Design Methodologies and Architecture Solutions for High-Performance Interconnects

Davide Pandini
STMicroelectronics, Central R&D
Agrate Brianza, 20041 Italy
+39 039 6036437
davide.pandini@st.com

Cristiano Forzan
STMicroelectronics, Central R&D
Bologna, 40136 Italy
+39 051 2093829
cristiano.forzan@st.com

Livio Baldi
STMicroelectronics, Central R&D
Agrate Brianza, 20041 Italy
+39 039 6035015
livio.baldi@st.com

ABSTRACT

In Deep Sub-Micron (DSM) technologies, interconnects play a crucial role in the correct functionality and largely impact the performance of complex System-on-Chip (SoC) designs. For technologies of 0.25μm and below, wiring capacitance dominates gate capacitance, thus rapidly increasing the interconnect-induced delay. Moreover, the coupling capacitance becomes a significant portion of the on-chip total wiring capacitance, and coupling between adjacent wires cannot be considered as a second-order effect any longer. As a consequence, the traditional top-down design methodology is ineffective, since the actual wiring delays can be computed only after layout parasitic extraction, when the physical design is completed. Fixing all the timing violations often requires several time-consuming iterations of logical and physical design, and it is essentially a *trial-and-error* approach. Increasingly tighter time-to-market requirements dictate that interconnect parasitics must be taken into account during all phases of the design flow, at different level of abstractions. However, given the aggressive technology scaling trends and the growing design complexity, this approach will only temporarily ameliorate the interconnect problem. We believe that in order to achieve gigascale designs in the nanometer regime, a novel design paradigm, based on new forms of regularity and newly created IP (Intellectual Property) blocks must be developed, to provide a direct path from system-level architectural exploration to physical implementation.

1. INTRODUCTION

In this era of DSM technologies the impact of interconnects is becoming increasingly important as it relates to integrated circuit (IC) functionality and performance. Wiring capacitance dominates the gate capacitance, thus rapidly increasing the interconnect-induced delay (as a percentage of the overall path delay) [1]. Therefore, the impact of interconnects on performances has to be carefully evaluated in order to satisfy the design constraints.

The growing weight of interconnections stems from two factors: on one side the global wirelength does not follow the feature size scaling, since it is related to the chip size, which does not shrink because more functionalities are integrated on the same chip. In contrast, the interconnect aspect ratio (height over width) is increased to control the wiring resistance (the vertical dimension of the wires is kept approximately constant to freeze the sheet resistance). In addition, the distance between adjacent wires shrinks, making the coupling capacitance dominate the ground capacitance. A larger cross-section aspect ratio yields a fringing capacitance (i.e., the capacitance between the side-walls of the wires and substrate) whose contribution to the overall ground capacitance cannot be neglected, as shown in Figure 1.

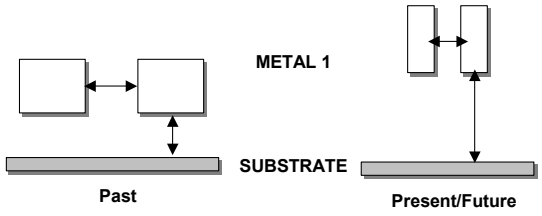

Figure 1. Interconnect capacitance trends

Although the original predictions on interconnect aspect ratio growth [3] have been significantly reduced (allowing a wiring resistance increment), an increase in aspect ratio from about 1.6 at 65nm to 1.7 at 35nm for the local and semi-global wires, and from 2.1 (65nm) to 2.2 (35nm) for global wires has been forecasted [4], thus confirming the coupling capacitance dominance over the ground capacitance.

Over the past few years, researchers have addressed the DSM interconnect problem from the architectural and circuit perspective, and at the technological level. In late 1990s copper interconnects were introduced. Copper has lower resistivity (2.2μΩ·cm) than the widely used aluminum (3.3μΩ·cm) and improved reliability with respect to electromigration. It is reported that by using copper a reduction of about 40% in wiring resistance can be obtained. Moreover, low-k dielectric materials have replaced the silicon dioxide to better insulate the tightly packed wires from each other, thus reducing the parasitic capacitance. However, the introduction of porous materials, in order to achieve very low-k values, poses new technology and reliability challenges such as an inferior mechanical stability, which will make practically impossible adding more metal layers on top of the nine layers currently available in 90nm technology. Hence, technological solutions will be beneficial, but it is doubtful if they can keep pace with increasing design complexity and routing density. Therefore, more effective approaches to the interconnect problem must be introduced both at the architectural and circuit level.

This paper is organized as follows: in Section 2 the impact of delay on the design of complex circuits is discussed and physical synthesis is introduced, while in Section 3 dynamic parasitic

effects, like crosstalk, are analyzed, and an approach for crosstalk-aware static timing analysis is presented. Section 4 analyzes why in block-based design the real problem is the global interconnects, and Section 5 outlines an interconnect-centric design methodology. In Section 6 clock distribution in synchronous circuits is discussed, and Section 7 presents a new approach to achieve gigascale designs based on regular fabrics. Finally, Section 8 summarizes a few conclusive remarks.

