\text{int}}\right)\frac{R_0}{t_{ox}} + 2.30\left(R_0 + R_{\text{int}}\right)\frac{C_0}{t_{ox}} \qquad (3.18)$$

$$\frac{\partial T_{Delay}}{\partial L_{eff}} = \frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial L_{eff}} + \frac{\partial T_{Delay}}{\partial C_0}\frac{\partial C_0}{\partial L_{eff}} = 2.30\left(C_0 + C_{\text{int}}\right)\frac{R_0}{L_{eff}} + 2.30\left(R_0 + R_{\text{int}}\right)\frac{C_0}{L_{eff}}$$

$$\frac{\partial T_{Delay}}{\partial W} = \frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial W} + \frac{\partial T_{Delay}}{\partial C_0}\frac{\partial C_0}{\partial W} = 2.30\left(C_0 + C_{\text{int}}\right)\frac{R_0}{W} + 2.30\left(R_0 + R_{\text{int}}\right)\frac{C_0}{W}$$

$$\frac{\partial T_{Delay}}{\partial T_{ILD}} = \frac{\partial T_{Delay}}{\partial C_{\text{int}}}\frac{\partial C_{\text{int}}}{\partial T_{ILD}} = \left(1.02R_{\text{int}} + 2.30R_0\right)\frac{C_{\text{int}}}{T_{ILD}}$$

$$\frac{\partial T_{Delay}}{\partial W_{\text{int}}} = \frac{\partial T_{Delay}}{\partial R_{\text{int}}}\frac{\partial R_{\text{int}}}{\partial W_{\text{int}}} + \frac{\partial T_{Delay}}{\partial C_{\text{int}}}\frac{\partial C_{\text{int}}}{\partial W_{\text{int}}} = \left(1.02C_{\text{int}} + 2.30C_0\right)\frac{R_{\text{int}}}{W_{\text{int}}} + \left(1.02R_{\text{int}} + 2.30R_0\right)\frac{C_{\text{int}}}{W_{\text{int}}}$$

$$\frac{\partial T_{Delay}}{\partial t_{\text{int}}} = \frac{\partial T_{Delay}}{\partial R_{\text{int}}}\frac{\partial R_{\text{int}}}{\partial t_{\text{int}}} = \left(1.02C_{\text{int}} + 2.30C_0\right)\frac{R_{\text{int}}}{t_{\text{int}}}$$

3.3.4. Summary of the models

- Parameters of the models

Model	Parameters	
1	• Interconnection resistance: R_{int} • Interconnection capacitance: C_{int} • Driving transistor on-resistance: R_0 • Driving inverter input capacitance: C_0 • Number of processing elements: N • Lowest level branch length: L_{int} • Power supply voltage: V_{DD} • Threshold voltage: V_T	Parameter deviations in % • Threshold voltage: σ_{VT} • Power supply voltage: σ_{VDD} • Gate oxide thickness: σ_{tox} • Effective channel length: σ_{Leff} • ILD thickness: σ_{TILD} • Wire width: σ_{wint} • Wire thickness: σ_{tint}
2	Process parameters: • Interconnection parameters: $r_{int}c_{int}$ • Threshold voltage of inverters: V_T • Power supply voltage: V_{DD} • Transistors energy gap: E_g Design parameters: • Buffer output resistance: R_0 • Die size: D • H-tree levels: n • Capacitive load of sub-blocks: C_L	Parameter deviations (in %): • Threshold voltage: σ_{VT} • Gate oxide thickness: σ_{tox} • Effective channel length: σ_{Leff} • Wire thickness: σ_{tint} • ILD thickness: σ_{TILD} • Power supply voltage: σ_{VDD} • Load capacitance: σ_{CL} • Temperature: $\sigma_{\mathcal{T}}$
3	• Interconnection resistance: R_{int} • Interconnection capacitance: C_{int} • Driving transistor on-resistance: R_0 • Driving inverter input capacitance: C_0 • H-tree levels: n • Lowest level branch length: L_{int} • Power supply voltage: V_{DD} • Threshold voltage: V_T	Parameter deviations in % • Threshold voltage: σ_{VT} • Charge carrier mobility: σ_{μ} • Gate oxide thickness: σ_{tox} • Transistor width: σ_W • Effective channel length: σ_{Leff} • Wire width: σ_{wint} • Wire thickness: σ_{tint} • ILD thickness: σ_{TILD}

- Equations of the models

Model	Equations
1	Clock skew expression (mean):

$$E[Skew] = \left(\sigma_b \log_2 N + \sigma_w 2\left(\sqrt{N}-1\right)\right)\left[\frac{4\ln N - \ln\ln N - \ln 4\pi + 2C}{(2\ln N)^{1/2}} + O\left(\frac{1}{\ln N}\right)\right]$$

Parameter calculation (using 90% time delay in Sakurai's model):

$$T_{Delay} = 1.02 R_{int} C_{int} + 2.30\left(R_0 C_0 + R_0 C_{int} + R_{int} C_0\right)$$

$$\sigma_b^2 = \left(\frac{\partial T_{Delay}}{\partial V_T}\right)^2 \sigma_{V_T}^2 + \left(\frac{\partial T_{Delay}}{\partial t_{ox}}\right)^2 \sigma_{t_{ox}}^2 + \left(\frac{\partial T_{Delay}}{\partial L_{eff}}\right)^2 \sigma_{L_{eff}}^2 + \left(\frac{\partial T_{Delay}}{\partial V_{DD}}\right)^2 \sigma_{V_{DD}}^2$$

$$\sigma_w^2 = \left(\frac{\partial T_{Delay}}{\partial T_{ILD}}\right)^2 \sigma_{T_{ILD}}^2 + \left(\frac{\partial T_{Delay}}{\partial W_{int}}\right)^2 \sigma_{W_{int}}^2 + \left(\frac{\partial T_{Delay}}{\partial t_{int}}\right)^2 \sigma_{t_{int}}^2$$

$$\frac{\partial T_{Delay}}{\partial V_T} = \frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial V_T} = 2.30\left(C_0 + C_{int}\right)\frac{R_0}{V_{DD} - V_T}$$

$$\frac{\partial T_{Delay}}{\partial t_{ox}} = \frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial t_{ox}} + \frac{\partial T_{Delay}}{\partial C_0}\frac{\partial C_0}{\partial t_{ox}} = 2.30\left(C_0 + C_{int}\right)\frac{R_0}{t_{ox}} + 2.30\left(R_0 + R_{int}\right)\frac{C_0}{t_{ox}}$$

$$\frac{\partial T_{Delay}}{\partial L_{eff}} = \frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial L_{eff}} + \frac{\partial T_{Delay}}{\partial C_0}\frac{\partial C_0}{\partial L_{eff}} = 2.30\left(C_0 + C_{int}\right)\frac{R_0}{L_{eff}} + 2.30\left(R_0 + R_{int}\right)\frac{C_0}{L_{eff}}$$

$$\frac{\partial T_{Delay}}{\partial V_{DD}} = \frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial V_{DD}} = 2.30\left(C_0 + C_{int}\right)\frac{R_0}{V_{DD} - V_T}$$

$$\frac{\partial T_{Delay}}{\partial T_{ILD}} = \frac{\partial T_{Delay}}{\partial C_{int}}\frac{\partial C_{int}}{\partial T_{ILD}} = \left(1.02 R_{int} + 2.30 R_0\right)\frac{C_{int}}{T_{ILD}}$$

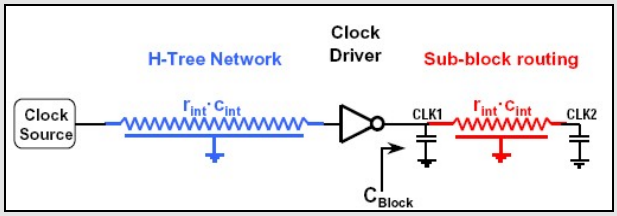
$$\frac{\partial T_{Delay}}{\partial W_{int}} = \frac{\partial T_{Delay}}{\partial R_{int}}\frac{\partial R_{int}}{\partial W_{int}} + \frac{\partial T_{Delay}}{\partial C_{int}}\frac{\partial C_{int}}{\partial W_{int}} = \left(1.02 C_{int} + 2.30 C_0\right)\frac{R_{int}}{W_{int}} + \left(1.02 R_{int} + 2.30 R_0\right)\frac{C_{int}}{W_{int}}$$

$$\frac{\partial T_{Delay}}{\partial t_{int}} = \frac{\partial T_{Delay}}{\partial R_{int}}\frac{\partial R_{int}}{\partial t_{int}} = \left(1.02 C_{int} + 2.30 C_0\right)\frac{R_{int}}{t_{int}}$$

Model	Equations
2	

$$T_{CSK} = \Delta T_{Delay} = \sum T_{CSK}(x) \approx \sum \left| \frac{\partial T_{Delay}}{\partial x} \right| \Delta x$$

$$T_{Delay} = T_{H\text{-}tree} + T_{Driver}$$



Parameter calculation (using 50% time delay in Sakurai's model):

$$T_{H-tree} = 0.4 \cdot \left(\frac{\rho \cdot \varepsilon_r}{t_{int} \cdot T_{ILD}} \right) \cdot D^2 \cdot \left(1 - \frac{1}{2^{n/2}} \right)^2 + \frac{\sqrt{\varepsilon_r}}{c_0} \cdot D \cdot \left(1 - \frac{1}{2^{n/2}} \right)$$

$$T_{driver} = 0.7 \cdot \left(\frac{L_{eff}/W}{\mu \cdot C_{ox} \cdot (V_{DD} - V_T)} \right) \cdot C_L$$

Physical parameter and derivation used	**Clock skew component**
Threshold voltage fluctuation	$T_{CSK}(V_T) = 0.7 \cdot R_0 \cdot C_L \cdot \left(\frac{V_T}{V_{DD} - V_T} \right) \cdot \frac{\Delta V_T}{V_T}$
Gate oxide thickness tolerance	$T_{CSK}(t_{ox}) = 0.7 \cdot R_0 \cdot C_L \cdot \frac{\Delta t_{ox}}{t_{ox}}$
Transistor channel length tolerance	$T_{CSK}(L_{eff}) = 0.7 \cdot R_0 \cdot C_L \cdot \frac{\Delta L_{eff}}{L_{eff}}$
Wire thickness variation	$T_{CSK}(t_{int}) = 0.4 \cdot (r_{int} \cdot c_{int}) \cdot D^2 \cdot \left(1 - \frac{1}{2^{n/2}} \right)^2 \cdot \frac{\Delta t_{int}}{t_{int}}$
ILD thickness variation	$T_{CSK}(T_{ILD}) = 0.4 \cdot (r_{int} \cdot c_{int}) \cdot D^2 \cdot \left(1 - \frac{1}{2^{n/2}} \right)^2 \cdot \frac{\Delta T_{ILD}}{T_{ILD}}$
IR drop	$T_{CSK}(V_{DD}) = 0.7 \cdot R_0 \cdot C_L \cdot \left(\frac{V_{DD}}{V_{DD} - V_T} \right) \cdot \frac{\Delta V_{DD}}{V_{DD}}$
Non uniform register distribution	$T_{CSK}(C_L) = 0.7 \cdot R_0 \cdot C_L \cdot \frac{\Delta C_L}{C_L}$
Temperature gradient	$T_{CSK}(\mathrm{T}) = 0.7 \cdot R_0 \cdot C_L \cdot \left(\frac{E_g/q + V_T}{V_{DD} - V_T} \right) \cdot \frac{\Delta \mathrm{T}}{\mathrm{T}}$

Model	Equations
3	Clock skew expression (mean):

$$E(\chi)=\frac{2}{\sqrt{\pi}}\sum_{i=1}^{n}\sqrt{\sum_{k=1}^{i}\left(\frac{\pi-1}{\pi}\right)^{k-1}\cdot D\left(d_{n-i+k}\right)}$$

For a *n* level well-balanced *H*-tree (with buffers at each split point), $D(d_i)$, $i=0, \ldots, n$, is the delay variance of the branch *i*.

$$\sigma^2_{T_{Delay}}=\left(\frac{\partial T_{Delay}}{\partial V_T}\right)^2\sigma^2_{V_T}+\left(\frac{\partial T_{Delay}}{\partial \mu}\right)^2\sigma^2_{\mu}+\left(\frac{\partial T_{Delay}}{\partial t_{ox}}\right)^2\sigma^2_{t_{ox}}+\left(\frac{\partial T_{Delay}}{\partial L_{eff}}\right)^2\sigma^2_{L_{eff}}+\left(\frac{\partial T_{Delay}}{\partial W}\right)^2\sigma^2_{W}$$
$$+\left(\frac{\partial T_{Delay}}{\partial T_{ILD}}\right)^2\sigma^2_{T_{ILD}}+\left(\frac{\partial T_{Delay}}{\partial W_{\text{int}}}\right)^2\sigma^2_{W_{\text{int}}}+\left(\frac{\partial T_{Delay}}{\partial t_{\text{int}}}\right)^2\sigma^2_{t_{\text{int}}}$$

where:

$$\frac{\partial T_{Delay}}{\partial V_T}=\frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial V_T}=2.30\left(C_0+C_{\text{int}}\right)\frac{R_0}{V_{DD}-V_T}$$

$$\frac{\partial T_{Delay}}{\partial V_T}=\frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial \mu}=2.30\left(C_0+C_{\text{int}}\right)\frac{R_0}{\mu}$$

$$\frac{\partial T_{Delay}}{\partial t_{ox}}=\frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial t_{ox}}+\frac{\partial T_{Delay}}{\partial C_0}\frac{\partial C_0}{\partial t_{ox}}=2.30\left(C_0+C_{\text{int}}\right)\frac{R_0}{t_{ox}}+2.30\left(R_0+R_{\text{int}}\right)\frac{C_0}{t_{ox}}$$

$$\frac{\partial T_{Delay}}{\partial L_{eff}}=\frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial L_{eff}}+\frac{\partial T_{Delay}}{\partial C_0}\frac{\partial C_0}{\partial L_{eff}}=2.30\left(C_0+C_{\text{int}}\right)\frac{R_0}{L_{eff}}+2.30\left(R_0+R_{\text{int}}\right)\frac{C_0}{L_{eff}}$$

$$\frac{\partial T_{Delay}}{\partial W}=\frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial W}+\frac{\partial T_{Delay}}{\partial C_0}\frac{\partial C_0}{\partial W}=2.30\left(C_0+C_{\text{int}}\right)\frac{R_0}{W}+2.30\left(R_0+R_{\text{int}}\right)\frac{C_0}{W}$$

$$\frac{\partial T_{Delay}}{\partial T_{ILD}}=\frac{\partial T_{Delay}}{\partial C_{\text{int}}}\frac{\partial C_{\text{int}}}{\partial T_{ILD}}=\left(1.02R_{\text{int}}+2.30R_0\right)\frac{C_{\text{int}}}{T_{ILD}}$$

$$\frac{\partial T_{Delay}}{\partial W_{\text{int}}}=\frac{\partial T_{Delay}}{\partial R_{\text{int}}}\frac{\partial R_{\text{int}}}{\partial W_{\text{int}}}+\frac{\partial T_{Delay}}{\partial C_{\text{int}}}\frac{\partial C_{\text{int}}}{\partial W_{\text{int}}}=\left(1.02C_{\text{int}}+2.30C_0\right)\frac{R_{\text{int}}}{W_{\text{int}}}+\left(1.02R_{\text{int}}+2.30R_0\right)\frac{C_{\text{int}}}{W_{\text{int}}}$$

$$\frac{\partial T_{Delay}}{\partial t_{\text{int}}}=\frac{\partial T_{Delay}}{\partial R_{\text{int}}}\frac{\partial R_{\text{int}}}{\partial t_{\text{int}}}=\left(1.02C_{\text{int}}+2.30C_0\right)\frac{R_{\text{int}}}{t_{\text{int}}}$$

4. Comparison between clock skew models

The main goal of this work is to analyse the existent clock skew models and to determine which gives us the most accurate clock skew estimation for a specific clock distribution network. The efficiency of these models is going to be studied applying them to a generic H-tree structure, which is often used in the global part of a clock distribution network (higher hierarchical level).

At first, in section 4.1, a specific design methodology is presented. An H-tree with optimized buffers and wire sizes is going to be implemented. In the next section 4.2, the technology parameters of the 130 nm technology are shown in different tables. In section 4.3, numerical results of the simulations are presented. They have been done thanks to the JAVA program *"equations"*. This program calculates clock skew estimations according to the equations of the three analysed models. In appendix 4, program operation is presented. Finally, in section 4.4, clock skew estimations for future technologies are presented.

4.1. Design methodology

There are different possibilities for the structure of an H-tree clock distribution network. There could be intermediate buffers at the branching points (models 1 and 3) or not (model 2). In the second case, wire widths must be widened to avoid signal reflections at these split points.

In this work, the H-tree utilized for models simulations is going to be designed optimizing their buffers and wire sizes (width, thickness and length) according to different optimization methods. The result of optimizing interconnection length is that if in any level of the H-tree any branch length is longer than the optimal length, it is divided in k sections inserting optimal size inverters. The topology resultant is illustrated in the following figure 4.1:

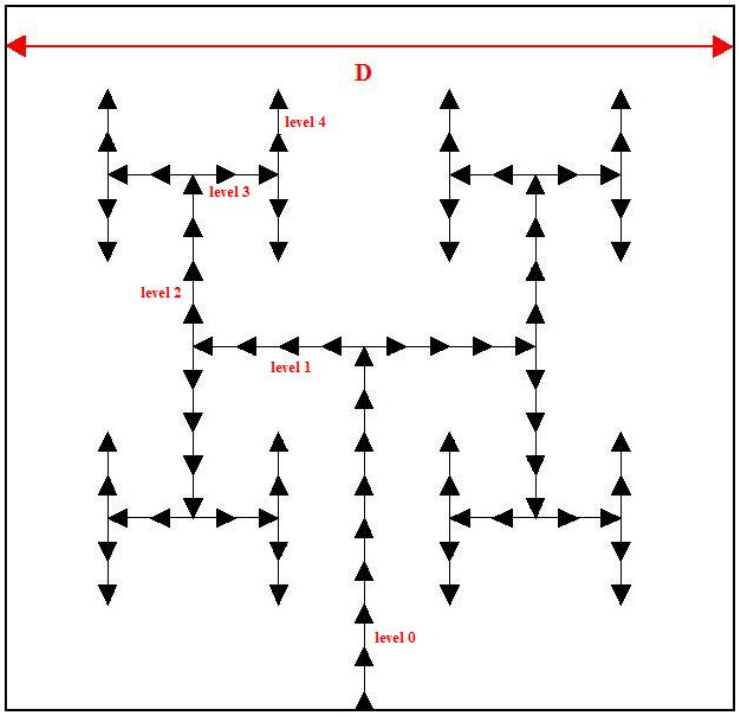


Figure 4.1: Buffered H-tree with optimized wire dimensions.

The first step is to characterize interconnection wires. Azad Naeemi, Raguraman Venkatesan and James D. Meindl present two methods (RC wire model and RLC wire model) [NAE-01], [NAE-02] where wire width is optimized for any technology. Since the inductance starts to become significant for long metal interconnections and high frequency operation, the RLC method has been chosen in this work. The optimal wire width is given by this expression:

$$W_{opt} = 1.88 c_0 \sqrt{\xi \rho \varepsilon_0 R_0 C_0} \qquad (4.1)$$

where c_0 is the speed of light in vacuum, ξ a constant depending on the wire geometry, ρ the metal resistivity, ε_0 the dielectric constant in vacuum, and R_0 and C_0 the output resistance and input capacitance of the minimum size repeater. For such interconnect width, wire thickness is calculated according the technology aspect ratio parameter. Then, using analytical models and formulas, the resistance, capacitance and inductance for wires are calculated.

The next step is to determine the optimum number and buffer size that we are going to insert in the H-tree. In this way, dividing branches in optimal length sections, their delay time is minimized [ISM-00]. If this time is smaller, then the clock skew is also smaller (because clock skew depends on the standard deviation of the path delay). The equivalent resultant wire after being divided is shown in figure 4.2:

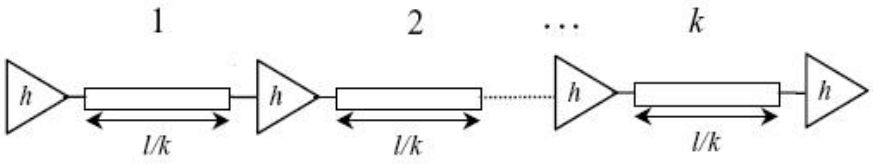


Figure 4.2: Repeaters inserted in an *RLC* line to minimize the propagation delay.

