The Impact of the Nanoscale on Computing Systems

Seth Copen Goldstein
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
Email: seth@cs.cmu.edu

Abstract— **Nanoscale technologies provide both challenges and opportunities. We show that the issues and potential solutions facing designers are technology independent and arise mainly from shrinking device sizes and an increase in the number of devices available. We explore how it is possible to use some of the devices that will be available to help ease design complexity as well as overcome process related challenges such as limited layout freedom, increased defect densities, timing constraints, and power dissipation.**

I. Introduction

Many challenges exist in the creation of complex artifacts with nanoscale features. However, there are also enormous opportunities. The challenges and opportunities arise not from a specific implementation technology, but come from the length scale. The challenges fall into two basic categories: changes in physical processes (from small feature size) and increases in complexity (from numerous devices). The question we must answer is how do we harness the massive number of very small devices[1] to improve computing systems without succumbing to the increased complexity inherent in working at the nanoscale?

As feature sizes shrink, manufacturability will have an ever increasing impact on the kinds of circuits that can economically be produced. In the realm of traditional CMOS, a combination of factors will converge to favor regular layouts, thus reducing a designer's freedom to create arbitrary circuits. Parametric variation will cut into the expected gains from scaling. Mask costs will drive out all but the highest volume designs. Alternative technologies, such as molecular scale electronics (MSE) offer no quick fix. MSE, if successful, should be able to create systems with less than 10nm pitch. However, the MSE fabrication methods will probably severely restrict layouts. In addition, defect densities may be significantly higher than what we expect today. On the positive side, both technologies incur these penalties because they produce many more devices per unit area. In Sections II and III we examine how reconfigurable fabrics and asynchronous circuits can use some of the devices to overcome the manufacturability issues mentioned above.

A potentially even more important limit is power consumption and heat dissipation. In Section IV, we examine how the massive numbers of small devices can be used to help reduce total power consumption while also making power densities more uniform.

In addition to the physical constraints imposed by future manufacturing processes, there are market pressures which will continue to exacerbate the "design productivity gap." Demands for increased functionality and faster time-to-market combined with the increases in the number of available devices could make the design problem, if not the verification problem, all but intractable. This is compounded by the need to remove traditional abstraction boundaries to overcome manufacturability issues such as the ones mentioned above. Novel nanoscale technologies bring their own set of complications. Almost all proposed technologies include CMOS, even if the CMOS is used only for support or interface functions. In addition, novel technologies generally include some devices with unusual I-V curves and/or state. In Section V we address how high-level CAD tools can be used to reduce the productivity gap and insulate the designers from changes in the underlying technology.

In the far future, it may be possible to engineer inexpensive nanoscale electronics by precisely placing individual atoms at deterministic locations. Such technology, if it is ever realized, would allow complex irregular circuits to be fabricated with devices which were all have identical properties—eliminating all of the problems related to physical processes addressed in this paper. This paper restricts itself to the near future, during which decreases in feature size will inevitably be accompanied by increased problems in manufacturability. We further restrict our attention to classical computing devices (as opposed to, e.g., quantum computing [1]) which use electrons as information carriers (as opposed to, e.g., DNA-based computing [2] or spintronics [3]). An excellent review of other nanoscale alternatives can be found in [4].

II. Logic Substrate

The traditional approach to manufacturing integrated circuits is to start with a silicon wafer, adding layers which define the logic devices and interconnect. Today, this top-down method requires over 35 masks and 700 steps for a 90nm process [5], [6]. An alternative approach actively being investigated is the bottom-up process of molecular scale electronics (MSE) which should one day be able to create circuits with pitches of less than 10nm [7]. MSE relies on bottom-up

[1]By devices we mean the parts that make up a fabricated circuit, e.g., transistors and wires. We use system to refer to an entire fabricated circuit.

0-7803-9254-X/05/$20.00 ©2005 IEEE.

self-assembly of novel devices (e.g., Carbon Nanotubes and various molecules). Both approaches share the characteristic that as features and pitch shrink, manufacturing becomes ever more difficult and designs more restricted.

A. Photolithography

The processes used in traditional top-down manufacturing at the nanoscale influence all levels of the design process. In particular, regular layouts are preferred. For example, as feature sizes become smaller than the illumination wavelength used in photolithography, resolution enhancement techniques (RETs) must be used to ensure printability. RETs are already restricting layouts that are manufacturable [8]–[11]. Chemical-Mechanical Polishing (CMP) is another example of a process/layout interaction [12]. To maintain a uniform thickness of the wafer, CMP requires a uniform layout density [13]. The complexity of generating a reliable mask set which produces reliable chips also limits the ability to create arbitrary patterns of wires [12]. There is ongoing research into solving these problems; for example, restricted design rules [8], [9] and process aware routing [14] aim to reduce the impact of RET on layout. Dummy features added to each layer help to make irregular layouts more uniform for CMP [12]). However, as scaling continues, designs with regular layouts will be favored.

