Design of Clock Distribution Networks in Presence of Process Variations

M. Nekili, Y. Savaria and G. Bois

Ecole Polytechnique de Montreal
Department of Electrical & Computer Engineering
P.O. Box 6079, Station “Centre-Ville”, Montreal, Quebec, Canada H3C 3A7
Phone: (514) 340-4711 ext. 4737; E-mail: nekili@vlsi.polymtl.ca

Abstract: Tolerance to process-induced skew remains one of the major concerns in the design of large-area and high-speed clock distribution networks. Indeed, despite the availability of some efficient exact-zero skew algorithms that can be applied during circuit design, the clock skew remains an important performance limiting factor after chip manufacturing, and is of increasing concern for sub-micron technologies. This tutorial reviews the importance of the problem, its sources, as well as typical examples of existing solutions. Solutions range from design rules strategies to built-in self-compensation methods.

I. Introduction

An ever-growing limitation for high-speed and large area clock distribution networks is process-induced skew. The present paper reviews the research progress for solving this problem since the early 80’s.

Minimizing clock skew in synchronous integrated systems has been approached from different angles. In section II, the importance of the problem is assessed. In section III, two examples of experimental characterizations of process variations point to the source of the problem. Existing solutions include rules incorporated early in the design process (section IV), optimistic exact-zero skew algorithms (section V) and built-in self-compensation techniques (section VI) which are activated after chip manufacturing.

II. Importance of the Problem

The importance of the problem can easily be studied in the case of clock nets distributed according to predefined geometries. Existing models of process variations are either probabilistic (Steiglitz & Kugelmass [1] and Afghahi & Svensson [2]), deterministic (Fisher & Kung [3] and Nekili et al. [4, 5]) or hybrid (Pelgrom et al. [6]).

With the probabilistic approach, Steiglitz & Kugelmass [1] deal with the signal delay along a given clock path as a sum of the delays of different segments of that path, each behaving according to a probabilistic law. Assuming an independence of segment delays, the total path delay as well as the clock skew tend to follow the normal law. For Afghahi & Svensson [2], the clock skew is assumed to be a dispersion of physical circuit parameters (geometrical dimensions) and process parameters (temperature sensitivity, ...).

Work of Fisher & Kung [3]

Fisher & Kung proposed two distinct clock skew models: a difference model (Fig. 1) and a summation model (Fig. 2). They assume an architecture composed of processors synchronized by the clock and organized as an array of a given size. The processors act as the leaves of a binary tree (Fig. 1).

For the difference model, the clock skew between two tree nodes C_i and C_j depends on the physical distance from each of the nodes to the root of the tree (hold line in Fig. 1). This model was proposed for high-speed systems made of discrete components, where clock trees are often wired so that the delay from the root is the same for all cells.

Fig. 1. Difference Model

Nevertheless, as system size increases, small variations in electrical characteristics along clock lines can build up unpredictably to produce skews even between wires of the same length. In the worst case, two wires can have propagation delays which differ in proportion to the sum of their lengths. This remark suggests a second model that simulates process variations which affect the clock tree.
In the summation model, the clock skew between two tree nodes \( C_i \) and \( C_j \) depends on the sum of their distances to their common ancestor (bold line in Fig. 2).

Beside the binary tree distributing the clock signal (Fig. 1 and Fig. 2), it is assumed that the processors exchange data through a communication graph superposed on the clock graph.

![Fig. 2. Summation Model](image)

Fisher & Kung reached the following conclusions:
- Under both models, if the communication graph is linear or one-dimensional, i.e., a direct connection links all adjacent processors laid out as a linear array, the processor array can be clocked with a skew that is independent of system size.
- For multi-dimensional processor arrays, i.e., each processor communicates with any other processor, the above conclusion remains valid under the difference model only, making the synchronization of large systems unfeasible under the summation model.

**Work of Pelgrom et al. [6]**

Despite the widely recognized importance of matching, there were only a limited number of specialized literature contributions to this field, up to late 80’s. Previously, Shyu et al. [7, 8] analyzed the variation in capacitor and current sources in terms of local and global variations. Later, Lakshmikumar et al. [9] described MOS-transistor matching by means of threshold-voltage and current-factor standard deviations.