2. WIRELOAD MODEL AND PHYSICAL SYNTHESIS

The size of modern VLSI ICs does not allow performing logic and physical design concurrently. Hence, the traditional ASIC (Application Specific IC) design flow is essentially top-down and broadly speaking consists of: high-level logic design, logic synthesis (i.e., mapping the logic design onto a library of pre-characterized library cells), physical design (placement and routing), and sign-off timing and functional verification. This methodology is based on the assumption that all internal signals reach the downstream logic within one clock cycle; hence, a proper estimation of internal delays is critical for the circuit functionality. For technologies above 0.25μm, the interconnect models were essentially based on fanout loading and predefined net configurations. Unfortunately, as technology moves deeper into the DSM regime, the standard cell driving resistance becomes comparable with the wiring resistance, and for accurate delay estimations a distributed RC representation of the interconnections is necessary.

A fanout-based model (i.e., the *wireload model*) can be highly inaccurate for wiring delay estimation, since by not considering the actual interconnect topology, it cannot accurately predict the distributed RC effects. As a consequence, several iterations between logic synthesis and physical design are necessary to meet the performance constraints and reach the timing closure, which is the complete synchronization of all internal signals. Unfortunately, this iterative process does not have any guarantee to converge, and significant changes to the high-level description of the design may be necessary, thus introducing a critical bottleneck in achieving tight time-to-market targets.

In performance-driven design, optimization based on the wireload often yields results that are significantly different from post-layout values. This model statistically predicts the gate load capacitance as a function of fanout based on technology data and design legacy information, but does not consider the actual wiring topology. Up to which level of complexity can the wireload model be effectively used to predict the wiring effect on timing? In [2] Sylvester and Keutzer employed technology roadmaps data with some predicted modifications [3] to estimate the logic block size below which the interconnect impact on performance is not significant. They concluded that the largest block size where wiring capacitance does not dominate gate capacitance, and traditional synthesis can still be used, is approximately 50k gates of logic. This 50k gate number is derived from the assumption of unlimited driving size capabilities. However, the results published in [9] demonstrate that for a block size of 20k gates of custom logic, the wireload model can be quite inaccurate for nets with a fanout number larger than four. More recently, Gopalakrishnan et al. [11] confirmed that the intrinsic wireload model inaccuracy could have a strong impact on predicting the lengths and delays also for local nets (intra-block nets) in industrial designs.

Anyhow, the assumption of unlimited driving size capabilities is questionable, because it implies a worst-case design for all cells, and therefore a dramatic increase in power consumption. This is not acceptable, since power is rapidly becoming one of the most important constraints in a variety of modern System-on-Chip (SoC) and wireless applications. Moreover, wiring congestion is another limiting factor that due to the growing design complexity is becoming more and more important. The larger the congestion, the higher is the number of interconnects that have to detour around critical areas, and longer detour paths will increase the wiring delay. In [12] it was demonstrated that for logic blocks below the 50k limit, congestion has a detrimental impact on timing and die area. Hence, routability is another optimization objective that must be considered during logic synthesis along with area, delay, and power, in order to avoid many design iterations. The traditional approach has been to use more metal layers to relieve congestion, and to insert buffers with high driving strength, to speed-up signal propagation in long interconnections. Both approaches are showing limits, since interconnection area dominates the active area, and power dissipation is one of the most critical issues in SoC design.

The traditional ASIC design flow must be significantly changed, and several approaches to physical synthesis have been proposed both in industry and academia. Physical synthesis attempts to combine logic synthesis with physical design in order to predict the interconnect effects more accurately. In particular, there has been a trend towards placement-aware synthesis, where logic synthesis has been integrated with placement, and wiring delay estimations are obtained from the physical locations of the placed cells. A first approximation of the interconnect topology is the bounding-box model, which considers the semi-perimeter of the rectangle enclosing the net terminals. But this approach is not enough accurate for long nets, where the metal resistance can be significantly larger than the driver resistance. Hence, more accurate estimation methodologies have been developed, based on coarse net routing (Steiner Tree models [14]). More recently, the Design Automation industry has proposed new approaches were some logic optimization is routing-driven, and the delay of the most critical nets is estimated from actual routing topologies.

While physical synthesis is the current solution to the interconnect prediction problem in industrial design flows in 0.13μm and 90nm technologies, we believe it is an *evolutionary* approach. The growing complexity of modern SoC designs and the aggressive technology scaling trends need a *revolutionary* solution to achieve gigascale designs, based on regularity exploitation at higher level of abstraction, and newly created customizable regular fabrics.

3. DYNAMIC EFFECTS: CROSSTALK IMPACT ON SIGNAL INTEGRITY

In Section 1 it was discussed that interconnect coupling capacitance dominates the overall parasitic capacitance, and crosstalk between neighboring wires can impact the functionality and performances of high-speed circuits. The most intuitive crosstalk effect is noise injection from switching aggressor signals on a quiet victim net. With technology scaling, noise immunity has rapidly become another critical metric in the design of high-performance circuits. Increased clock frequencies and interconnect densities, continued threshold voltage scaling, and using more aggressive design styles, i.e., dynamic logic and pass

transistor logic, strongly impair the noise immunity and consequently the circuit functionality. In fact, noise glitches may alter the logic information stored in state elements, like a latch or a flip-flop, or on a dynamic node, thus causing a functional failure. Furthermore, large noise glitches may generate reliability problems, such as transistor aging, due to hot electrons injection. However, an approach to detect the noise-induced failures based on static noise margins can be too conservative, and dynamic noise margins must be considered [8].

Equally important, is the crosstalk impact on performances. In fact, when aggressor signals switch in opposite directions with respect to the victim, they introduce an extra-delay that will affect both the timing critical paths, and the set-up time of registers. In contrast, aggressor and victim signals switching with the same phase generate a speed-up on the victim net that might violate the hold-time constraints of sequential modules. Therefore, crosstalk analysis must be based on the logical and temporal relations of the switching signals.

A detailed analysis of crosstalk impact on signal integrity cannot be performed with circuit simulations, given the prohibitive size of the interconnect networks of complex SoC designs. Model Order Reduction (MOR) techniques have been proposed to obtain reduced order macromodels of the original interconnect distributed RC representation [5]. The accuracy of these methods is comparable with circuit simulations, and they have been successfully used for a post-layout crosstalk analysis of large VLSI (Very Large Scale Integrated) circuits [6]. However, due to the dynamic behavior of crosstalk, a purely static analysis based on the worst-case coupling, would introduce an unaffordable level of pessimism. Hence, the temporal interactions between the victim and the aggressor signals, and their phase relations must be considered. In [7] an approach for crosstalk evaluation during static timing analysis was presented, where crosstalk-aware timing windows are computed with an iterative procedure that reduces the level of pessimism and performs an accurate crosstalk analysis on timing critical nets. This approach, integrating MOR-based macromodels with a static timing analysis engine, has become mainstream, and has been implemented in state-of-the-art signal integrity analysis tools currently used in industrial design flows for timing verification.

4. BLOCK-BASED DESIGN

In [13] it was observed that the statistical distribution of wirelengths in VLSI ICs is a bimodal distribution as shown in Figure 2, where the peaks are at the average local interconnect length (intra-block nets), and at the average global interconnect length (inter-block nets).

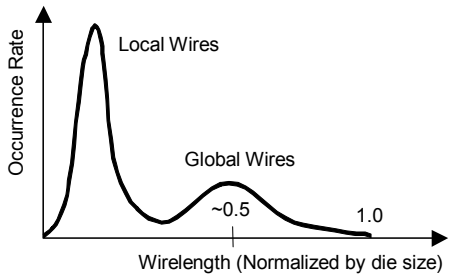

Figure 2. Wirelength distribution

Such distribution clearly indicates the difficulty with timing closure when several IP blocks must be assembled on the same die, as illustrated in Figure 3.

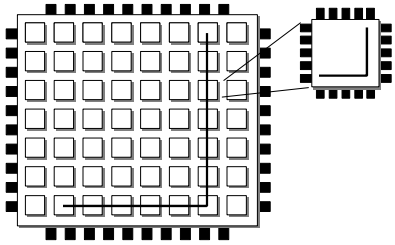

Figure 3. Global and local interconnects in block-based design

What happens is that feature size scaling normally does not result in smaller chips (which would make the interconnect delays roughly constant), but in the integration of more functions in the same die area. Therefore, while wireload models can predict timing of some, if not all, local interconnects within the logic blocks, which are scaling with the technology, the global interconnects traveling across large regions of the chip are strongly design-dependent, and do not scale. In [10] it was analyzed that as DSM technology moves deeper into the nanometer regime, the number of global wires will increase in absolute value, since there are more blocks to be assembled on the chip. Moreover, with routing congestion and buffering, the global wirelength will also increase relative to the local wirelength. In contrast, the complexity of each hierarchical block will remain relatively fixed as predicted in [2], and the local average wirelength will move to the left, as shown in Figure 4, while the global wires will cover a larger portion of the normalized wirelength distribution.

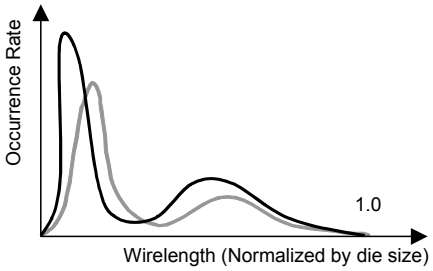

Figure 4. Wirelength distribution trends

In next generation designs the interconnect prediction problem will be mainly caused by global wires, and accurate delay estimations will be more strongly dependent on detailed routing information. The intrinsic physical hierarchy defined by the local interconnects (for intra-block connections) and the global interconnects (for inter-block connections) is exploited by routing the global interconnections (power and clock distribution networks, inter-block communication) on the upper metal layers, where wiring sizing and spacing can be increased to reduce the interconnect resistance and coupling capacitance. Implementation of copper and low-k materials allows scaling of the intermediate wiring levels and minimizes their impact on delay. While local wires are relatively unaffected by scaling, RC delay is dominated by global interconnects and the benefit of material changes alone

cannot sustain increasingly tighter performance constraints. It is worth noticing that the number of buffers used to drive long interconnections in high-speed design will grow dramatically. Therefore, the associated power dissipation along with the increased active area and routing congestion will soon limit the effectiveness of this approach, and a new design paradigm based on structured and regular communication channels, and innovative clock distribution techniques will have to be developed.

5. INTERCONNECT-CENTRIC DESIGN

Given the growing importance of interconnects in current and future generations of VLSI systems, interconnect optimization must be considered at all levels of abstraction. In conventional IC design much emphasis was given to device and logic optimization, while the interconnections were left to automatic layout tools. In contrast, in *interconnect-centric* design, which includes three major tasks: interconnect planning, interconnect synthesis, and interconnect layout, all phases of the design flow are focused on interconnect optimization.

5.1 Interconnect Planning

The architectural-level phase of interconnect-centric design is *interconnect planning*, and since it is applied early in the design flow, it can have a significant impact on the final result. The first step is to generate a physical hierarchy, which defines the global, semi-global, and local wires. A physical hierarchy is more embeddable onto a two-dimensional layout surface than a simple logical hierarchy, which only represents the relationship among the various logic functions of the design high-level architectural description, and as a rough approximation, it corresponds to the definition of the IP blocks, as sketched in Figure 3. A this point, the connections between the different blocks obtained from top-level partitioning are the global interconnects, while the connections between different modules within the same block occupy a lower level in the wiring hierarchy. With a recursive process, all the levels down to local interconnect are defined. After the wiring physical hierarchy is generated, the second step is floorplanning, where some of the interconnect synthesis techniques described in Section 5.2 are applied to the global and semi-global interconnections to determine the best topology, layer assignment, wire width and spacing, which meet the performance constraints.

The advances in DSM technology leave room for optimization that is not solely dictated by the manufacturing capability. Hence, a step of interconnect architecture planning may determine various interconnect parameters for system-level performance and reliability optimization, subject to the manufacturing constraints. Ideally, these parameters will include the number of routing layers, the wire thickness, and the nominal width and spacing in each layer. Interconnect architecture planning would consider a design characterization in terms of target clock rate and interconnect distribution, obtained after the physical hierarchy generation and interconnect planning, and predetermine a small number of common interconnection widths in each metal layer that can be used during interconnect synthesis.

Performance estimation is a critical task carried out during interconnect planning, where a large number of different floorplan configurations must be explored. However, important detailed information such as the granularity of wire segmentation, and the buffer locations and sizes, is not available at this level of abstraction. Consequently, interconnect planning that simply uses wirelength-based delay models may not correlate with the optimization techniques in interconnect synthesis. As a consequence, the performance estimation models used in interconnect planning, in addition to the estimated wirelength, should also include both technology parameters such as the wiring sheet resistance and the unit area and fringing capacitance coefficients, and design parameters like the loading capacitance and the driving resistance.

After interconnect planning is completed, synthesis and placement under the wiring physical hierarchy are executed. During this step, physical synthesis can be used to synthesize complex blocks, since the wireload model may be inaccurate even for local nets, as it was discussed in Section 2.

5.2 Interconnect Synthesis

The second major task in interconnect-centric design is *interconnect synthesis*, which determines the (near-) optimal interconnect structure for each net in terms of topology, wire ordering, buffer locations and sizes, wire width and spacing, that meet the performance and signal integrity requirements under area and routability constraints. The main techniques used in interconnect synthesis are timing-driven global routing for performance optimization and for relieving routing congestion, and wire ordering and spacing to reduce crosstalk noise.

The first step in topology optimization is to minimize or control the pathlengths from the driver to the timing-critical sinks in order to reduce the interconnect distributed RC delays. Several algorithms have been proposed to minimize both the total wirelength and the pathlength in a routing tree. The Bounded-Radius Bounded-Cost (BRBC) algorithm [14] bounds the radius (the maximum pathlength from a net driver to a net sink) in the routing tree while minimizing the total wirelength. Alternative routing algorithms are described in [15]. Further optimization of interconnect topology in physical design is based on more accurate delay models, such as the Elmore delay model [16], which is a first-order dominant-pole approximation to the interconnect distributed RC delay, and can be efficiently computed [17].

In DSM design wiring resistance cannot be neglected; hence, wire sizing effectively reduces the interconnect delay. It is important to notice that many approaches to wire sizing assume a fixed coupling capacitance or lump the coupling capacitance into the fringing capacitance. The hypothesis of a fixed coupling capacitance in wire sizing is not realistic under the assumption of a fixed pitch between adjacent wires, since if the width of one wire changes, then its spacing to adjacent wires also changes, often resulting in different coupling capacitance values. Therefore, wire spacing should be considered concurrently during wire width optimization [18]. For global interconnects, wire sizing and spacing alone are not sufficient to limit the quadratic growth of the interconnect delay with respect to the wirelength. Buffer insertion (also called repeater insertion) is a very effective approach that trades off the active device area with a reduction of interconnect delay. With optimal buffer insertion, the growth of the interconnect delay becomes linear with the wirelength. This technique is widely used in performance-driven optimization, both during routing and in post-layout incremental resynthesis.

Moreover, the global router should also handle congestion minimization. The methods usually employed for this task are:

1. *rip-up and reroute*, which is used to find alternative routes for blocked nets, and iteratively converges to a low-congestion solution: the routing engine searches the layout region around the congested area, and finds an alternative connection for the net;
2. *iterative deletion*, which begins with multiple routes for each net, and iteratively removes redundant routing paths with the highest congestion, until each net has only one route.

The two strategies can be combined: rip-up and reroute can be initially used to obtain multiple routing solutions for some (or all) nets, and iterative deletion can be used to determine the best routing solution for each net. Wiring congestion is a critical design factor that strongly impact routability and timing that should be considered in logic synthesis as it was demonstrated in [12], where structurally less congested circuits can be routed within a smaller die size and with fewer metal layers.

5.3 Interconnect Layout

Aggressive interconnect synthesis and optimization often result in complex interconnect structures with many buffers, variable wire widths, and different spacing rules between adjacent wires to minimize capacitive coupling. These requirements must be taken into account during detailed routing. Ideally, the routing algorithm would support multilayer, variable-width, and variable-spacing interconnections.

To overcome the ordering problem associated with a *net-by-net* routing, and to support efficient rip-up and reroute, first the available routing resources are estimated, and then a multi-iteration approach evenly distributes the nets in the routing regions, in order to minimize wiring congestion. It is important to point out that during this phase, the resynthesis capabilities of layout tools are limited, and cannot significantly modify the wiring structures previously obtained. As a consequence, the most effective interconnect optimization techniques must be applied at higher level of abstraction.

5.4 Custom Techniques in ASIC Design

In [19] Dally and Chang proposed to use custom techniques to improve the performance, power, and area of ASIC designs without affecting the development cycle. Traditionally, custom design was restricted to microprocessors, but with an increasing impact of wiring delay and design complexity, custom techniques will be more important as fully automatic flows fail to meet aggressive design constraints, as it was discussed in Section 2. In custom design it is possible to control the physical structure of the design, while in automated design the layout is generated automatically with little or no control on the structure of the physical implementation. Automatic tools first place the logic modules and then route the signals, thus losing the intrinsic structure of the design. In contrast, in custom design first the critical signals are routed and then modules are placed. The fundamental idea of applying custom design techniques to an ASIC is based on structuring the most critical wires and leaving the rest to automatic tools, and then placing the logic modules and completing the routing. Critical wires that can be structured are global signals (like clock signals), busses, datapath bits and word lines. Structured wiring will be an important phase in high-performance ASICs, and it is consistent with interconnect-centric design.

6. CLOCK DISTRIBUTION AND SYNCHRONIZATION

In synchronous digital systems, clock signals are used to define a time reference for data transferring, and much attention must be given to their characteristics and distribution networks. Clock signals are often considered as simple control signals; however, they have some special properties, as they are typically loaded with the largest fanout, travel over the longest distances across the die, and operate at higher speeds then any other signal within the entire system. Since the temporal reference of data signals is provided by the clock signals, their waveforms must be particularly clean and precise.

Clock signals are particularly affected by technology scaling, since long global interconnect lines become more resistive. Such increased line resistance is one of the primary reasons for the growing impact of clock distribution on synchronous circuit performance. Moreover, any differences in the clock signal delay can also create catastrophic race conditions, where an incorrect data value may be latched within a register. Most synchronous digital circuits consist of cascaded banks of sequential registers with combinational logic between each set of registers. The functional requirements are satisfied by the logic stages, while the global performance and local timing requirements are satisfied by the careful insertion of pipeline registers into equally spaced time windows to meet critical worst-case timing constraints.

A careful design of the clock distribution network ensures that no race conditions exist, and synchronous system performance can increase, surpassing the potential performance advantages of asynchronous circuits by allowing synchronous performance to be based on average path delays rather than worst-case path delays, without incurring the handshaking protocol delay penalties required in most asynchronous circuits. In a synchronous system, each data signal is typically stored in a bistable register waiting the incoming clock signal, which determines when the data signal leaves the register, propagates through the combinational network and, for a properly working system, enters the next register and is fully latched before the next clock appears. Therefore, the delay components in a synchronous system are composed of:

- memory storage elements;
- logic elements;
- clocking circuitry and distribution network;

and the relationships among these three subsystems are critical to achieving maximum performance and reliability.

The difference in clock signal arrival time between two *sequentially-adjacent* registers[1] is the clock skew[2], which is originated by the following causes:

[1] Two registers are sequentially-adjacent when they are driven by the same clock signal.

[2] The clock skew is the difference in clock delay from the clock source of two sequentially-adjacent registers.

1. differences in line lengths from the clock source to the clocked register, and consequent different distributed RC delay;
2. differences in delays of any active buffers in the clock distribution network [due to 3) and 4) below];
3. process variations in passive interconnect parameters, such as wire resistivity, dielectric constant and thickness, via/contact resistance, ground and fringing capacitance, and line dimensions;
4. process variations in active device parameters, such as MOS threshold voltages and channel mobilities, which affect the delay of the active buffers.

6.1 Clock Distribution Networks

Several approaches have been developed for designing clock distribution networks in synchronous digital systems. The most common and general approach to uniform clock distribution is based on buffered trees, where buffers are inserted both at the clock source and along the clock paths, forming the tree structure shown in Figure 5.

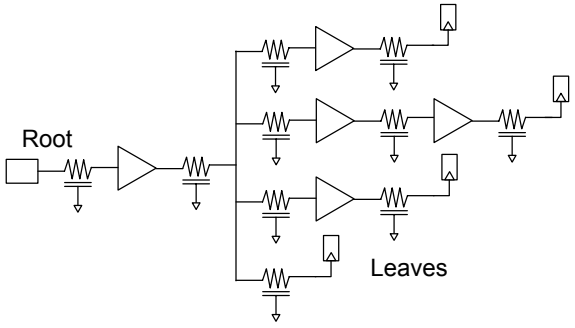

Figure 5. Buffered tree clock distribution network

The unique clock source is described as the *root* of the tree, while the registers driven by the distribution network are the *leaves*. When a single buffer is used to drive the entire clock distribution network, it should provide sufficient current to drive the network capacitance (both interconnect and fanout), while maintaining high-quality waveform shapes (i.e., short transition times) and minimizing the effects of the interconnect resistance. An alternative approach to using only a single buffer at the clock source is to distribute buffers throughout the network. This approach requires additional area but greatly improves the precision and control of the clock signal waveforms, and is necessary when the interconnect resistance cannot be neglected. The distributed buffers serve the double function of amplifying the clock signals degraded by the distributed interconnect impedances and isolating the local clock nets from upstream load impedances. Alternatively, a mesh version of the clock tree structure can be used to minimize the interconnect resistance within the clock tree.

Another approach for distributing clock signals utilizes a hierarchy of symmetric H-tree structures illustrated in Figure 6. This clock distribution network guarantees zero clock skew by maintaining identical distributed interconnect and buffers from the clock signal source to the clocked register of each clock path. In this approach, the primary clock driver is connected to the center of the main "H" structure and the clock signal is transmitted to the four corners of the main "H". These four "close to identical" clock signals provide the inputs to the next level of the H-tree hierarchy, represented by the four smaller "H" structures. The distribution process continues through several levels of progressively smaller "H" topologies. The final destination points of the H-tree are used to drive the local registers or are amplified by local buffers driving the local registers. Thus, each clock path from the clock source to a clocked register has practically the same delay.

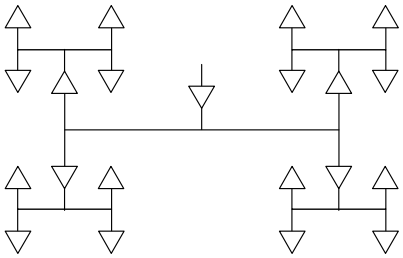

Figure 6. H-tree clock distribution network

The primary source of clock skew within an H-tree structured clock distribution network is due to variations in process parameters that affect the interconnect impedance and, more importantly, any distributed active buffer amplifiers. Hence, clock skew depends upon the control of the semiconductor process, and the degree to which active buffers and clocked latches are uniformly distributed within the H-tree topology. The conductor widths in H-tree structures are designed to progressively decrease as the signal propagates to lower levels of the hierarchy. This strategy minimizes reflections of the high-speed clock signals at the branching points.

It is worth noticing that interconnect capacitance (and therefore the power dissipation) is much greater for H-trees since the total wirelength tends to be much greater. Such increased capacitance of the H-tree structure dictates an important trade-off between clock delay and clock skew in the design of high-speed clock distribution networks. Symmetric structures are used to minimize clock skew; however, an increase in clock signal delay is incurred. Therefore, the increased clock delay must be considered when choosing between buffered trees and H-trees. Furthermore, since clock skew only affects sequentially-adjacent registers, the obvious advantages of highly symmetric structures to distribute clock signals are significantly degraded. However, there may be certain sequentially-adjacent registers distributed across the integrated circuit. For this situation, a symmetric H-tree structure may be appropriate, particularly to locally distribute the global clock. Another consideration in choosing a clock distribution topology is that H-trees are difficult to implement in those VLSI-systems that are irregular in nature. In these circuits, buffered tree topologies, integrated with structured custom design methodologies, should be used in the design of the clock distribution network in order to maximize system clock frequency, minimize clock delay, and control all detrimental effects of local clock skew.

6.2 Beyond Synchronous Design

A fundamental limit to on-chip signal propagation is given by the speed of light. A 750mm^2 die cannot support a global frequency above 7.75GHz, which is reduced to about 5.5GHz in practice due to Manhattan-like routing. As a consequence, operating

frequencies above 10GHz present fundamental limits to current communication schemes, since the entire die cannot be reached within one clock cycle. Gigascale designs will require multiple clock cycles, and in 65nm technology about 16 clock cycles are estimated to be necessary to cross a chip size of about 25mm, with a repeater every 0.5mm. Therefore, achieving global synchronization will be more difficult in future nanometer technologies. Moreover, the clock overhead in terms of power consumption will be unacceptable.

Although a fully asynchronous design was described in [20], the design methodology for asynchronous systems is far from mature for general acceptance [21]. A design style called *Globally Asynchronous and Locally Synchronous* (GALS) was proposed in [22]. The GALS architecture is composed of several synchronous blocks of different size, which communicate on an asynchronous basis by means of a handshake protocol. Each synchronous block can be clocked independently, and there is no global clock. Hence, a GALS design is globally skew tolerant, and the global power consumption is significantly reduced. However, there is a communication overhead and an area and power penalty introduced by the handshake protocol implementation. This architecture is very promising, especially for a block-based design methodology.

7. REGULAR FABRICS: A NEW PARADIGM FOR GIGASCALE DESIGN

Interconnect-centric design and physical synthesis have significantly changed traditional ASIC design. However, the evolutionary trend of IC design has demonstrated that the productivity gap between what we could implement in next generation technology and what we can afford to build in that technology is growing as technology scales down. The most advanced CAD (Computer-Aided Design) tools that implement the algorithms and methodologies presented in Sections 2, 3, 5, and 6 can only partially fill the gap depicted in Figure 7, which is widening, thus suggesting that CAD tools are not keeping pace with technology and design complexity.

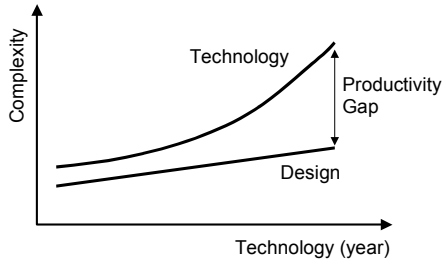

Figure 7. Design productivity gap

The typical DSM effects such as crosstalk, power supply voltage fluctuations, and process variations make performance prediction during the front-end (i.e., logic synthesis) phase of the design flow unreliable. Furthermore, manufacturability is another major challenge in nanometer technologies. Modern lithography must use compensation schemes on masks, such as Optical Proximity Correction (OPC), or advanced illumination schemes to define feature sizes that are smaller than wavelength, but the effectiveness of these techniques largely depends on the geometry of the patterns to be defined, and it is definitely anisotropic. Such trends favor the use of a limited number of regular layout structures. In spite of all these interconnect-related effects, most of the design efforts must be focused at the architectural level to manage the complexity of modern SoC applications. Hence, an alternative design paradigm based on a new class of ICs with some regularity and structured on-chip communication, and with some application-specific customization, would replace the traditional standard cell-based ASIC design style for a wide range of mid-volume applications that can afford to surrender some performance, area, and power, for a much faster turn-around time, specific customization, and reusability. By exploiting regularity both at the logical and physical level, we will obtain an early interconnect effect predictability, thus drastically reducing the number of iterations necessary to achieve the design closure. In addition, more regularity in the physical implementation introduces a *correct-by-construction* approach that greatly facilitates the verification phase, and a reduced number of regular layout patterns also improve the OPC efficiency. Hence, regular fabrics will allow focusing the design efforts at the system level, providing a direct path to physical implementation.

In [23] a noise-immune interconnect fabric was proposed, where the signal wires are alternated with the power supply and ground lines acting as shields, and the wires on one layer are perpendicular to the wires on adjacent layers (Figure 8). In this regular interconnect pattern the coupling capacitance between signal wires is negligible. Although the routing area penalty of this approach make it practical only to connect a network of PLAs (Programmable Logic Arrays), ideally, the basic concept of exploiting regularity to design correct-by-construction interconnect structures can be further expanded to develop a library of pre-characterized, parameterized, and scalable noise-immune wiring topologies for a predictable and structured on-chip communication.

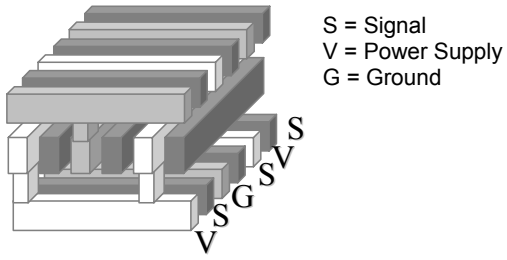

Figure 8. Noise-immune interconnect fabric

Moreover, the regular interconnect structures can also be reconfigured by means of programmable active switches, or via-customizable through mask patterns, where by surrendering the flexibility of field-programmable switches, and using vias to implement the connections, significant improvements in terms of area and performances can be obtained. In [24] a via-customizable crossbar was presented, and in [25] the benefits in terms of area saving and performance of using via-programmable interconnect structures to create a new class of hybrid mask- and field-programmable architectures was demonstrated. More recently, in [26] regular and structured routing architectures were proposed. In the structured routing architecture the router is restricted to an underlying grid template, where each grid point is a "potential via" site, and metal lines can be cut to desired length without producing undesirable patterns. There is some performance penalty with this model, since density and total interconnect resistance and capacitance will be worse than ASIC routing.

However, this loss of performance is traded off against better printability and manufacturability. By only allowing the router the selection of vias to complete the connections, there are additional performance penalties. However, in such via-configurable model there are two important advantages: 1) application-specific customization for low-volume products only requires a set of customized via masks; 2) the number of via patterns can be controlled and optimized for printability significantly better than ASIC or even structured routing. The results presented in [26] show a limited performance loss compared with the potential improvement in manufacturability and yield.

In [27] the concept of regular fabrics was extended to customizable logic blocks, and a detailed study on the impact of logical and physical regularity on manufacturability, performance, and design methodology was presented.

8. CONCLUSIONS

In this paper we have addressed several interconnect-related problems in DSM technologies from the design methodology perspective. Other physical effects are becoming important, and next generation designs will have to consider inductive coupling due to the increasing operating frequencies and low-resistivity materials, and reliability problems caused by electromigration and thermal effects. Moreover, the power supply voltage drop will have to be carefully evaluated due to the increasing complexity of on-chip power distribution and scaling power supply values.

Although the technological progress will ameliorate the detrimental impact of wires on performance and functionality of high-speed SoC designs, we believe the most effective and manufacturable solutions will be obtained from new interconnect architectures and structured on-chip communication paradigms (Network-on-Chip), from innovative design strategies (block-based design and regular fabrics), and from new CAD tools supporting such methodologies.

REFERENCES

[1] H. B. Bakoglu, *Circuits, Interconnections and Packaging*. Reading, MA: Addison Wesley, 1990.

[2] D. Sylvester and K. Keutzer, "Getting to the Bottom of Deep Submicron," in *Proc. Intl. Conf. on Computer-Aided Design*, Nov. 1998, pp. 203-211.

[3] Semiconductor Industry Association, *National Technology Roadmap for Semiconductors*, 1997.