One possible solution to these issues as well as the high cost of mask sets [15] is to share some or all the masks between designs. "Structured" ASICs [16] and FPGAs are two possible approaches. A Structured ASIC is based on a prefabricated (regular) set of logic blocks which can be customized by each user. The customization requires significantly fewer masks than a traditional ASIC [17]. Furthermore, the prefabricated layers are generally the ones which have the highest critical dimensions, removing the most expensive masks from the customization process. SRAM-based FPGAs are even more regular and all customization occurs post-manufacturing—completely eliminating the need for custom masks. Irregular layouts are a natural byproduct of ASIC or custom design. Making layouts more regular generally requires more area and hurts performance. However, at the nanoscale this tradeoff may be worthwhile.

B. Molecular Scale Electronics

Layout restrictions are even more onerous in bottom-up manufacturing processes such as those used for MSE. The key to understanding the restrictions imposed by MSE is that, unlike traditional semiconductor manufacturing, the manufacture and assembly of the devices occur in separate steps. First, the devices are created and then they are assembled into a system. Due to the scale of the individual devices, the assembly techniques favor regular patterns.

We divide the MSE assembly methods into three broad categories: Probe-based methods, nanoimprint techniques, and self-assembly based approaches. Probe-based methods use the tip of a scanning tunneling microscope (STM) or an atomic force microscope (AFM) to write on a surface. One of the most promising of these methods is dip-pen lithography (DPL); which uses an AFM tip to deliver molecules to a surface [18]. This technique can create arbitrary patterns, but is limited by the write time of the probe. Parallel versions of DPL have been demonstrated [19], [20], but either the linewidths increase or the individual probes are not individually addressable. In the latter case, complex patterns may be generated, but each pattern must be replicated by all the probes.

Nanoimprint techniques use a master to stamp a pattern directly into the substrate [21]. E-beam lithography or DPL would be used to create a master with irregular patterns. However, issues relating to lift-off, master creation, and contact printing may, for irregular patterns, limit the pitch to 60nm [22]. On the other hand, printing by superlattice nanowire pattern transfer (SNAP), can create parallel arrays of wires with a pitch of $<$20nm [22]. The SNAP master is formed by layering (through molecular beam epitaxy (MBE)) materials of different hardness. The composite is cleaved and the cleaved edge is used as the master. The softer of the two materials is partially etched, leaving a parallel set of voids on the cleaved edge. Metal can be placed in the voids and then contact printed onto a surface. The resulting wires have a very large aspect ratio.

There are a multitude of self-assembly based approaches, which include flow-based alignment, electro-field alignment, self-assembled monolayers, Langmuir-Blodgett (LB) films, etc. (see [23], [24] for an overview). All of these approaches create highly regular assemblies. For example, 2-D meshes of nanowires have been made by combining self-assembly and photolithography [25]. The nanowires are placed in a fluid which is compressed, which in turn causes the wires to align along their long axis. The nanowires are then transfered, en masse, to another surface. A second set of aligned nanowires is placed orthogonally on top of the first. Photolithography is then used to create individual meshes from the two layers.

A more deterministic approach to self-assembly uses the hybridization of DNA to guide the assembly process [26]. DNA-directed synthesis proceeds by attaching an unpaired DNA strand to a device and the complementary strand to another device. The two stands will hybridize assembling the two devices. This method has been used to join nanoscale devices [27] as well as micronscale devices [28]. A more complex process (based on DNA crossover), has been used to attach a single-wall carbon nanotube FET to a particular location and then use the DNA as a template for the creation of metallic contacts [29]. Another approach uses the DNA to create a template on which to assemble other structures [30]. A final example "programs" self-assembly patterns through the use of DNA tiles [31]. Each tile has multiple single-stranded ends that lie in the plane and act as programmable binding sites. The tiles are designed so that they self-assemble into a desired shape, e.g., a demultiplexor or a 2D-mesh [32]. Additionally, the tiles can be functionalized with molecules in the third dimension allowing them to serve as a scaffold onto which additional structures can be placed [33].

The MSE fabrication and assembly primitives will most readily create highly regular structures. To date, the most com-

plex MSE systems created are crossbars [34]–[36]. Clearly, post-fabrication customization is required to implement custom circuits. By lucky coincidence, a two-terminal programmable non-volatile switch can be implemented at a single crossing of two wires, e.g., [37]–[40]. Furthermore, the state of the switch can be programmed using the same wires as carry the signal during execution of the circuit. Thus, the area overhead associated with solid-state reconfigurable fabrics is not present in their molecular cousins.

C. Defect Tolerance

Defect tolerance is another reason to move to reconfigurable fabrics. As features shrink the expected defect rate will increase. This is true of traditional manufacturing [41] and even more so for MSE. While defect rates for a mature MSE technology are unknown, the stochastic nature of the assembly processes combined with the scale of the devices leads most researchers to believe that they will be significantly higher than we see in today's processes.

