SuperSized VLSI:
A Recipe for Disaster

Patrick H. Madden
SUNY Binghamton and the University of Kitakyushu
pmadden@acm.org

Designing a large integrated circuit is difficult. As transistor counts increase, the task of designing a SuperSized chip that is two, four, or eight times larger than current chips seems insurmountable. In this paper, we argue that the level of difficulty for such a task is irrelevant–it is unwise to even attempt it. Problems related to interconnect delay are well known, and have become a barrier to larger "monolithic" circuit designs. In response to this, there has been a surge in interest in designs that have multiple relatively independent processing cores; while design size scaling in this manner is sometimes portrayed as a new concept, it in fact has more than thirty years of commercial failure behind it.

In contrast to most semiconductor roadmaps, we argue that total transistor counts on typical chips will remain stable, or even decrease–despite an increase in fabrication capacity. The semiconductor industry is undergoing a fundamental change; design problems cannot be approached as "more of the same, but larger."

Categories and Subject Descriptors

J.6 [**Computer-Aided Engineering**]: CAD

General Terms

Algorithms

Keywords

Placement, Rent's Rule, Interconnect

1. INTRODUCTION

Fabrication density has been growing exponentially since the early predictions by Moore[20]; while progress is not easy, most technology roadmaps show transistor counts continuing to grow for the next decade or so. With design complexity exceeding the abilities of current EDA tools, the challenge of designing next generation chips seems daunting.

In this paper, we argue that next generation chips will not be "SuperSized," bigger and faster versions of current designs. Interconnect delay has become a major barrier, and we present analysis based on empirical observations and computational complexity theory to explain how this barrier came about.

The difficulty of "scaling up" traditional designs is widely known–this has resulted in growing interest in designs that utilize multiple independent processor cores. By breaking the size of design problems down to a more manageable size, and utilizing a high-speed on-chip interconnection network, one might hope to successfully construct a large chip capable of unprecedented levels of computation. While the approach is intellectually appealing, the commercial prospects for the approach are poor.

We do not wish to suggest that there is any major impediment to increased fabrication capacity. The central argument is that there will be extremely few designs (primarily memory and graphics coprocessors) that can use this capacity effectively. For the research community, the relevant question is not how one will design a circuit with size $2X$, but instead, how one should one do a better job with a $1X$ sized design.

2. SUPERSIZE BARRIERS

In this section, we will focus primarily on microprocessor design–but the basic arguments also apply to large Application Specific Integrated Circuits (ASICs). Transistor counts in microprocessors have been growing exponentially for years. Moore[20] covered from 1965 to the mid 1970s accurately. Gelsinger[12] also projected growth with reasonable accuracy from the late 1980s until the late 1990s. For roughly 40 years, a bigger-and-faster strategy has been successful. Recently, however, the strategy has resulted in the delay or cancellation of a number of designs.

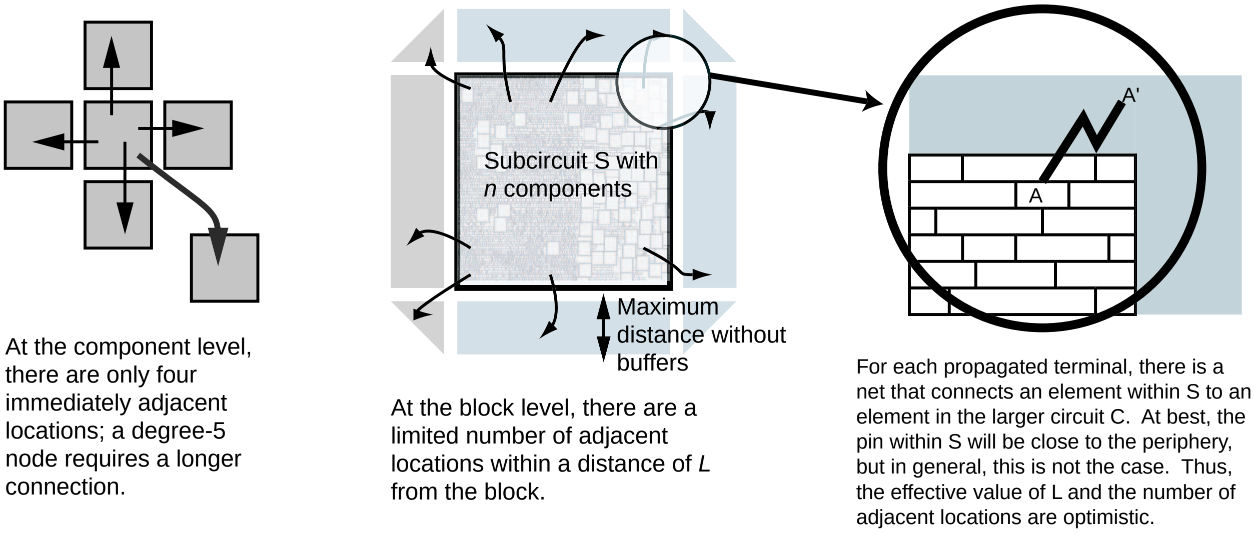
While there are a number of barriers to continued size scaling, one of the most interesting barriers is not dependent on device technology–and thus, we should not expect a fabrication magic bullet to address the problem.