[4] Semiconductor Industry Association, *International Technology Roadmap for Semiconductors*, 2003.

[5] L. T. Pillage and R. A. Rohrer, "Asymptotic Waveform Evaluation for Timing Analysis," *IEEE Trans. on Computer-Aided Design*, vol. 9, pp. 352-366, Apr. 1990.

[6] D. Pandini, P. Scandolara, and C. Guardiani, "Network Reduction for Crosstalk Analysis in Deep Submicron Technologies," in *Proc. Intl. Symp. on Timing Issues*, pp. 280-289, Dec. 1997.

[7] B. Franzini, C. Forzan, D. Pandini, P. Scandolara, and A. Dal Fabbro, "Crosstalk Aware Static Timing Analysis: a Two Step Approach," in *Proc. ISQED*, Mar. 2000, pp. 499-503.

[8] K. L. Shepard, V. Narayanan, and R. Rose, "Harmony: Static Noise Analysis of Deep Submicron Digital Integrated Circuits," *IEEE Trans. on Computer-Aided Design*, vol. 18, pp. 1132-1150, Aug. 1999.

[9] S. Hojat and P. Villarrubia, "An Integrated Placement and Synthesis Approach for Timing Closure of PowerPC Microprocessors," in *Proc. Intl. Conf. on Computer Design*, Oct. 1997, pp. 206-210.

[10] L. T. Pileggi, "Achieving Timing Closure for Giga-Scale IC Designs," in *Proc. Intl. Symp. on Timing Issues*, Mar. 1999, pp. 25-28.

[11] P. Gopalakrishnan, A. Odabasioglu, L. T. Pileggi, and S. Raje, "Overcoming Wireload Model Uncertainty During Physical Design," in *Proc. Intl. Symp. on Physical Design*, Apr. 2001, pp. 182-189.

[12] D. Pandini, L. T. Pileggi, and A. J. Strojwas, "Congestion-Aware Logic Synthesis," in *Proc. DATE*, Mar. 2002, pp. 664-671.

[13] S. M. Kang, "Metal-Metal Matrix (M3) for High-Speed MOS VLSI Layouts," *IEEE Trans. on Computer-Aided Design*, vol. CAD-6, pp. 886-891, Sep. 1987.

[14] J. Cong, A. B. Kahng, G. Robins, M. Sarrafzadeh, and C. K. Wong, "Provably Good Performance-Driven Global Routing," *IEEE Trans. on Computer-Aided Design*, vol. 16, pp. 739-752, Jun. 1992.

[15] A. B. Kahng and G. Robins, *On Optimal Interconnections for VLSI*. Boston, MA: Kluwer, 1994.

[16] W. C. Elmore, "The Transient Response of Damped Linear Networks with Particular Regard to Wide-Band Amplifiers," *Jr. of Applied Physics*, vol. 19, pp. 55-63, Jan. 1948.

[17] J. Rubinstein, P. Penfield Jr., and M. A. Horowitz, "Signal Delay in RC Tree Networks," *IEEE Trans. on Computer-Aided Design*, vol. CAD-2, pp. 202-211, July 1983.

[18] J. Cong, L. He, C.-K. Koh, and Z. Pan, "Global Interconnect Sizing and Spacing with Considerations of Coupling Capacitance," in *Proc. Intl. Conf. on Computer-Aided Design*, Nov. 1997, pp. 628-633.

[19] W. J. Dally and A. Chang, "The Role of Custom Design in ASIC Chips," in *Proc. Design Automation Conf*., Jun. 2000, pp. 643-647.

[20] G. M. Jacobs and R. W. Brodersen, "A Fully Asynchronous Digital Signal Processor Using Self-Timed Circuits," *IEEE Jr. Of Solid-State Circuits*, vol. 25, pp. 1526-1537, Dec. 1990.

[21] S. Hauck, "Asynchronous Design Methodologies: An Overview," *IEEE Proceedings*, vol. 83, pp. 69-93, Jan. 1995.

[22] A. Hemani, et al., "Lowering Power Consumption in Clock by Using Globally Asynchronous Locally Synchronous Design Style," in *Proc. Design Automation Conf*., Jun. 1999, pp. 873-878.

[23] S. P. Khatri, A. Mehrotra, R. K. Brayton, R. H. J. M. Otten, and A. Sangiovanni-Vincentelli, "A Novel VLSI Layout Fabric for Deep Sub-Micron Applications," in *Proc. Design Automation Conf*., Jun. 1999, pp. 491-496.

[24] C. Patel, A. Cozzie, H. Schmit, and L. T. Pileggi, "An Architectural Exploration of Via Patterned Gate Arrays," in *Proc. ISPD*, Apr. 2003, pp. 184-189.

[25] L. Macchiarulo, C. F. Caccamo, and D. Pandini, "A Comparison Between Mask- and Field-Programmable Routing Structures on Industrial FPGA Architectures," in *Proc. GLSVLSI*, Apr. 2004, pp. 436-439.

[26] V. Kheterpal, A. J. Strojwas, and L. T. Pileggi, "Routing Architectures for Regular Fabrics," in *Proc. Design Automation Conf*., Jun. 2004, pp. 204-207.

[27] L. T. Pileggi, et al. "Exploring Regular Fabrics to Optimize the Performance-Cost Trade-Off," in *Proc. Design Automation Conf*., Jun. 2003, pp. 782-787.