Now time expressions are transformed in the next ones:

$$T_{delay} = k \cdot T_{segment}$$
$$T_{segment} = 2.3\frac{R_0}{h}\left(\frac{C_{\text{int}}}{k} + hC_0\right) + \frac{R_{\text{int}}}{k}\left(\frac{C_{\text{int}}}{k} + 2.3hC_0\right) \qquad (4.2)$$

where $T_{segment}$ is the delay per line segment, C_0 and R_0 are the input capacitance and output resistance of the minimum size inverter, C_{int} and R_{int} are the capacitance and resistance of the interconnection line in the path, k is the number of segments and h is the number of times that the inverters are bigger than the minimum size inverter. According to the expressions presented in [ISM-00], the optimal values of k and h taking into account the inductive wire effects are:

$$h_{opt} = \sqrt{\frac{R_0 C_{int}}{R_{int} C_0}} \frac{1}{\left[1 + 0.16\left(T_{L/R}\right)^3\right]^{0.24}} \qquad (4.3)$$

$$k_{opt} = \sqrt{\frac{R_{\text{int}} C_{int}}{2R_0 C_0}} \frac{1}{\left[1 + 0.18\left(T_{L/R}\right)^3\right]^{0.3}} \qquad (4.4)$$

where $T_{L/R}$ is:

$$T_{L/R} = \sqrt{\frac{L_{\text{int}} / R_{int}}{R_0 C_0}} \qquad (4.5)$$

In those expressions, R_{int}, L_{int} and C_{int} are the total resistance, inductance and capacitance of the line.

By other hand, in VLSI circuits, clocks lines are usually shielded by ground and power lines to reduce the noise coupled into signal lines from the clock lines. Furthermore, in this way the modelling of the capacitive and inductive wire characteristics is easier, resulting predictable equations that allow control the delay of each clock signal path and the clock skew between them.

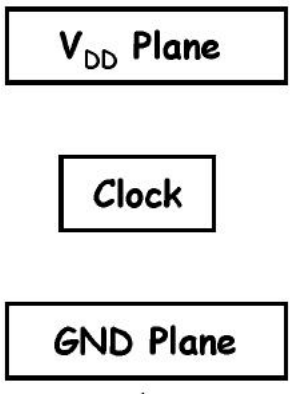


Figure 4.3: Wiring structure.

Models 1 and 3 can be applied to same clock distribution network topology: An H-tree with intermediate buffers at split points and same wire width at each level of the tree. Model 2 is applied to the other H-tree possibility, a tapered H-tree without intermediate buffers and widened wire widths. Therefore, some changes have to be done to compare the results of that model with the other two models results.

- Model 2 differences

The main difference in the equations of this model, comparing to models 1 and 3, is the parameter $r_{int}c_{int}$. It is the distributed capacitance and resistance of the line from the root to the leaf because no buffer is placed between them.

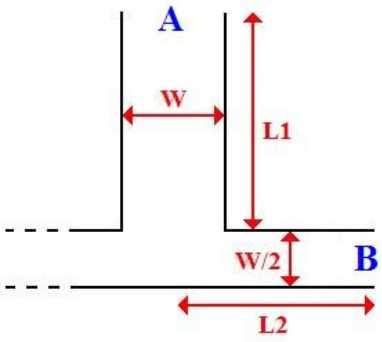


Figure 4.4: Wire diagram in model 2.

In an H-tree, wire width is doubled at each higher lever. In figure 4.4, the parameters of a tapered section are shown. The total resistance and capacitance from point A to point B are:

$$\begin{aligned} r_1 c_1 &= rc \\ r_2 c_2 &= 2r \cdot \frac{c}{2} = rc \\ RC_{total} &= rc \cdot \left(L_1 + L_2\right)^2 \end{aligned} \qquad (4.6)$$

where r and c are the resistance and capacitance per wire length unit. We can see that parameter $r_{int}c_{int}$ is the same in each level of the clock H-tree. It is a constant value. Therefore, in model 2, to calculate $r_{int}c_{int}$, we only have to multiply the parameters R_{int} and C_{int} of the models 1 and 3.

The parameter C_L in the model 2 equations is the capacitive load of the following sub-block in the hierarchical clock distribution network. We are going to assume that it is the input capacitance of the buffer that drives the next level sub-block, so we can use the same C_0 that we have in models 1 and 3. According to model 2 equations, the variance of C_L is going to be assumed 0 because we decided before that C_L is input capacitance of a buffer, so with the variance of V_t, t_{ox}, L_{eff} and V_{dd} the variance of C_L is defined.

The resultant tapered H-tree structure is shown in picture 4.5. The lowest level wire width is the minimal and it doubles at each higher level. No intermediate buffers are placed at split points.

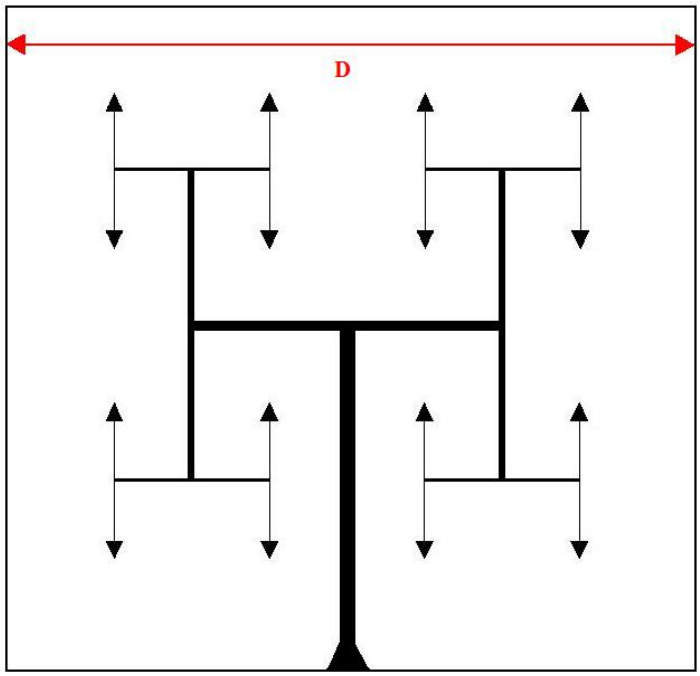


Figure 4.5: Tapered H-tree.

4.2.H-tree parameters (technology parameters)

In this section, the parameters of the 8 levels H-tree that is going to be utilized in simulations are presented. The chosen technology is 130 nm. Parameters are extracted from the ITRS roadmap and from the result of applying the design methodology equations. The first table shows the primary parameter of the 130 nm technology, like CMOS device parameters and minimal size interconnection wire parameters.

Parameter	Description	Value
V_{DD}	Supply voltage	1.2 V
V_T	Threshold voltage	0.19 V
T_{ox}	Gate oxide thickness	3.3 nm
D	Die size	17.32 mm

ε_{ILD}	Dielectric constant (interlevel)	3.6
W_{int} (min)	Minimal wire width	335 nm
A/R	Wire aspect ratio (t_{int} / W_{int})	2
t_{int} (min)	Minimal wire thickness	670 nm
R_0 (min)	Buffer output resistance (min. size inv.)	3.94 kΩ
C_0 (min)	Buffer input capacitance (min. size inv.)	0.77 fF

Table 4.1: 130 nm technology parameters [ITRS].

After applying the formulas to calculate optimized buffers and wire dimensions, next table shows buffer and wire parameters that are going to be used in the simulations.

Parameter	Description	Value
W_{int} (opt)	Optimal wire width	1250 nm
t_{int} (opt)	Optimal wire thickness	2500 nm
L_{int} (opt)	Optimal segment length	2.88 mm
W (opt)	Optimal buffer size (optimal width)	322 μm

Table 4.2: Optimized parameters.

The following table shows the number of required buffers in an 8 levels H-tree according to the optimal wire partition and the die size of the 130 nm technology.

Level	Branch length	Required buffers
0	D/2 = 8.66 mm	3
1	D/4 = 4.33 mm	2
2	D/4 = 4.33 mm	2
3	D/8 = 2.17 mm	1
4	D/8 = 2.17 mm	1
5	D/16 = 1.08 mm	1
6	D/16 = 1.08 mm	1
7	D/32 = 0.54 mm	1
8	D/32 = 0.54 mm	1

Table 4.3: Required buffers at each level.

Finally, next table shows the definitive parameters of the clock distribution network that is going to be utilized in the simulations. It is an 8 levels H-tree with optimized buffers and wire dimensions.

Parameter	Description	Value
N	Number of nodes	256
D	Chip size	17.32 mm
R_{int}	Interconnection resistance (optimal)	7.04e3 Ω/m
C_{int}	Interconnection capacitance (optimal)	3.4e-10 F/m
$C_{w\text{-}total}$	Total wire capacitance (397.76 mm)	135.24 nF
R_0	Buffer output resistance (optimal)	1.59 Ω
C_0	Buffer input capacitance (optimal)	1.91e-12 F
$C_{b\text{-}total}$	Total buffer capacitance (775 buffers)	1.48 nF

Table 4.4: H-tree parameters.

Finally, in the last table, process and circuit parameter tolerances are shown. It is necessary to know them because they are the input parameters of the JAVA program for the simulations. The transistor width deviation is calculated according to the minimal size technology deviation. It means that the value given for the 130 nm technology is referred to the minimum size transistor lithography. For optimal size transistors, the absolute value of the deviation is the same. Consequently, the percentage of deviation is smaller. The same calculations are done for the optimal wire width and thickness according to the deviation of minimal size wires.

Parameter	Description	Standard Deviation
σ_{VDD}	Power supply voltage	3.3 % [KAH-01]
σ_{VT}	Threshold voltage	4.2 % [KAH-01]
σ_{μ}	Charge carrier mobility	2 % [KAH-01]
σ_{tox}	Gate oxide thickness	1.3 % [KAH-01]
σ_{W}	Transistor width	5% of 130 nm → [KAH-01] 2.02e-3 % of 322 μm
σ_{Leff}	Transistor effective channel length	5 % [KAH-01]
σ_{Wint}	Wire width	3% of 335 nm → [FAN-98] 0.804% of 1250 nm
σ_{tint}	Wire thickness	3% of 670 nm → [FAN-98] 0.804% of 2500 nm
σ_{TILD}	ILD thickness	3 % [FAN-98]
σ_{T}	Temperature	8% [GRO-98]

Table 4.5: Parameter tolerances.

As was said in the previous section, model 2 requires some different parameters. First of them is $r_{int}c_{int}$, the distributed capacitance and resistance of the line from the root to the leaves. It is calculated multiplying the same optimal R_{int} and C_{int} that were used in models 1 and 3. C_L is the same input capacitance than C_0 in the previous models. Wire width variation σ_{Wint} is calculated for the worst case (last level width = optimal width).

Parameter	Description	Value
$\mathbf{r_{int}c_{int}}$	Distributed capacitance and resistance	2.39e-6 s/m^2
$\mathbf{C_L}$	Load capacitance	1.91e-12 F

Table 4.6: Model 2 parameter differences.

4.3. Simulations

To analyse the different clock skew models with the proposed H-tree described in the previous section (256 nodes, 130 nm technology), three different simulations have been realised. In the first of them, the value of the clock skew according to each model is compared. Secondly, each clock skew component is individually analysed to determine with parameter deviation has higher influence on the total clock skew value. The third simulation shows the clock skew estimation for different number of H-tree end points (number of levels).

We have to take into account that each model considers different clock skew sources. For example, only model 2 considers the temperature variation, or moreover the case that model 3 considers transistor width variations but model 1 does not.

4.3.1. First simulation

Clock skew estimations for the three models in a 256 nodes H-tree and 130 nm technology.

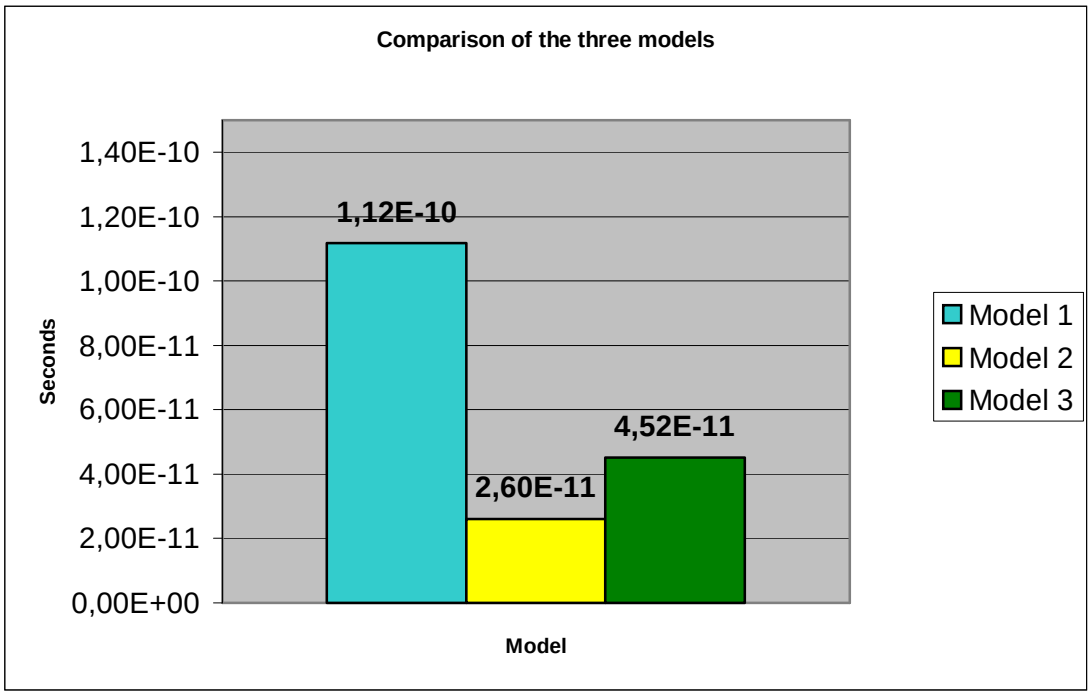


Figure 4.6: Clock skew estimations according to each model.

We can observe that the results are the expected. Model 1 gives us a value of *112 ps* for an H-tree with intermediate buffers and model 3 gives a value of *45.2 ps*. It is correct because the first model estimates an upper bound for the clock skew and the third model calculates the clock skew more accurately since it takes into account the correlation between overlapped parts of clock paths. Model 2 estimation is realized with a different clock distribution network topology, a tapered H-tree without intermediate buffers. The value of *26 ps* is accepted because it must be smaller than the values for buffered H-trees, where more sources of clock skew are present along the paths (intermediate buffers). The reliability of these estimations depends on how accurately the clock system parameters have been predicted.

4.3.2. Second simulation

Clock skew components according to each model in a 256 nodes H-tree and 130 nm technology.

- MODEL 1

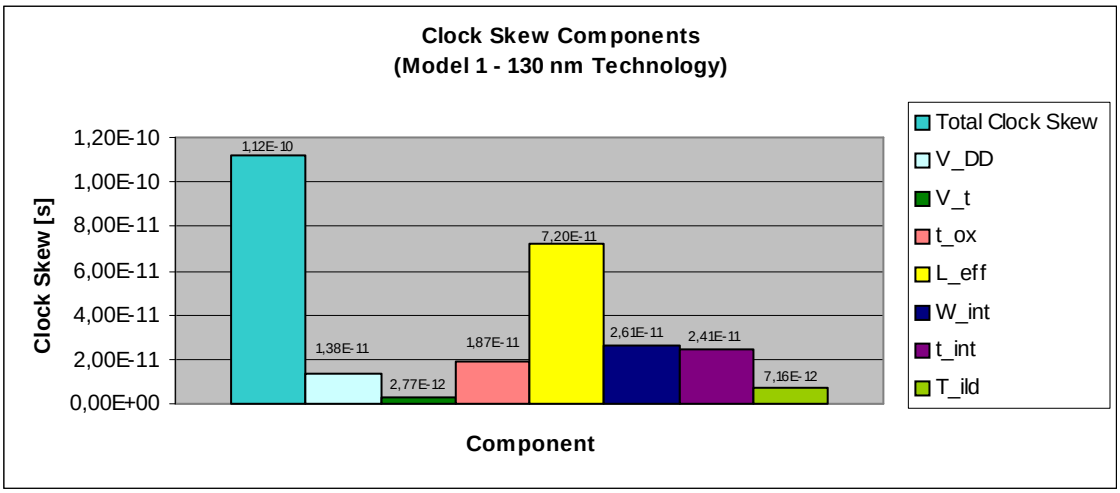


Figure 4.7: Components of the clock skew in model 1.

According to model 1 (and the parameter tolerance predictions), an important conclusion can be drawn: the tolerance of L_{eff}, the effective channel length, should be specially controlled because it is the main source in the total clock skew. The influence of this parameter (tolerance of *5 %*) is higher compared to the gate oxide thickness, t_{ox}, another transistor source of skew, but with smaller tolerance (*1.3 %*). As it was said in previous sections, buffers are the main source of clock skew. Their parameters may be accurately controlled to obtain low skew H-trees. Wire sources of skew are less important if their tolerances are not particularly significant.

- MODEL 2

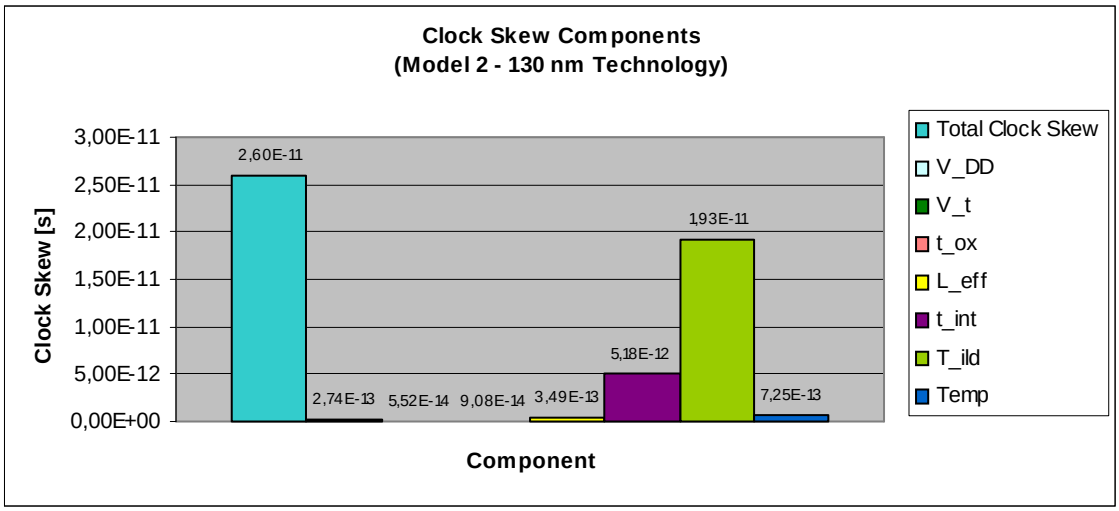


Figure 4.8: Components of the clock skew in model 2.

The main conclusion that can be extracted after simulate the model 2 in a tapered H-tree is that wire clock skew sources are more important than buffer sources. It is consequent because a tapered H-tree with *256* nodes has only *257* buffers and the optimized H-tree utilized in models 1 and 3 simulations has *576* buffers. By other hand, the ILD thickness is the main source among wire skew sources. It is due to a higher variance (*3 %)* than wire thickness variation (*0.804 %*).