2.1 The Rent Limit

Interconnect delay has been rising steadily, making timing closure more difficult to achieve. In recent years, a debate revolving around interconnect delay and buffering has developed[23]. Interconnect delay now comprises a significant portion of system delay, and growing wire lengths have created a need for extensive buffer and repeater insertion. In this debate, there are two main positions. One group argues that the percentage of cells used as repeaters will reach 70% by the 32*nm* technology node[28]; this is clearly an unacceptable design point. Contrasting arguments suggest limiting block sizes may reduce the impact of the problem[33, 34]. Additionally, there is active research on improved buffer and repeater design, and novel global signaling techniques.

While one might view the interconnect problem as the result of inadequate quality in design tools, the problem has much deeper

EDP'05 April,2005,Monterey,California,USA.



At the component level, there are only four immediately adjacent locations; a degree-5 node requires a longer connection.

At the block level, there are a limited number of adjacent locations within a distance of *L* from the block.

For each propagated terminal, there is a net that connects an element within S to an element in the larger circuit C. At best, the pin within S will be close to the periphery, but in general, this is not the case. Thus, the effective value of L and the number of adjacent locations are optimistic.

Figure 1: As the number of components in a block increases, so does the area. Additionally, the number of "nearby" locations for components also increases. The growth rate is at the square root of the number of components. Each terminal on a subcircuit connects an element within the subcircuit to a component outside the subcircuit–the length of these connections limits achievable design complexity.

roots. Interconnect problems in general are caused by geometry and circuit complexity; we present our analysis here.

At roughly the same time as Moore's projection, an observation was made by F. Rent, regarding the number of "terminal" connections on a piece of computing hardware. "Rent's Rule", $T = k \times n^p$, was reported in 1969 by Radke[26], and by Landman and Russo[18] in 1971. The number of terminals T was seen to correlate with the number of components n within a portion of hardware. The *Rent parameter* p can be viewed as a measure of circuit complexity; the number of connecting terminals on a subcircuit increases as device complexity increases. A constant factor k accounts for component fan-in or fan-out.

The observation by Rent has been confirmed by a number of studies. An early survey on the rule is [10]; more recent surveys and tutorials include [4, 32]. The rule has been used extensively for congestion, wire length, and net degree estimation (for example, [13, 24, 42, 30, 19]), as well as cut sequence optimization for recursive bisection placement[40].

The value of the Rent parameter p is extremely important. In heirarchical board-level decompositions (which led to Rent's observation), p is typically around 0.6. By extracting rectangular portions of standard cell placements, [7] observed p values around 0.7, or higher. [7] also notes that there is a difference between the "circuit topology" Rent parameter and "layout" Rent parameter, with the "layout" parameter being strictly greater.

In [19], Rent parameters of around 0.6 were used for congestion estimation; it was noted that the PEKO[3] benchmarks have unusually low values of p. [35] compared Rent parameters from different placement approaches, with values of p approaching 0.8 for some circuits. In [41], different circuit structures were compared, with many blocks involved in instruction fetch, instruction decoding, and floating point operations having values of p around 0.6. For nearly all the analysis of computing architectures, the value of p is from 0.6 to 0.8. A pipeline may have a Rent parameter of 0, while a perfect mesh has $p = 0.5$; at worst, $p = 1$.

We note that the Rent parameter is architecture-dependent, and may vary within a design. At the highest levels of heirarchy, Rent's Rule can be inaccurate, due to limited numbers of inputs and outputs at the periphery of a chip; one recent work to consider this is [36]. Rent's Rule has proven effective for characterizing circuit interconnect demands at a coarse level; it has been known to be inaccurate for fine details[29].