Pelgrom et al. defined mismatch as the process that causes time-independent random variations in physical quantities of identically designed devices. An analysis is performed for a general parameter \( P \), which can be, in particular, the delay of an electrical device.

Pelgrom et al. assumed that the value of a parameter \( P \) is composed of a constant part and a random part, which results in different values of \( P \) for different pairs of coordinates \((x, y)\) over the wafer. If the variations are small, the average value of the parameter \( P \) over a given area can be represented by the integral function of \( P(x, y) \), as well as the mismatching \( \Delta P \) between two identical areas of logic at different coordinates on the wafer.

Based on a two-dimensional Fourier transform, Pelgrom et al. expressed the mismatch \( \Delta P \) as the product of a geometry-dependent function and a process-dependent function. The latter being the combination of a short-distance “white” noise and a long-distance stochastic phenomenon acting on the wafer scale.

For two rectangular and identical electrical components with an area of \( W \times L \) each at an horizontal distance \( D_x \) from each other, the variance \( \Delta P \) writes:

\[
\sigma^2(\Delta P) = \frac{A_p^2}{W \times L} + S_p^2 \times D_x^2
\]

where \( A_p \) is the area proportionality constant for parameter \( P \), while \( S_p \) describes the variation of parameter \( P \) with spacing. The proportionality constants can be measured and used to predict the mismatch variance of a circuit.

As an application example, Pelgrom et al. used the threshold voltage \( V_T \) and the current factor \( \beta \), two parameters that determine the drain current and hence the transistor time constant. Another direct application of Pelgrom’s modeling was proposed recently by Karhunen et al. [10]. It is a compensation technique based on centroid configurations.

**III. Source of the Problem**

The source of the problem of process variations is investigated through characterization. Two typical examples are provided in this paper: the work of Pavasovic and Andreou [11, 12] on VLSI scale and the work of Gnetting and Jalowiecki [13] on WSI scale. Recently, Nekil et al. [14] conducted a series of experiments in order to build a map of variations in CMOS transistor time constants on die and wafer levels. For a detailed analysis of the source of disturbances in IC manufacturing process, see Maly et al. [15]. Also, Shyu et al. [8] analysis of current mismatch pointed out four physical causes: edge effects, implantation and surface-state charges, oxide effects and mobility effects.

**Work of Pavasovic and Andreou [11,12]**

CMOS combined with the subthreshold operation of the transistor has been a technology of choice for implementing low-power devices. However, with voltage reduction and scaling, subthreshold operation faces substantial parametric
variations in the process. Pavasovic and Andreou addressed
the measurement of process variations through the subthresh-
hold transistor drain current [11] and the above-threshold drain
current [12]. The drain current is a parameter determining
transistor time constant and hence affecting the delay in elec-
tronic devices.

Their experiment is based on a set of four transistor arrays.
Transistors belonging to different arrays have different sizes,
while transistors of the same array have the same size. Pavas-
ovic and Andreou reported the presence of three quasi-deter-
mministic effects ("edge", "striation" and "gradient" effects)
and a random phenomenon.

The "edge" effect manifests itself as a drastic increase or
decrease of drain current at the borders of the transistor
arrays. According to Pavasovic and Andreou, a probable
cause to this effect is strain-induced shifts in the characteris-
tics of the devices. During annealing processes, stress in the
middle of the array will tend to propagate outward and accu-
mulate in the periphery. The "striation" effect appears as a
spatial, sinusoidal oscillation of slowly varying frequency in
the device current. For Pavasovic and Andreou, a possible
cause of this effect is the threshold adjustment ion implanta-
tion process. It could be the resultant doping concentration
spatial distributions due to summation of the Gaussian doping
profile at each pass of the scan. Regarding the "gradient"
effect, it appears as a position-dependent spatial variation of
much lower frequency acting on long distance. When system-
atic effects are removed from the data, the random variations
follow an inverse linear dependence on the square root of
transistor area.