- MODEL 3

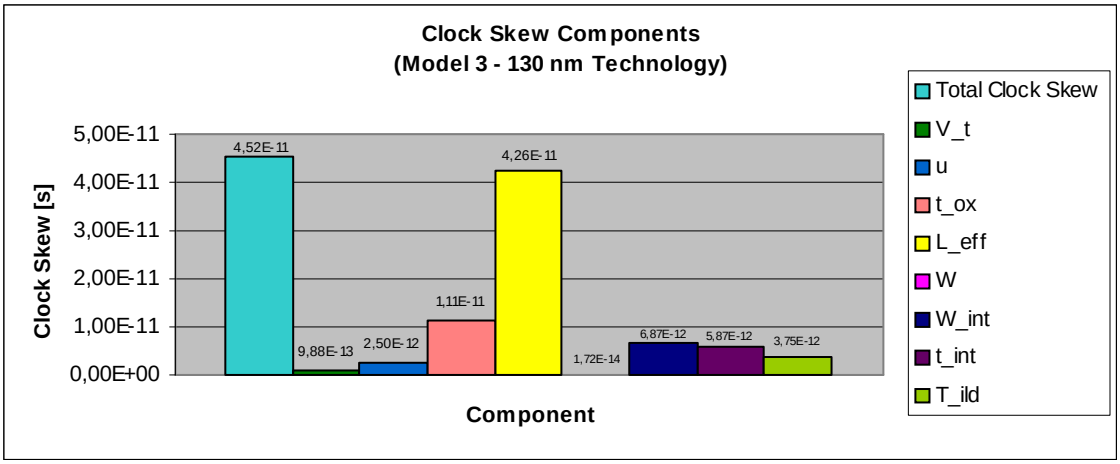


Figure 4.9: Components of the clock skew in model 3.

Practically the same conclusions than were drawn for the first model simulation can be drawn. Buffers parameters, principally L_{eff}, are the main source of clock skew. A different parameter, which model 1 does not consider, is transistor width *W*. The effect of it is almost negligible due to the very low tolerance value of this parameter (*2.02e-3 %*).

4.3.3. Third simulation

Clock skew estimations for different number of nodes in an H-tree implemented in the 130 nm technology.

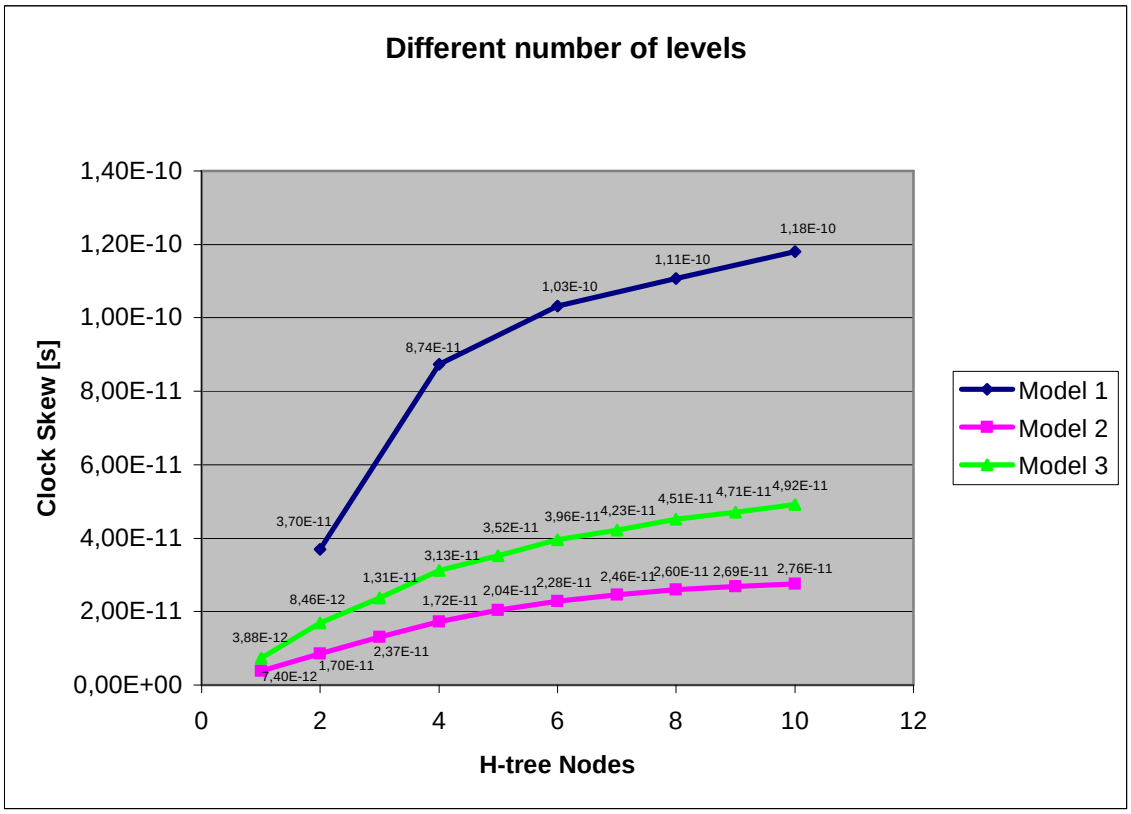


Figure 4.10: Clock skew in the 130 nm technology for different number of levels.

The results are again the expected. In every case, the clock skew value given by the model 3 is lower than the given by the model 1. It is logical because model 1 only provides an upper bound for the clock skew. Model 3 gives us a value considering the correlation between paths, so the value is less conservative. Estimations for model 2 are always the smallest. It is because the tapered H-tree has less clock skew sources than the equivalent optimized buffered H-tree.

We can observe how the clock skew increase as number of levels in the H-tree is bigger. This increase is less appreciable for high number of levels because when we add a new one, we are just adding a new buffer and a very small interconnection section.

On the other hand, model 1 estimations are made only for a even number of levels. Their equations have no meaning when they are applied to H-trees with an odd number of levels.

4.4. Clock skew for future technologies

Once the three models have been analysed and simulated, and several conclusions about them have been drawn, it is possible to apply them to a global clock distribution network designed with future technology parameters: an 8 levels H-tree implemented in 100, 70 and 45 nm technologies. We can verify how the clock skew will be in the future for this topology and if an H-tree like that will be appropriate within a global clock distribution network.

At first, technology parameters of these technologies are presented in the following table. We can compare them to the 130 nm parameters.

Parameter	130 nm	100 nm	70 nm	45 nm
V_{DD}	1.2 V	1 V	0.9 V	0.6 V
V_T	0.19 V	0.15	0.06	0.021
T_{ox}	3.3 nm	2.5 nm	1.6 nm	1.4 nm
D	17.32 mm	17.32 mm	20 mm	21.21 mm
ε_{ILD}	3.6	3.3	3	3.6
W_{int} (min)	335 nm	237 nm	160 nm	103 nm
A/R	2	2.1	2.2	2.3
t_{int} (min)	670 nm	500 nm	352 nm	235 nm
R_0 (min)	3.94 kΩ	4.95 kΩ	8.02 kΩ	15.7 kΩ
C_0 (min)	0.77 fF	0.66 fF	0.457 fF	0.34 fF

Table 4.7: Technology parameters for the chosen technologies [ITRS].

After applying the equations, the optimized parameters for the buffer and wire size are in the next table:

Parameter	130 nm	100 nm	70 nm	45 nm
W_{int} (opt)	1250 nm	1005 nm	850 nm	620 nm
t_{int} (opt)	2500 nm	2110 nm	1870 nm	1426 nm
L_{int} (opt)	2.88 mm	2.16 mm	1.67 mm	1.32 mm
W (opt)	322 µm	354 µm	496 µm	648 µm

Table 4.8: Optimized parameters.

The following table shows the number of required buffers in a 256 nodes H-tree according to the optimal segment partition and the die size in the 130 nm technology.

		130 nm		100 nm		70 nm		45 nm	
Level	**Section**	**Length**	**Buffs.**	**Length**	**Buffs.**	**Length**	**Buffs.**	**Length**	**Buffs.**
0	D/2	8.66mm	3	8.66mm	4	10mm	6	10.6mm	8
1	D/4	4.33mm	2	4.33mm	2	5mm	3	5.30mm	4
2	D/4	4.33mm	2	4.33mm	2	5mm	3	5.30mm	4
3	D/8	2.17mm	1	2.17mm	1	2.5mm	2	2.65mm	2
4	D/8	2.17mm	1	2.17mm	1	2.5mm	2	2.65mm	2
5	D/16	1.08mm	1	1.08mm	1	1.25mm	1	1.33mm	1
6	D/16	1.08mm	1	1.08mm	1	1.25mm	1	1.33mm	1
7	D/32	0.54mm	1	0.54mm	1	0.63mm	1	0.67mm	1
8	D/32	0.54mm	1	0.54mm	1	0.63mm	1	0.67mm	1

Table 4.9: Required buffers at each level.

Finally, next table shows the definitive parameters of the clock distribution network that is going to be utilized in the simulations. It is a 8 level H-tree (256 nodes) with optimized buffers and wire dimensions.

Parameter	**130 nm**	**100 nm**	**70 nm**	**45 nm**
N	256	256	256	256
D	17.32 mm	17.32 mm	20 mm	21.21 mm
R_{int}	7.04e3 Ω/m	10.04e3 Ω/m	13.8e3 Ω/m	24.9e3 Ω/m
C_{int}	3.4e-10 F/m	3.11e-10 F/m	2.92e-10 F/m	2.64e-10 F/m
$C_{w\text{-}total}$	135.24 nF	123.70 nF	134.32 nF	128.73 nF
R_0	1.59 Ω	1.40 Ω	1.13 Ω	1.09 Ω
C_0	1.91e-12 F	2.34e-12 F	3.24e-12 F	4.90e-12 F
$C_{b\text{-}total}$	1.48 nF	1.82 nF	2.62 nF	4.00 nF

Table 4.10: H-tree parameters.

One of the biggest problems presented during the elaboration of this work was the difficulty in searching the exact parameters for different technologies. Due to this reason, tolerances of the required parameters are going to be chosen the same than the 130 nm case. Therefore, the following table presents them.

Par.	130 nm	100 nm	70 nm	45 nm
σ_{VDD}	3.3 % [KAH-01]	3.3 % [KAH-01]	3.3 % [KAH-01]	3.3 % [KAH-01]
σ_{VT}	4.2 % [KAH-01]	4.2 % [KAH-01]	4.2 % [KAH-01]	4.2 % [KAH-01]
σ_{μ}	2 % [KAH-01]	2 % [KAH-01]	2 % [KAH-01]	2 % [KAH-01]
σ_{tox}	1.3 % [KAH-01]	1.3 % [KAH-01]	1.3 % [KAH-01]	1.3 % [KAH-01]
σ_{W}	5% of 130 nm → 2.02e-3 % of 322 µm [KAH-01]	5% of 130 nm → 2.02e-3 % of 322 µm [KAH-01]	5% of 130 nm → 2.02e-3 % of 322 µm [KAH-01]	5% of 130 nm → 2.02e-3 % of 322 µm [KAH-01]
σ_{Leff}	5 % [KAH-01]	5 % [KAH-01]	5 % [KAH-01]	5 % [KAH-01]
σ_{Wint}	3% of 335 nm → 0.804% of 1250 nm [FAN-98]	3% of 237 nm → 0.707% of 1005 nm [FAN-98]	3% of 160 nm → 0.565% of 850 nm [FAN-98]	3% of 103 nm → 0.498% of 620 nm [FAN-98]
σ_{tint}	3% of 670 nm → 0.804% of 2500 nm [FAN-98]	3% of 670 nm → 0.711% of 2110 nm [FAN-98]	3% of 352 nm → 0.565% of 1870 nm [FAN-98]	3% of 235 nm → 0.494% of 1426 nm [FAN-98]
σ_{TILD}	3 % [FAN-98]	3 % [FAN-98]	3 % [FAN-98]	3 % [FAN-98]
σ_{T}	8% [GRO-98]	8% [GRO-98]	8% [GRO-98]	8% [GRO-98]

Table 4.11: Parameter tolerances.

The next figure 4.11 shows the estimation for the 256 nodes H-tree (8 levels) designed according to four different technologies: 130 nm, 100 nm, 70 nm and 45 nm.

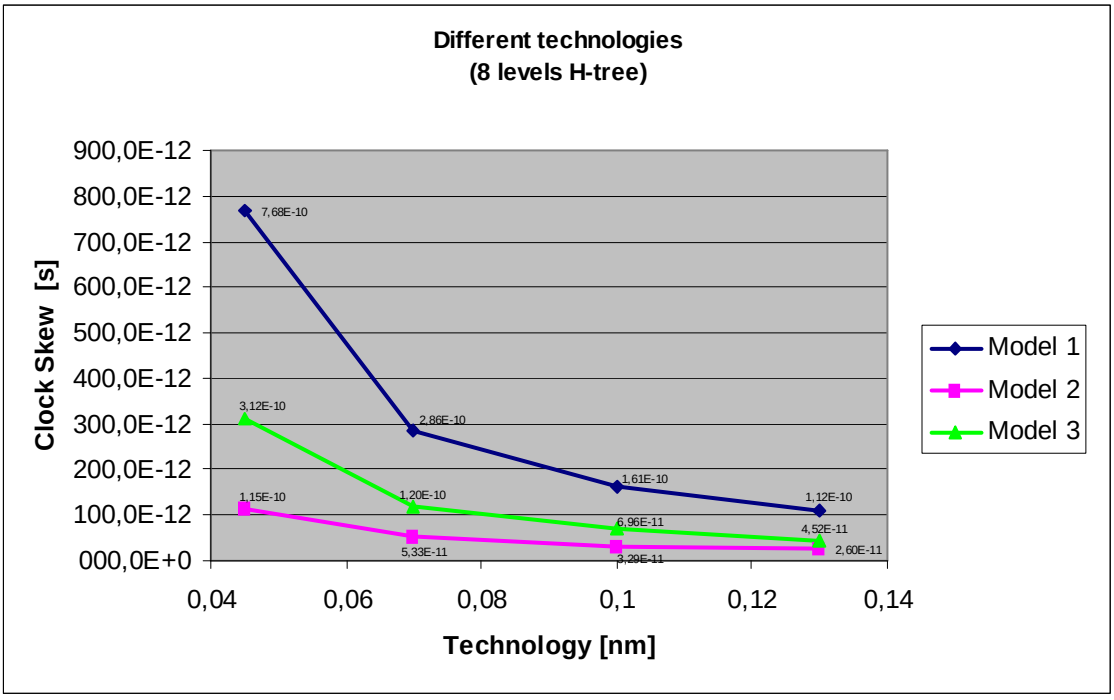


Figure 4.11: Clock skew for different technologies (130, 100, 70, 45 nm).

In every case, the clock skew value given by model 3 is lower than the given by model 1. It is logical because model 1 only provides an upper bound for the clock skew. Model 3 gives us a value considering the correlation between paths, so the value is less conservative. Model 2 estimations for the tapered H-tree are always smaller than the estimations for the H-tree with intermediate buffers. It is reasonable because there are less clock skew sources in a tapered H-tree (less buffers).

On other hand, an increasingly clock skew is observed as technology size is decreasing. It is due to the increase of the RC constant (both wires and inverters) when technology size decreases. The raise is exponential, so for each new technology the clock skew in an H-tree increases considerably. That presents a big problem because one of the main goals of a newer technology is to achieve higher clock frequencies. To do that, clock skew has to be kept proportional to the clock period (10 % approximately). In the next table and in figure 4.12, the relation between the clock skew and the clock period is shown. Only an H-tree implemented in the 130 nm technology can satisfy the design rule of 10 %.

Parameter	130 nm	100 nm	70 nm	45 nm
Clock frequency [ITRS]	1.6 GHz	3.0 GHz	9 GHz	15 GHz
Clock period	625 ps	333 ps	111 ps	66.7 ps
Clock skew (model 3)	45.2 ps	69.6 ps	120 ps	312 ps
% (Clock skew vs. period)	6.8 %	20.9%	108.1 %	467.8 %

Table 4.12: Relation between clock skew and clock period.

An important conclusion can be drawn from this simulation: a single H-tree like that is not appropriate if we want small clock skew values for a global clock distribution network. Additional methods like deskew circuits or feedbacks are necessary to obtain low skew clock distribution networks in modern technologies.

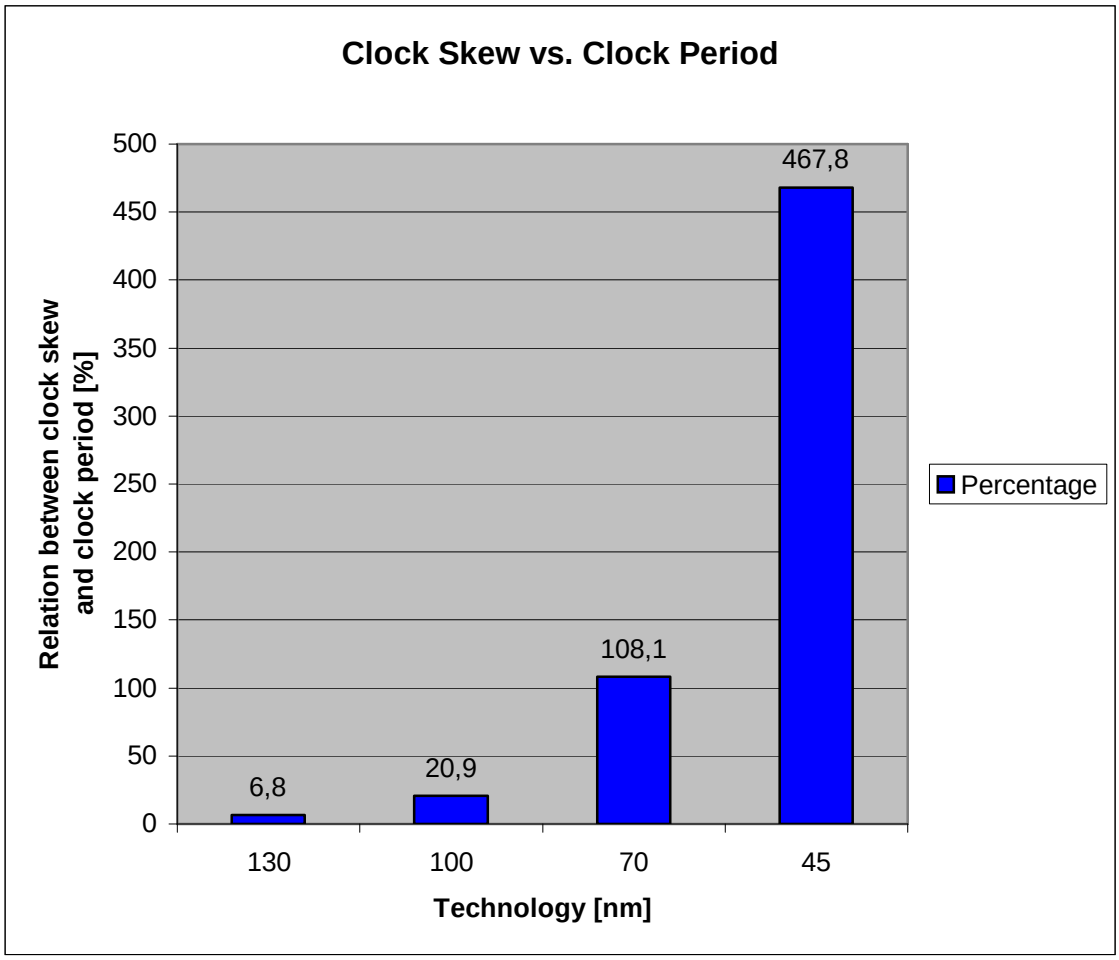


Figure 4.12: Clock skew vs. Clock period.

5. Conclusions

In this work several aspects about the clock skew within a synchronous digital system have been considered. Nowadays, clock skew is among the main parameters that imposes restrictions in a digital system performance and reliability. It is actually one of the issues that requires more time of study.

At first, the role of a clock distribution network within a synchronous digital system was presented. A hierarchical clock distribution network, starting with the global part and finishing with the local clocking, was decided as the best option to implement a clock scheme since it allows the partition of the system in different functional sub-blocks. The advantages and disadvantages of the different strategies and topologies to implement a clock distribution network were discussed.

Secondly, an extensive analysis of the clock skew was performed. The theoretical background was described and their main sources were presented. One of the main goals of this work was to look for the existing models to estimate the clock skew in a clock distribution network. Three different models (statistical models according process and circuit variations) were studied and their equations were presented (models 1, 2 and 3).