2.1.1 Rent Limit on System Size

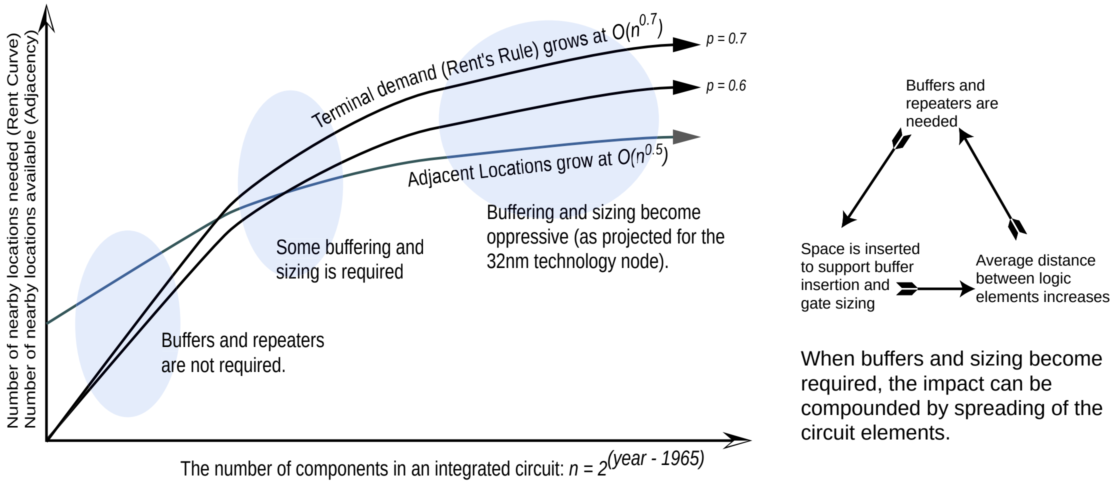
Any computing system for any technology must have a physical representation; the logical elements must be embedded in at most three dimensions. This embedding places constraints on the system, and it is at this point that the value of the Rent parameter p becomes critical.

Our analysis uses an intuitive concept of "adjacency." For any given component in a planar layout, there are four "immediately adjacent" locations–above, below, and to the left and right. For three dimensions, there are six adjacent locations.

Consider a circuit C, which has a subcircuit S that contains n components; we will assume that S has been embedded in a plane, and is roughly square. The number of immediately adjacent locations to S is $4 \times n^{0.5}$. If we assume that the maximum distance that an average size gate can drive a wire is L (which is related to the physical size of the gate), we have $4 \times L \times n^{0.5} + 2 \times L^2$ locations that are within L of the subcircuit S. The number of adjacent locations to a region, and constraints this adjacency places on circuit topology, are illustrated in Figure 1.

Rent's Rule specifies that there will be T terminal connections on S; thus, we have T pairs of components that must be connected. For each pair, one of the components will be located somewhere within S, and the other will be located somewhere in $C - S$.

The impending trouble should be apparent to anyone familiar with the computational complexity work of Hartmanis and Stearns[14]. The basic principle is illustrated by the degree-5 node in Figure 1; the number of components connected together exceeds the physical space–therefore, at least one of the connections must be longer



When buffers and sizing become required, the impact can be compounded by spreading of the circuit elements.

Figure 2: Computational complexity should make the following obvious: if the number of adjacent locations increases at $O(n^{0.5})$, and the terminal demand for the block increases at $O(n^p)$ for $p > 0.5$, the two curves *will* cross–resulting in both high interconnect delay and the need for buffer insertion. Buffer insertion and gate sizing can compound the problem, by further spreading apart the components.

than one unit.

If one has a computing architecture similar to those designed in the past fourty years, the number of terminals T required for a block with n components is $k \times n^p$ for $p \geq 0.6$; this is $O(n^p)$. The number of locations in which an "external" computing element can be placed grows at $L \times n^{0.5} + L^2$, which is $O(n^{0.5})$.

Suppose we have a large circuit that contains a subcircuit of size n. Without question, for some value of n, there will be external connections from a subcircuit that *must* travel a distance of greater than L. As n increases, the percentage of these "long connections" increases. This is true for any value of L, and for any degree of component fan-in or fan-out. This is illustrated in Figure 2.

The value of n is tied to Moore's law. Increasing the transistor count of a system inevitably increases the length of connections until the majority are "worst case," and require extensive buffering and repeater insertion. This is in fact the behavior projected in recent empirical studies from Intel[28][1] We will refer to the necessity of long wires as the Rent Limit on system size. At some point, the benefit gained by increasing the number of computing elements within a system is surpassed by the increased delay of the interconnecting wires. As circuit sizes increase, the percentage of interconnect wires that can be considered "worst case" increases towards 100%, and excessive numbers of buffers and repeaters are required.