Reconfigurable fabrics provide a natural path to defect tolerance. Teramac [42], [43] is an example of a defect-tolerant system. The Teramac was built from essentially untested fine-grained reconfigurable chips—75% of which had some kind of observable defect. Defect tolerance was provided by creating a map of the defects and then using that map to place-and-route designs around the defects. A more recent, if ad-hoc, example of defect-tolerance in reconfigurable fabrics is the Easypath program from Xilinx [44] in which state-of-the-art FPGA chips with known defects can be purchased at a discount [44]. Customers can ship designs to Xilinx which are tested against the chips. As long as the defects in the chip do not affect the customer's design, it is as if the chip were defect-free. A comparison of three defect-tolerance methods shows that if the defects can be mapped, then reconfiguration-based defect-tolerance is significantly better than N-modular redundancy or von-Nuemann's multiplexing [45].

The testing strategy employed by Teramac and Easypath rely on the fact that defect densities are very low. As defect densities increase, deterministic testing (or defect mapping) methods become too time consuming. However, probabilistic mapping methods can reduce the time complexity of testing while still finding almost all of the usable devices [46]. Active CAD research is needed for improving methods to represent the defect map and creating place-and-route tools which can — in the field—quickly place-and-route circuits around defects. Final place-and-route needs to be done in the field so that a single configuration can be shipped for all systems, in spite of the fact that each system will have a different set of defects.

In recent years there have been several proposals for reconfigurable defect-tolerant MSE architectures, e.g., [47]–[51]. They all rely on post-fabrication customization. They are all based on bottom-up assembly of cross-bars.

D. A Higher-Level Substrate

In this section we have shown that independent of the implementation technology, reconfiguration—at some level—is a key enabler for nanoscale systems. It increases regularity of layout which improves (or in the case of MSE enables) manufacturability. Post-fabrication reconfiguration increases yield by allowing "mostly" good chips to be used as if they were defect-free. And, in the case of traditional manufacturing, it decreases costs by sharing the cost of the masks between many different customers. Finally, it decreases time to market. The disadvantages to using reconfigurable fabrics over custom circuits are an increase in the number of devices needed to implement a circuit and a potential decrease in performance. However, we believe that the benefits outweigh the disadvantages, especially since reconfigurable fabrics enable the manufacture of smaller devices and finer pitches.

III. Clocking Methodology

Today, the standard circuit design methodology is synchronous clocked-design. As technology scales, correct clocked design will become significantly harder. Parametric variation will increase as devices become smaller. This forces designers to be overly conservative or to adopt statistical approaches [13], [52], [53]. Some even argue that parametric variation will essentially eliminate the performance gains typically expected when feature sizes shrink [54]. Timing closure, already a problem, will become worse with increased wire delay [55]. This is especially true for global wires, which may take many cycles to cross the chip—it is estimated that less than 2% of the transistors will be reachable in one clock cycle [56]. Defect tolerance through reconfiguration will also exacerbate the problem as defect locations could affect timing due to changes in place-and-route. This is all compounded by the increased design complexity and verification challenges as the number of devices per design grows.

Asynchronous design eliminates the global clock using local control to synchronize operations [57]. Asynchronous methodologies, such as quasi-delay insensitive (QDI) circuits [58], implement the local synchronization by designing circuits which detect when a computation has completed or when new data is ready. This can lead to improved performance as operations are triggered as soon as the data is ready [59]. When the data being processed allows an operation to finish quickly (e.g., multiply by 0), there is no need to wait for the clock. Instead of designing a system for worst case delays, in order to ensure correct data at the clock edge, designers may instead design for the average case. Asynchronous circuits also adapt to changes in their physical environment which will naturally make parametric variation less of a problem. Likewise, wire delay and timing closure becomes less of an issue as QDI circuits are designed to work as long as the gates and wires have an arbitrary, but finite, delay. Of course, realtime deadlines will still require some form of timing verification. Defect tolerant place-and-route is also naturally handled. Since each operation is self-timed asynchronous systems are naturally composable, in that they are correct by construction. This should help to ease design and verification issues. Globally asynchronous locally synchronize design

methods are an example of using asynchronous protocols to ease design of large scale systems. [60], [61].

There is evidence, though not conclusive, that asynchronous circuits can also reduce power consumption. In some sense, asynchronous circuits naturally implement clock-gating. Further power savings arise if the circuits are designed to exploit the data-dependencies in the algorithm [62], [63]. For example, an asynchronous instruction decoder for the Pentium-II exploits the fact that the average instruction length is 3 bytes to reduce latency and power by a factor of two [64].

The main cost of asynchronous design is in increased area. QDI circuits, for example, are often between 150% and 200% as large as synchronous designs. Asynchronous circuits are also perceived to be significantly harder to design. However, we believe this is largely in part because the asynchronous CAD tools are less mature. Furthermore, the composability of asynchronous circuits should allow more reuse. Nanoscale technologies will provide massive numbers of devices, but also increase physical constraints. It is worth using a small fraction of the devices for asynchronous design in order to reduce the constraints that come with the technology.

IV. Power and Parallelism

Power dissipation and heat removal are seen by many as the main threat to the continuation of Moore's law [65], [66]. This is seen in the recent move to dual-core processor designs and a slowdown in the rate of increase in clock frequency [67]. One way to reduce power is to reduce the clock rate by using a spatial computing model which eliminates reuse and virtualization by laying out the circuits in space. This in effect trades off more devices for lower power. In other words, it decreases power by increasing parallelism.

