The Scaling Challenge: Can Correct-by-Construction Design Help?

Prashant Saxena Noel Menezes Pasquale Cocchini Desmond A. Kirkpatrick
{prashant.saxena, noel.menezes, pasquale.cocchini, desmond.a.kirkpatrick}@intel.com

Intel Labs (CAD Research)
2111 NE 25th Ave, Hillsboro, OR 97229 USA

ABSTRACT

We present the results of scaling studies in the context of typical block-level wiring distributions, and study the impact of the identified trends on the post-RTL design process. In particular, we look at the implications of exponentially increasing repeater and clocked repeater counts on the algorithms and methodologies used for logic synthesis, technology mapping, layout, and full-chip assembly, and identify several new research problems relevant to future designs. Next, we introduce the basic principles of correct-by-construction (CbC) design. We look at some techniques for post-RTL design meeting CbC philosophy, and then construct a case for flexible, abstract fabrics. Finally, we suggest CbC approaches to tackle the new synthesis and layout challenges identified in this paper.

Categories and Subject Descriptors

B.7.1 [**Integrated Circuits**]: Types and Design Styles – *advanced technologies, VLSI (very large scale integration).*

General Terms

Algorithms, Measurement, Performance, Design, Experimentation

Keywords

Clocked Repeaters, Correct-by-construction Design, Design Fabrics, Interconnect, Logic Synthesis, Placement, Post-RTL Design, Repeaters, Routing, Scaling, Technology Mapping.

1. INTRODUCTION

Although CMOS scaling has enabled VLSI designers to realize Moore's law [1], it has also created several new design concerns. Primary among these is the increasing dominance of interconnects [2] and leakage [3]. Under ideal scaling, all dimensions of the wires are shrunk 0.7x per generation. Therefore, although the wire capacitance per micron remains invariant, the wire resistance per micron doubles every process generation, resulting in a wire delay degradation per scaled micron of 1.4x every generation (modulo improvements in materials). Since the RC-delay of an unbuffered interconnect grows quadratically with wire length, repeaters have traditionally been used to linearize the dependence of delay on interconnect length. In an optimally buffered interconnect, the delay of any given stage is approximately equally divided between the repeater and the wire. However, the wire delay degradation during process scaling disturbs this balance by increasing the proportion of the wire delay in an optically shrunk buffered interconnect [4]. In order to re-optimize this interconnect, additional repeaters need to be added. Our work quantifies this increasing number of repeaters in the context of the wiring distribution of a typical synthesizable block, and explores its impact on the algorithms and methodologies used for post-RTL design. Although we omit the details here, our experimental results are in accordance with first-order scaling theory. There have been several works highlighting the impact of scaling on power (see, for instance, [3]); in this paper, we focus on the new post-RTL CAD concerns arising from scaling.

2. EXPERIMENTAL METHODOLOGY

We opted for a SPICE-level simulation based study of scaling in contrast to a first-order theoretical approach because a greater range of electrical effects are captured in simulations with a higher confidence level in the quantitative predictions. However, rather than using an existing technology exploration system like GTX [5], we decided to create new process-independent, behavioral models for the devices in our study. Our device and wire models were closely calibrated with existing process technology files used for microprocessor designs at Intel, and approximated future technology files using scaling trends. Further details on these models are provided in the Appendix.

In this paper, *critical repeater length* refers to the minimum distance beyond which inserting an optimal-sized repeater makes the interconnect delay smaller than that of the corresponding unrepeated wire. Similarly, the *critical sequential length* is the maximum distance that a signal can travel in an interconnect that has been optimally sized and optimally buffered uniformly, within a single clock period (whose duration is determined by the frequency scaling assumptions). We derived critical repeater and sequential lengths on an infinite buffered wire for different metal layers under various Miller coupling factors (MCFs). We also studied the sensitivities of the buffered wire delays to variations in driver size, MCF and inter-repeater length. Our infinite buffered wire assumption allowed us to ignore the complexities of repeater

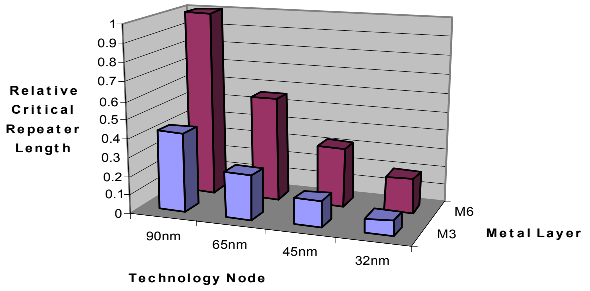
ISPD'03, April 6-9, 2003, Monterey, California, USA.

tapering, yielding results that were easier to interpret. In practice, most repeaters on a reasonably long interconnect end up being identically sized even under optimal repeater tapering with a discrete cell library. Furthermore, this uniformity tends to increase as wires become more resistive. Although we optimized the interconnect buffering for delay, the same trends hold true for power-constrained designs also (the value of the first-order expression for inter-repeater distance is almost invariant across different repeater sizes in a given process technology). In this paper, we report data for metal layers M6 and M3; M5 and M6 are primarily used for communication at the full-chip level (and are usually designed to have similar electrical properties), while M3 and M4 usually form an electrically similar pair used for "semi-global" communication (i.e. within large synthesizable blocks). While future process technologies will provide additional metal layers, it has been predicted that these upper layers will be reserved primarily for the power grid, global clocking and a few critical handcrafted busses, thus leaving automated synthesis to operate primarily in the M1-M6 range. Furthermore, the data reported here assumes a commonly used MCF of 1.5; other realistic MCF values do not change the trends significantly.

We obtained the wiring distribution from a microprocessor block designed for a 90nm process technology using a standardized physical synthesis flow. This wiring distribution was obtained *after stripping off the repeaters* that had been added by the physical synthesis engine during design convergence. This block contained 79K cells (excluding the removed repeaters) and 92K (pre-buffering) nets. As we scaled this histogram across different process technology nodes, we studied the migration of the critical repeater and sequential lengths across the histogram as well as their impact on the total cell count within the synthesizable block.

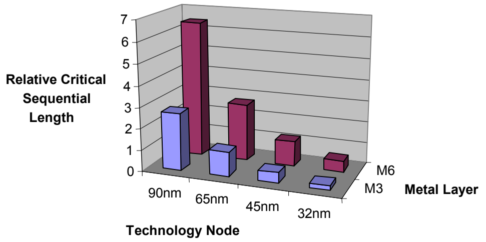
2.1 Scaling Assumptions

We assume that minimum feature sizes will continue to shrink at the rate of 0.7x per generation (similar to that assumed by ideal scaling theory) with approximately 30% speedup per generation in the devices as described in the Appendix. Based on historical trends, we also assume that the actual area of the largest block size that can be synthesized efficiently remains invariant (see Section 4.3 for further justification of this assumption). This implies that the number of cells in the block doubles every generation, and therefore, by Rent's rule, the number of wires in the block also approximately doubles every generation [6]. Furthermore, we assume that the shape of the wiring histogram of the block remains invariant [6]. Recent history and ITRS trends indicate a doubling of the frequency of leading edge designs every generation [7,3]. We use this frequency scaling assumption in our studies, starting with a 5GHz frequency at the 90nm technology node (extrapolating from 3GHz microprocessors commercially available currently at the 130nm node). However, as will be pointed out later in this paper, many of the most important quantitative conclusions of our repeater studies are independent of the actual operating frequency and the frequency scaling assumptions. Furthermore, while it is likely that future designs will be constrained by the power envelope rather than the maximum obtainable performance, it has been empirically observed and analytically shown that scaling under the delay-power product is asymptotically identical to ideal scaling that is targeted towards pure performance. Thus, even if we tradeoff performance for power, the upcoming post-RTL design crisis can not be avoided.



	90nm	65nm	45nm	32nm
🟦 M3	0.43	0.24	0.14	0.08
🟥 M6	1	0.56	0.32	0.19

Figure 1. Relative critical repeater lengths.



	90nm	65nm	45nm	32nm
🟦 M3	2.72	1.15	0.50	0.21
🟥 M6	6.44	2.71	1.17	0.51

Figure 2. Critical sequential lengths (relative to critical repeater length for M6 in 90nm).

Our simulations ignore the exponential increase per generation in the *resistivity* of the Cu material used for the wires because of the impact of surface boundary scattering and the poor scaling of the barrier layer used to prevent the Cu from diffusing into the dielectric [8]. These phenomena will only exacerbate a situation that is dire even with our optimistic assumptions.

3. EXPERIMENTAL RESULTS

3.1 Critical Wire Lengths

Figure 1 plots the relative critical repeater lengths for various technology nodes, normalized to the critical repeater length for M6 at the 90nm node. Note that the lengths for M6 and M3 shrink at the (geometric average) rate of 0.57x per generation (in contrast to normal scaling of 0.7x). This quantifies the observations in Section 1 that additional repeaters need to be added during a

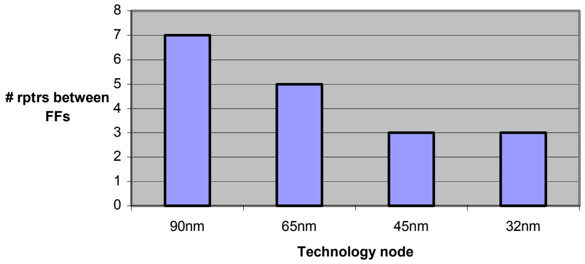


Figure 3. Unclocked repeater count per interconnect pipeline stage.

shrink of an optimal repeated interconnect from one process generation to the next.

Next, we present the critical sequential lengths for various technology nodes in Figure 2. As in Figure 1, these lengths are also normalized to the critical repeater length for M6 at the 90nm node. The critical sequential lengths shrink at the rate of 0.43x per generation. Not only is this shrinking much faster than normal (0.7x) scaling, it is also much faster than the rate of decrease in critical repeater lengths. This implies that ideally shrunk interconnects will not only require new repeaters as shown by Figure 1, but that many of these new repeaters will need to be clocked. The ratio between clocked and unclocked repeaters on a buffered interconnect will continue to grow, as it is scaled across technology nodes. This changing ratio is reflected in Figure 3 that shows the number of unclocked repeaters between two successive clocked repeaters on an infinite buffered interconnect that has been optimized to maximize the distance between the clocked repeaters. This number shrinks at the average rate of 0.75x per generation, and is predicted to be as small as 3 at the 45nm node itself. (It remains 3 at the 32 nm node due to discretization).

3.2 Repeated Wires

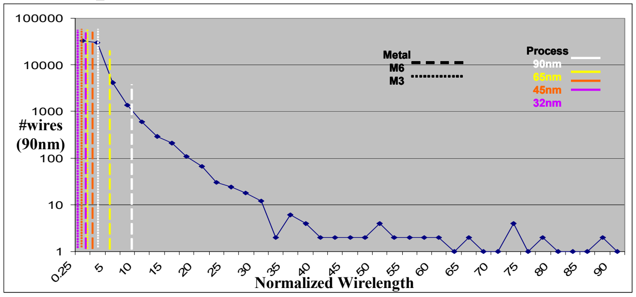


Figure 4. Migration of critical repeater lengths across scaled wiring histogram of testcase.

In order to study the impact of the shrinking critical repeater and sequential lengths, we ran an experiment to see how they migrated across the wiring histogram of a typical synthesized block across process generations. The histogram shape was kept invariant across process generations [6] (as justified in Section 2.1), even as the absolute wire count kept doubling every generation. Figures 4 and 5 depict this migration for metal layers M3 and M6. The number of nets requiring repeaters or clocked repeaters in any given process generation is proportional to the area under the histogram curve to the right of the line representing the corresponding critical length. Observe that as the critical lengths move to the left, they impact an exponentially increasing number of nets, as is clear from the increasing slope of the histogram curve in spite of a logarithmic scale on the "# wires" axis.

We next attempted to quantify this leftward migration of the critical lengths by measuring the percentage of nets that was impacted by repeaters and clocked repeaters in successive process generations. This data is presented in Figures 6 and 7. Observe that although the number of block-level nets impacted by clocked repeaters is negligible at the 90nm technology node, it becomes quite substantial by the time we reach the 32nm node. Furthermore, the rate at which the percentage of impacted nets is increasing also starts accelerating (as indicated by the upwardly concave curves, especially in the clocked repeater chart).

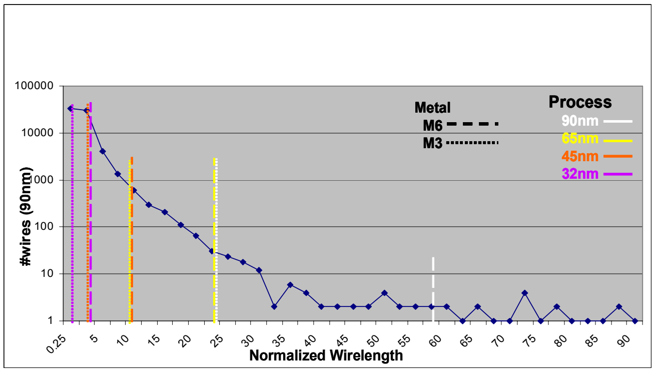
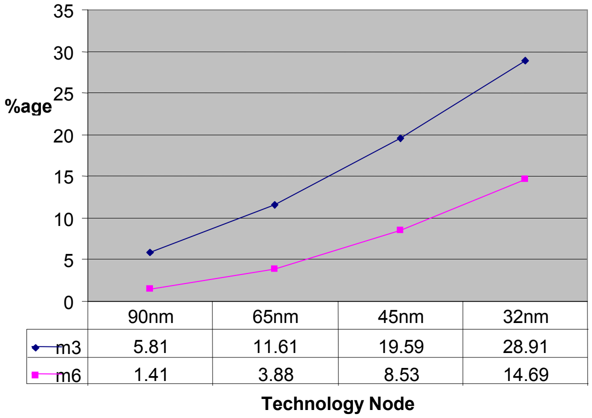
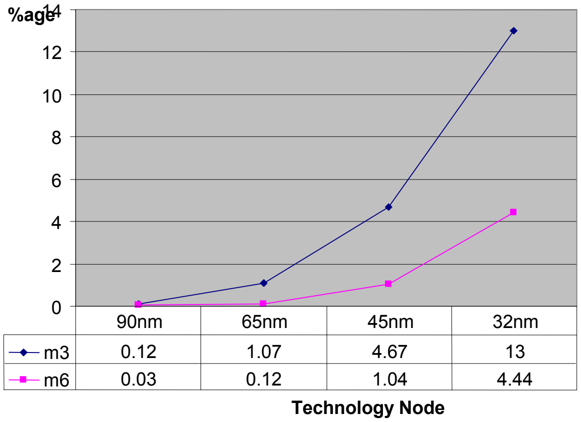


Figure 5. Migration of critical sequential lengths across scaled wiring histogram of testcase.



	90nm	65nm	45nm	32nm
m3	5.81	11.61	19.59	28.91
m6	1.41	3.88	8.53	14.69

Figure 6. Percentage of block-level nets requiring repeaters.

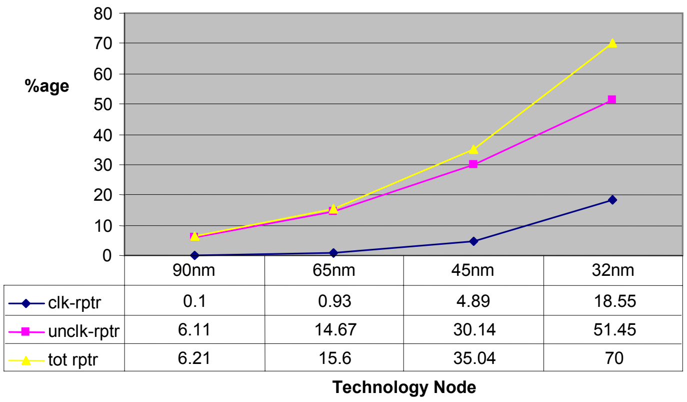


	90nm	65nm	45nm	32nm
m3	0.12	1.07	4.67	13
m6	0.03	0.12	1.04	4.44

Figure 7. Percentage of block-level nets requiring clocked repeaters.

3.3 Repeater Count

While the number and proportion of nets that require repeaters grows rapidly, the number of repeaters in the block and their proportion of the total cell count grows even more rapidly. This is because the longer nets require a disproportionately large number



	90nm	65nm	45nm	32nm
clk-rptr	0.1	0.93	4.89	18.55
unclk-rptr	6.11	14.67	30.14	51.45
tot rptr	6.21	15.6	35.04	70

Figure 8. Intra-block communication repeaters as a fraction of the total cell count.

of repeaters. In order to estimate the repeater count, we conservatively represented all the intra-block wiring on layers M1-M4 by M3 and that on layers on M5-M7 by M6 before measuring the number of repeaters required by the block. This yielded the number of repeaters and clocked repeaters required by intra-block nets on the lower and upper layers separately. The total number of cells in the block can be easily obtained from our block scaling assumptions. This data is put together in Figure 8, which shows the percentage of the total block cell count that is made up of repeaters required for the intra-block communication. This percentage reaches 35% by the 45nm technology node and an alarming 70% by the 32nm node. Furthermore, note that this is a conservative estimate – not only because of our optimistic scaling assumptions but also because *it does not include the repeaters that must be placed within the block to support the inter-block global interconnects that are routed above the block*. (The repeaters required by these nets cannot all be placed in repeater banks on the periphery of the block because the critical repeater distance is smaller than the block dimensions even on the upper layers). Observe that *the total number of repeaters is independent of the frequency scaling assumptions*, but depends only on the process and block size scaling assumptions that we have outlined in Section 2.1. The only aspect of this experiment that is impacted by the frequency scaling assumptions is the ratio of the numbers of clocked and unclocked repeaters. This ratio is also growing alarmingly, as can be seen from Figure 8.

4. IMPACT ON POST-RTL CAD

This explosion in the number of clocked and unclocked repeaters has a profound impact on post-RTL design, and will need to be tackled at both the methodological and algorithmic levels. The primary methodological impact is in the dynamic management of the physical design hierarchy that must now manage netlists that change on the fly due to repeaters, as well as handle repeaters from other levels of the hierarchy. Another major problem arises because of the increasing instances of pipelined interconnects [9]. A small error in the early prediction of the sequential latency of an interconnect can lock the logic at its sink into a sub-optimal or infeasible pipeline stage, whereas retiming some logic at a late phase can invalidate early architectural performance simulations, dynamic validation suites, formal verification proofs and RTL accuracy (due to back-annotation issues) under many current methodologies. Thus, methodologies will have to evolve to handle sequential optimization seamlessly during post-RTL design.

Sections 4.1 and 4.2 describe how each of the post-RTL design problems is changing in a fundamental way that cannot be handled with incremental changes to current algorithms. The huge fraction of post-layout repeaters implies that *mere capacity improvements to today's physical synthesis technologies will not suffice*. We expect that part of the scaling challenge will eventually be met at the algorithmic level by tackling the new versions of these design problems, while the rest of the design gap will be filled by improved design methodologies as well as by migrating towards more interconnect-friendly architectures.

4.1 Logic Synthesis and Technology Mapping

There have been many recent attempts to integrate logic synthesis and technology mapping with placement (that approximate global routes by bounding boxes or trees) (e.g., [10,11]). However, although this physical synthesis tries to comprehend the capacitive load of wires, the metrics that drive it are still the traditional literal or gate counts and fanout-based wire load metrics, sometimes augmented with estimated wirelengths. But these metrics do not take into account the large numbers of repeaters that will be required by the interconnects. This is particularly true in the frequent use of the number of logical levels as a predictor for the eventual delay of the longest path through a logic cone during performance-driven logic synthesis. With more and more of the delay migrating to the repeated interconnect, the gate count metric can lead to wrong heuristic choices during early synthesis. Even in terms of the area metric, it is often the wiring (rather than the gate count) that determines the area of a block, and the correlation between the block area and the gate count available during logic synthesis becomes even weaker as the proportion of repeaters grows. Furthermore, fanout-based load metrics can be very misleading because of the isolation of some of the sinks of an interconnect from its driver by repeaters. Thus, the current trend of building placement awareness into synthesis algorithms is necessary but not sufficient; these algorithms will also need to understand congestion aware routing and its buffering. However, a naïve integration of full-fledged global routing within the inner loop of placement is prohibitively expensive. A way to break this impasse may be through top-down constraints to control the load variation and keep it predictable.

Another aspect in which the nature of logic synthesis changes in response to frequency scaling is in the amount of logic available for optimization in a single pipe stage. With the number of logic levels per pipe stage shrinking dramatically, the maximum possible benefit of a good logic synthesis solution also reduces. However, at the same time, the smaller size of the set of all possible combinational synthesis solutions means that the solution space can be searched far more thoroughly at the same runtime cost as earlier. In this deeply pipelined regime, the major benefits arise by synthesizing across sequential boundaries, although this opens the methodological can of worms described earlier.

At a more abstract level, the increasing dominance of repeated interconnect delays and the difficulty of efficiently getting signals to processing logic opens up several new trade-offs between encoding/decoding effort and communication channel width. Dense encodings become more appealing, whereas 1-hot encodings that are currently preferred because of their device-centered delay optimality lose some of their attractiveness. It starts

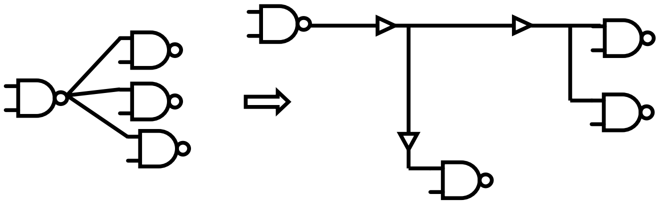


Figure 9. Fanout isolation by repeaters.

making sense to replicate processing logic for which the output bit count exceeds the input bit count (e.g., decoders), at each of the sinks where the data generated by this logic is consumed (in order to reduce the interconnect overhead). At the same time, deep and narrow logic may be better implemented by spreading it out over its communication channel (by distributing the processing over the repeaters within its communication channel).

Pre-layout sizing of gates ceases to have any significant impact on the path delays because of the isolating effect of repeaters. Experiments show that the delay of a net is increasingly insensitive to pre-layout (and pre-buffering) variation in the size of its driver and receiver, and this insensitivity grows with longer wire lengths as well as with process scaling. This argues for an iterative approach in which a coarse layout is obtained for the critical paths in the partially synthesized circuit prior to their first-pass mapping and sizing, and the logic, sizing and layout of the circuit is then refined successively.

4.2 Placement and Routing

The post-RTL design stage that sees the biggest impact due to the exploding repeater count is that of placement. Observe that the number of repeaters required by an interconnect is strongly dependant on the placement of the cells. Furthermore, although one may statistically estimate the total number of repeaters required in a block, the exact set of nets that will require the repeaters (and the number of repeaters required by each of them) is extremely hard to determine prior to placement. Current placement algorithms handle repeaters by reserving a certain fraction of the block area for repeaters prior to placement, and then inserting them into long nets using ECOs after the placement. However, empirical evidence shows that ECO techniques break down when greater than 5-10% of the nets in the netlist are changed. In such cases, it is better to do the placement again from scratch – but this results in a totally new set of nets that now require repeaters. Thus, in a regime where up to one third of the nets within a synthesizable block require repeaters and three-quarters of the cells are repeaters that did not exist prior to placement, repeater insertion will no longer be viable as a post-placement afterthought. Instead, it will have to be inserted into the very core of the placement engines. This is easy to achieve in placement engines based on simulated annealing [12], but the runtime cost of doing so becomes prohibitive. On the other hand, it is harder to incorporate robust buffering at the required scale into the more widely used mathematical formulations for placement based on force-directed quadratic programming [13] or recursive multi-level partitioning [14,15]. These algorithmic approaches will also need to be augmented by improvements in the ability of placement engines to do robust hierarchical placement in order to estimate the interconnect lengths and repeater demands through successive refinement, without sacrificing the placement quality excessively. Further placement complications arise from the fact that the block placement engine at any level of the hierarchy now needs to deal with obstructions arising from the repeater requirements for nets at other levels of the hierarchy – and several of these levels may not yet have been implemented in the layout. This requires improvement in the ability of placement engines to handle blockages as well as a widely varying range of placeable block sizes.

A completely new level of complication is added to placement when the critical sequential lengths shrink below the dimensions of the synthesizable block. At that stage, the placement engine has to implement the clocked repeater insertion strategy also. However, in contrast to mere buffering, the insertion of clocked repeaters has many methodological implications, as mentioned at the beginning of Section 4. One can either opt for latency-constrained placement, or else allow the placement engine to optimize the cycle latency of the interconnects on the fly. While the former option bypasses much of the methodological concerns, it does so at the cost of micro-architectural sub-optimality caused by conservative predictions of interconnect latency prior to design implementation. Furthermore, the span of any interconnect is now upper-bounded because of the latency constraints. Although this is superficially similar to the problem of wire lengths constrained by timing budgets, the sequential constraints are far more rigid. A timing budget that is hard to satisfy locally can be relaxed by optimizing some other part of the path that lies in a less critical block. However, such an option is not available with cycle latency constraints – any local relaxation of these constraints results in a different micro-architecture. Therefore, issues like accurate prediction of path topology, layer assignment, and path routing become much more important during placement because one can no longer rely on the averaging effect over the length of the path to reduce the error in these estimations.

The other approach one can adopt is that of allowing variable latency on the interconnects. While less sub-optimal than the pre-determined interconnect latency case, this requires overhauling the methodology to support post-RTL micro-architectural changes (along with its impact on architectural performance prediction, dynamic validation, ATPG, formal verification and RTL accuracy). Designing the clock tree at the synthesizable block level becomes a challenge because the total number and locations of the clocked repeaters cannot be finalized until after the placement. In fact, going down this path, one soon begins to enter the domains of delay-insensitive and asynchronous designs [16] –

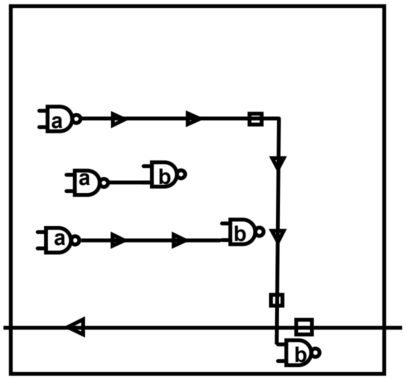


Figure 10. Dependence of repeater and clocked repeater requirements of an interconnect on the placement.

domains that require considerable expertise and CAD support to which most commercial designers do not have access.

On the routing front, the primary impact is that routers can no longer operate in a purely geometric world (even if augmented with parasitics and simple delay models), but must understand buffering intimately. Once repeaters have been inserted, the subnets that the router must handle are short, two-pin segments. At the same time, the large number of repeaters leads to a correspondingly large number of via blockages. Thus, while detailed routers can avoid many of the complications caused by multi-pin nets, they must become better at handling a large number of blockages. Furthermore, the increasingly resistive wires and vias imply that the power grid will have to become fine-grained (on the lower metal layers due to cell design and local routability issues, and on the upper layers due to inductive noise concerns). This implies that the routing process can now be thought of in templated terms rather than as a generic maze search. There are several advantages to this view of routing. Since each of the power grid lines is effectively a shield available for free, routing algorithms can control the switching capacitance of critical nets by intelligent track assignment [17]. In fact, the permutation problem underlying the noise-driven ordering of nets reduces to a simpler bin assignment problem, as does the global topology embedding problem. Finally, there is a need to develop yet higher abstractions of the routing process in order to enable *accurate* estimation of the eventual repeated interconnect topology and route within the placement process itself.

4.3 Chip Assembly and “Small” Blocks

So far, we have assumed that the non-cache area of the synthesizable blocks remains invariant across process generations. However, the synthesis challenges can instead be abstracted to the level of the assembly of the physical hierarchy of the chip. This approach, presented in [18], assumes that size of each synthesizable block is kept small enough that wires within the block do not become a problem, thus enabling successful block design with current synthesis tools. Therefore, because of the rapidly shrinking lengths beyond which wires cannot be ignored, the area of the synthesizable block also keeps shrinking with each process generation under this methodology. However, since the overall non-cache chip size does not shrink, the number of blocks to be managed and interconnected must grow cubically every process generation. This approach leads to a huge increase either in the levels of hierarchy or the number of blocks that need to be assembled together at the full chip level. Traditionally, it is the full chip level assembly methodology that has been the most problematic aspect of doing large designs. The assembly of a large number of blocks results in much loss of quality because of fragmentation of paths and difficulties in resource budgeting, whereas the creation of many levels of hierarchy results in much loss of quality due to lack of visibility across the hierarchy levels. Thus, pushing the worsening interconnect problem into what is already perhaps the hardest aspect of design does not seem like a winning strategy. Even if the hierarchical techniques in use today in custom design can be enhanced to handle the data management aspect of this problem, interconnection of these blocks is a concern that runs (literally) orthogonal to hierarchy. The number of repeaters and clocked repeaters grows rapidly on the inter-block interconnects and must still be implemented – either within the blocks themselves or in separate repeater banks that will soon dominate the entire chip (so that the small synthesizable blocks end up as islands in a sea of repeaters). While some new abstractions have been proposed to bypass this problem, they do not seem very practical. For instance, the inversely scaled global interconnect architecture proposed in [19,20] assumes that the number of global interconnects is small enough to permit using "fat" wires. This runs counter to our experience: designs tend to be metal-limited, especially on the upper layers used for global interconnects. Indeed, [21] shows that such architectures require an unrealistic number of metal layers. Thus, although the “small block” approach bypasses the post-RTL design challenges described earlier, its non-scalability makes it infeasible. Therefore, the new post-RTL design challenges described in Section 4.1 and 4.2 cannot just be wished away, and must be tackled with new algorithmic and methodological innovations.

5. CORRECT-BY-CONSTRUCTION DESIGN

Correct-by-construction (CbC) design refers to the downstream enforcement of specifications used in early design through top-down constraints. Examples include prerouted busses, repeater grids, power planes and PLD-based architectures. The core philosophy underlying CbC design is that of *trading off optimality for predictability*. While this is not a profitable trade-off when the relevant solution space can be searched efficiently, the increasing complexity of the post-RTL solution spaces outlined in Section 4 makes it a feasible alternative to the scattershot solution sampling that heuristics must resort to if their runtimes are to be kept feasible. A promising sub-space can be explored more profitably than sampling a larger, higher-dimensional solution space. For instance, restricting buffered net topologies to S- or P-trees only can often be as effective as the more general SP-trees at a fraction of the runtime [22]. Furthermore, the restricted transformations of CbC approaches yield designs that are more predictable. A collateral advantage of this predictability is that high level optimizations become more effective because of decreased error margins in the estimations used by them. Another advantage of CbC approaches is that, ideally, they *break the design-verification loop* by guaranteeing that the design implementation will meet the predefined specifications. CbC design can be thought of as a sequence of small, guaranteed-correct design transformations, in contrast to the more widely prevalent construct-by-correction methodology that encourages integration of different phases of the design process in large iterative loops. CbC approaches benefit design convergence considerably by avoiding unpleasant surprises in previously non-critical metrics at the end of an unconstrained optimization of the current bottleneck. CbC design techniques *avoid the micro-engineering of every wire and device* that can become prohibitively expensive in custom design (where the sheer volume of fixes can overwhelm the designers) as well as in automated flows (where the simplification of the solution space allows for more efficient optimization algorithms), thus directly addressing the design productivity gap[1]. Thus, it is evident that CbC design techniques are a worthy goal to strive for.

However, past attempts at CbC design have usually resulted in rigid fabrics that sacrifice excessive resources for predictability.

[1] The number of available raw transistors increases by 58% per year, whereas the designer’s capability to design them grows by only 21% per year [7].

Thus, regular logic architectures (e.g., [23]) have not proven very effective for high performance design. Another good example of such rigid fabrics is the dense wiring fabric proposed in [24]. While it does make the load driven by any gate completely predictable by eliminating both switching capacitive and inductive noise, it does so at the cost of much of the routing resources (by making every alternate wire a shield) – a cost that is hard to justify in wire-limited designs. However, one can apply CbC principles without being tied down with a rigid structural fabric. Thus, [25] abandons some of this rigidity while successfully raising the level of abstraction of the interconnects to focus on the throughput of busses, whereas [26] lays out a interconnect fabric that classifies each net by criticality into buckets parameterized by pitch, spacing and buffering and shielding strategies, thus avoiding the micro-engineering of every interconnect while still maintaining flexibility. Most noise problems can be fixed using repeaters with the help of top-down CbC constraints on maximum permissible wire length that are not too resource-intensive or too hard to build into standard backend algorithms (unlike [24]). Another example of a widely used rigid fabric is a regular power grid. In contrast, [17] allows the block-level power grid to be non-uniform while still avoiding on-the-fly verification of the grid. It does so by enforcing a "minimum power pitch" abstraction that is pre-characterized to be correct. Indeed, the most promising approach to apply CbC principles to high-performance design seems to be at an abstract level, in what can be thought of as "dirty" or "loose" fabrics. There has also been some recent promising work [27] on the development of predictability as a design metric.

5.1 Block Construction

As mentioned earlier, the most promising approach to the repeater prediction problem during post-RTL design seems to be that of successive refinement. In this approach, the design would be clustered after early logic synthesis, and the area of various clusters estimated. The clusters would then be spread out on the placement area with sufficient sparseness to be able to accommodate the subsequently added repeaters. All *a priori* latency constraints would be incorporated into the coarse placement at this level. The physical separation of the clusters would be used to plan the routes (including layer assignment, buffering and possibly the shielding) for the inter-cluster nets. These buffered nets would, in turn, be used to continue the logic synthesis and repeater-aware technology mapping of the clusters with realistic loads. We expect that this successive refinement process would iterate between logic synthesis/technology mapping and placement a few times, with the clusters being divided further after each converged iteration, until all the cells and repeaters have been placed successfully. While this approach is sub-optimal because cell placements are restricted even before the cells have been created during the synthesis and mapping of the clusters, it allows the cluster implementation to be more predictable because of the realistic, controlled loads that the route planning enables. This also helps break the design-verification loop, replacing one big synthesis-techmap-placement-buffering-routing iteration with a sequence of "light" iterations, each of which refines the design incrementally, but hopefully leads to the final design without any unpleasant surprises. Thus, without actually using any CbC structural fabric, this approach embodies the core principles of CbC design. There has been some early work in this direction (e.g., [10]), but it needs to be extended with a consideration of extensive unclocked and clocked repeaters.

5.2 Chip Assembly

Mis-predicted cycle latency of global interconects is even more damaging than such mis-predictions within blocks. In order to avoid this, one can use conservative "staging" (i.e. more interconnect flops than necessary) to make the design tolerant of early estimation errors in block area or global routing congestion and the inevitable contortions of chip assembly. This trade-off of predictability for optimality cannot be blindly applied on all global interconnects because of excessive performance degradation, but must be selectively applied to the most critical global routes. Furthermore, design methodology should be extended to support cycle latency *ranges* for the next-most critical set of global interconnects. Enabling early decisions on the cycle latency ranges for the global interconnects based on their anticipated criticality not only can ease their design effort by avoiding excessive local optimizations, but can also avoid downstream surprises because of infeasible cycle latency constraints (hard latency constraints are easier to satisfy for a few nets that can be given preferential treatment, but this is infeasible for a set that contains a significant fraction of all the global nets).

Another application of CbC design at the global level is in the use of fabrics that allow movement of underlying blocks without their having to be resynthesized. Such fabrics currently include power and occasionally clock grids, but can be extended to the grids for the communication requirements (viz., repeaters and clocked repeaters) of the global interconnects also. They would allow the design to become more tolerant to errors in early estimation by enabling offending blocks to unstitch from the grid, move, and then re-stitch and still be correct.

6. CONCLUDING REMARKS

In this paper, we have made the case that repeaters, which are already a problem at the full-chip level, will become critical at the synthesizable block level also. However, since blocks actually implement the logic (unlike the full-chip level that focuses primarily on block assembly and the handling of global nets), the exploding repeater count at the block level has numerous algorithmic implications for post-RTL design. Most of these design problems change in a fundamental way that mere tweaks to existing algorithms or flows cannot handle. If this design gap is to be filled, it will require a combination of algorithmic and methodological advances in the near future. It is in this context that CbC design approaches are a promising alternative. Trading off design quality for predictability, they address the upcoming crisis at both the flow and the engine levels. However, most CbC approaches to date have not been applicable to high performance designs because of excessive resource utilization in an attempt to create a structural fabric. Instead, we feel that viable CbC approaches of the future must abstract the CbC philosophy to the level of heuristics that break the design-verification loop, without necessarily translating it into a rigid structural fabric.

7. REFERENCES

[1] Moore, G.E. Cramming more components onto integrated circuits, in Electron. Mag., Apr. 1965, 114-117.

[2] Cong, J. An interconnect-centric design flow for nanometer technologies. Proc. IEEE, 89(4), Apr. 2001. 505-528.

[3] Borkar, S. Obeying Moore's law beyond 0.18 micron, in Proc. Intl. ASIC/SOC Conf., 2000, 26-31.

[4] Bakoglu, H.B. Circuits, Interconnections and Packaging for VLSI, Addison-Wesley: Reading MA, 1990.

[5] Caldwell, A.E., Cao, Y., Kahng, A.B., Koushanfar, F., Lu, H., Markov, I.L., Oliver, M.R., Stroobandt, D., and Sylvester, D. GTX: the MARCO GSRC technology exploration system, in Proc. Design Automation Conf., June 2000, 693-698.

[6] Davis, J.A., De, V.K., and Meindl, J.D. A stochastic wire-length distribution for gigascale integration (GSI), Parts I and II. IEEE Trans. Electron Devices, 45(3), Mar. 1998, 580-597.

[7] ITRS 2001 edition. http://public.itrs.net.

[8] Kapur, P., McVittie, J.P., and Saraswat, K. Realistic copper interconnect performance with technological constraints, in Proc. IEEE Interconnect Tech. Conf., June 2001, 233-235.

[9] Cocchini, P. Concurrent flip-flop and repeater insertion for high-performance integrated circuits, in Proc. Intl. Conf. Computer-aided Design, Nov. 2002, 268-273.

[10] Salek, A.H., Lou, J., and Pedram, M. An integrated logical and physical design flow for deep submicron circuits. IEEE Trans. Computer-aided Design, 18(9), Sep. 1999, 1305-1315.

[11] Stenz, G., Riess, B.M., Rohfleisch, B., and Johannes, F.M. Performance optimization by interacting netlist transformations and placement. IEEE Trans. Computer-aided Design, 19(3), Mar. 2000, 350-358.

[12] Swartz, W., and C. Sechen. Timing driven placement for large standard cell circuits, in Proc. Design Automation Conf., June 1995, 211-215.

[13] Eisenmann, H., and Johannes, F.M. Generic global placement and floorplanning, in Proc. Design Automation Conf., June 1998, 269-274.

[14] Caldwell, A.E., Kahng, A.B., and Markov, I.L. Optimal partitioners and end case placers for standard cell layout. IEEE Trans. Computer-aided Design, 19(11), Nov. 2000, 1304-1313.

[15] Wang, M., Yang, X., Sarrafzadeh, M. Dragon2000: standard-cell placement tool for large industry circuits, in Proc. Intl. Conf. Computer-aided Design, Nov. 2000, 260-263.

[16] Hauck, S. Asynchronous design methodologies: an overview. Proc. IEEE 83(1), Jan. 1995, 69-93.

[17] Saxena, P., and Gupta, S. On integrating power and signal routing for shield count minimization in congested regions. IEEE Trans. Computer-aided Design, 22(4), Apr. 2003.

[18] Sylvester, D., and Keutzer, K. Getting to the bottom of deep submicron, in Proc. Intl. Conf. Computer-aided Design, Nov. 1998, 203-211.

[19] Sai-Halasz, G.A. Performance trends in high-end processors. Proc. IEEE, 83(1), Jan. 1995, 20-36.

[20] Sylvester, D., and Keutzer, K. Getting to the bottom of deep submicron II: a global wiring paradigm, in Proc. Intl. Symp. Physical Design, Apr. 1999, 193-200.

[21] Davis, J.A., Venkatesan, R., Bowman, K.A., and Meindl, J.D. Gigascale integration (GSI) interconnect limits and n-tier multilevel interconnect architectural solutions, in Proc. Intl. Workshop System Level Interconnect Prediction, Apr. 2000, 147-148.

[22] Hrkic, M., and Lillis, J. Buffer tree synthesis with consideration of temporal locality, sink polarity requirements, solution cost and blockages, in Proc. Intl. Symp. Physical Design, Apr. 2002, 98-103.

[23] Mo, F., and Brayton, R.K. River PLAs: a regular circuit structure, in Proc. Design Automation Conf., June 2002, 201-206.

[24] Khatri, S.P., Mehrotra, A., Brayton, R.K., Sangiovanni-Vincentelli, A.L., and Otten, R.H.J.M. A novel VLSI layout fabric for deep submicron applications, in Proc. Design Automation Conference, June 1999, 491-496.

[25] Lin, T., and Pileggi, L.T. Throughput-driven IC communication fabric synthesis, in Proc. Intl. Conf. Computer-aided Design, Nov. 2002, 274-279.

[26] Kahng, A.B., Muddu, S., and Sarto, E. Tuning strategies for global interconnects in high-performance deep-submicron ICs. VLSI Design, 10(1), 1999, 21-34.

[27] Srivastava, A., and Sarrafzadeh, M. Predictability: definition, analysis and optimization, in Proc. Intl. Conf. Computer-aided Design, Nov. 2002, 118-121.

[28] Cheng, Y., Jeng, M-C., Liu, Z., Huang, J., Chan, M., Chen, K., Keung, K.P., and Hu, C. A physical and scalable I-V model in BSIM3v3 for analog/digital circuit simulation. IEEE Trans. Electron Devices, 44(2), Feb. 1997, 277-287.

[29] Talkhan, E.A., Manour, I.R., and Barboor, A.I. Investigation of the effect of drift-field-dependent mobility on MOSFET characteristics, Parts I and II. IEEE Trans. Electron Devices, 19(8), 1972, 899-916.

[30] Liang, M.S., Choi, J.Y., Ko, P.K., and Hu, C. Inversion-layer capacitance and mobility of very thin gate-oxide MOSFETs. IEEE Trans. Electron Devices, 33, 1986, 409.

Appendix: Process-independent Device Models

One of the main requirements of the device and interconnect simulation models adopted in our studies was the ability of easily scaling the model parameters to represent speculative smaller feature size processes according to specified scaling scenarios and benchmark circuit target performance, maintaining a direct relation between process parameters and device performance. Moreover, another important requirement was easy calibration and integration within state-of-the-art internal simulation engines. To this purpose, we generated a custom behavioral model for MOS devices based on the BSIM3 models [28]. As a compromise in the model accuracy and ease of scalability, only a subset of all second order effects considered in [28] were included. Moreover, since run-time was not critical, we used analytical expressions, when possible, instead of fitted formulae or series expansions, so as to limit the dependency of the model to only a few physically meaningful process parameters with known scaling properties.

At the core of our model lies the following equation for the channel current in strong inversion:

$$I_{ds} = \frac{W_{eff}\mu_{eff}(V_{gs},V_{th})C_{ox}V_{ds}\left(V_{gs} - V_{th}(V_{ds}) - \frac{V_{ds}}{2}\right)}{L_{eff}\left(1 + \frac{\mu_0 V_{ds}}{2\upsilon_{sat}L_{eff}}\right)}$$

This expression handles carrier velocity saturation [29] via the parenthesized term in the denominator as in [28], and mobility, vertical field and surface scattering dependence via the term $\mu_{eff}(V_{gs},V_{th})$ .as in [30]. For simplicity, the threshold voltage is modeled as $V_{th}(V_{ds}) = V_{th0} - \sigma V_{ds}$, where σ is an empirical parameter representing the dependency of Vth on the horizontal field due to more complex effects such as DIBL and channel length modulation.

The model parameters were extracted from available 180nm, 130nm, and 90nm process technologies and calibrated to closely match the performance of a set of simulated benchmark circuits using more detailed in-house simulation models. The scaling trends of the model parameters (including the empirical σ) were computed using these calibration results and used to produce new sets of parameters for speculative process technologies nodes at 65nm, 45nm and 32nm, each one featuring devices approximately 30% faster than the ones of the previous node. The device models were than integrated into a state-of-the-art in-house simulation engine and used in the scaling studies described in this paper.