The analysis we have summarized here can be viewed as optimistic. First, the number of adjacent locations is made assuming that the connection is made from the periphery of the subcircuit–in general, however, a connecting terminal may be deep within the subcircuit. The components placed outside the subcircuit also have constraints–even if space is available in the adjacent region, it may not be practical to place certain components there.

2.1.2 *Compounding the Problem*

In the past few years, physical synthesis has become a key element of circuit design. Buffer insertion, gate sizing, and repeater insertion, are all widely used. Unfortunately, they can also compound the Rent limit problem. The Rent parameter has traditionally been tied to components that "do something." By inserting buffers into a design, we do not increase the complexity of computations performed, but we do increase the total area.

If we amortize the area demand for buffers and repeaters, the "average size" of each component increases, thereby decreasing the effective value of L. Additionally, the increased average size of components means that wires internal to the block also increase in length–and may themselves need additional buffers and repeaters.

Buffer and repeater insertion, as well as gate sizing, can be viewed as having diminishing returns. As we increase the "average" size of a computing element, the buffer demand increases as well. This matches with the projections of [28].

In some designs, there is a significant amount of "white space[39, 1]" within the design. This is typically used to simplify routing, and to provide space for gate sizing and buffer insertion. One might amortize this white space across the computing elements–and again we see that this results in lower effective L, and the need for further buffering, repeaters, and gate sizing.

2.2 Architecture Observations

The discussion in the prior section may appear a bit pessimistic, and from the "chicken little" perspective[23]. There is also skepticism in the microarchitecture community on the prospects for design size increases.

In 2000, for example, Agarwal[2] projected that conventional microprocessor architectures could not support the annual 50% perfomance improvements projected. Instead, performance gains were projected to be in the 12% range, despite steady progress in device technology.

The lack of improvement is due to a variety of factors (including increased interconnect delay). Clock rate scaling was expected to slow–and this in fact has occurred. Similarly, top end "conventional" microprocessors offer only modest performance improvements over the chips made a few years ago.

[1] In [28], by scaling a design through different technology generations, a rapid increase in the number of repeaters was observed. While the total number of computing logic gates remained constant, the effective value of L decreased, pushing the "crossover" point to the left. The jump in repeater counts shows that the crossover point is not asymptotic behavior for some arbitrarily large design in the distant future, but rather a design limiting factor of current chips.

2.3 The Writing on the Wall

It should be clear from recent design starts that the current state is not business as usual. The impact of the Rent limit has already been felt; the buffer explosion projected in [28] is a symptom of this problem. One might also view the move from CISC (complex instruction set computer) architectures to RISC (reduced instruction set computer) architectures as further evidence of the limit.

There is also broad adoption of multi-core designs, rather than conventional single core designs. This reduces the effect of the Rent limit, as each core has a lower device count, and they are embedded in a mesh structure that has $p = 0.5$, which can be sustained easily.

Multi-core designs are currently being produced a number of companies, including IBM, AMD, and Sun. Intel recently announced a multi-core design[21], and has apparently abandoned a quest towards higher clock rates. The CELL architecture[25], jointly developed by IBM, Toshiba, and Sony, features large numbers of independent vector processors.

Recent attempts to SuperSize conventional CPUs have not been successful. This is not due to inadequate design tools or device technology. Instead, design sizes are approaching fundamental limits based on geometry and circuit complexity. Only a few classes of circuits will be able to utilize the increased capacity of the next few technology nodes.

3. PARALLEL ARCHITECTURES

The prior section paints a bleak picture for further transistor count scaling in traditional computing architectures. The Rent limit makes designing a large single-instruction single-data (SISD) microprocessor progressively less attractive, and in fact many of the major microprocessor manufacturers have moved towards multiprocessor designs.

Multi-processor systems are appealing on a number of levels. They clearly simplify the design process, allowing smaller individual components–this also avoids the “Rent limit” described in the prior section. Through replication of processors, arbitrarily large amounts of silicon area can be utilized. With a well designed on-chip interconnection network, it is clearly possible to construct a “server farm on a chip[27].” In terms of the number of floating point operations possible per second, energy per operation, and a variety of other metrics, parallel computing has many advantages.

The advantages are so clear, in fact, that one might wonder why this approach was not pursued earlier. A brief survey of prior work, however, shows that in fact, parallel processing has been investigated thorougly. Furht[11], in 1994 paper, notes the following: “a decade ago, university researchers were in love with parallel computers, and the US government amorously responded. Those were the days of glory, but times have changed: the market for massively parallel computers has collapsed, and many companies have gone out of business, but the researchers are still in love with parallel computing.”