Spatial Computation (SC) is a model of computation optimized for wires at the expense of increased computation units. SC replicates computation units to simplify interconnect, building a system which uses very simple, completely dedicated communication channels. As a consequence, communication on the datapath never requires arbitration; the only arbitration required is for accessing memory. SC relies on very simple hardware primitives, using no associative structures, no multiported register files, no scheduling logic, and no broadcast. Furthermore, each function block is optimized to the context it is used in; allowing significant simplification of many of the units. As a consequence, SC hardware is fast and extremely power efficient. Our implementation of SC converts C programs directly into hardware [68]. Circuits produced by our compiler exhibit substantially better (by more than two orders of magnitude) energy-delay than either simple single-issue processors or aggressive superscalar cores and are comparable in terms of energy efficiency per operation to custom hardware [69].

Spatial computing is a special case of a fundamental result of early VLSI research; that for many computational functions [70]–[72] there is a tradeoff between the implementation area (A) and the time it takes to compute the function (T) of a form such as:

$$AT^{\sigma} = O(n^{\sigma}), \tag{1}$$

where σ has tended to lie between 1 and 2 for traditional circuit design [71]–[74]. σ can be viewed as an indicator of how inherently sequential a circuit or algorithm is. A higher value of σ means that increasing area will not allow the overall time to be reduced. If we fix the size of the computation (i.e. n^{σ} is constant), then:

$$A^{-1} \propto T^{\sigma} \Rightarrow T \propto A^{-1/\sigma} \qquad \text{for } \sigma > 0. \tag{2}$$

This observation describes the area-time tradeoffs that are possible for a planar circuit: within bounds, one can increase circuit area to reduce circuit delay. On the other hand, total computation time is inversely proportional to clock frequency (F), so if we fix completion time we get:

$$F \propto A^{-1/\sigma} \tag{3}$$

In other words, as the area dedicated to a circuit increases, it is be possible to reduce the overall frequency of operation in order to control power usage.

In CMOS, dynamic power is $P = aFC_{\mathrm{L}}V^2$, where a is the activity factor, C_{L} is the capacitance, and V is voltage. Combining this with (3) leads to:

$$P \propto C_{\mathrm{L}}V^2A^{-1/\sigma} \tag{4}$$

In scaled CMOS, switching speed is a function of supply voltage and if load capacitance is held constant, frequency scales with voltage as $F \propto (V-V_{\mathrm{TH}})^{\frac{5}{4}}/V$ [75]. If we fix V_{TH}, then $F \propto V$ [76]. The constant load capacitance is valid for computational models such as spatial computing that exhibit small Rent exponents for which the fanout and interconnect length do not depend greatly on the size of the circuit [77]. We can thus conclude that $P \propto C_{\mathrm{L}}F^3$. Combining this with (3) leads to $P \propto C_{\mathrm{L}}A^{-3/\sigma}$. Since total capacitance is roughly proportional to area, the dynamic power becomes:

$$P \propto A^{(\sigma-3)/\sigma} \tag{5}$$

Thus, as long as $\sigma < 3$, utilizing more area (or equivalently more devices) will result in using less dynamic power [78]. Whether this holds for static power in CMOS or MSE is still an open question. The implicit CAD tool challenge is to determine σ for a particular specification. If $\sigma < 3$, the tool must then the specification to a parallel implementation taking into account increases in wire delay which might arise due to the increased area each unit will consume.

V. System Complexity

Design and verification complexity is already a significant issue: while the number of available transistors grows by 58% annually, designer productivity only grows by 21% [41]. This exponentially increasing productivity gap has been historically covered by employing larger and larger design and verification teams, but human resources are economically hard to scale. With ever more devices and demands for faster turn around time, designers will be hard-pressed to meet demand.

Furthermore, as was pointed out in Section II, additional physical constraints will make the design problem even harder to manage. Indeed, many of the current proposed solutions to these problems require the removal of abstraction barriers, introducing more information about the manufacturing process into the entire tool flow.

Historically, custom design was the preferred method of creating chips. As technology scaled and design cycles shortened, the efficiency of the design process became more important than the efficient use of silicon. Custom design gave way to standard cell design. Nanoscale technologies will push this even farther. High-level synthesis will be essential towards harnessing the nanoscale. The ability to leverage compiler technology and the semantic information in the design will vastly improve designer productivity. The initial cost of this approach will be less efficient designs. However, just as today's high-level language compilers probably generate more efficient assembly code than most humans could create, we believe that as high-level tools become more prevalent, they will eventually out-perform most hand designs most of the time. This will be particularly true as technologies change.

As technologies scale, the amount of process-related knowledge a designer needs is increasing. High-level tools should also insulate designers from technology changes. In the same way in which software developers do not need to know how many registers are in a processor ISA, hardware designers could, for example, be insulated from the particular restricted design rules necessary to avoid forbidden pitches. This will also promote technology independence—allowing designs to more readily be retargeted to new technologies.