The idea behind this experiment is to characterize, on WSI
scale, the effect of process variations on a ring-oscillator
designed using 51 inverter stages. It involved 25 wafers from
different lots. Copies of the ring-oscillator were dispatched
over the different dies in a wafer and over different wafers.
Oscillation frequencies ranging from 60 to 90 MHz were
measured, with variation coefficients (\( \sigma \)) of 11%, where \( \mu \)
and \( \sigma \) respectively represent the average and the standard
deviation.

The above data combined with other process parameters
extracted from V-I curves allowed studying the impact of
process variations on delay dispersion and skew in a set of
typical WSI clock distribution networks inspired from a
literature review in [16].

One of their main observations is the negligible
contribution of variations due to interconnections (as low as
10.6% for delay and 17.3% for skew). Moreover, the H-tree
based clock network illustrated in Fig. 3, seems to be the most
affected in terms of delay, while it is the least affected in
terms of skew.

IV. "Design Rules" Solution

An approach proposed by Shoji in [17] is based on a first-
order model of process variations while, in [18], Theune et al.
exploit electro-magnetic phenomena. The goal is to elaborate
rules which will allow circuit designers to minimize clock
skew. In [19], Vittoz summarizes a set of rules applied by
designers of high-performance analog circuits.

![Fig. 3. H-tree Clock](image)

**Work of Shoji [17]**

Shoji starts from the architecture of Fig 4, which shows
two clock paths for a clock signal I. The first one contains an
even number of inverters, while the second contains an odd
number of inverters. The two paths feed identical processors
represented by a load.

![Fig. 4. A two-phase clock circuit](image)

Shoji makes the following reasoning:

Let us call \( T_1 \) and \( T_2 \) the delays of inverters 1 and 2 and \( T_A \),
\( T_B \) and \( T_C \) the delays of inverters A, B and C. In order to
make this architecture process-insensitive, the circuit is
designed such that the electrical length of the clock path that
inverts signal I (i.e., \( T_I \)) is equal to that of the path which
does not invert signal I (i.e., \( T_{NI} \)). In other words:

\[
T_I = T_{NI} \quad \text{(Eq. 1)}
\]

with \( T_I = T_A + T_B + T_C \) \quad \text{(Eq. 2)}

and \( T_{NI} = T_1 + T_2 \) \quad \text{(Eq. 3)}

According to Shoji, a circuit designer assumes that his
circuit will be implemented in a typical process noted M. In
practice, however, the transistors P and N are subject to
variations of their time constants. Shoji reflects this
phenomenon by considering two other processes: a high-
current process (noted H) and a low-current process (noted
L). These processes are distinct from the typical process in
guard to the time constants of P-type and N-type transistors.
The time constants will be lower for the H-process and higher
for the L-process (Table 1).

<table>
<thead>
<tr>
<th>Process</th>
<th>NFET (ps)</th>
<th>ratio ( f_N )</th>
<th>PFET (ps)</th>
<th>ratio ( f_P )</th>
</tr>
</thead>
<tbody>
<tr>
<td>High current</td>
<td>88.0</td>
<td>0.556</td>
<td>201.0</td>
<td>0.620</td>
</tr>
<tr>
<td>Typical</td>
<td>158.0</td>
<td>1.000</td>
<td>324.0</td>
<td>1.000</td>
</tr>
<tr>
<td>Low current</td>
<td>273.0</td>
<td>1.730</td>
<td>530.0</td>
<td>1.630</td>
</tr>
</tbody>
</table>

As an example, let us consider the case of a low-to-high
transition on signal I and a circuit that is subject to a high-
current process H. Table 1 suggests that the delay of inverter
B (determined by the P-type transistor time constant)
becomes:

\[
T_B(H) = \frac{T_B(M)}{f_P(H)}
\]

where \( f_P(H) \) is the \( T_B(M) \) and \( T_B(H) \) ratio. The factor \( f_N \)
is defined similarly from Table 1 based on inverter A which
delay is determined by its N-type transistor time constant.
Applying a similar reasoning to the rest of inverters, one can
re-write equations 2 and 3 as:

\[
T_I(H) = \frac{T_A(M) + T_C(M)}{f_N(H)} + \frac{T_B(M)}{f_P(H)}
\]

\[
T_{NI}(H) = \frac{T_1(M)}{f_N(H)} + \frac{T_2(M)}{f_P(H)}
\]