Almost all commercial ventures into the construction of large scale multi-processor systems have met little success. Perhaps most dramatic was the attempt by Thinking Machines in the mid 1980’s; despite having an experienced architecture team with excellent credentials, as well as computer scientists and physicists of note, the company failed to become profitable. The price and performance advantage of multi-processor systems is undeniable; to date, however, there has been little success in harnessing this power for “general purpose” computing.

While in principle, a computer with X microprocessors can do X times the work of a single processor system, *in practice*, the amount of work that can be performed is far less than that. Most problems that *average computer customers* wish to solve are inherently serial. At most, a single human user might be able to take advantage of a few microprocessors–but not the hundreds or thousands proposed as a means of continuing Moore’s law scaling.

Attempts to develop new software paradigms to harness the computing power of parallel machines have also not met with success. Parallel-optimizing compilers have been in existance for many years, and there have been numerous attempts to develop better mathematical frameworks. As an example, one might consider the programming language *occam*, developed in the early 1990s–carefully designed, flexible, portable, and essentially non-existant in modern computing.

To be clear on this point–there is a seemingly insatiable demand for *single processor* performance. Typical consumers can rapidly harness all the available compute power available of a traditional microprocessors; the demand results in high profit margins for this market. Multi-processor systems, by contrast, have only a handful of customers; we would suggest that the commercial prospects for a “server farm on a chip” are bleak.

The only notable exception, in which one encounters large commercial demand for parallel computing, is in graphics coprocessing. In the rendering of three-dimensional scenes, large numbers of triangles need to be transformed and rendered–this is commonly done in parallel.

4. FUTURE DESIGN PROBLEMS

While much of the discussion has revolved around microprocessors, we note that ASICs in general encounter the same problems–with a few years of lag. Because these problems are based on geometry and circuit complexity, attempts to design extremely large monolithic circuits are almost certainly doomed–thus, the nature of design problems that must be solved will change to reflect the nature of designs that are attempted.

4.1 (Almost) No SuperSized Designs

If one considers current microprocessors, there is a clear trend away from larger transistor count “traditional” single-core designs. This is a significant shift–prior to 2001, large-volume microprocessors were all single-core. We would argue that the “Rent limit” is an underlying motivation for the architectural decisions made by experienced design teams at IBM, AMD, Sun, and Intel – and that these decisions were not made lightly.

An exception to this would be memory chips–which have an inherent Rent parameter of 0.5, and thus do not face the same scaling challenges of more complex circuitry.

4.2 Some Small-Scale Multi-Core Designs

Emerging architectures such as the multi-core designs from IBM, Sun, AMD, and Intel, feature a few processing cores–normally 8 or fewer. There is a need for design tools to handle on-chip communication networks, and methods to assign process tasks to individual blocks. It seems unlikely, however, that these designs will scale to larger sizes.

4.3 Few Large-Scale Multi-Core Designs

By repeatedly increasing the number of processing cores, one might expect to be able to utilize all fabrication potential, without encountering the Rent Limit. A mesh of processors has a Rent parameter of $p = 0.5$, which is easily sustainable.

While there is currently a great deal of enthusiasm for parallel processing, it is perhaps reminiscent of the views of Enslow[16] or Jones and Schwarz[15], from a few decades ago. Enslow, for example, suggested “the future will find multiprocessing, as well as the other concepts of parallel processing, in much wider use than does the present.”

By now, it should obvious that despite numerous attempts, parallel processing has failed to expand much beyond compute-intensive scientific tasks that can be made parallel easily.

4.4 Some Three Dimensional Circuits

Motivated in part by the “interconnect problem,” a number of groups are investigating the fabrication of circuits in three dimensions. While this might appear to improve the “adjacency parameter” from 0.5 to 0.66, this is not in fact the case.

The technologies being investigated (for example, [8, 9, 37, 31]) explore only a limited number of layers. If one considers the number of layers as a constant factor, we clearly regress into the original difficult situation. With k layers, the number of “adjacent” locations improves to $k \times (L \times n^{0.5} + L^2)$, which is still $O(n^{0.5})$.

Three dimensional circuits are clearly advantageous; tools will need to adapt to handle this type of design.

4.5 Non-Manhattan Design

The interconnect length advantages given by non-Manhattan routing architectures are well known[17, 38]. As designs are less likely to be differentiated based on process technology or transistor counts, the gains available through alternative wiring become more attractive. Non-Manhattan design breaks many current tool flows, and might be considered as a last resort; extensive tool development is required.

4.6 Structured Circuitry

The gap in performance between custom design and synthesized design is well known[6]. For human generated designs, there is commonly a great deal of regularity and structure–bit and control lines are spread in regular patterns, as it is not possible for a designer to manage a chaotic semi-random design.