VI. Conclusions

The move to nanoscale technology will not be an abrupt change; one could easily argue that it is already here. Instead, it will be the continued gradual shrinking of feature size. This "slippery slope" could lull us into continually adapting our methods, tools, and systems by small, evolutionary steps. However, to get the most out of nanoscale electronics a different approach is required.

Our suggested solution is to leverage the massive numbers of components that will become available to help reduce the problems involved in shrinking feature size; i.e., use some to support reconfigurable fabrics, use some to implement asynchronous circuit methodology, use even more to increase parallelism through spatial computing, and finally, allow for slightly inefficient designs with high-level specification. The result of this solution will be a two-fold gain. First, design time will be shortened. Second, designs will be able to exploit the high-density of nanoscale technologies. In the long run, investments in higher-level tools will continue to pay dividends as the tools improve and we can all take advantage of them.

Acknowledgment

I would like to thank Mihai Budiu, Tim Callahan, Tiberia Chelcea, Mahim Mishra, Dan Vogel, and Iris Bahar. This work was supported in part by National Science Foundation under Grants No. CCR-9876248 and CCR-0205523 by DARPA under contracts MDA972-01-03-0005 and N000140110659 and by the SRC.

References

[1] M. Oskin, F. T. Chong, and I. L. Chuang, "A practical architecture for reliable quantum computers," *IEEE Computer*, vol. 35, no. 1, pp. 79–87, Jan. 2002.

[2] L. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol. 266, no. 5187, pp. 1021–1024, November 1994.

[3] S. A. Wolf, D. D. Awschalom, R. A. Buhrman, J. M. Daughton, S. von Molnar, M. L. Roukes, A. Y. Chtchelkanova, and D. M. Treger, "Spintronics: A Spin-Based Electronics Vision for the Future," *Science*, vol. 294, no. 5546, pp. 1488–1495, 2001.

[4] S. I. Assoication, *The International Technology Roadmap for Semiconductors (2003 Edition)*. International SEMATECH, 2003, ch. Emerging Research Devices. [Online]. Available: http://public.itrs.net/Files/2003ITRS/Home2003.htm

[5] H. Stork, "Economies of cmos scaling," in *ULSI Metrology Conference*. NIST, March 2005, http://www.eeel.nist.gov/812/conference/2005_Talks/Stork.pdf.

[6] W. Trybula, "Technology acceleration and the economics of lithography (cost containment and roi)," *Future Fab Intl.*, vol. 14, no. 19, 2003.

[7] M. Butts, A. DeHon, and S. Goldstein, "Molecular electronics: Devices, systems and tools for gigagate, gigabit chips," in *ICCAD-2002*, Nov. 2002.

[8] L. W. Liebmann, "Layout impact of resolution enhancement techniques: impediment or opportunity?" in *ISPD '03: Proceedings of the 2003 international symposium on Physical design*. New York, NY, USA: ACM Press, 2003, pp. 110–117.

[9] L. Capodieci, P. Gupta, A. B. Kahng, D. Sylvester, and J. Yang, "Toward a methodology for manufacturability driven design rule exploration," in *Proc.ACM/IEEE Design Automation Conf*, June 2004, pp. 311–316.

[10] A. B. K. P. Gupta and C.-H. Park, "Detailed placement for improved depth of focus and cd control," in *Proc. Asia and South Pacific Design Automation Conf.*, January 2005, pp. 343–348.

[11] F. M. Schellenberg and L. Capodieci, "Impact of ret on physical layouts," in *ISPD '01: Proceedings of the 2001 international symposium on Physical design*. New York, NY, USA: ACM Press, 2001, pp. 52–55.

[12] V. Singh, "The importance of layout density control in semiconductor manufacturing," in *Proc. Electronic Design Processes Workshop*. IEEE CS Press, April 2003, pp. 70–74.

[13] P. Gupta and A. Kahng, "Manufacturing-aware physical design," in *Proc. IEEE/ACM Intl. Conference on Computer-Aided Design*, November 2003, pp. 681–687.

[14] H. K.-S. Leung, "Advanced routing in changing technology landscapes," in *Proceedings of the 2003 International Symposium on Physical Design (ISPD'03)*, Monterey, CA, USA, Apr. 2003, pp. 118–121.

[15] A. B. Kahng, "The next implementation fabric: Issues and considerations," *FPGA and Programmable Logic Journal*, vol. 3, no. 7, May 2004.

[16] L. Pileggi, H. Schmit, A. J. Strojwas, P. Gopalakrishnan, V. Kheterpal, A. Koorapaty, C. Patel, V. Rovner, and K. Y. Tong, "Exploring regular fabrics to optimize the performance-cost trade-off," in *DAC '03: Proceedings of the 40th conference on Design automation*. New York, NY, USA: ACM Press, 2003, pp. 782–787.

[17] J. Lipman, "The "missing link" of SoC design – platform and structured ASICs," Editorial, SemiView, an Application Adaptable IC Newsletter. [Online]. Available: http://www.semiview.com/articles/v1-3_missinglink.php

[18] R. D. Piner, J. Zhu, F. Xu, S. Hong, and C. A. Mirkin, ""Dip-Pen" Nanolithography," *Science*, vol. 283, no. 5402, pp. 661–663, 1999.

[19] M. Zhang, D. Bullen, S. Chung, S. Hong, K. Ryu, Z. Fan, C. Mirkin, and C. Liu, "A mems nanoplotter with high-density parallel dip-pen nanolithography probe arrays," *Nanotechnology*, vol. 13, pp. 212–217, 2002.

[20] D. Bullen, X. Wang, J. Zou, S. Chung, C. Mirkin, and C. Liu, "Design, fabrication, and characterization of thermally actuated probes arrays for dip pen nanolithography," *J of Microelectrmechanical Systems*, vol. 13, no. 4, pp. 594–602, 2004.

[21] B. D. Gates, Q. Xu, J. C. Love, D. B. Wolfe, and G. M. Whitesides, "Unconventional nanofabrication," *Annual Review of Materials Research*, vol. 34, no. 1, pp. 339–372, 2004.

[22] N. A. Melosh, A. Boukai, F. Diana, B. Gerardot, A. Badolato, P. M. Petroff, and J. R. Heath, "Ultrahigh-Density Nanowire Lattices and Circuits," *Science*, vol. 300, no. 5616, pp. 112–115, 2003.

[23] G. M. Whitesides and M. Boncheva, "Supramolecular Chemistry And Self-assembly Special Feature: Beyond molecules: Self-assembly of mesoscopic and macroscopic components," *PNAS*, vol. 99, no. 8, pp. 4769–4774, 2002.

[24] B. Amir Parviz, D. Ryan, and G. Whitesides, "Using self-assembly for the fabrication of nano-scale electronic and photonic devices," *IEEE Transactions on Advanced Packaging*, vol. 26, no. 3, pp. 233–241, 2003.

[25] D. Whang and C. M. Lieber, "Nanolithography using hierarchically assembled nanowire masks," *Nano Letters*, vol. 3, no. 7, pp. 951–4, 2003.

[26] Y. Sun and C.-H. Kiang, "DNA-based artificial nanostructures: Fabrication, properties, and applications," in *Handbook of Nanostructured Biomaterials and Their Applications in Nanobiotechnology*, H. S. Nalwa, Ed. American Scientific Publishers, 2005, ch. 5.

[27] C. Mirkin, "Programming the Assembly of Two- and Three-Dimensional Architectures with DNA and Nanoscale Inorganic Building Blocks," *Inorg. Chem.*, vol. 39, pp. 2258–72, 2000.

[28] J. K. N. Mbindyo, B. R. Reiss, B. R. Martin, C. D. Keating, M. J. Natan, and T. E. Mallouk, "DNA-Directed Assembly of Gold Nanowires on Complementary Surfaces," *Advanced Materials*, vol. 13, pp. 249–254, 2001.

[29] K. Keren, R. S. Berman, E. Buchstab, U. Sivan, and E. Braun, "DNA-Templated Carbon Nanotube Field-Effect Transistor," *Science*, vol. 302, no. 5649, pp. 1380–1382, 2003.

[30] H. Yan, S. H. Park, L. Feng, J. Reif, and T. H. LaBean, "4x4 dna tile and lattices: Characterization, self-assembly and metallization of a novel dna nanostructure motif," in *Ninth International Meeting on DNA Based Computers (DNA9)*, June 2003.

[31] P. W. Rothemund, N. Papadakis, and E. Winfree, "Algorithmic self-assembly of dna sierpinski triangles," *PLoS Biology*, vol. 2, no. 12, p. e424, 2004.

[32] M. Cook, P. W. Rothemund, and E. Winfree, "Self-assembled circuit patterns," in *DNA Computers 9*, vol. 2943, no. 91. LNCS, 2004, pp. 91–107.

[33] T. H. LaBean, "Introduction to self-assembling DNA nanostructures for computation and nanofabrication," in *Computational Biology and Genome Informatics*, J. T. L. Wang, C. H. Wu, and P. P. Wang, Eds. World Scientific Publishing, Singapore, 2003, ch. 2.

[34] Y. Luo, C. Collier, K. Nielsen, J. Jeppesen, J. Perkins, E. DeIonno, A. Pease, J. F. Stoddart, , and J. R. Heath, "Molecular electronics random access memory circuits," *ChemPhysChem*, vol. 3, pp. 519–25, 2002.

[35] Y. Chen, G.-Y. Jung, D. A. A. Ohlberg, X. Li, D. R. Stewart, J. O. Jeppesen, K. A. Nielsen, J. F. Stoddart, and R. S. Williams, "Nanoscale molecular-switch crossbar circuits," *Nanotechnology*, vol. 14, no. 4, pp. 462–468, 2003. [Online]. Available: http://stacks.iop.org/0957-4484/14/462

[36] G. Y. Jung, S. Ganapathiappan, D. A. A. Ohlberg, D. L. Olynick, Y. Chen, W. M. Tong, and R. S. Williams, "Fabrication of a 34 x 34 crossbar structure at 50 nm half-pitch by uv-based nanoimprint lithography," *Nano Letters*, vol. 4, no. 7, pp. 1225–1229, July 2004.

[37] X. Duan, Y. Huang, and C. Lieber, "Nonvolatile memory and programmable logic from molecule-gated nanowires," *Nano Letters*, vol. 2, no. 5, pp. 487–490, 2002.

[38] C. N. Lau, D. R. Stewart, R. S. Williams, and M. Bockrath, "Direct observation of nanoscale switching centers in metal/molecule/metal structures," *Nano Letters*, vol. 4, no. 4, pp. 569–572, April 2004.

[39] C. P. Collier, G. Mattersteig, E. W. Wong, Y. Luo, K. Beverly, J. Sampaio, F. M. Raymo, J. F. Stoddart, and J. R. Heath, "A [2]Catenane-Based Solid State Electronically Reconfigurable Switch," *Science*, vol. 289, no. 5482, pp. 1172–1175, 2000.

[40] T. Rueckes, K. Kim, E. Joselevich, G. Tseng, C. Cheung, and C. Lieber, "Carbon nanotube based nonvolatile random access memory for molecular computing," *Science*, vol. 289, pp. 94–97, 2000.

[41] S. I. Assoication, "The international technology roadmap for semiconductors (2003 edition)," 2003. [Online]. Available: http://public.itrs.net/Files/2003ITRS/Home2003.htm

[42] B. Culbertson, R. Amerson, R. Carter, P. Kuekes, and G. Snider, "Defect tolerance on the Teramac custom computer," in *Proceedings of the 1997 IEEE Symposium on FPGA's for Custom Computing Machines (FCCM '97)*, Napa Valley, CA, April 16-18 1997.

[43] J. R. Heath, P. J. Kuekes, G. S. Snider, and R. S. Williams, "A defect-tolerant computer architecture: Opportunities for nanotechnology," *Science*, vol. 280, pp. 1716–1721, 12 June 1998.

[44] I. Xilinx, "Virtex-ii series easypath: Frequently asked questions," "http://www.xilinx.com/publications/products/v2/faq/faq101_easypath.pdf", January 2003.

[45] K. Nikolic, A. Sadek, and M. Forshaw, "Fault-tolerant techniques for nanocomputers," *Nanotechnology*, vol. 13, no. 3, pp. 357–362, 2002. [Online]. Available: http://stacks.iop.org/0957-4484/13/357

[46] M. Mishra and S. Goldstein, "Defect tolerance at the end of the roadmap," in *International Test Conference (ITC) '03*, September 2003.

[47] J. Heath, P. Kuekes, G. Snider, and R. Williams, "A defect-tolerant computer architecture: Opportunities for nanotechnology," *Science*, vol. 280, p. 1716, 1998.

[48] S. Goldstein and M. Budiu, "NanoFabrics: Spatial computing using molecular electronics," in *Proceedings of the 28th Annual International Symposium on Computer Architecture*, June 2001, pp. 178–189.

[49] A. DeHon, "Array-based architecture for FET-based, nanoscale electronics," *IEEE Transactions on Nanotechnology*, vol. 2, no. 1, pp. 23–32, Mar. 2003.

[50] J. Patwardhan, C. Dwyer, A. Lebeck, and D. Sorin, "Circuit and system architecture for dna-guided self-assembly of nanoelectronics," in *Proceedings of the 1st Conference on the Foundations of Nanoscience: Self-Assembled Architectures and Devices*, April 2004, pp. 344–358.

[51] G. Snider, P. Kuekes, and R. S. Williams, "Cmos-like logic in defective, nanoscale crossbars," *Nanotechnology*, vol. 15, no. 8, pp. 881–891, 2004. [Online]. Available: http://stacks.iop.org/0957-4484/15/881

[52] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variations and impact on circuits and microarchitecture," in *DAC '03: Proceedings of the 40th conference on Design automation*. New York, NY, USA: ACM Press, 2003, pp. 338–342.

[53] J. Tschanz, K. Bowman, and V. De, "Variation-tolerant circuits: circuit solutions and techniques," in *DAC '05: Proceedings of the 42nd annual conference on Design automation*. New York, NY, USA: ACM Press, 2005, pp. 762–763.

[54] K. A. Bowman and J. D. Meindl, "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration," *IEEE JSSC*, vol. 37, no. 2, Feb. 2002.

[55] L. Trevillyan, D. Kung, R. Puri, L. N. Reddy, and M. A. Kazda, "An integrated environment for technology closure of deep-submicron IC designs," *IEEE Design and Test of Computers*, pp. 14–22, January–February 2004.

[56] V. Agarwal, M. S. Hrishikesh, S. W. Keckler, and D. Burger, "Clock rate versus ipc: the end of the road for conventional microarchitectures," in *ISCA '00: Proceedings of the 27th annual international symposium on Computer architecture*. New York, NY, USA: ACM Press, 2000, pp. 248–259.

[57] A. Davis and S. M. Nowick, "An introduction to asynchronous circuit design," Dept. of Computer Science, University of Utah, Tech. Rep. UUCS-97-013, Sept. 1997.

[58] A. J. Martin, "The limitations to delay-insensitivity in asynchronous circuits," in *AUSCRYPT '90: Proceedings of the sixth MIT conference on Advanced research in VLSI*. Cambridge, MA, USA: MIT Press, 1990, pp. 263–278.

[59] P. A. Beerel, S. Kim, P.-C. Yeh, and K. Kim, "Statistically optimized asynchronous barrel shifters for variable length codecs," in *International Symposium on Low Power Electronics and Design*, Aug. 1999, pp. 261–263.

[60] A. Iyer and D. Marculescu, "Power and performance evaluation of globally asynchronous locally synchronous processors," in *ISCA '02: Proceedings of the 29th annual international symposium on Computer architecture*. Washington, DC, USA: IEEE Computer Society, 2002, pp. 158–168.

[61] K. Niyogi and D. Marculescu, "System level power and performance modeling of GALS point-to-point communication interfaces," in *Proceedings of the 2005 International Symposium on Low Power Electronics and Design (ISLPED'05)*, San Diego, CA, Aug. 2005, pp. 381–386.

[62] R. Manohar, "A case for asynchronous computer architecture," in *the ISCA Workshop on Complexity-Effective Design*, June 2000. [Online]. Available: http://vlsi.cornell.edu/~rajit/abstracts/async-case.html

[63] L. S. Nielsen and J. Sparsø, "Designing asynchronous circuits for low power: an ifir filter bank for a digital hearing aid," in *Proceedings of*

IEEE, vol. 87, no. 2. IEEE Computer Society Press, Feb. 1999, pp. 268–281.

[64] K. S. Stevens, S. Rotem, R. Ginosar, P. Beerel, C. J. Myers, K. Y. Yun, R. Koi, C. Dike, and M. Roncken, "An asynchronous instruction length decoder," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 2, pp. 217–228, Feb. 2001.

[65] D. J. Frank, "Power-constrained cmos scaling limits." *IBM Journal of Research and Development*, vol. 46, no. 2-3, pp. 235–344, 2002.

[66] M. J. Flynn and P. Hung, "Microprocessor design issues: Thoughts on the road ahead," *IEEE Micro*, vol. 25, no. 3, pp. 16–31, May 2005.

[67] K. Krewell, "Intel's pc roadmap sees double," *Microprocessor Report*, May 24 2004.

[68] M. Budiu, G. Venkataramani, T. Chelcea, and S. C. Goldstein, "Spatial computation," in *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Boston, MA, October 2004. [Online]. Available: http://www.cs.cmu.edu/~mihaib/research/asplos04.pdf

[69] G. Venkataramani, M. Budiu, and S. C. Goldstein, "C to asynchronous dataflow circuits: An end-to-end toolflow," in *International Workshop on Logic Syntheiss*, Temecula, CA, June 2004. [Online]. Available: http://www.cs.cmu.edu/~mihaib/research/iwls04.pdf

[70] C. D. Thompson, "Area-time complexity for vlsi," in *Proc. Eleventh Annual ACM Symposium on Theory of Computing*, 1979, pp. 81–88.

[71] H. Abelson and P. Andreae, "Information transfer and area-time tradeoffs for vlsi multiplication," *Communications of the ACM*, vol. 23, no. 1, pp. 20–23, 1980.

[72] R. J. Lipton and R. Sedgewick, "Lower bounds for vlsi," in *Proc. Thirteenth Annual ACM Symposium on Theory of Computing*, 1981, pp. 300–307.

[73] R. P. Brent and H. T. Kung, "The chip complexity of binary arithmetic," in *Proc. Twelfth Annual ACM Symposium on Theory of Computing*, Los Angeles, CA, USA, 1980, pp. 190–200.

[74] ——, "The area-time complexity of binary multiplication," *Journal of the ACM*, vol. 28, no. 3, pp. 521–534, 1981.

[75] K. Chen, C. Hu, P. Fang, R. Lin, and D. Wollesen, "Predicting cmos speed with gate oxide and voltage scaling and interconnect loading effects," *IEEE Trans. Electron Devices*, vol. 44, no. 11, pp. 1951–57, Nov 1997.

[76] M. J. Flynn, P. Hung, and K. W. Rudd, "Deep-submicron microprocessor design issues," *IEEE Micro*, July 1999.

[77] P. Zarkesh-Ha, J. A. Davis, W. Loh, and J. D. Meindl, "Prediction of interconnect fan-out distribution using rent's rule," in *Proc. Int'l Workshop on System-level Interconnect Prediction (SLIP00)*, 2000, pp. 107–112.

[78] P. Beckett and S. Goldstein, "Why area might reduce power in nanoscale CMOS," in *Proceedings of the 2005 IEEE International Symposium on Circuits and Systems (ISCAS'05)*, Kobe, Japan, May 2005.

[79] A. Edman and C. Svensson, "Timing closure through a globally synchronous, timing partitioned design methodology," in *Proc. ACM/IEEE Design Automation Conference*, 2004, pp. 71–74.