The third part of this work was probably the most important. An extensive analysis of the models was realised thanks to different simulations in a generic H-tree (with optimized buffers and wires). They are usually utilized in the global part of a hierarchical clock distribution network. A java program, "*equations.java*", was specifically developed to calculate the clock skew estimations according to each model equations. Clock skew prediction for future technologies were made in a H-tree.

Concerning the models, model 1, which gives us an upper bound for the clock skew, and model 3, which gives an estimation that takes into account the correlation between clock paths, were compared in a H-tree with intermediate buffers. The main conclusion that could be drawn is that model 1 gave us a too conservative estimation compared to model 3. However, the validity of the simulations is dependent upon the accuracy of the

process and circuit variation parameters, which was probably the greatest difficulty found during the elaboration of the work. Model 2, which is applied to a tapered H-tree with widened wires, gave estimations smaller than the other two models. The reason is that a tapered H-tree has less clock skew sources (there are not buffers at the split points).

If models are applied to an H-tree for futures technologies, an important conclusion was drawn. Since clock skew increases rapidly for newer technologies due to the RC constant of buffers and wires increases considerably, a single H-tree is not appropriated for a low skew global clock distribution network. Additional techniques, like deskew circuits, are necessary to keep a low skew value within the system.

To finish, we can say that main goals of the work were achieved, but with the uncertainty of knowing if the clock skew value is realistic or not due to the difficulties of finding parameter values. The qualitative behaviour of the models has been described and checked.

Appendix 1

Clock Skew Model 1

Steven D. Kugelmass, Kenneth Steiglitz [KUG-88]

1. Introduction

The accumulation of clock skew, the differences in arrival times of signal in digital systems with a central clock, is one of the factors that limit the speed in these systems.

These authors present a probabilistic model for the accumulation of clock skew in synchronous systems. Using this model, it's possible to estimate upper bounds for expected clock skew between processing elements in a processor array, and it's variance, in tree distribution systems with *N* synchronously clocked processing elements.

These results can be applied to two specific models for clock distributions:

- The first, *metric-free* model, the skew is in a buffer stage is Gaussian with a variance independent of wire length. In this case, the upper bound on skew grows as $\Theta(ln^{3/2} N)$ for a system with *N* processing elements.

- The second, *metric* model, is intended to reflect VLSI constraints: the clock skew in a stage is Gaussian with a variance proportional to wire length. In this case the upper bound on expected skew is $\Theta\left(\sqrt{N \ln N}\right)$ for a system with *N* processors.

Thus the probabilistic model is more optimistic than the deterministic summation model of Fisher and Kung [FIS-85], which predict a clock skew *Θ(N)* in this case, and is also consistent with their lower bound of $\Omega\left(\sqrt{N}\right)$ for planar embeddings. Fisher and Kung's model ignore a fundamental property of clock skew: its origins in the random variations of propagation time through buffers and wires.

2. General model of signal distribution

A global signal, such as clock, is distributed throughout a processing system by a signal distribution system. This, is composed of a number buffers (amplifiers) and wires which may be organized in a number of different ways. Two common structures are a bus and a tree.

A clock distribution system can be represented by a graph. It has a single distinguished vertex called the *source*. This is the origin of the global signal, and it's the only input to the distribution system. This distribution system can have multiple destinations, but for practical reasons, there is exactly one path from the source to each destination, that we are going to call *processing elements* (PE). PEs may have their own internal signal distribution system. This internals systems can be modelled in a similar manner to the global signal distribution system.

Each buffer and wire in the clock distribution system propagates and delays its incoming signal. Therefore, it's natural to associate every delay element in the signal distribution system, either a buffer or a wire, with a random variable, d_j. The value of the random variable gives the delay contribution of that element. The delay at any point is a real random variable, which is the sum of all the random variables along the path from the source to that point.

These definitions constitute de essence of the model. They make it very simple. But extremely general, and allow one to model any clock distribution system. Geometry can be incorporated into the model by attaching an appropriate probability density function to wire delays. There is also the freedom to analyze as much or as little as desired by

creating simplified models, in which buffer delays or wire delays can be ignored entirely.

The primary interest is the skew, the distribution of arrival times of a particular clock pulse to al PEs that communicate. Because of the model is probabilistic, it's not possible to give an expression for the worst case skew, but we derive an expression for the expected maximum skew by assuming that all PEs communicate.

We haven't made any mention of any particular probability density function. The total delay through the distribution system is the sum of a number of random variables. In many cases, it quickly converges to a normal distribution, by the Central Limit Theorem. That's why, in the case of the skew computations, the actual distributions attached to buffers and wires are usually relatively unimportant.

The arrival times of a signal to the N PEs constitute a random sample of size N. From this sample, find the difference between the largest of them, A_{max}, and the smallest, A_{min}. The random variable $R = A_{max} - A_{min}$ is called *range* of the sample. The range is equivalent to the skew in the signal distribution system.

3. Analysis and upper bounds

There is a lot of literature that describe techniques to compute the expected range of a set of independent identically distributed (iid) random variables. However, little is described about the case when variables are dependent, as they are in a clock tree. Fortunately, it is possible to the statistics of iid variables to predict an upper bound on the statistics of the dependent variables. The relationship is given by the following theorem:

Theorem 1: *"The expected range of a set of random variables, which are dependent because they are sums of overlapping variables, is no greater than the expected range of the corresponding set of independent random variables"*.

We define y_i,, $i=1,2,\ldots,N$ as iid real random variables, with $N=kn$, and σ_j, $j=1,2,..,n$ as n disjoint subsets of different y_i, each of cardinality k. Now, we define τ_i similarly, with k distinct elements each, except that they are not necessarily disjoint. Define the corresponding sums of the y_i, by:

$$s_j = \sum_{y \in \sigma_j} y, \qquad t_j = \sum_{y \in \tau_j} y$$

The **theorem 1** demonstrates that **the expected range of the s_j dominates the expected range of the t_j,** and so any upper bound for the first also holds for the second. It's demonstrated using the following lemmas.

Lemma 1: *"If $F_A(x)$ and $F_B(x)$ are the probability distribution function for random variables A and B, and we suppose that $F_A(x)$ and $F_B(x)$ are differentiable and A and B have finite means and variances. Then, if $F_A(x) \geq F_B(x)$ for all, then $E(A) \leq E(B)$".*

Lemma 2: *"For any α and β, without any restriction in them, and any continuous probability density P: $P(y \leq \alpha \mid y \leq \beta) \geq P(y \leq \alpha)$. This lemma expresses that if $y \leq \beta$, then $y \leq \alpha$".*

Now that we have demonstrated this theorem, any bound for iid variables is an upper bound for the variables that arises in the clock distribution system model.

At this point we are going to assume that the arrival times are Gaussian, motivated by the Central Limit Theorem. Although no closed-form expression is known for the expected value and the variance of the range of N iid Gaussian distributed random variables, it is possible to obtain asymptotic expressions. This is shown by the following theorem.

Theorem 2: *"Let x_i, $i=1,2,\ldots N$ be random samples from an $n(\mu,\sigma^2)$ normal distribution, and let $R = x_{max} - x_{min}$ be the difference of the largest and smallest x_i. Then the expected value of R is asymptotically:*

$$E[R] = \sigma \left[\frac{4\ln N - \ln\ln N - \ln 4\pi + 2C}{(2\ln N)^{1/2}} + O\left(\frac{1}{\log N}\right) \right] \qquad (1)$$

where $C \approx 0.5772$ is Euler constant. The variance of R is given by:

$$Var[R] = \frac{\sigma^2}{\ln N}\frac{\pi^2}{6} + O\left[\frac{1}{\log^2 N}\right]" \qquad (2)$$

Equation (1) is therefore asymptotic upper bound on the expected skew in a clock distribution tree with N leaves.

4. Examples

The model is applied to two different global signal distribution systems. The examples represent what it's considered to be common, typical clock distribution systems, but they are not intended to represent the full scope of all possibilities.

4.1. Metric-free tree

The first example is a metric-free tree. This type of topology could be used to implement a large-scale distribution system, which would provide a clock to chips on a board or to boards in a system. It doesn't constrain the circuit to be planar, so it's possible to equalize the lengths of all the wires in the tree. Therefore, every wire has the same probability distribution for delay, which can be lumped with the delay of the buffer that follows it. This results in a model of a tree of buffers without wires.

The root of the tree (node *A*) is the source of the signal, the PEs are placed at the leaves (nodes *M-Z*), and the intervening levels consist of buffers and wires. Figure 1 represents the tree:

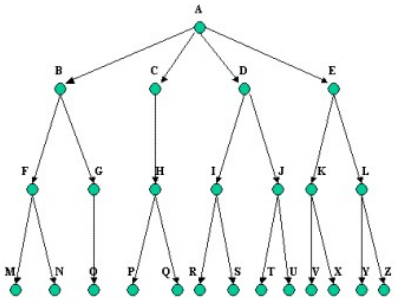


Figure 1: Clock distribution tree.

In Figure 1, internal nodes are buffers, which retransmit the clock signal. The leaf nodes of the tree are the PEs that communicates among them. Lines connecting nodes of the clock tree represent wires that conduct the clock signal to all the PEs.

The delay through a buffer of the clock distribution tree can be modelled as a real random variable d_i, The arrival time of the clock signal to any PE is the sum of the delays along the path from the root of the tree to the PE. The buffers in this metric-free model cause all the delays. The effect of a wire is absorbed by the line effect of the buffer that follows it. The arrival time at leaf i, A_i, is therefore the random variable:

$$A_i = \sum d_j \text{ , where the } d_j \text{ lie on the path from the root to the leaf } i.$$

In order to apply theorem 2, we must estimate the underlying distribution of the A_i.

Assuming that there are N PEs, each A_i is the sum of $log_2 N$ d_j's and that each d_j has mean μ_b and variance σ_b^2. By our Gaussian assumption, the A_i have the distribution $n(\mu_b log_2 N, \sigma_b^2 log_2^2 N)$. Applying theorem 2 with distribution, we find that the expected skew is:

$$E[Skew] = \sigma_b \log_2 N \left[\frac{4\ln N - \ln\ln N - \ln 4\pi + 2C}{(2\ln N)^{1/2}} + O\left(\frac{1}{\log N}\right) \right]$$
$$= \sigma_b \frac{4\ln^{3/2} N}{\sqrt{2\ln 2}} + lower\ \ order\ terms \qquad (3)$$
$$= \Theta(\ln^{3/2} N)$$

The variance of the skew is:

$$Var\left[Skew\right] = \sigma_b{}^2 \frac{\pi^2}{6} + O\left[\frac{1}{\log^2 N}\right] \qquad (4)$$

which is a constant when $N \to \infty$.

The following simulations corroborate the asymptotic skew results. Figure 2 shows the asymptotic curve of *E[Skew]*, with dashed line. The solid line represents the result of the Monte Carlo simulation after 100 trials. It shows that the bound is relatively tight, despite the fact that dependence between variables is ignored in the equation of *E[Skew]*. The gap between the asymptote and simulation results decreases steadily from about 20% for $N=2^2$ to about 10% for $N=2^{14}$.

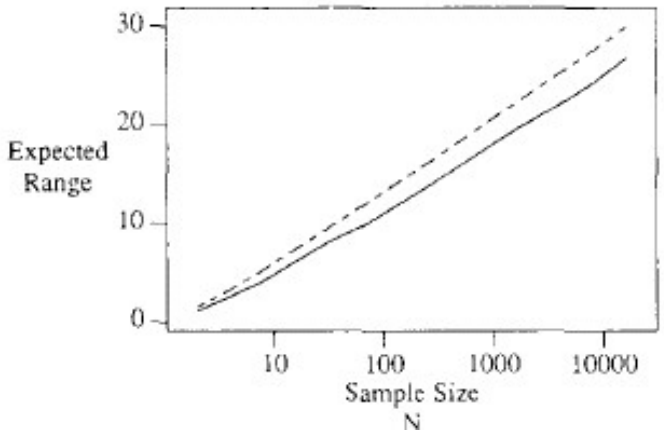


Figure 2: Metric-free simulations.

Even for trees of small depth, the expected range in both cases is nearly identical. This is due to the rapid convergence of the sums of random variables to a normal distribution.

The explicit inclusion of wire delays into the model (not to include wire delays with buffer delays) does not significantly modify the results. Wires can be considered to contribute an additional random delay at each level of the tree. Assuming that wire delays are distributed similarly to the buffer delays, the effect is to increase the variance

of the variance of the distribution of arrival times by a constant factor. This doesn't alter the asymptotic behaviour of skew.

4.2.Metric tree

The second example is a *metric* tree. This is the type of system that is typical of systems described for VLSI. The central assumption of this topology is that circuit must be embedded in the plane. If the embedding is to be area efficient, then the wires that connect buffers cannot be the same length everywhere. The delay through a wire therefore depends its location in he tree, and cannot be lumped with a buffer delay. A common tree of this type is the H-tree (Figure 3).

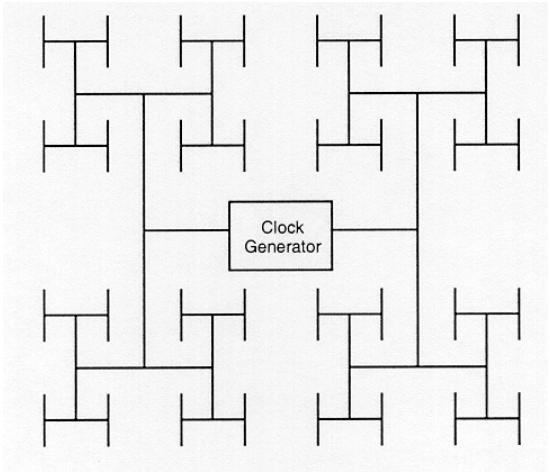


Figure 3: H-tree clock distribution system.

There are two distinct views of the effects of increasing system size (number of PEs) under the metric assumption. The first is to assume that a tree with an arbitrary number of leaves can be embedded in the fixed area of the integrated circuit. The alternative to this view sets a lower limit on the size of the smallest feature: in this case the size of a wire at the tree's leaves. Each preceding level is progressively larger (to avoid reflections in the wires) and the area of the entire clock tree grows with increasing system size. This view ignores the effects of shrinking feature size but its compatible with increases in this chip die size. We are going to adopt this second point of view.

Now we assume that buffer delay, d_j, is $n(\mu_b,\sigma_b^2)$. For this analysis we will also assume that wire delay, w_j, is Gaussian distributed with a mean value and a variance proportional to its length. The linear relationship for the variance can be justified by considering a long wire to be equivalent to two shorter wires placed end to end. The propagation delay of the long wire is equal to the sum of the propagation delays of the two shorter wires. The expected values add, as do the variances because the delays of he short wires are independent.

The wire delay at leaves of the tree is $n(\mu_w,\sigma_w^2)$ distributed. Because wire length doubles at each higher level of the tree, the distribution of w_j can be written as function of the depth, d, of the wire. We find that w_j, is $n\left(\mu_w \frac{N}{2^d}, \sigma_w^2 \left(\frac{N}{2^d}\right)^2\right)$, where $1 \le d \le log_2 N$ and μ_w and σ_w^2 are the starting wire (lowest level) time delay mean and variance.

The total delay, A_i, is the sum of $log_2 N$ buffers delay and the sum delays from each level of the tree. In a H-tree, the root to leave distance, starting with a unit length at leaves, follows geometric series $\left(1+1+2+2+4+4+\ldots+\frac{\sqrt{N}}{2}+\frac{\sqrt{N}}{2}\right) = 2\left(\sqrt{N}-1\right)$. The total delay, A_i, therefore has the distribution: $n\left(\mu_b \log_2 N + \mu_w 2\left(\sqrt{N}-1\right),\ \sigma_b^2 \log_2^2 N + \sigma_w^2 \left(2\left(\sqrt{N}-1\right)\right)^2\right)$.

As $N \to \infty$, the linear (wire) term dominates. The expected skew is therefore:

$$\begin{aligned} E[Skew] &= \sigma_w 2\left(\sqrt{N}-1\right)\left[\frac{4\ln N - \ln\ln N - \ln 4\pi + 2C}{(2\ln N)^{1/2}} + O\left(\frac{1}{\ln N}\right)\right] \\ &= \sigma_w \frac{8}{\sqrt{2}}\left(\sqrt{N}-1\right)\sqrt{\ln N} + lower\ \ order\ terms \\ &= \Theta\left(\sqrt{N \ln N}\right) \end{aligned} \qquad (5)$$

and the variance is given by:

$$Var[Skew] = \frac{\sigma_w^2 2\sqrt{N}}{\ln N}\left(\frac{\pi^2}{6}\right) + O\left[\frac{1}{\log N}\right] \qquad (6)$$

Next Figure 4 shows the comparison of the H-tree asymptotic bound (dashed line) an Monte Carlo simulations (solid line) after 100 trials. In this case, the bound is not as tight as in the metric-free case, because the shared variables, representing deviation from the independence assumption, are near the root of the tree, where the wire lengths are longer.

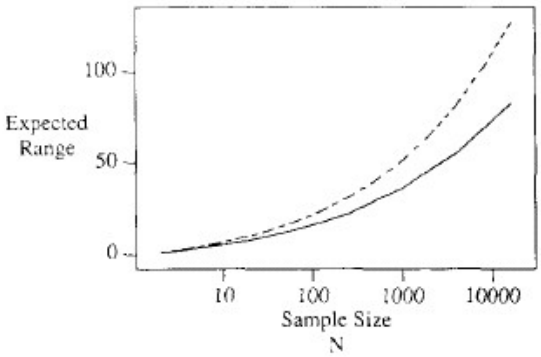


Figure 2: Metric simulations.

4.3.Mistake probability

It's possible to give an estimate on the probability that the sample value of the skew is outside a certain range. Assume that X is a random variable with real mean μ and variance σ^2. Then, using the one-sided Chebyshev inequality, we can predict un upper bound on the probability of exceeding the mean skew by an amount a:

$$P(X > (\mu + a)) \leq \frac{\sigma^2}{\sigma^2 + a^2} \qquad (7)$$

Now let $a=\alpha\mu$, where α is the fractional deviation from the expected value of skew. Then, using the estimate, in both the metric and metric-free case, we can have an estimate of an upper bound as $N \rightarrow \infty$:

$$P(X > (\mu + a)) \leq \frac{\frac{\pi^2}{6}}{\frac{\pi^2}{6} + 8\alpha^2 \ln^2 N} \qquad (8)$$

5. Calculation

Now, we are going to show how to calculate the parameters necessaries to apply de model.

5.1.Metric-free tree

In this model, I need the parameter σ_b^2, that is the delay variance through a buffer of the clock distribution tree (effect of the wire is absorbed by effect of the buffer).

Using Sakurai's model for interconnection delays, the delay of a stage composed of a wire interconnecting two buffers (inverters) is:

$$T_{Delay} = 1.02 R_{int} C_{int} + 2.30\left(R_0 C_0 + R_0 C_{int} + R_{int} C_0\right) \qquad (9)$$

where R_{int} and C_{int} are the total resistance and capacitance of the interconnecting wire, R_0 is the on-resistance of the driving transistor, and C_0 is the input capacitance of the driving inverter.

$$R_0 \approx \frac{L_{eff}/W}{\mu C_{ox}\left(V_{DD} - V_T\right)}, \quad C_o = C_{ox} \cdot W \cdot L_{eff}, \quad R_{int} = \frac{\rho}{W_{int} t_{int}} L_{int}, \quad C_{int} = \frac{\varepsilon_r W_{int}}{T_{ILD}} L_{int} \qquad (10)$$

Here, L_{int} is the length of the wire at the last stage of the H-tree.

We are going to consider the following variables to calculate T_{delay}:

- V_T: threshold voltage.
- t_{ox}: gate oxide thickness.
- L_{eff}: transistors effective length.
- V_{DD}: power supply voltage.
- T_{ILD}: interlevel dielectric thickness.
- W_{int}: wire width variation.
- t_{int}: wire thickness variation.

We also consider them independent. Therefore, variance of T_{delay} (σ_b^2) can be determined in terms of variances of these independent random variables.

$$\sigma_b^2 = \sigma_{T_{Delay}}^2 = \left(\frac{\partial T_{Delay}}{\partial V_T}\right)^2 \sigma_{V_T}^2 + \left(\frac{\partial T_{Delay}}{\partial t_{ox}}\right)^2 \sigma_{t_{ox}}^2 + \left(\frac{\partial T_{Delay}}{\partial L_{eff}}\right)^2 \sigma_{L_{eff}}^2 + \left(\frac{\partial T_{Delay}}{\partial V_{DD}}\right)^2 \sigma_{V_{DD}}^2 + \left(\frac{\partial T_{Delay}}{\partial T_{ILD}}\right)^2 \sigma_{T_{ILD}}^2 + \left(\frac{\partial T_{Delay}}{\partial W_{\text{int}}}\right)^2 \sigma_{W_{\text{int}}}^2 + \left(\frac{\partial T_{Delay}}{\partial t_{\text{int}}}\right)^2 \sigma_{t_{\text{int}}}^2 \quad (11)$$

where:

$$\begin{aligned}
\frac{\partial T_{Delay}}{\partial V_T} &= \frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial V_T} = 2.30\left(C_0 + C_{\text{int}}\right)\frac{R_0}{V_{DD} - V_T} \\
\frac{\partial T_{Delay}}{\partial t_{ox}} &= \frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial t_{ox}} + \frac{\partial T_{Delay}}{\partial C_0}\frac{\partial C_0}{\partial t_{ox}} = 2.30\left(C_0 + C_{\text{int}}\right)\frac{R_0}{t_{ox}} + 2.30\left(R_0 + R_{\text{int}}\right)\frac{C_0}{t_{ox}} \\
\frac{\partial T_{Delay}}{\partial L_{eff}} &= \frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial L_{eff}} + \frac{\partial T_{Delay}}{\partial C_0}\frac{\partial C_0}{\partial L_{eff}} = 2.30\left(C_0 + C_{\text{int}}\right)\frac{R_0}{L_{eff}} + 2.30\left(R_0 + R_{\text{int}}\right)\frac{C_0}{L_{eff}} \\
\frac{\partial T_{Delay}}{\partial V_{DD}} &= \frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial V_{DD}} = 2.30\left(C_0 + C_{\text{int}}\right)\frac{R_0}{V_{DD} - V_T} \\
\frac{\partial T_{Delay}}{\partial T_{ILD}} &= \frac{\partial T_{Delay}}{\partial C_{\text{int}}}\frac{\partial C_{\text{int}}}{\partial T_{ILD}} = \left(1.02R_{\text{int}} + 2.30R_0\right)\frac{C_{\text{int}}}{T_{ILD}} \\
\frac{\partial T_{Delay}}{\partial W_{\text{int}}} &= \frac{\partial T_{Delay}}{\partial R_{\text{int}}}\frac{\partial R_{\text{int}}}{\partial W_{\text{int}}} + \frac{\partial T_{Delay}}{\partial C_{\text{int}}}\frac{\partial C_{\text{int}}}{\partial W_{\text{int}}} = \left(1.02C_{\text{int}} + 2.30C_0\right)\frac{R_{\text{int}}}{W_{\text{int}}} + \left(1.02R_{\text{int}} + 2.30R_0\right)\frac{C_{\text{int}}}{W_{\text{int}}} \\
\frac{\partial T_{Delay}}{\partial t_{\text{int}}} &= \frac{\partial T_{Delay}}{\partial R_{\text{int}}}\frac{\partial R_{\text{int}}}{\partial t_{\text{int}}} = \left(1.02C_{\text{int}} + 2.30C_0\right)\frac{R_{\text{int}}}{t_{\text{int}}}
\end{aligned} \quad (12)$$

5.2.Metric tree

In the second model, we need the parameter σ_w^2 that is the delay variance through a wire of the clock distribution tree. Buffer delay is not necessary to be taken into account because, in this model, the wire effect dominates.

Using Sakurai's model for interconnection delays, the delay of a stage composed of a wire interconnecting two buffers (inverters) is:

$$T_{Delay} = 1.02 R_{int} C_{int} + 2.30\left(R_0 C_0 + R_0 C_{int} + R_{int} C_0\right) \quad (13)$$

where R_{int} and C_{int} are the total resistance and capacitance of the interconnecting wire, R_0 is the on-resistance of the driving transistor, and C_0 is the input capacitance of the driven inverter.

$$R_0 \approx \frac{L_{eff}/W}{\mu C_{ox}\left(V_{DD} - V_T\right)}, \quad C_o = C_{ox} \cdot W \cdot L_{eff}, \quad R_{int} = \frac{\rho}{W_{int} t_{int}} L_{int}, \quad C_{int} = \frac{\varepsilon_r W_{int}}{T_{ILD}} L_{int} \quad (14)$$

Here, L_{int} is the length of the wire at the last stage of the H-tree.

We are going to consider the following variables (Only the variables that affect to R_{int} and C_{int}, because we only consider the wire's delay) to calculate T_{delay}:

- T_{ILD}: interlevel dielectric thickness.
- W_{int}: wire width variation.
- t_{int}: wire thickness variation.

We also consider them independent. Therefore, variance of T_{delay} can be determined in terms of variances of these independent random variables.

$$\sigma_w^2 = \sigma_{T_{Delay}}^2 = \left(\frac{\partial T_{Delay}}{\partial T_{ILD}}\right)^2 \sigma_{T_{ILD}}^2 + \left(\frac{\partial T_{Delay}}{\partial W_{int}}\right)^2 \sigma_{W_{int}}^2 + \left(\frac{\partial T_{Delay}}{\partial t_{int}}\right)^2 \sigma_{t_{int}}^2 \quad (15)$$

where:

$$\begin{aligned}
\frac{\partial T_{Delay}}{\partial T_{ILD}} &= \frac{\partial T_{Delay}}{\partial C_{int}} \frac{\partial C_{int}}{\partial T_{ILD}} = \left(1.02 R_{int} + 2.30 R_0\right) \frac{C_{int}}{T_{ILD}} \\
\frac{\partial T_{Delay}}{\partial W_{int}} &= \frac{\partial T_{Delay}}{\partial R_{int}} \frac{\partial R_{int}}{\partial W_{int}} + \frac{\partial T_{Delay}}{\partial C_{int}} \frac{\partial C_{int}}{\partial W_{int}} = \left(1.02 C_{int} + 2.30 C_0\right) \frac{R_{int}}{W_{int}} + \left(1.02 R_{int} + 2.30 R_0\right) \frac{C_{int}}{W_{int}} \\
\frac{\partial T_{Delay}}{\partial t_{int}} &= \frac{\partial T_{Delay}}{\partial R_{int}} \frac{\partial R_{int}}{\partial t_{int}} = \left(1.02 C_{int} + 2.30 C_0\right) \frac{R_{int}}{t_{int}}
\end{aligned} \quad (16)$$

6. Conclusions

- The model is probabilistic.

- It give us an estimation of upper bounds for expected clock skew in tree distribution systems with *N* synchronously clocked processing elements.

- Three basic assumptions in our model:

  - The paths from the clock source to the processing elements are identical, in the sense that the paths are equal of length, contain the same number of wires and buffer stages, and equally loaded.

  - The clock arrival times are random variables, and are the sums of uncertain independent delays through many wires and buffers. By the Central Limit Theorem, the arrival time of clock signals at processors can be modelled asymptotically as a Gaussian random variable.

  - For the purpose of arriving at an upper bound on clock skew that is independent of topology, we make the additional assumption that any processor can communicate with any other.

- Results are applied to two different cases:

  - Metric-free model: Total delay in each buffer stage is Gaussian with a variance independent of stage number. Upper bound grows as *Θ(log N)*.

  - Metric model: Clock delay in a stage is Gaussian with a variance proportional to wire length. H-tree case. Upper bound grows $\Theta\left(\sqrt{\sqrt{N}\log N}\right)$.

- To apply the model it's necessary to know:

- Metric-free model
  - N: Number of processing elements.
  - σ_b^2: Delay variance through a buffer of the clock distribution tree (effect of the wire is absorbed by effect of the buffer).

- Metric model
  - N: Number of processing elements.
  - σ_w^2: Delay variance through a wire (effect of the buffers is not taken into account).

- We can calculate an upper bound of the clock skew for a H-tree clock distribution network taking into account the buffer delay. It is very recommendable, because the model doesn't consider a real case, where to increase the number of levels doesn't imply that the die size increases. To increase the number of levels supposes that all the branches are shorter, so their delay is reduced. It makes no sense the metric-free supposition that when $N \rightarrow \infty$, the wire term dominates.

  In this case, including buffer delays, the upper bound clock skew expression would be:

$$E[Skew] = \left(\sigma_b \log_2 N + \sigma_w 2\left(\sqrt{N} - 1\right)\right)\left[\frac{4\ln N - \ln\ln N - \ln 4\pi + 2C}{(2\ln N)^{1/2}} + O\left(\frac{1}{\ln N}\right)\right] \qquad (17)$$

  Now, to calculate σ_w^2, *we* have to calculate the variance of T_{Delay} as function of buffer parameter variations (V_T, t_{ox}, L_{eff}, and V_{DD}). To compute this variations, we do in the same way that in the metric-free model (equations 12).

- Parameters we need to know:

  - Interconnection resistance: R_{int}
  - Interconnection capacitance: C_{int}
  - On-resistance of the driving transistor: R_0

- Input capacitance of the driving inverter: C_0
- Threshold voltage of inverters: V_T
- Power supply voltage: V_{DD}

- Threshold voltage deviation (in %): σ_{VT}
- Power supply voltage deviation (in %): σ_{VDD}
- Gate oxide thickness deviation (in %): σ_{tox}
- Effective channel length deviation (in %): σ_{Leff}
- ILD thickness deviation (in %): σ_{TILD}
- Wire width deviation (in %): σ_{wint}
- Wire thickness deviation (in %): σ_{tint}

- Number of processing elements: N
- Lowest level branch length: L_{int}

Appendix 2

Clock Skew Model 2

Payman Zarkesh-Ha, Tony Mule', James D. Meindl [ZAR-98]

1. Introduction

Performance of high-speed synchronous digital systems is reduced significantly by clock skew of the clock distribution network. The concept of zero clock skew balanced networks has been proposed, however clock skew created by process parameter variations is still unavoidable. It is, therefore, imperative to characterize the clock skew components due to process parameter variations.

These authors describe a compact model to enable first-order estimation for on-chip clock skew as a function of device, interconnect and system parameter variations. Unlike previous models that describe qualitative behaviour of clock skew components, the new model provides a closed form expression for each clock skew component. This model provides a statistical expression (in function of both, process and design parameters, % of variation) for the clock skew in a balanced clock network (H-Tree), but this expression can be easily modified and applied to any clock network.

2. Clock skew components

Clock skew appears principally from unequal clock path lengths from the clock source to the clocked registers. To equalize line lengths, and thus reduce the clock skew, a common practice is the use of a balanced clock network. In this way the nominal value of skew becomes zero and clock skew reduces to the variations of the clock path from the clock generator to the registers.

These variations are originated by process and circuit parameter tolerances. We can distinguish the following parameter variations:

- ***Device Parameter Variations***

In the IC fabrication process, all device parameters are subject to deviations from their nominal values. Statistical models have been developed for transistor parameters such as **threshold voltage** (ΔV_T), **gate oxide thickness** (Δt_{ox}), and **effective channel length** (ΔL_{eff}).

- ***Interconnect Parameter Variations***

Interconnect width (ΔW_{int}) and **thickness** (Δt_{int}) and **interlevel dielectric thickness** (ΔT_{ILD}) variations are the main parameters of interest. As technology advances, the number of interconnect layers increases, and the interconnect lines become more non-uniform. This non-uniformity, which is caused by manufacturing processes, produces large variations of interconnect parameter values.

Chemical mechanical polishing (CMP) is the manufacturing process for planarization of metal and ILD layers that greatly reduces the ILD non-uniformity in multilayer structures. However, the CMP process still doesn't eliminate interconnect parameter variations completely.

- ***System Parameter Variations***

Besides process parameter variations, which are mainly the tolerances of device and interconnect physical parameters, system level fluctuations may create clock skew. **Power supply voltage fluctuation** (ΔV_{DD}), **temperature variations** ($\Delta \mathcal{T}$), and **non-uniform distribution of clocked registers** (ΔC_L) are considered as system level parameter variations.

3. Clock skew components

The goal of the derivation of a clock skew model is to understand the impact of process and system variations in an ideally zero skew clock network distribution. The most common strategy to ensure zero nominal clock skew, which is often used for distributing high frequency clock signals in digital systems, is a symmetric H-tree structure.

Although a model is derived here especially for the symmetric H-tree structure, the model can be easily modified for any balanced clock tree network.

3.1. Assumptions

Although the growing importance of on-chip transmission line effects has been predicted, the difficulty of modelling and simulating them, non-uniform transmission lines using existing CAD tools has prevented consideration of these effects in most cases. There are some methods, however, to reduce the inductance effects in actual design.

For example, there are clock distribution networks described, in which ground return path wiring has been implemented on the two metal levels above and below the clock wire to reduce inductance effects. To simplify the derivation of a clock skew model, good return path wiring has been assumed to surround the clock wiring network. Therefore, in this simplified study **the inductance effect is ignored**. Moreover, without lose of generality, it is assumed that **the clock network is a balanced H-tree structure**. This model, however, can be easily modified for any balanced clock tree network.

3.2. The Complete Clock Skew Model

Figure 1 shows a symmetric H-tree clock distribution with n=4 levels of H-tree branches. At the end of the 4th level, drivers are implemented to feed the clock signal to

all registers in the sub-blocks. It's important to underline that, in this model, the H-tree clock distribution system doesn't have any buffer at the split points where the i-level branch is divided in two (i+1)-level branches. Therefore, the wires width has to be designed so that i-level branch width has to be double as wide as (i+1)-level branch. It's necessary to avoid signal reflections in the split points.

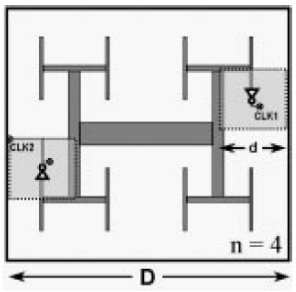


Figure 1: Symmetric H-tree structure.

The total clock skew, by definition, is the time difference between the maximum and minimum delays as illustrated in Figure 2.

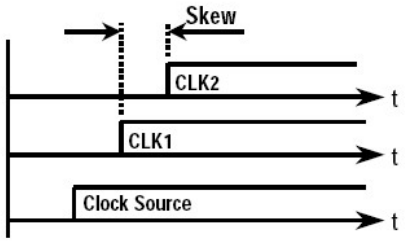


Figure 2: Clock skew between the points CLK1 and CLK2 (Fig 1).

Any H-tree circuit (Figure 1) can be simplified in the following equivalent circuit shown in Figure 3.

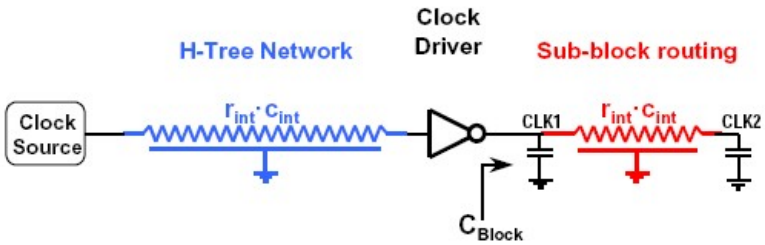


Figure 3: Equivalent circuit of clock H-tree network.

Using the equivalent circuit, the delay of the entire clock network of Figure 1 is divided into three parts:

- ***Interconnect delay from the clock source located at the centre of the chip through the H-tree to the driver***

Assuming that the H-tree network is driven by a single driver and the line capacitance of the H-tree network is much greater than the transistor input capacitance of a sub-block clock driver, then the interconnect delay expression for a distributed RC line using Sakurai's model (*50%* of time delay) is:

$$T_{H-tree} = 0.4 \cdot (r_{\text{int}} c_{\text{int}}) \cdot l^2 + \frac{\sqrt{\varepsilon_r}}{c_0} \cdot l \qquad (1)$$

where l is the length of the H-tree network, r_{int} and c_{int} are the distributed resistance and capacitance of the line, ε_r is the relative dielectric constant of the ILD material, and c_o is the speed of light in free space.

Because of the wires in clock distributions are often much wider than the minimum wire width, the fringing capacitance is negligible compared to parallel plate wiring capacitance. Using the expression for the length of the H-tree versus die size, D, and the number of H-tree levels, n, then (1) becomes:

$$r_{\text{int}} \cdot c_{\text{int}} = \frac{\rho \cdot \varepsilon_r}{t_{\text{int}} \cdot T_{ILD}}, \quad l = D \cdot \left(1 - \frac{1}{2^{n/2}}\right) \quad \Rightarrow$$

$$T_{H-tree} = 0.4 \cdot \left(\frac{\rho \cdot \varepsilon_r}{t_{\text{int}} \cdot T_{ILD}}\right) \cdot D^2 \cdot \left(1 - \frac{1}{2^{n/2}}\right)^2 + \frac{\sqrt{\varepsilon_r}}{c_0} \cdot D \cdot \left(1 - \frac{1}{2^{n/2}}\right) \qquad (2)$$

where t_{int} is interconnect thickness, T_{ILD} is interlevel dielectric thickness, and ρ is the line resistivity.

- ***Transistor delay of the sub-block clock driver***

The clocked registers within sub-blocks are assumed to be randomly placed and routed, therefore the delay expression for the time delay of the sub-block drivers is very simple (50% of time delay in Sakurai's Model):

$$T_{driver} = 0.7 \cdot R_0 \cdot C_L \qquad (3)$$

where R_0 is the average inverter resistance and C_L is the capacitive load. Using the expression for the output resistance of an inverter in saturation, expression (3) becomes:

$$T_{driver} = 0.7 \cdot \left(\frac{L_{eff} / W}{\mu \cdot C_{ox} \cdot (V_{DD} - V_T)} \right) \cdot C_L \qquad (4)$$

where, L_{eff} and W are transistor channel length and width respectively, μ is the mobility, C_{ox} is the gate oxide capacitance, and V_{DD} and V_T are supply and transistor threshold voltage respectively.

- ***Internal wire routing delay within the sub-block from the clock driver to registers***

The wiring delay inside the sub-block is computed in a similar way from (1) except that the length of wire, *l*, is the distance from centre to the corner of the sub-block ($d = D/2^{n/2}$).

$$T_{Sub-Blk} = 0.4 \cdot (r_{\text{int}-sub} c_{\text{int}-sub}) \cdot d^2 + \frac{\sqrt{\varepsilon_r}}{c_0} \cdot l = 0.4 \cdot \left(\frac{\rho \cdot \varepsilon_r}{t_{\text{int}-sub} \cdot T_{ILD-sub}} \right) \cdot \frac{D^2}{2^n} + \frac{\sqrt{\varepsilon_r}}{c_0} \cdot \frac{D}{2^{n/2}} \qquad (5)$$

Because of, in general, the placement of clocked registers is not uniform, the routing length inside the sub-block is not the same always. For example, in Figure 1 the clock signal at the point CLK2 arrives later than CLK1. This delay, which is often called ***internal clock skew***, in the worst case is given by (5):

The overall delay of the entire clock distribution network, from the clock source to the clocked registers, is $T_{Delay} = T_{H\text{-}tree} + T_{Driver} + T_{Sub\text{-}Blk}$. Since the sub-block size is often much less than the chip size, the wiring delay within the sub-block, $T_{Sub\text{-}Blk}$, can be ignored. Therefore the total delay is given by:

$$T_{Delay} \approx T_{H-tree} + T_{driver} = 0.4 \cdot \left(\frac{\rho \cdot \varepsilon_r}{t_{\text{int}} \cdot T_{ILD}} \right) \cdot D^2 \cdot \left(1 - \frac{1}{2^{n/2}} \right)^2 + \frac{\sqrt{\varepsilon_r}}{c_0} \cdot D \cdot \left(1 - \frac{1}{2^{n/2}} \right) + 0.7 \cdot \left(\frac{L_{eff} / W}{\mu \cdot C_{ox} \cdot (V_{DD} - V_T)} \right) \cdot C_L \quad (6)$$

Equation (6) contains all device, interconnect and system parameters described previously. Assuming that these parameters have small variations compared to their nominal values, the clock skew, T_{CSK}, can be evaluated by:

$$T_{CSK}(x) = \Delta T_{Delay} \approx \left| \frac{\partial T_{Delay}}{\partial x} \right| \Delta x \quad (7)$$

where T_{Delay} is the complete delay function of (6), and x is any variation of clock skew components such as ΔV_T, Δt_{ox}, ΔL_{eff} , ΔH_{int}, ΔT_{ILD}, ΔV_{DD}, $\Delta \mathcal{T}$ and ΔC_L. Table 1 shows the closed form equations for each individual clock skew component by using (7):

Physical parameter and derivation used	**Clock skew component**
Threshold voltage fluctuation	$T_{CSK}(V_T) = 0.7 \cdot R_0 \cdot C_L \cdot \left(\frac{V_T}{V_{DD} - V_T} \right) \cdot \frac{\Delta V_T}{V_T}$
Gate oxide thickness tolerance	$T_{CSK}(t_{ox}) = 0.7 \cdot R_0 \cdot C_L \cdot \frac{\Delta t_{ox}}{t_{ox}}$
Transistor channel length tolerance	$T_{CSK}(L_{eff}) = 0.7 \cdot R_0 \cdot C_L \cdot \frac{\Delta L_{eff}}{L_{eff}}$
Wire thickness variation	$T_{CSK}(t_{\text{int}}) = 0.4 \cdot (r_{\text{int}} \cdot c_{\text{int}}) \cdot D^2 \cdot \left(1 - \frac{1}{2^{n/2}} \right)^2 \cdot \frac{\Delta t_{\text{int}}}{t_{\text{int}}}$
ILD thickness variation	$T_{CSK}(T_{ILD}) = 0.4 \cdot (r_{\text{int}} \cdot c_{\text{int}}) \cdot D^2 \cdot \left(1 - \frac{1}{2^{n/2}} \right)^2 \cdot \frac{\Delta T_{ILD}}{T_{ILD}}$

IR drop	$T_{CSK}(V_{DD}) = 0.7 \cdot R_0 \cdot C_L \cdot \left(\frac{V_{DD}}{V_{DD} - V_T} \right) \cdot \frac{\Delta V_{DD}}{V_{DD}}$
Non uniform register distribution	$T_{CSK}(C_L) = 0.7 \cdot R_0 \cdot C_L \cdot \frac{\Delta C_L}{C_L}$
Temperature gradient	$T_{CSK}(\mathrm{T}) = 0.7 \cdot R_0 \cdot C_L \cdot \left(\frac{E_g / q + V_T}{V_{DD} - V_T} \right) \cdot \frac{\Delta \mathrm{T}}{\mathrm{T}}$
Internal Clock skew	$T_{CSK}(sub) = 0.4 \cdot (r_{\mathrm{int}-sub} c_{\mathrm{int}-sub}) \cdot \frac{D^2}{2^n} + \frac{\sqrt{\varepsilon_r}}{c_0} \cdot \frac{D}{2^{n/2}}$

Table 1: Clock skew components.

3.3. Clock Skew for Temperature Variation

The clock skew due to temperature gradient on a chip, in general, is more complex since there are three main parameters that vary with temperature: resistivity of interconnect $\rho(\mathcal{T})$, threshold voltage $V_T(\mathcal{T})$, and mobility $\mu(\mathcal{T})$. Assuming that the variation of threshold voltage is greater than that of mobility and resistivity of lines, then the clock skew due to temperature difference is given by:

$$T_{CSK}(\mathrm{T}) = \Delta T_{Delay}(\mathrm{T}) = \left| \frac{\partial T_{Delay}}{\partial V_T} \cdot \frac{\partial V_T}{\partial \mathrm{T}} \right| \cdot \Delta \mathrm{T} \qquad (8)$$

where $\Delta\mathcal{T}$ is temperature difference of two points in the chip. The first expression, $\partial \mathrm{T}_{\mathrm{Delay}} / \partial \mathrm{V}_{\mathrm{T}}$, is computed from (6) as:

$$\frac{\partial T_{Delay}}{\partial V_T} = 0.7 \cdot R_0 \cdot C_L \cdot \frac{1}{V_{DD} - V_T} \qquad (9)$$

Also the second expression, $\partial\, \mathrm{V_T} / \partial\mathcal{T}$, is:

$$\frac{\partial V_T}{\partial \mathrm{T}} = \frac{1}{\mathrm{T}} \cdot \left(2 - \frac{Q_B}{2 C_{ox} \varphi_f} \right) \cdot \left(\varphi_f + \frac{E_g}{2q} \right) \qquad (10)$$

where Q_B is the depletion-region charge, C_{ox} is the gate oxide capacitance, φ_f is the Fermi level potential, E_g is the energy gap of Si, and q is the charge of electron. In order to simplify (10), the threshold voltage can be written as $V_T \approx \varphi_{Si} - Q_B/C_{ox}$. Moreover assuming that the substrate doping concentration is relatively high, then the surface potential of MOSFET transistor is given by $\varphi_{Si} = \varphi_f \approx 2(E_g/2)$. Therefore the first order approximation of (9) is given by:

$$\frac{\partial V_T}{\partial \mathrm{T}} = \frac{E_g / q + V_T}{\mathrm{T}} \qquad (11)$$

where $Eg/q = 1.12\ V$ is the energy gap of Si in volts, and $\mathcal{T}$ is the temperature in degrees Kelvin. Equation (8) along with the results of (9) and (11) give:

$$T_{Delay}(\mathrm{T}) = 0.7 \cdot R_{tr} \cdot C_L \cdot \left(\frac{E_g / q + V_T}{V_{DD} - V_T} \right) \cdot \frac{\Delta \mathrm{T}}{\mathrm{T}} \qquad (12)$$

4. Example

In order to illustrate the new clock skew model, a design example for 0.18 μm technology has been studied using the design parameters illustrated in Table 2. The H-tree clock distribution is assumed to be routed by the 4th metal level shielded with the 3rd and 5th metal levels as shown in Figure 4. This wiring structure ensures minimal inductive effect.

Parameters		Values
Process parameters	L_{eff}	0.18 μm
	V_T	0.32 V
	V_{DD}	1.8 V
	$r_{int}c_{int}$	115 ps/cm^2
Design parameters	R_0	12.0 Ω
	C_L	6.25 pF
	D	2.0 cm
	n	4

Table 2: Process and design parameters.

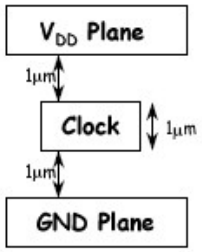


Figure 4: Clock distribution wiring structure.

Equation (1) is, therefore, valid for interconnect propagation delay with $r_{int}c_{int}=115$ *ps/cm*^{2}, assuming that Cu/SiO_2 materials are used in wiring network. In this example, the total number of clocked registers is set to *20,000*. With an input register capacitance of *5fF*, the total capacitance of each sub-block (2^4 = 16 in total), on average, is computed as $C_L=6.25$ *pF*. A driver with $R_0=12.0\Omega$ output resistance is selected to ensure fast rise time for the total sub-block loading capacitance.

Parameters	% of variation	Clock Skew Component [ps]
V_T	5 %	0.55
t_{ox}	1.2 %	0.64
L_{eff}	5 %	2.65
H_{int}	3 %	3.12
T_{ILD}	3 %	3.12
V_{DD}	10 %	6.40
C_L	20 %	10.60
τ	8 %	4.08
Internal Clock Skew [ps]		61.7
Total Clock Skew [ps]		92.9

Table 3: Clock Skew Components.

Using the expressions of Table I, the complete set of clock skew components are evaluated as shown in Table 3. The second column contains percentage of variations, which are technology and design dependent. It means the following: if % of variation of V_T is *5%*, then ΔV_T is $0.05V_T$.

The tolerances of interconnect and ILD thickness is roughly *3%* for a well-controlled CMP process. Statistical modelling extracted from the measurement data of

0.18μm technology shows that the fluctuations of threshold voltage, MOS effective channel length, and gate oxide thickness are about *5%*, *5%* and *1.2%* respectively. The tolerance of supply voltage is usually limited to *10%* of total supply voltage. We are going to suppose a *30ºC* temperature gradient over the entire chip (as the thermal image of the Alpha microprocessor), which gives a temperature variation of about *8%*. The variations of loading capacitance of clock drivers highly depend on the uniformity of architecture. A specific investigation of a microprocessor design shows that about *20%* variation exists on the sub-block loading capacitance.

The clock skew components are evaluated based on the amount of variations in the third column of Table 3. Figure 5 illustrates a graphical view of clock skew components. A skew of *10.6ps* is created just by clock driver load mismatch. Also, IR drop, temperature gradient, interconnect and ILD thickness variations, and MOS channel length tolerance create *6.4ps*, *4.08ps*, *3.12ps* and *2.65ps* respectively.

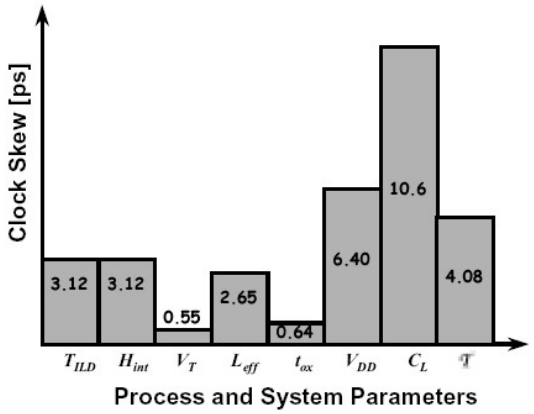


Figure 5: Clock skew components.

5. Conclusions

- The model provides a statistical expression for the clock skew in a H-Tree clock network, where there aren't intermediate buffers at the split points.

- Its easily applicable to any clock network, modifying the expressions obtained.

- It's a statistical model because the total clock skew is obtained in function of both, process and design parameters, % of variation.

- To ignore the inductance effects between wires, system design have to include a good return path wiring surrounding the clock-wiring network.

- To apply the model it's necessary to know:

  - Process parameters:
    - Interconnection parameters: $r_{int}c_{int}$
    - Threshold voltage of inverters: V_T
    - Power supply voltage: V_{DD}
    - Transistors energy gap: E_g

  - Design parameters:
    - Output resistance of driving buffers: R_0
    - Die size: D
    - H-tree levels: n
    - Capacitive load of sub-blocks: C_L
    - Relative dielectric constant of ILD material (only to calculate sub-block delay): ε_r

  - Parameter variations (in %):
    - Threshold voltage: σ_{VT}
    - Gate oxide thickness: σ_{tox}
    - Transistor effective channel length: σ_{Leff}
    - Wire thickness: σ_{tint}
    - ILD thickness: σ_{TILD}
    - Power supply voltage: σ_{VDD}
    - Load capacitance: σ_{CL}
    - Temperature: $\sigma_{\mathcal{T}}$

- The model equations can be easily modified to be more similar to other models, where the Sakurai's expressions are used with 90% of time delay. It only supposes to change the coefficients of $T_{H\text{-}tree}$, T_{Driver} and $T_{Sub\text{-}Blk}$.

  - $T_{H-tree}: \quad 0.4 \rightarrow 1.02 \Rightarrow T_{H-tree} = 1.02 \cdot (r_{\text{int}} c_{\text{int}}) \cdot l^2 + \frac{\sqrt{\varepsilon_r}}{c_0} \cdot l$
  - $T_{driver}: \quad 0.7 \rightarrow 2.3 \Rightarrow T_{driver} = 2.3 \cdot R_0 \cdot C_L$
  - $T_{Sub-Blk}: \quad 0.4 \rightarrow 1.02 \Rightarrow T_{Sub-Blk} = 1.02 \cdot (r_{\text{int}-sub} c_{\text{int}-sub}) \cdot d^2 + \frac{\sqrt{\varepsilon_r}}{c_0} \cdot l$

Appendix 3

Clock Skew Model

Xiaohong Jiang and Susumu Horiguchi [JIA-01]

1. Introduction

The evolution of VLSI chips toward larger die sizes and faster clock speeds makes clock distribution an increasingly important issue. Clock skew modelling is important in the performance evaluation and prediction of clock distribution networks, because at high speeds clock skew becomes a very significant problem.

Clock skew may arise mainly from unequal clock path lengths to various modules and process variations that cause clock path delay variations. There are worst-case and statistical skew models not suitable for modelling the clock skews of general clock distribution networks in which clock paths are not identical.

The worst-case approach (Kung and Fisher model) can usually cause an unnecessarily long clock period. The statistical models handle the problem from the point of view that all clock paths are assumed to be identical and independent. Kugelmass and Steiglitz model predicts an upper bound of expected clock skew. Zarkesh-Ha, Mule' and Meindl model is too conservative for estimating the clock skew of a well-balanced clock network that has identical but strongly correlated clock paths (for example, a well-balanced H-tree).

In order to provide a more accurate and more general statistical skew model for general clock (in which clock paths can be not identical), these authors propose a new approach to estimate the mean value and variance of clock skew of general clock distribution networks. Based on the new approach, a closed-form model is also obtained for well-balanced H-tree clock distribution networks. The paths delay correlation caused

by the overlapped parts of path lengths is considered in the new approach, so the mean value and the variance of clock is accurately estimated for general clock distribution networks.

2. Clock skew modelling

For a given CDN (clock distribution network), let $t(l_o, l_i)$ denote the signal propagation time on the unique path from the clock source l_o to the sink l_i. The maximal clock delay ξ and the minimal clock delay η of the CDN can be defined as:

$$\xi = \max_i \{t(l_0, l_i)\} \qquad (1)$$

$$\eta = \min_i \{t(l_0, l_i)\} \qquad (2)$$

The clock skew between two sinks l_i and l_j is the delay difference $|t(l_0, l_i) - t(l_0, l_j)|$ and clock skew χ of the CDN is in general defined as the maximum value of $|t(l_0, l_i) - t(l_0, l_j)|$ over all sink pairs l_i and l_j and in the CDN. Thus, χ is given by

$$\chi = \max_{i \neq j} \left| t(l_0, l_i) - t(l_0, l_j) \right| = \xi - \eta \qquad (3)$$

Process variations are subject to two sets of factors: systematic factors, like power supply fluctuations, which can be controlled by proper techniques and factors that are random, and therefore uncontrollable by improved techniques. Therefore, the random factors determine the achievable performance of a circuit. We want to model the clock skew of general CDNs when the random factors are considered.

When random process variations are considered, variations of paths delay are modelled by normal distributions. To model the clock skew χ, random variables ξ and η should be first characterized. The model is based on the following two assumptions:

- ***Assumption 1:*** A CDN can in general be represented by a binary tree. We assume that both the maximal clock delay and the minimal clock delay in each subtree (and

also the whole binary tree) of the CDN can be modelled by normal distributions when process variations are considered. The assumption makes it easy to analyze the correlation that exists between the maximal and the minimal delay in a subtree. This correlation analysis is critical in determining the variance of skew in each subtree (and also the whole binary tree) of the CDN.

- ***Assumption 2:*** The delay along a clock path is the sum of the uncertain independent delays of the branches along the given path. Correlation between the delay of any two paths is determined only by the overlapped parts of their length.

The clock paths of a CDN usually have some common branches over their length, and these common branches cause correlation among the delays of these paths. The above assumption enables a complete analysis of this kind of correlation.

In addition to the delay correlation described in Assumption 2, the correlation among paths delay may also be caused by the correlated intra-die variations of these parameters involved in that delay (e.g., threshold voltages, resistances, etc.). However, finding the correlation coefficient of these parameters is, in practice, quite uncomfortable and difficult. So authors neglect these kinds of correlation in as indicated in Assumption 2. In general, the intra-die process parameters' correlation will lead to the paths delay in the same chip tending to be positive dependent. In this case, Assumption 2 will guarantee that the expected value of clock skew will still be upper bounded by the corresponding values estimated using our approach.

Compared to the old upper bound of expected skew of a well-balanced CDN where all the clock paths are assumed completely independent, our estimates are enhanced significantly, because the paths delay correlation caused by the common branches of paths length are completely considered. Furthermore, the new approach is applicable to general CDNs, whereas the old models is only applicable to the well-balanced CDNs in which clock paths are identical.

From (3), the mean value and the variance of χ are given by:

$$E(\chi) = E(\xi) - E(\eta) \quad (4)$$

$$D(\chi) = D(\xi) + D(\eta) - 2\rho\sqrt{D(\xi) \cdot D(\eta)} \quad (5)$$

Here, $E(\cdot)$ and $D(\cdot)$ represent the mean value and the variance of a random variable, respectively, and ρ is the correlation coefficient of ξ and η. The parameters $E(\xi)$, $E(\eta)$, $D(\xi)$, $E(\eta)$ and ρ should be accurately estimated for a CDN to allow the accurate modelling of clock skew.

3. Parameter estimation

A recursive approach for evaluating the parameters $E(\xi)$, $E(\eta)$, $D(\xi)$, $D(\eta)$ and ρ of general CDNs is presented here. Based on this algorithm, closed-form expressions of clock skews and the maximal clock delay of well-balanced *H*-tree CDNs are also developed.

3.1. Algorithm for general CDNs

A CDN can generally be represented by a binary tree, so a simplified binary tree shown in Figure 1 is taken as an example to illustrate the evaluating process of these parameters. The evaluating process is then applied to general CDNs. All the paths in Figure 1 are partitioned into independent branches s_0, s_1, s_2, … by the branch split points in the clock tree, where d_i is the actual delay of s_i. The branch split point (i, j) in the clock tree is associated with a set of random variables $(\xi_{ij}, \eta_{ij}, \chi_{ij})$, here ξ_{ij}, η_{ij} and χ_{ij} are the maximal clock delay, the minimal clock delay and the clock skew of the subtree starting from the split point, respectively. Each random variable here is characterized by both its mean value and its variance.

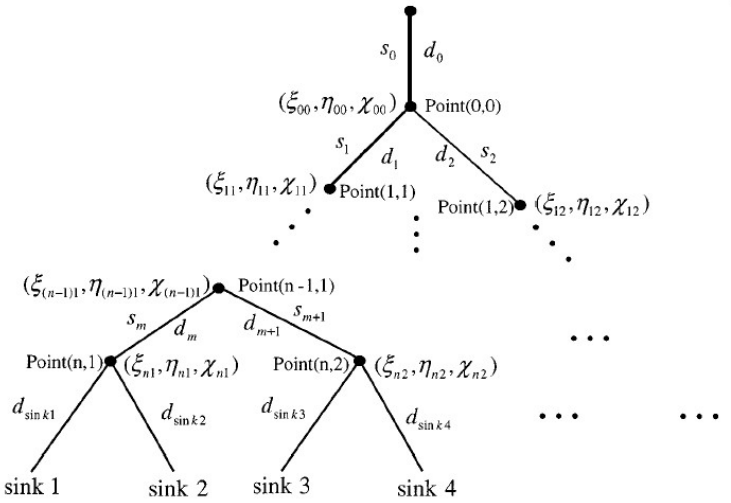


Figure 1: Binary clock tree.

To illustrate that the parameters $E(\xi)$, $E(\eta)$, $D(\xi)$, $D(\eta)$ and ρ of the simplified binary clock tree can be evaluated recursively, we begin with the evaluating process of $(\xi_{00},\eta_{00},\chi_{00})$. Let branch s_i also be associated with a set of random variables, (ξ_i, η_i, ρ_i), with ξ_i being the maximal clock delay and η_i being the minimal clock delay of the subtree starting from branch s_i, and being ρ_i the correlation coefficient of ξ_i and η_i. Thus

$$\begin{aligned} \xi_1 &= \xi_{11} + d_1, \quad \xi_2 = \xi_{12} + d_2 \\ \eta_1 &= \eta_{11} + d_1, \quad \eta_2 = \eta_{12} + d_2 \end{aligned} \tag{6}$$

$$\begin{aligned} \rho_1 &= \frac{D(\xi_1) + D(\eta_1) - D(\chi_{11})}{2\sqrt{D(\xi_1)\cdot D(\eta_1)}} \\ \rho_2 &= \frac{D(\xi_2) + D(\eta_2) - D(\chi_{12})}{2\sqrt{D(\xi_2)\cdot D(\eta_2)}} \end{aligned} \tag{7}$$

Then we have:

$$\begin{aligned} \xi_{00} &= \max\{\xi_1, \xi_2\} = \frac{\xi_1 + \xi_2 + |\xi_1 - \xi_2|}{2} \\ \eta_{00} &= \min\{\eta_1, \eta_2\} = \frac{\eta_1 + \eta_2 - |\eta_1 - \eta_2|}{2} \end{aligned} \tag{8}$$

$$\chi_{00} = \xi_{00} - \eta_{00} \tag{9}$$

Equations (6)–(9) indicate that the results of branch split point *(0, 0)* are determined by and can be evaluated from both the corresponding results [$(\xi_{11}, \eta_{11}, \chi_{11})$ and $(\xi_{12}, \eta_{12}, \chi_{12})$] of the next lower level split points *(1, 1), (1, 2),* and the delay of the branches [s_1 and s_2] connecting the point to those next lower level split points. So once the results of $(\xi_{ij}, \eta_{ij}, \chi_{ij})$ are obtained for each lowest-level split point (i.e., the split point from which no further branch split points can be found in the subtree starting from that split point – points *(n, 1), (n, 2), …*), the process above can be used recursively to evaluate the mean values and the variances of clock skew, the maximal clock delay and the minimal clock delay of a general CDN in a bottom-up manner.