Taking into account Eq. 1, we have:

\[
T_I(H) - T_{NI}(H) = \left( \frac{T_B(M)}{f_P(H)} - \frac{1}{f_N(H)} \right)
\quad \text{(Eq. 4)}
\]

Therefore, since \( f_P(H) \) is in general different from \( f_N(H) \), a
condition for the circuit to be process-insensitive, i.e.,
\( T_I(H) = T_{NI}(H) \), is to satisfy \( T_B(M) = T_2(M) \). This
equality between \( T_B(M) \) and \( T_2(M) \) can also be expressed as:

\[
T_A(M) + T_C(M) = T_1(M)
\]

Thus, Shoji concludes by stating the following design
rules to be applied by designers aiming at process-insensitive
circuits:

1) match the sum of pull-up delays of the P-type
transistors with the pull-up delays of other clock paths.

2) match the sum of pull-down delays of the N-type
transistors with the pull-down delays of other clock paths.

**Work of Vittoz [19]**

Most analog circuit techniques are based on the matching
properties of similar components. For a given process,
maching of critical devices may be improved by enforcing
the set of rules that are summarized below. These rules are not
specific to CMOS and are applicable to all kinds of IC
technologies. The relevance and the quantitative importance
of each of these rules depend on the particular process and on
the particular device under consideration.

1) Devices to be matched should have the same structure.
   For instance, a junction capacitor cannot be matched with an
   oxide capacitor.

2) They should have same temperature, which is no
   problem if power dissipated on chip is very low. Otherwise,
   devices to be matched should be located on the same
   isotherm, which can be obtained by a symmetrical
   implementation with respect to the dissipative devices.

3) They should have same shape and same size. For
   example, matched capacitors should have same aspect ratios,
   and matched transistors or resistors should have same width
   and same length, and not simply same aspect ratios.

4) Minimum distance between matched devices is
   necessary to take advantage of spatial correlation of
   fluctuating physical parameters.

5) Common-centroid geometries should be used to cancel
   constant gradients of parameters. Good practical examples
   are the quad configuration used to implement a pair of
   transistors, and common-centroid sets of capacitors.

6) The same orientation on chip is necessary to eliminate
   dissymmetries due to anisotropic steps in the process, or the
   anisotropy of the silicon substrate itself. In particular, the
   source to drain flows of current in matched transistors should
be strictly parallel.

7) Devices to be matched should have the same surroundings in the layout. This is to avoid for instance the end effect in a series of current sources implemented as a line of transistors, or the street effect in a matrix of capacitors.

8) Using non-minimum size devices is an obvious way of reducing the effect of edge fluctuations, and to improve spatial averaging of fluctuating parameters.

V. Exact-Zero Skew Algorithms

In the competitive market of integrated circuits, the algorithmic approach is a fast and usually inexpensive design tool. Early work has focused on global clock routing which addressed geometrical issues exploiting the notions of Steiner minimal tree and rectilinear minimal tree [20, 21]. Later, researchers [22, 23, 24, 25] began taking into account electrical aspects of the clock tree such as skew, by balancing and minimizing the length of clock paths. A disadvantage of this approach is that it uses minimum width lines which are very susceptible to variations in the etch rate of the metal lines, as well as to mask misalignment or local spot defects [26]. As a consequence, effective interconnect impedance and delay can vary greatly from wafer to wafer. To design process-insensitive interconnects, Pullen et al. [27, 28, 29] have developed an automated layout algorithm that widens rather than lengthens the interconnect.

The work of Tsay [30, 31] announced a new generation of algorithms [32, 33, 34]. Algorithms where delay balancing is based on electrical length rather than geometrical length, and that takes into account clock net loads. Though the primary concern of this algorithm is not process-induced skew, to this date, it is the representative of a class of algorithms which illustrate the limit of what amount of clock skew can be optimized prior to chip manufacturing.