Recently, there has been an interest in generating circuits which can be mapped to a regular structure easily[5]. If the circuit structure matches a physical embedding, the benefits are enormous.

Regular or semi-regular structures initially were quite challenging to traditional placement tools, but some have been adapted to handle the new circuit structures. The PEKO[3] benchmarks, initially used to show suboptimality of placement methods, have a great deal of structure. While initial experiments showed that placement tools were 50% or more away from optimal, this gap has been closed rapidly, and fast methods that are within 22% of optimal[22] (or less) are available.

We need not restrict a circuit topology to be perfectly mesh-like; if most nets are semi-local, global nets are clearly marked, and the synthesis approach considers physical embedding, large scale design can be performed automatically. Circuit performance will be close to that of classic hand-designed datapaths.

Design tools that can perform synthesis with regularity, or that can extract regularity from a design, will be extremely useful.

4.7 Little Big Design

Fabrication capacity will continue to increase–and so the transistor counts that are leading edge today well be commonplace tomorrow. If chip fabrication for designs with a few hundred million gates becomes a commodity market, this will drive down costs. Similarly, it will be possible to extract more dies per wafer, reducing individial die costs.

With stable fabrication processes that are not on the leading edge, current design problems such as variability control or OPC/RET become much less difficult. The performance advantage of a semi-custom design over FPGA implementations are considerable; thus, we anticipate an explosion in designs that are still relatively large, but that target a technology generation that gives a good price to performance ratio. The designs will be small relative to fabrication capacity, but large by current standards.

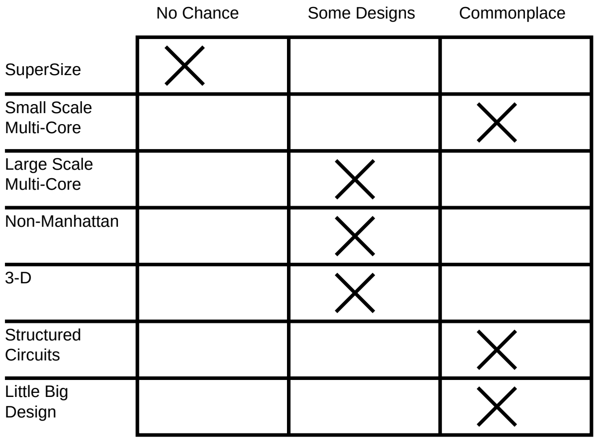
4.8 Summary

Figure 3 summarizes our expectations for future design problems. The wall that has been hit by the microprocessor design community (resulting in multi-core CPUs) will be encountered shortly by the mainstream ASIC community. Thus, there is little future for “supersized” monolithic designs.

Small scale multi-core can be used in average consumer applications, but large scale multi-core designs will be useful only for a handful of applications. There are relatively few customers for supercomputers, and thus there will be little need for single chip server farms.

Non-Manhattan routing architectures, three-dimensional fabrication, and circuit structuring all provide benefit to interconnect delay. Three dimensional fabrication is the most expensive solution, but relatively easy to take advantage of by current design tools. Non-Manhattan routing requires tool development–but there appears to be little impact to fabrication cost or manufacturing difficulty. Structured circuitry offers the potential of large performance gain–but only if the functionality can be mapped to a regular structure.

We anticipate an explosion in the number of “little-big” designs. Rather than pushing the limits of fabrication technology, they will target stable processes that offer lower total gate counts, but with greatly reduced fabrication cost. These designs will not be trivial in size–perhaps a few hundred million gates–but closer to what can be currently handled by design tools. The performance advantages of ASIC over an FPGA implementation will remain large–and with reduced manufacturing cost, there may not be a mass exodus of designs to the FPGA market.



	No Chance	Some Designs	Commonplace
SuperSize	X		
Small Scale Multi-Core			X
Large Scale Multi-Core		X	
Non-Manhattan		X	
3-D		X	
Structured Circuits			X
Little Big Design			X

Figure 3: A summary of future design types.

5. CONCLUSION

In this paper, we have presented a theoretical foundation which explains why large monolithic designs are unattractive. Similar conclusions have been reached in the microarchitecture community. Recent chips from microprocessor vendors indicate that they have voted with their feet, and are rapidly moving away from increased core sizes. Fundamental size limits have been reached, and changes are obvious across the industry.