In fact, the results of $(\xi_{n1}, \eta_{n1}, \chi_{n1})$ of one lowest-level split point in Figure 1 can be obtained as follows: The mean values and the variances of ξ_{n1} and η_{n1} can be evaluated by using their distribution functions, respectively. The mean value and the variance of χ_{n1} are given by:

$$\begin{aligned} E(\chi_{n1}) &= E(\xi_{n1} - \eta_{n1}) = E\left(\left|d_{\sin k1} - d_{\sin k2}\right|\right) \\ &= \frac{2\sqrt{D_{\sin k}}}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{E_{\sin k}}{\sqrt{D_{\sin k}}}\right)^2\right) + \frac{2\left|E_{\sin k}\right|}{\sqrt{2\pi}} \int_0^{\left|E_{\sin k}\sqrt{D_{\sin k}}\right|} \exp\left(-\frac{1}{2}t^2\right) dt \end{aligned} \tag{10}$$

$$D(\chi_{n1}) = D\left(\left|d_{\sin k1} - d_{\sin k2}\right|\right) = E_{\sin k}^2 + D_{\sin k} - \left[E(\chi_{n1})\right]^2 \tag{11}$$

where

$$\begin{aligned} E_{\sin k} &= E(d_{\sin k1}) - E(d_{\sin k2}) \\ D_{\sin k} &= D(d_{\sin k1}) + D(d_{\sin k2}) \end{aligned} \tag{12}$$

Based on the results of $(\xi_{00}, \eta_{00}, \chi_{00})$, the parameters, $E(\xi)$, $E(\eta)$, $D(\xi)$, $D(\eta)$ and ρ of the whole binary tree are then given by:

$$\begin{aligned} E(\xi) &= E(\xi_{00}) + E(d_0), \quad E(\eta) = E(\eta_{00}) + E(d_0) \\ D(\xi) &= D(\xi_{00}) + D(d_0), \quad D(\eta) = D(\eta_{00}) + D(d_0) \end{aligned} \tag{13}$$

$$\rho = \frac{D(\xi) + D(\eta) - D(\chi_{00})}{2\sqrt{D(\xi) \cdot D(\eta)}} \tag{14}$$

The pseudocode for the parameter estimation algorithm is the following:

- **Parameter estimation for general CDNs**

```
{
Initialization:
      for each Lp ∈ V do
      {
            ξ^b ← d^b
            η^b ← d^b
            ρ^b ← 1
      }
Algorithm:
      while (V not empty) do
      {
            for each Lp ∈ V do
            {
                  p_{ξ^LP}(x) = p(ξ_1^b < x) · p(ξ_2^b < x)
                  p_{η^LP}(x) = 1 − p(η_1^b > x) · p(η_2^b > x)
                  E(ξ^LP) = ∫_{−∞}^{+∞} x · d(p_{ξ^LP}(x))
                  D(ξ^LP) = ∫_{−∞}^{+∞} [x − E(ξ^LP)]^2 · d(p_{ξ^LP}(x))
                  E(η^LP) = ∫_{−∞}^{+∞} x · d(p_{η^LP}(x))
                  D(η^LP) = ∫_{−∞}^{+∞} [x − E(η^LP)]^2 · d(p_{η^LP}(x))
                  Cov(ξ^LP, η^LP) = Cov((ξ_1^b + ξ_2^b + |ξ_1^b − ξ_2^b|)/2, (η_1^b + η_2^b − |η_1^b − η_2^b|)/2)
                  E(χ^LP) = E(ξ^LP) − E(η^LP)
                  D(χ^LP) = D(ξ^LP) + D(η^LP) − 2 · Cov(ξ^LP, η^LP)

                  Remove G^LP = (V^LP, E^LP) from G = (v, E)
```

$$\xi^b \leftarrow \xi^{LP} + d^{LP}$$
$$\eta^b \leftarrow \eta^{LP} + d^{LP}$$
$$E\left(\xi^b\right) \leftarrow E\left(\xi^{LP}\right) + E\left(d^{LP}\right)$$
$$E\left(\eta^b\right) \leftarrow E\left(\eta^{LP}\right) + E\left(d^{LP}\right)$$
$$D\left(\xi^b\right) \leftarrow D\left(\xi^{LP}\right) + D\left(d^{LP}\right)$$
$$D\left(\eta^b\right) \leftarrow D\left(\eta^{LP}\right) + D\left(d^{LP}\right)$$
$$\rho^b \leftarrow \frac{D\left(\xi^b\right) + D\left(\eta^b\right) - D\left(\chi^{LP}\right)}{2\sqrt{D\left(\xi^b\right) \cdot D\left(\eta^b\right)}}$$

```
            }
        }
}
```

In the pseudocode, a CDN is represented by graph $G = (V, E)$ with vertex (split point) set V and edge (branch) set E. The lowest level split point L_P of the graph is associated with random variables $(\xi^{LP},\eta^{LP},\chi^{LP})$, as defined above. (ξ^b,η^b,χ^b), are the random variables associated with the branches starting from a lowest level splitting point, with $(\xi_1^b,\eta_1^b,\chi_1^b)$ and $(\xi_2^b,\eta_2^b,\chi_2^b)$ representing the random variables associated with the two branches starting from L_P, respectively. For a CDN, the initial values of ξ^b and η^b are just the actual delay d^b of the branches that support sinks. d^{Lp} is the actual delay of the branch L_P connecting to its parent splitting point, and $G^{Lp} = (V^{LP}, E^{LP})$ is the subgraph starting from L_P.

Since the algorithm carries out the same amount of computation for each split point, the following conclusion can be obtained:

***Theorem 1**: "The parameter estimation algorithm given above computes a network G=(V, E) in O(|V|) time".*

The theorem indicates that the parameter estimation algorithm is computationally effective in estimating the parameters, $E(\xi)$, $E(\eta)$, $D(\xi)$, $D(\eta)$ and ρ of a general CDN.

3.2. Clock skew estimation for H-tree CDNs

The H-tree technique is widely used to reduce the clock skew. Due to the very symmetric structure of *H*-tree CDNs, it is possible for us to get a closed form model for clock skew and the maximal clock delay of *H*-tree CDNs.

Before developing the models, the *H*-tree itself must first be defined. Without loss of generality, a well-balanced *H*-tree has *n* hierarchical levels, where *n* denotes the tree depth. The level *0* branch corresponds to the root branch, and level *n* branches to the branches that support sinks. A level *i* branch begins with a level split *i* point and ends with level *i+1* split point. The *H*-tree illustrated in Figure 2 is drawn for *n=6*, which is used to distribute the clock signals to *64* processors.

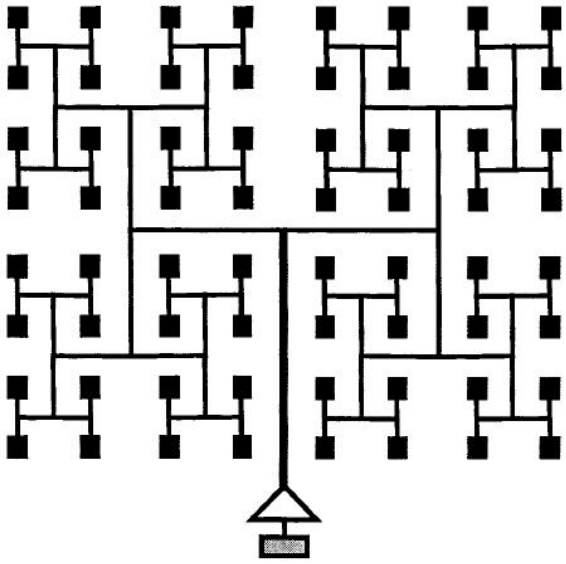


Figure 2: A well-balanced *H*-tree clock distribution network for 64 processors

For a *n* level well-balanced *H*-tree, let d_i, $i=0, \ldots, n$ be the actual delay of branch *i* of a clock path. The mean values and the variances of the maximal clock delay ξ and the minimal clock delay, η, of the *H*-tree are then given by following equations:

$$E(\xi) = \sum_{i=0}^{n} E(d_i) + \frac{1}{\sqrt{\pi}} \sum_{i=1}^{n} \sqrt{\sum_{k=1}^{i} \left(\frac{\pi - 1}{\pi} \right)^{k-1} \cdot D(d_{n-i+k})} \qquad (15)$$

$$E(\eta) = \sum_{i=0}^{n} E(d_i) - \frac{1}{\sqrt{\pi}} \sum_{i=1}^{n} \sqrt{\sum_{k=1}^{i} \left(\frac{\pi - 1}{\pi}\right)^{k-1} \cdot D(d_{n-i+k})} \tag{16}$$

$$D(\xi) = D(\eta) = \sum_{i=0}^{n} \left(\frac{\pi - 1}{\pi}\right)^{i} \cdot D(d_i) \tag{17}$$

Results (15)–(17) and (3) indicate that the expected clock skew $E(\chi)$ and skew variance $D(\chi)$ of the n level well-balanced H-tree are given by:

$$E(\chi) = E(\xi) - E(\eta) = \frac{2}{\sqrt{\pi}} \sum_{i=1}^{n} \sqrt{\sum_{k=1}^{i} \left(\frac{\pi - 1}{\pi}\right)^{k-1} \cdot D(d_{n-i+k})} \tag{18}$$

$$\begin{aligned} D(\chi) &= D(\xi) + D(\eta) - 2 \cdot \rho \cdot \sqrt{D(\xi) \cdot D(\eta)} = \\ &= 2 \cdot (1 - \rho) \cdot \sum_{i=0}^{n} \left(\frac{\pi - 1}{\pi}\right)^{i} \cdot D(d_i) \end{aligned} \tag{19}$$

where ρ is the correlation coefficient of ξ and η, and ρ can be recursively evaluated for a network as discussed in Section 3.1. The closed-form expressions (15)–(19) indicate clearly how the clock skew is accumulated along the clock paths and with the increase of H-tree size. This enables a suitable H-tree size to be selected for a specified clock frequency, and also enables minimization of the clock period, improving the speed for a fixed size H-tree network.

3.3. Yield of clock skew model

The clock period of a CDN is in general determined by the clock skew of the network. With the estimates of mean values and variances χ in hand, it is possible for us to estimate its yield. Here, the yield of a random variable means the probability that the variable is less than a specified value. For general CDNs, ξ and η are positively correlated (i.e., $\rho>0$) normal variables, and clock skew χ can be modelled by log-normal distribution as verified by available extensive simulation results. The clock skew yield, i.e., the probability that the actual skew of the network, χ, is less than a skew specification x ($P(\chi<x)$), can then be evaluated as:

$$P(\chi < x) = \int_0^x \frac{\log e}{\sqrt{2\pi} \cdot \delta_1 \cdot t} \exp\left[-\frac{1}{2}\left(\frac{\log t - \mu_1}{\delta_1} \right)^2 dt \right] \quad (20)$$

where parameters μ_1 and δ_1 are given by:

$$\mu_1 = \log\left(\frac{\left[E(\chi)\right]^2}{\sqrt{D(\chi) + \left[E(\chi)\right]^2}} \right) \quad (21)$$

$$\delta_1 = \sqrt{\log e \cdot \log\left(\frac{D(\chi) + \left[E(\chi)\right]^2}{\left[E(\chi)\right]^2} \right)} \quad (22)$$

Once the algorithm developed in Section 3 estimates the mean value of χ of a CDN, the yield of χ can be approximated by a lognormal distribution.

4. Clock skew calculation in function of its components

4.1. Calculation

The delay of a branch may then be obtained by averaging the rise and fall times. Using Sakurai's model for interconnection delays (90 % of time delay), the delay of a stage composed of a wire interconnecting two buffers (inverters) is:

$$T_{Delay} = 1.02 R_{int} C_{int} + 2.30\left(R_0 C_0 + R_0 C_{int} + R_{int} C_0 \right) \quad (23)$$

Here R_0 is the output resistance of the driving transistor of minimum size inverter, C_0 is the input capacitance of the driven minimum size inverter, C_{int} and R_{int} are the capacitance and resistance of the interconnection line in the branch. Their expressions are given by:

$$
\begin{aligned}
&R_0 = \frac{1}{K \cdot (V_{DD} - V_T)}, \quad K = \frac{\mu \cdot C_{ox} \cdot W}{L_{eff}}, \quad C_{ox} = \frac{\varepsilon_{ox}}{t_{ox}} \\
&C_0 = C_{ox} \cdot W \cdot L_{eff} \\
&R_{\text{int}} = \frac{\rho}{W_{\text{int}} \cdot t_{\text{int}}} \cdot L_{\text{int}} \\
&C_{\text{int}} = \varepsilon_{ILD} \cdot \frac{W_{\text{int}}}{T_{ILD}} \cdot L_{\text{int}}
\end{aligned}
\tag{24}
$$

where:

- W and L_{eff}: width and effective length of the transistor.
- C_{ox}: gate unit area capacitance.
- t_{ox}: gate oxide thickness.
- μ: charge carrier mobility.
- V_T: threshold voltage.
- ρ: metal resistivity.
- ε_{ox}: oxide dielectric constant.
- ε_{ILD}: interlevel dielectric constant.
- W_{int}, L_{int} and t_{int}: width, length and thickness of the interconnection line.
- T_{ILD}: Interlevel dielectric thickness.

In the C_{int} expression, we don't take into account the contribution of fringing fields.

The process parameters and their standard deviations used here are based on the *0.25 μm* CMOS technology predicted by the International Technology Roadmap for Semiconductors (ITRS) and the MOSIS parametric test results of a typical *0.25 μm* technology. The mean values and intra-die standard deviations (SD) of these process parameters are presented in Table 1.

Parameters	**Mean**	**SD**
L_{eff} *(μm)*	0.25	0.0075
V_{DD} *(V)*	2.5	0
V_{TN} *(V)*	0.51	0.02
V_{TP} *(V)*	-0.51	0.02

t_{ox} (A^o)	50	0.65
μ_n ($cm^2/V{\cdot}s$)	391	7.82
μ_p ($cm^2/V{\cdot}s$)	122	2.44
t_{int} (μm)	0.1	0.0013
T_{ILD} (μm)	0.1	0.0013

Table 1: *0.25 μm* Process parameters (mean and standard deviation).

Here, V_{DD} is not considered as a random variable since the power supply in a system is globally controlled. Furthermore, the standard deviation of the width of a transistor is 0.02 μm for n-MOS and 0.05 μm for p-MOS as estimated from MOSIS. The n-MOS transistor and p-MOS transistor in the minimum inverter of the technology are assumed to have the gate width/length of *0.37 μm /0.25 μm* and 1.1 μm /0.25 μm, respectively.

As indicated in Assumption 2, the intra-die parameters correlations are neglected in this model. Thus, R_0 in (24) will be independent from C_0. One approach to calculating the delay variance of a branch due to the variations of process parameters is to first express the relations (24) in terms of independent variables. The delay variance of the branch can then be determined in terms of variances of these independent random variables. For example, the variance, σ_z^2 , of a random variable z that is a function of independent random variables, $z=f(x,y,\ldots)$, may be obtained from:

$$\sigma_z^2 = \left(\frac{\partial f}{\partial x}\right)^2 \cdot \sigma_x^2 + \left(\frac{\partial f}{\partial y}\right)^2 \cdot \sigma_y^2 + \ldots \qquad (25)$$

We are going to consider the following variables to calculate the variance of T_{delay}: V_T, μ, t_{ox}, L_{eff}, W, T_{ILD}, W_{int}, t_{int}.

Therefore, the variance of the delay in a branch is the following:

$$\sigma_{T_{Delay}}^2 = \left(\frac{\partial T_{Delay}}{\partial V_T}\right)^2 \sigma_{V_T}^2 + \left(\frac{\partial T_{Delay}}{\partial \mu}\right)^2 \sigma_{\mu}^2 + \left(\frac{\partial T_{Delay}}{\partial t_{ox}}\right)^2 \sigma_{t_{ox}}^2 + \left(\frac{\partial T_{Delay}}{\partial L_{eff}}\right)^2 \sigma_{L_{eff}}^2 + \left(\frac{\partial T_{Delay}}{\partial W}\right)^2 \sigma_W^2 + \left(\frac{\partial T_{Delay}}{\partial T_{ILD}}\right)^2 \sigma_{T_{ILD}}^2 + \left(\frac{\partial T_{Delay}}{\partial W_{\text{int}}}\right)^2 \sigma_{W_{\text{int}}}^2 + \left(\frac{\partial T_{Delay}}{\partial t_{\text{int}}}\right)^2 \sigma_{t_{\text{int}}}^2 \qquad (26)$$

where:

$$\frac{\partial T_{Delay}}{\partial V_T} = \frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial V_T} = 2.30\left(C_0 + C_{\text{int}}\right)\frac{R_0}{V_{DD} - V_T}$$

$$\frac{\partial T_{Delay}}{\partial V_T} = \frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial \mu} = 2.30\left(C_0 + C_{\text{int}}\right)\frac{R_0}{\mu}$$

$$\frac{\partial T_{Delay}}{\partial t_{ox}} = \frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial t_{ox}} + \frac{\partial T_{Delay}}{\partial C_0}\frac{\partial C_0}{\partial t_{ox}} = 2.30\left(C_0 + C_{\text{int}}\right)\frac{R_0}{t_{ox}} + 2.30\left(R_0 + R_{\text{int}}\right)\frac{C_0}{t_{ox}} \qquad (27)$$

$$\frac{\partial T_{Delay}}{\partial L_{eff}} = \frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial L_{eff}} + \frac{\partial T_{Delay}}{\partial C_0}\frac{\partial C_0}{\partial L_{eff}} = 2.30\left(C_0 + C_{\text{int}}\right)\frac{R_0}{L_{eff}} + 2.30\left(R_0 + R_{\text{int}}\right)\frac{C_0}{L_{eff}}$$

$$\frac{\partial T_{Delay}}{\partial W} = \frac{\partial T_{Delay}}{\partial R_0}\frac{\partial R_0}{\partial W} + \frac{\partial T_{Delay}}{\partial C_0}\frac{\partial C_0}{\partial W} = 2.30\left(C_0 + C_{\text{int}}\right)\frac{R_0}{W} + 2.30\left(R_0 + R_{\text{int}}\right)\frac{C_0}{W}$$

$$\frac{\partial T_{Delay}}{\partial T_{ILD}} = \frac{\partial T_{Delay}}{\partial C_{\text{int}}}\frac{\partial C_{\text{int}}}{\partial T_{ILD}} = \left(1.02R_{\text{int}} + 2.30R_0\right)\frac{C_{\text{int}}}{T_{ILD}}$$

$$\frac{\partial T_{Delay}}{\partial W_{\text{int}}} = \frac{\partial T_{Delay}}{\partial R_{\text{int}}}\frac{\partial R_{\text{int}}}{\partial W_{\text{int}}} + \frac{\partial T_{Delay}}{\partial C_{\text{int}}}\frac{\partial C_{\text{int}}}{\partial W_{\text{int}}} = \left(1.02C_{\text{int}} + 2.30C_0\right)\frac{R_{\text{int}}}{W_{\text{int}}} + \left(1.02R_{\text{int}} + 2.30R_0\right)\frac{C_{\text{int}}}{W_{\text{int}}}$$

$$\frac{\partial T_{Delay}}{\partial t_{\text{int}}} = \frac{\partial T_{Delay}}{\partial R_{\text{int}}}\frac{\partial R_{\text{int}}}{\partial t_{\text{int}}} = \left(1.02C_{\text{int}} + 2.30C_0\right)\frac{R_{\text{int}}}{t_{\text{int}}}$$

Also we have to consider that every branch has a different length. It doubles it length each two levels.