Work of Tsay [30, 31]

This algorithm is based on the Elmore delay model [35]. It uses a bottom-up recursive procedure which is illustrated in Fig. 5. In this figure, it is assumed that the algorithm has reached an intermediate stage in the clock tree hierarchy. Also, two zero-skew subtrees are already built, i.e., for each subtree, all root-to-leaf delays are equal. This assumption is obvious in the case the subtree reduces to a leaf, which is the initial condition of the algorithm.

To interconnect two zero-skew subtrees using a wire, while ensuring the resulting tree is zero-skew, the problem is to find a location (tapping point) on the wire so that the delays from all leaves to the new tree root are equal. The tapping point does not necessarily cut the wire into two equal halves, but into two different segments. Each segment is represented by a \( \pi \) interconnect model, where \( r_1 \) and \( c_1 \) are the resistance and capacitance of the first segment. Parameters \( r_2 \) and \( c_2 \) are similarly defined for the second segment. Each subtree is replaced by an input capacitance \( C_i \) (accounts for load differences) and a branch delay (accounts for capability of clock drivers).

To ensure the resulting clock tree is zero-skew, the following equation has to be satisfied:

\[
r_1 \left( \frac{c_1}{2} + C_1 \right) + t_1 = r_2 \left( \frac{c_2}{2} + C_2 \right) + t_2 \quad (\text{Eq. 5})
\]

The total wire length is noted \( l \). If the length of the wire segment between the new tree root and the root of subtree 1 is \( x \times l \), then, the length of the wire segment between the new tree root and the root of subtree 2 will be \( (1-x) \times l \).

![Fig. 5. Interconnecting two zero-skew subtrees [30]](image)

Let \( \alpha \) and \( \beta \) be the resistance and capacitance per unit of wire length. Hence, we get: \( r_1 = \alpha \ell \), \( r_2 = \alpha (1-x) \ell \). We also have \( c = \beta \ell \), \( c_1 = \beta x \ell \) and \( c_2 = \beta (1-x) \ell \).

Solving Eq. 5 ensures the zero-skew condition and fixes the tapping point location at:

\[
x = \frac{(t_2-t_1) + \alpha \ell \left( C_2 + \frac{\beta \ell}{2} \right)}{\alpha \ell (\beta \ell + C_1 + C_2)}
\]

If \( 0 \leq x \leq 1 \), the tapping point lies somewhere on the wire, otherwise (\( x<0 \) or \( x>1 \)), no tapping point satisfying the zero-skew condition can be found on the wire. For \( x<0 \), subtree 1 features a delay greater than what the wire can...
balance. Therefore, the tapping point should be fixed at the root of subtree. Then, a wire elongation is explored following a way similar to the above. A similar scheme is adopted for x>1 which corresponds to an unbalanced delay lying on the subtree 2 side. In case the delays of the two subtrees are too different, so that a reasonable wire elongation cannot balance them. Tsay recommends the use of drivers, delay lines or capacitors.

As far as circuit design is concerned, in order to complete this approach, a consistent strategy for buffer insertion was needed [29, 36, 37, 38, 39]. One can understand the importance of such a strategy if we know that up to 90% of process variations affecting delay are due to buffers [13]. The notion of iso-radius levels [37] is worth mentioning here. Previous methods insert buffers level by level at the branch split points of the clock tree. Unfortunately, this works only in a full binary tree where all sinks have the same number of levels (e.g., processor arrays). In general, to ensure equalized root-to-leaf delays, one sometimes needs to insert a buffer somewhere between consecutive levels, at every iso-radius level.

Despite the far extent to which skew is reduced by exact-zero skew algorithms, this strategy faces two main limitations [40]:

1- It is vulnerable to process variations. The clock routing produced in [30, 31] is guided by estimated interconnect and logic RC parameters. It therefore achieves zero-skew only in regard to the parameter values available during the design process. After chip manufacturing, any variation in these parameters will cause skew to appear again.

2- It lacks in design flexibility. If designers want to modify the location of any clock pin, the entire clock routing needs to be redone. Modifications commonly occur because designers cannot know the exact clock pin locations until late in the physical design process.

To overcome the above drawbacks, Lin and Wong [40] proposed a hierarchical two-stage multiple-merge approach which relies on a zero-skew center chunk (fat wire).