We would argue that this change is very significant. From 1965 to 2001, the industry used a "single-core" architectural model, in which microprocessors consumed all available transistors. All the resources of each technology generation were rapidly utilized, and there was great consumer demand at each step along the way. From 2001 to current designs, there has been a shift to multi-core designs. With a brief consideration of the history of parallel processing, one finds the unbridled optimism of researchers, engineering teams, and investors, juxtaposed against stunning, relentless, commercial failure.

This is an interesting time for electronic design automation researchers, and for the semiconductor industry as a whole. While further increases in fabrication technology can be made, few designs will be able to use leading-edge technology. The only traditional designs not to face scaling challenges are memory and graphics chips; while not unimportant, they are only a portion of the entire semiconductor market. Microprocessors have moved to dual-core; but further increases in this manner are unwise.

While there is a great deal of concern for how design tools will scale to larger transistor counts–we have reached a point where there are more transistors available than can be effectively used. Thus, rather than increasing capacity, we would argue that increased quality is the proper objective for future research.

6. REFERENCES

[1] S. N. Adya, I. L. Markov, and P. G. Villarrubia. On whitespace in mixed-size placement and physical synthesis. In *Proc. Int. Conf. on Computer Aided Design*, pages 311–318, 2003.

[2] V. Agarwal, M. S. Hrishikesh, S. W. Keckler, and D. Burger. Clock rate versus IPC: The end of the road for conventional microarchitectures. In *Proc. Int'l Symp. on Computer Architecture*, 2000.

[3] C. C. Chang, J. Cong, and M. Xie. Optimality and scalability study of existing placement algorithms. In *Proc. Asia South Pacific Design Automation Conf.*, pages 621–627, 2003.

[4] P. Christie. Managing interconnect resources. In *Proc. System Level Interconnect Prediction Workshop*, pages 1–51, 2000.

[5] M. Chrzanowska-Jeske and A. Mischenko. Synthesis for regularity using decision diagrams. In *Proc. IEEE Int. Symp. on Circuits and Systems*, page to appear, 2005.

[6] W. J. Dally and A. Chang. The role of custom design in ASIC chips. In *Proc. Design Automation Conf*, pages 643–647, 2000.

[7] J. Dambre, P. Verplaetse, D. Stroobandt, and J. Van Campenhout. On Rent's rule for rectangular regions. In *Proc. System Level Interconnect Prediction Workshop*, pages 49–56, 2001.

[8] S. Das, A. Fan, K.-N. Chen, C. S. Tan, N. Checka, and R. Reif. Technology, performance, and computer-aided design of three-dimensional integrated circuits. In *Proc. Int. Symp. on Physical Design*, pages 108–115, 2004.

[9] Y. Deng and W. P. Maly. Interconnect characteristics of 2.5-D system integration scheme. In *Proc. Int. Symp. on Physical Design*, pages 171–175, 2001.

[10] W. E. Donath. Placement and average interconnection lengths of computer logic. *IEEE Trans. on Circuits and Systems*, CAS-26:272–277, 1979.

[11] B. Furht. Parallel computing: Glory and collapse. *IEEE Computer*, 27(11):74–75, 1994.

[12] P. P. Gelsinger, P. A. Gargini, G. H. Parker, and A. Y. C. Yu. Microprocessors circa 2000. *IEEE Spectrum*, pages 43–47, 1989.

[13] C. V. Gura and J. A. Abraham. Average interconnection length and interconnection distribution based on Rent's rule. In *Proc. Design Automation Conf*, pages 574–577, 1989.

[14] J. Hartmanis and R. E. Stearns. On the computational complexity of algorithms. *Trans. AMS*, 117:285–306, 1965.

[15] A. K. Jones and P. Schwarz. Experience using multiprocessor systems – a status report. *Computing Surveys*, 12(2):121–165, 1980.

[16] P. H. Enslow Jr. Multiprocessor organization – a survey. *Computing Surveys*, 9(1):103–129, 1977.

[17] C.-K. Koh and P. H. Madden. Manhattan or non-Manhattan? a study of alternative VLSI routing architectures. In *Proc. Great Lakes Symposium on VLSI*, pages 47–52, 2000.

[18] B. Landman and R. Russo. On a pin versus block relationship for partitioning of logic graphs. *IEEE Trans. on Computers*, C-20:1469–1479, December 1971.

[19] Q. Liu and M. Marek-Sadowska. Pre-layout wire length and congestion estimation. In *Proc. Design Automation Conf*, pages 582–588, 2004.

[20] G. E. Moore. Cramming more components onto integrated circuits. *Electronics Magazine*, 38(8):114–117, April 1965.

[21] S. Naffziger, T. Grutkowski, and B. Stackhouse. The implementation of a 2-core multi-threaded Itanium family processor. In *Proc. IEEE Int. Solid-State Circuits Conference*, page 10.1, 2005.

[22] S. Ono and P. H. Madden. On structure and suboptimality in placement. In *Proc. Asia South Pacific Design Automation Conf.*, pages 331–336, 2005.

[23] P. Osler, L. Scheffer, P. Saxena, and D. Sylvester. The great interconnect buffering debate: Are you a chicken or an ostrich? In *Proc. Int. Symp. on Physical Design*, page 61, 2004.

[24] G. Parthasarathy, M. Marek-Sadowska, A. Mukherjee, and A. Singh. Interconnect complexity-aware FPGA placement using Rent's rule. In *Proc. System Level Interconnect Prediction Workshop*, pages 115–121, 2001.

[25] D. Pham, S. Asano, M. Bolliger, M. Day, H. Hofstee, C. Johns, J. Kahle, A. Kameyama, J. Keaty, Y. Masubuchi, M. Riley, D. Shippy, D. Stasiak, M. Wang, J. Warnock, S. Weitzel, D. Wendel, T. Yamazaki, and K. Yazawa. The design and implementation of a first-generation CELL processor. In *Proc. IEEE Int. Solid-State Circuits Conference*, page 10.2, 2005.

[26] C. E. Radke. A justification of, and an improvement on, a useful rule for predicting circuit-to-pin ratios. In *Proc. Design Automation Conf*, pages 257–267, 1969.

[27] M. Santarini. EE Times article: Cal Berkeley dean predicts server-farm-on-a-chip (http://www.eedesign.com/article/showarticle.jhtml?articleid=51000480, 2004.

[28] P. Saxena, N. Menezes, P. Cocchini, and D. A. Kirkpatrick. Repeater scaling and its impact on CAD. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 23(4):451–463, 2004.

[29] L. Scheffer and E. Nequist. Why interconnect prediction doesn't work. In *Proc. System Level Interconnect Prediction Workshop*, pages 139–144, 2000.

[30] N. Selvakkumaran, P. N. Parakh, and G. Karypis. Perimeter-degree: A priori metric for directly measuring and homogenizing interconnection complexity in multilevel placement. In *Proc. System Level Interconnect Prediction Workshop*, pages 53–59, 2003.

[31] S. J. Souri, K. Banerjee, A. Mehrotra, and K. C. Saraswat. Multiple Si layer ICs: motivation, performance analysis, and design implications. In *Proc. Design Automation Conf*, pages 213–220, 2000.

[32] D. Stroobandt. A priori system-level interconnect prediction: Rent's rule and wire length distribution models. In *Proc. System Level Interconnect Prediction Workshop*, pages 3–21, 2001.

[33] D. Sylvester and K. Keutzer. Getting to the bottom of deep submicron. In *Proc. Int. Conf. on Computer Aided Design*, pages 203–211, 1998.

[34] D. Sylvester and K. Keutzer. Getting to the bottom of deep submicron II: a global wiring paradigm. In *Proc. Int. Symp. on Physical Design*, pages 193–200, 1999.

[35] P. Verplaetse, J. Dambre, D. Stroobandt, and J. Van Campenhout. On partitioning vs. placement Rent properties. In *Proc. System Level Interconnect Prediction Workshop*, pages 33–40, 2001.

[36] T. Wan and M. Chrzanowska-Jeska. Prediction of interconnect net-degree distribution based on Rent's rule. In *Proc. System Level Interconnect Prediction Workshop*, pages 107–114, 2004.

[37] L. Wei, R. Zhang, K. Roy, Z. Chen, and D. B. Janes. Vertically integrated SOI circuits for low-power and high-performance applications. *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 10(3):351–362, 2002.

[38] X Initiative. Home page. http://www.xinitiative.org.

[39] X. Yang, B.-K. Choi, and M. Sarrafzadeh. Routability driven white space allocation for fixed-die standard cell placement. In *Proc. Int. Symp. on Physical Design*, pages 42–50, 2002.

[40] M. C. Yildiz and P. H. Madden. Improved cut sequences for partitioning based placement. In *Proc. Design Automation Conf*, pages 776–779, 2001.

[41] P. Zarkesh-Ha, J. A. Davis, W. Loh, and J. D. Meindl. Prediction of interconnect fan-out distribution using rent's rule. In *Proc. System Level Interconnect Prediction Workshop*, pages 107–112, 2000.

[42] T. Zhang and S. S. Sapatnekar. Optimize pin assignment for lower routing congestion after floorplanning phase. In *Proc. System Level Interconnect Prediction Workshop*, pages 17–21, 2002.