Once the mean delay value and the delay variance of each branch are evaluated for a network, the theoretical approach developed can be used to estimate the mean value, the variance and the yield of the clock skew of the network.

4.2. Verification

To verify the new approach, transistor level Monte Carlo simulations are also conducted. In the simulation, each basic parameter in (24) is simulated by a normal random variable. To agree with the conditions used in the theoretical approach, the correlation between K and V_T, between μ and C_{ox} is neglected as discussed above. The actual delay of a branch is then evaluated from the random values of these basic parameters using (24). The actual delay of a path is the sum of the actual delays of the

branches along that path. The actual maximal clock delay, minimal clock delay and clock skew of the network are then determined by (1)–(3). For a specified value, the yield of a parameter (clock skew) is estimated by the ratio of number of simulations in which the parameter is less than the specified value to the total number of simulations.

The first network considered is well known as the *H*-tree approach shown in Fig. 2 (for brevity, inverters are not illustrated in the following networks). Due to the very symmetrical design of *H*-tree clock networks, all clock paths within the *H*-tree are identical, and the old statistical model can be used to get an upper bound of its expected clock skew when all the paths are assumed to be independent. According to the old model, an upper bound of expected clock skew of a well-balanced *H*-tree is asymptotically given by:

$$E^{upper}(\chi) = \sigma \cdot \left[\frac{4\ln N - \ln\ln N - \ln 4\pi + 2C}{(2\ln N)^{1/2}} + O\left(\frac{1}{\log N}\right) \right] \quad (28)$$

with the variance of clock skew being given by:

$$D^{upper}(\chi) = \frac{\pi^2\sigma^2}{6\ln N} + O\left(\frac{1}{\log^2 N}\right) \quad (29)$$

where:

- σ: standard deviation of path delay.
- $C \approx 0.5772$: Euler's constant.
- N: number of paths (number of processing elements).
- $O(\cdot)$: higher order term.

In a n-level *H*-tree, there are a total of $2^{n+1}-1$ branches, and it can be used to distribute clock signals to 2^n elements. For two combinations of parameters h and W_{int}, in a wide range, and when the numbers of processors are 4, 8, 16, 32, and 64, the theoretical results (obtained using both the old and new models) and simulation results of clock skews of corresponding *H*-trees are summarized in Figure 3.

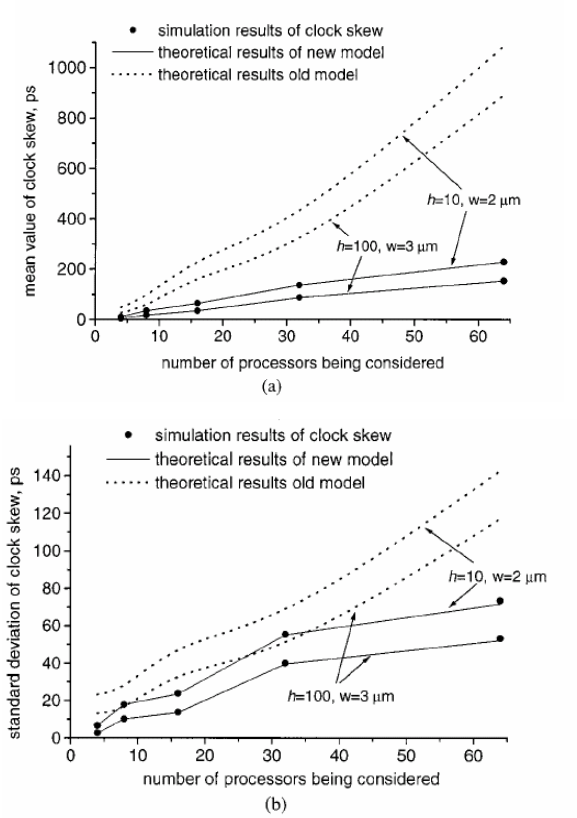


Figure 3: Simulation results and theoretical results of clock skew of *H*-tree networks when different numbers of processors are considered. (a) Mean value of clock skew. (b) Standard deviation of clock skew.

The results in Figure 3 show that the new model is accurate in estimating the mean values and standard deviations of the clock skew of *H*-tree CDNs, where the delay correlation determined by the overlapped parts of path lengths has been considered as indicated in Assumption 2. Compared to the old estimates of expected clock skew where all the paths in the *H*-tree are assumed completely independent, the new estimates based on Assumption 2 are a significant enhancement. For different sized *H*-trees, the expected clock skews estimated using the old model (26) are at least 2.6 times the expected clock skews estimated using our approach. This is shown in Fig. 3(a). In cases where clock frequency is limited by skew rather than by the minimum time between two successive events propagated through the *H*-tree, an unnecessarily long clock period will result from using the old skew model.

Based on the above estimated results of the mean values and the variances of χ, the yields of χ and can be further estimated as discussed in Section4. For the six-level *H*-tree shown in Figure 2, the theoretical yield results and the simulation yield results of χ is summarized in Figure 4. For comparison, we also present in Figure 3 the simulation results of yield of clock skew when all paths are assumed independent.

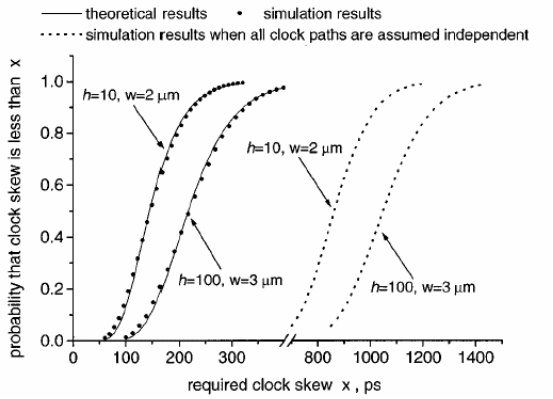


Figure 4: Simulation yield results and theoretical yield results of clock skew of an *H*-tree network for two combinations of parameters *h* and *W*.

The results in Figure 4 indicates that the old model's independent assumption leads to very conservative estimates of clock skew yields of *H*-tree CDNs that have identical but strongly correlated clock paths. The results in Figure 5 also show that when the mean values and the variances of χ of the *H*-tree CDNs are accurately estimated by our approach, the yields of their χ is further approximated by log-normal distribution.

Due to the assumption that all clock paths are identical, the old statistical skew model could not be used to model the clock skews of general clock networks with non-identical paths. The new model developed, however, can be used to accurately estimate the mean values and the variances of clock skew of these general CDNs. For a well-balanced CDN (e.g., *H*-tree clock network), the expected clock skews estimated by old model are very conservative because correlation among paths delay are completely neglected. On the other hand, the expected clock skews estimated by the new model are enhanced significantly. For the well-balanced *H*-tree network shown in Figure 2, the results in Figure 3 show that when parameters $h=100$ and $W_{int}=3\ \mu m$, the expected clock skew estimated by old model is about *892.5 ps*, a value *5.8* times larger than the actual value. For a traditional clocking mode and when the *10%* rule of thumb relating the

skew to the clock period is used, the actual clock period should be dominated by the maximal clock delay, and the mean value of clock period will be *6.96 ns*. However, when the old skew model is used in the skew estimate, the clock period should be determined by the clock skew rather than the maximal clock delay, and the mean value of clock period will be *8.925 ns*. The old model will thus mislead efforts to reduce the clock period of the network. For a pipelined clocking mode, the clock periods of well-balanced *H*-tree networks will be dominated by the clock skew rather than by the minimum time between two successive events propagated through the *H*-tree, an unnecessarily long clock period will result from using the old skew model.

5. Conclusions

- The old model (upper bound model) is too conservative in estimating the expected skew of a well-balanced CDN. Also, in not general enough to model the clock skew of a non-balanced CDN.

- A closed form model (statistical) of clock skew is presented for well-balanced H-tree CDNs.

- The path delay correlation determined by the overlapped parts of path lengths is completely considered in this approach. Therefore, the mean value and variance of clock skew is accurately estimated for general CDNs.

- Considering process variations in designing a clock distribution network, mean values and variances of delays for all branches should carefully be estimated, then the approach presented here will be useful in evaluating and predicting the network's performance of clock skew.

- Two main assumptions:

  - ***Assumption 1:*** A CDN can generally be represented by a binary tree. We assume that both the maximal clock delay and the minimal clock delay in

each subtree (and also the whole binary tree) of the CDN can be modelled by normal distributions when process variations are considered. The assumption makes it easy to analyze the correlation that exists between the maximal and the minimal delay in a subtree. This correlation analysis is critical in determining the variance of skew in each subtree (and also the whole binary tree) of the CDN.

- ***Assumption 2:*** The delay along a clock path is the sum of the uncertain independent delays of the branches along the given path. Correlation between the delays of any two paths is determined only by the overlapped parts of their length.

- To apply the model in a H-tree CDN, it's necessary to know:

  - Interconnection resistance: R_{int}
  - Interconnection capacitance: C_{int}
  - On-resistance of the driving transistor: R_0
  - Input capacitance of the driving inverter: C_0
  - Power supply voltage: V_{DD}
  - Threshold voltage: V_T

  - Threshold voltage deviation (in %): σ_{VT}
  - Charge carrier mobility deviation (in %): σ_{μ}
  - Gate oxide thickness deviation (in %): σ_{tox}
  - Transistor width deviation (in %): σ_W
  - Effective channel length deviation (in %): σ_{Leff}
  - Wire width deviation (in %): σ_{wint}
  - Wire thickness deviation (in %): σ_{tint}
  - ILD thickness deviation (in %): σ_{TILD}

  - Lowest level branch length: L_{int}
  - H-tree levels: n

Appendix 4

Simulator: equations.java

1. Introduction

An important part of this work is related to the analysis of the three proposed clock skew models. It was necessary to program a tool to make the simulations required to check the behaviour of each model. In this appendix, the operation of a JAVA program specifically developed to calculate clock skew estimations according to model equations is presented.

The programming language chosen is JAVA. The main reason is the characteristic of platform independence. It means that this program written in the Java language runs similarly on diverse hardware.

The operation can be summarized as a program that takes parameters (technology parameters) from an input file (with a specific format) or the keyboard and apply over them the model equations. The results are the clock skew estimations, which are presented in an output file and the computer screen.

2. Program operation

This program is executed from the command line. We need to have installed in the computer the Java Runtime Environment or JRE, which is the software required to run any application deployed on the Java Platform.

To start the program, we execute it in this way:

```
C:\Borland\Projects\equations>java equations
```

After this, a menu is showed on the screen:

```
Introduce what do you want to do:
      1. Utilize Kugelmass & Steiglitz Clock-Skew Model
      2. Utilize Zarkesh-Ha, Mule & Meindl Clock-Skew Model
      3. Utilize Jiang & Horiguchi Clock-Skew Model
      4. Utilize Kugelmass & Steiglitz Clock-Skew Model (without
optimal lenths)
      5. Utilize Jiang & Horiguchi Clock-Skew Model (without optimal
lenths)
```

Five options are possible. Options 1, 2 and 3 correspond with the three analysed clock skew model as they were described (buffered H-tree with buffers at the split points and tapered H-tree without intermediate buffers). Options 4 and 5 are the models 1 and 3 respectively adapted to the specific H-tree proposed to analyse the models in this work, where wires are partitioned with buffers to optimize their delay.

Next step is to choose the parameter input. It can be a file, which name is required to type through the command line, or the keyboard, where each parameter has to be individually introduced through the command line.

```
Choose input:
      1. File
      2. Keyboard
      3. Get input format file
```

There is a third option that allows us to get a file with the required input file format. This format is different for each model because they consider different parameter variations to calculate the clock skew. The format must be correct; otherwise the programs will show an error message. Next, an example of configuration file is shown. It is the case of the input file that model 3 requires in the 130 nm technology.

```
R_int=26.27e3
C_int=0.91e-10
R_0=3.94e3
C_0=0.77e-15
V_DD=1.2
V_t=0.19
D=17.32e-3
Dev_V_t=0.042
Dev_u=0.02
Dev_t_ox=0.013
Dev_W=0.05
Dev_L_eff=0.05
Dev_W_int=0.03
Dev_t_int=0.03
Dev_T_ild=0.03
```

Input File: Model 3, 130 nm parameters.

Next step in the program execution is to input the number of levels of the H-tree.

```
Number of levels in the H-tree:
```

After that, the name of the output file has to be introduced.

```
Type the name of the output file
```

After this step, results are displayed on the screen. Also, they can be read in the output file.

```
Total Clock-Skew: 1.6509922986937226E-9
```

Output example.

Model 2 has an important particularity. Authors propose the equations with 50 % of time delay in Sakurai expressions, but if we want that clock skew estimations be comparable with the estimations of models 1 and 3, equations must be calculated with 90 % of time delay. It is possible since during model 2 execution, we have to choose which time delay we want.

```
Choose Sakurai's model:
   1. 50 % time delay
   2. 90 % time delay
```

To type all the options required for each simulation can be laborious if several simulations have to be realised. There is an alternative way to execute this program by introducing options and file names as additional parameters in the command line. It is shown in the following commands.

```
java equations <In file> <Out file> <Model> <No. levels> {<delay>}

java equations in.txt out.txt 1 8
java equations in.txt out.txt 3 8
java equations in.txt out.txt 4 8
java equations in.txt out.txt 5 8
java equations in.txt out.txt 2 8 2
```

3. Recommendations

When many simulations have to be realised, to execute the program lots of times can be laborious. It is recommendable to type a macro that helps us to make similar simulations. An example is when a model is analysed for different number of levels. The required macro is the following.

```
java equations in.txt out.txt 4 1
java equations in.txt out.txt 4 2
java equations in.txt out.txt 4 3
java equations in.txt out.txt 4 4
java equations in.txt out.txt 4 5
java equations in.txt out.txt 4 6
```

Other recommendation that makes our work easy is to redirect the output flow from the screen to a file thanks the parameter to redirect “>”.

```
Macro > results.txt
```

In this way, after executing the macro, all the results can be displayed in a single file facilitating their post-processing.

```
C:\>java equations in.txt out.txt 4 1
   Clock-Skew: 5.418359660062558E-10

C:\>java equations in.txt out.txt 4 2
   Clock-Skew: 1.7925917276162354E-9

C:\>java equations in.txt out.txt 4 3
   Clock-Skew: 1.9237312208357593E-9

C:\>java equations in.txt out.txt 4 4
   Clock-Skew: 3.427741470061498E-9

C:\>java equations in.txt out.txt 4 5
   Clock-Skew: 2.774911991533342E-9

C:\>java equations in.txt out.txt 4 6
   Clock-Skew: 4.255477662366857E-9
```

Macro output example.

References

[AFG-89] M. Afghahi and C. Svensson, “Calculation of clock path delay and skew in VLSI synchronous systems,” in Proc. IEEE Eur. Conf. Circuit Theory and Design, Sept. 1989, pp. 265–269.

[BAI-98] D.W. Bailey and B. J. Benschneider, “Clocking design and analysis for a 600-MHz alpha microprocessor”, IEEE J. Solid-State Circuits, vol. 3, pp. 1627–1633, Nov. 1998.

[FAN-98] S. J. Fang et al, “Control of dielectric CMP using an interferometry based endpoint sensor”, IEEE Int’l Inter. Tech. Conf., pp. 76-78, 1998.

[FIS-85] A.L. Fisher, H. T. Kung, “Synchronizing large VLSI processors arrays”, IEEE Transactions on Computers, Vol. c-34, No. 8, pp 734-740, August 1985

[FRI-86] E. G. Friedman and S. Powell, “Design and analysis of a hierarchical clock distribution system for synchronous standard cell/macrocell VLSI”, IEEE J. Solid-State Circuits, vol. SC-21, pp. 240–246, Apr. 1986.

[FRI-00] E. G. Friedman, “Clock Distribution Networks in Synchronous Digital Integrated Circuits”, Proceedings of the IEEE, Vol. 89, No. 5, pp 665-692, May 2000.

[GRO-98] P. E. Gronowski, W. J. Bowhill, R. P. Preston, M. K. Gowan, and R. L. Allmon, “High-performance microprocessor design,” IEEE J. Solid-State Circuits, vol. 33, pp. 676–686, May 1998.

[HOR-92] M. Horowitz, “Clocking strategies in high performance processors”, in Proc. IEEE Symp. VLSI Circuits, June 1992, pp. 50–53.

[JAC-90] M. A. B. Jackson, A. Srinivasan, and E. S. Kuh, “Clock routing for high performance ICs,” in Proc. ACM/IEEE Design Automation Conf., June 1990, pp. 573–579.

[JIA-01] X. Jiang and S. Horiguchi, “Satistical skew modeling for general clock distribution networks in presence of process variations”, IEEE Transaction on VLSI systems, Vol. 9, No. 5, October 2001, pp 704-711.

[IBM-85] “Method of deskewing data pulses,” IBM Tech. Disclosure Bull, vol. 28, pp. 2658–2659, Nov. 1985.

[ITRS] Semiconductor Industry Association, “The International Technology Roadmap for Semiconductors”, San Jose CA, USA.

[ISM-00] Y.I.Ismail and E.G.Friedman, “Effects of Inductance on the Propagation Delay and Repeater Insertion in VLSI Circuits”, IEEE Transactions on Very Large Scale Integration Systems Vol.8 No.2 pp. 195-206 2000.

[KAH-91] A. B. Kahng, J. Cong, and G. Robins, “High-performance clock routing based on recursive geometric matching”, in Proc. ACM/IEEE Design Automation Conf., June 1991, pp. 322–327.

[KAH-01] A. Kahng, "Design and Other ITRS Technologies: Sharing Brick Walls", IEEE SCCS Kansai Chapter, October 17, 2001.

[KUG-88] S. D. Kugelmass and K. Steiglitz, “A probabilistic model for clock skew,” in Proc. Int. Conf. Systolic Arrays, San Diego, CA, May 1988, pp. 545–554.

[NAE-01] A.Naeemi J.D.Meindl “An optimal partition between on-chip and on-board interconnects” Proceedings on the IEEE International Technology Conference. San Francisco, CA: pp.131-133, 2001.

[NAE-02] A.Naeemi, R.Venkatesan J.D.Meindl “System-on-a-chip global interconnect optimization” Proceedings on ASIC/SoC Conf. pp.399-403 2002.

[RAM-89] P. Ramanathan and K. G. Shin, “A clock distribution scheme for nonsymmetric VLSI circuits,” in Proc. IEEE Int. Conf. Computer-Aided Design, Nov. 1989, pp. 398–401.

[RUS-00] S. Rusu and S. Tam, “Clock generation and distribution for the first IA-64 microprocessor,” in Proc. IEEE Int. Solid-State Circuits Conf., Feb. 2000, pp. 176–177.

[SAK-93] T.Sakurai “Closed-Form Expressions for Interconnection Delay, Coupling and Crosstalk in VLSI's”, IEEE Transactions on Electron Devices Vol.40 No.1 pp.118-124 1993.

[STE-97] S. Stetson, “Low Jitter Clock Distribution Networks”, Dissertation Proposal, The University of Michigan, July 1997.

[TAM-04] S. Tam, Rahul Dilip Limaye, and Utpal Nagarji Desai, “Clock Generation and Distribution for the 130-nm Itanium 2 Processor With 6-MB On-Die L3 Cache”, IEEE Journal of solid-state circuits, Vol. 39, No. 4, April 2004.

[WES-94] N. H. E.Weste and K. Eshraghian, Principles of CMOS VLSI Design: A Systems Perspective. Reading, MA: Addison-Wesley, Apr. 1994.

[YEH-06] C. Yeh, G. Wilke, H. Chen, S. Reddy, H. Nguyen, T. Miyoshi, W. Walker R. Murgai “Clock Distribution Architectures: A Comparative Study”, Proceedings of the 7th International Symposium on Quality Electronic Design (ISQED’06).

[ZAR-98] P. Zarkesh-Ha, T. Mule and J.D. Meindl, “Characterization and Modeling of clock skew with process variations”, submitted to IEEE Custom Integrated Circuit Conf., May 1998.