Another attempt to design process-insensitive circuits [37] aimed at constructing a clock distribution tree by automatically and separately sizing PMOS and NMOS transistors of the clock buffers (Shoji’s technique [17]). However, it was shown [45] that worst-case analysis is often carried out in terms of a correlated set of parameters, which results in a design that is unnecessarily pessimistic. This remark was confirmed in [14]. Nekili et al. noticed that active devices, such as inverters, are subject to large dispersions in their time constants depending on the position of the transistor in the wafer, and on the nature of its surroundings. As a consequence, only a small proportion of the transistor population is close to the worst or best case. Therefore, designing a clock system based on transistor worst-case performance will cause a substantial penalty to the clock frequency, whereas in practice, one might take advantage of knowledge on the variations of transistor performance throughout the wafer.

To overcome the difficulty of controlling clock path delays due to process parameter variations, Neves and Friedman [46] reduce minimum clock period using intentional skew. A permissible clock skew range is calculated for each local data path while incorporating process dependent delay values on the clock signal paths.

Based on a concept initially proposed by Lee and Murphy [47], Nekili et al. [48] recently proposed zero-skew techniques to minimize process-induced clock skew using delay calibration in buffered clock trees. In a bottom-up fashion, an algorithm proceeds by a set of clock signal measurements performed at a limited number of clock nets and then the delays of tree branches are balanced by affecting the electrical features of either the interconnections or the buffers. A first technique consists of connecting several capacitors to the branches of the clock tree that are closest to the root. The clock skew is measured between two clock nets and accordingly, a laser alternately cuts capacitors from the corresponding pair of tree branches, until both skew and delay are minimized at the current tree level. A 500-MHz clocked experimental chip was fabricated with tree branches laid out over a 1.8cm by 1.8cm die, using the Nortel 0.8μm BiCMOS technology. A second technique implements a clock buffer as a number of minimum-sized inverters in parallel, all connected using the top metal layer. According to the skew measurement at a pair of clock nets, a laser cuts inverters from one of the corresponding pair of buffers until the skew is minimized at the current tree level. Once the whole calibration phase is finished, the circuit designer reports the final buffer sizes into the circuit design using an adapted layout mask of the top metal layer and vias only, which allows rapid and relatively inexpensive development of production masks with calibrated clock trees. With this method, skews induced by deterministic effects such as environment-induced process variations and power supply drops may be compensated, even though no method may be available to accurately predict these effects before manufacturing.

VI. Built-in Self-Compensation

The literature available in support of this approach comes mainly from industry, which tends to adopt a "pragmatic" approach. With this approach, the emphasis is put on the detection or measurement of the resultant of all process variations, in order to compensate them either on the device level [41, 42] or by self-adjustment [43, 44].

Cox et al. [41] have developed a control circuit
methodology, which measures the relative performance of a CMOS chip, and transmits a digital state code to output drivers and clock generation circuits, in order to monitor the device characteristics in presence of process variations, temperature and voltage fluctuations.

Chengson et al. [43] developed a self-adjusting synchronization system for clock distribution. This system receives a digital and periodic clock signal as a reference, and generates multiple clock signals that dynamically synchronize with the clock reference. This structure was the critical part of an error-tolerant computer system.

Beside these types of variations, Watson et al. [44] focus on compensating load variations when drivers are used to increase the fanout of a clock distribution network. Watson et al. use a sort of delay adjustment called absolute to ensure that a driver delay remains a multiple of clock period.

The effect of process variations being more important with scaled technologies, Asahina et al. [42] use AC and DC fluctuations of macrocells characteristics in ASICs in order to detect such variations and compensate them. With this method, clock delay variations (skew) as well as noise are reduced.

VII. Conclusion

To our knowledge, this is the first tutorial addressing the design of clock distribution networks in presence of process variations. Based on deterministic as well as probabilistic modeling, it was observed that process-induced skew makes synchronous multi-dimensional processor arrays not feasible. Various sources to this problem were pointed out, including local and global variations. Solutions to the problem range from design rules, applied in digital design as well as analog devices, to built-in self-adjustment techniques. The evolution of CAD tools and their limitations in skew minimization was also reviewed.

References:


