NanoComputing

subtitle

Contents

1 Executive Summary

Over the next few decades, extending or further accelerating Moore's law will become increasingly dependent on the success of nanoelectronics and computers built with nanodevices. As deep submicron MOSFETs are displaced by nanoscale MOSFETs, which in turn are supplemented or displaced by currently emerging nanoscale devices, the way computers are designed is expected to change drastically.

The purposes of this NSF-sponsored workshop were to provide further definition of the problems that need to be solved by Computer Scientists and Engineers to build practical nanocomputers, to build other systems involving nanotechnology, such as sensor networks; and to explore how to effectively engage this community in solving these problems. This workshop builds on the NSF-sponsored workshop in 2001, The Molecular Architecture Workshop.

Nanocomputing, as defined in this report, refers to computing systems which are constructed from nanoscale components. The issues that need to be faced for successful realization of nanocomputers relate to the scale and integration of the components. The issues of scale relate to the dimensions of the components; they are at most a few nanometers in at least two dimensions. The issues of integration are twofold. First, the manufacture of complex arbitrary patterns may be economically infeasible. Second, nanocomputers may include massive quantities of devices. The key to these definitions is that many of the issues facing nanocomputing can dealt with in a technology independent manner and the solutions will work well for many different technologies.

This report outlines a broad set of research agendas which will enable nanocomputing. As much as possible the research was framed in a manner which would make it technology independent. In fact, the report describes a number of problem areas that have common ground between computers built from nanoscale MOSFETs and those built with emerging devices. This will hopefully help encourage computer scientists and engineers to pursue the research in spite of the fact that the underlying technology is changing rapidly. Among the many research areas identified the three most critical to the success of nanocomputing are:

- **Fault and Defect tolerance** At the nanoscale both defect tolerance (in order to reduce the requirements on the manufacturing process) and fault tolerance (in order to compensate for the small devices and wires) are crucial. Solving these problems will enable low-cost manufacture of extremely complex nanocomputing systems.
- **Abstractions** Abstractions are required to control complexity and permit progress to be made at many different levels simultaneously. Practical nanocomputers will entail new abstractions.
- **Device simulation** Better numerical solvers are needed to predict the macroscale behavior of nanoscale devices and ensembles of nanoscale devices.

In addition, important and potentially high impact research problems were identified in a number of other areas such as circuit design, computing and circuit paradigms, computer organization, microarchicture, and compilation.

In order to motivate the community and provide some understanding of the impact of nanocomputing a set of challenge problems were identified. As nanocomputing evolves more complex and more tightly integrated systems will be feasible. The report lists a series of nanocomputing systems which define a set of goals ranging from near term goals of nanoscale memory integrated to traditional logic to a long term goal of a nanocomputing-based sensor that integrates nanoscale memory, logic, sensing, and actuation. It then describes some possible applications that are engendered by such systems. Nanocomputing will create an opportunity to bring computing to the problem. Nanocomputing will continue the process of making computers more useful that began with mainframes which were in another building, to desktops in our offices, to PDAs in our pockets—Nanocomputing will enable computers to move in situ. This will enable applications such as embedded medical diagnostic and treatment.

The report concludes with recommendations that should increase the multidisciplinary activity needed to make progress in this area. It is imperative that computer scientists and engineers become involved or there is a danger that the technology being developed will not prove useful for building computers. One strong recommendation is that next years NSEC RFP should focus on application-driven research, with nanocomputing applications being explicitly identified as one potential area. In addition, NSF needs to establish mechanisms that promote joint work between computer scientists and engineers with nanotechnologists, otherwise the former do not develop a deep understanding of the opportunities and constraints, and the latter are unable to relate their work with the final goal: a working nanocomputer.

Finally, the committee examined some education issues. A near term goal should be to provide graduate students with solid exposure to Nanoscience so that they will be better prepared to solve future problems. A longer term goal should be the development of undergraduate curricula that give students the ability to understand how advances in other fields affect their own.

2 Introduction

We are approaching the end of a remarkably successful era in computing: the era where Moore's Law reigns, where processing power per dollar doubles every year. This success is based in large part on advances in complementary metal-oxide semiconductor (CMOS)-based integrated circuits. Although we have come to expect, and plan for, the exponential increase in processing power in our everyday lives, today Moore's Law faces imminent challenges both from the physics of deep-submicron CMOS devices and from the costs of both chip masks and next-generation fabrication plants. In recent years there has been intense investigation of technologies which hope to replace photolithographically manufactured CMOS as the basis for future computing devices.

One of the main reasons for the successes of the last thirty years has been the ability to effectively design and implement computing systems based on CMOS-based transistors. The relative ease of designing ICs is a combination of many factors including the elegance of the three-terminal CMOS transistor, the successful separation of circuit design and circuit manufacturing, and the combination of fabrication and manufacture inherent in photo-lithography. As VLSI technology pushes ever deeper into the deep-submicron regime designing circuits is becoming significantly harder. In fact, transistors no longer behave near the ideal, circuit design is becoming ever more entangled with manufacturing concerns, and the cost of fabricating the device is soaring.

This report summarize the results of an NSF-sponsored workshop on Nanocomputing. The workshop was an outgrowth of a previously sponsored workshop the Molecular Architecture Workshop held at Notre Dame University in November 2001. At the previous workshop explored some of the many alternatives to CMOS currently being investigated. For example, recent advances in molecular switches lead to the promise of nanocomputers extending Moore's law beyond the end of the CMOS road map. The promise of computation being performed on self-assembled computers promises to help us breakthrough the lithography barriers facing the semiconductor industry. However, One of the main findings of that workshop is that nano-based design methodology needs to be investigated to more effectively promote the science and engineering of these alternative technologies. In this workshop we explored how to more effectively engage computer scientists and electrical engineers in this process, so that research in the underlying technology can continue to make effective progress.

The report is structured as follows. First, we outline some of the potentially important research agendas whose solution is needed to make Nanocomputing a reality. Second, we identify some potential grand challenge problems and applications, that can serve as drivers for motivating researchers. Finally, we discuss structural issues and education.

3 Research Agenda

In this section we outline some of the research directions necessary to realize the potential of nanocomputing. In addition, we identify suitable abstractions that permit research at all levels of the hierarchically to be conducted simultaneously. One of the difficulties in engaging computer scientists and engineers in nanocomputing is that progress at in the device and fabrication areas is rapid. This creates a situation in which researchers may feel that any progress they make may be nullified by future advances in the field. Thus, we have made an effort to identify the abstractions that can promote research on both sides of the abstraction. This is possible because many of the important research questions that need to be answered are related not to the underlying technology, but rather the scale at which the technology operates. In other words, the fact that nanocomputing is based on atomic scale components presents certain challenges that are related directly to the atomic scale and are independent of the final implementation. For example, sub-90 nm CMOS devices share a number of similar attributes to non-traditional nanodevices (e.g. higher parametric variations, greater quantum level effects).

We begin by outlining the attributes that are common to all technologies at the atomic scale. Thus, any abstractions based on these features will apply to technologies as different as photolithography-based CMOS and self-assembled molecular electronics. Using these features we describe a set of abstractions which can be the basis of independent research at different levels in the abstraction hierarchy. We also outline some important research agendas based on some less general, but none the less, common attributes to many different approaches currently being pursued by nanocomputing device scientists, for example, cross-bar architectures.

We also examine some of the issues involved in using nanocomputing to interface with the physical world at the atomic scale. Nanosensors and nano-actuators have the potential for improving the capability of sensor networks, and possibly enable an exciting range of yet-to-be-consider applications. This can greatly extend the potential of Ubiquitous Computing.

In the process of arriving at the abstractions and research directions they enable the committee identified some that overlap with traditional research directions. Others have a lot of uniqueness to nanocomputing.

3.1 Common Features

The ITRS provides an excellent summary of both future nanoscale CMOS devices, and potential devices for continuing electronic scaling after CMOS reaches the 22 nm half pitch node (10 nm gate length), as predicted to occur around 2016. The reader is strongly encouraged to refer to that document [**?**]. Many of the potential research problems that will need to be solved in non MOSFET nano devices share common characteristics with nanoscale MOSFETs. These include the following:

- *Device Modeling complexity.* NanoFETs will require atomic level precise modeling, just like other quantum devices will. Many of the underlying codes and techniques are similar.
- *Higher defect and fault rates.* Several types of defects and faults were explored. Parametric defects refer to transistors behaving differently than expected. Permanent 'hard' defects include failed opens, shorts and bridges. Post-fabrication permanent faults can be caused by phononic disruption, oxidation, etc. Temporary faults can be caused by high energy particles introducing energy, or by statistical quantum effects.
  For example, parametric defect rates are expected to increase in all devices. For example, in nanoFETs, dopant variations, or gate oxide variations on the atomic scale will affect device performance. Molecular devices currently have poor control of contact parasitics.
  However, permanent 'hard' defects are likely to be more prevalent in non-lithographed nanocomputers than those built with lithography
- *High Design Complexity.* Designing chips with billion+ devices is going to present enormous challenges. Though most of these challenges are familiar to the Design Automation and Architecture communities, some highly novel approaches, that exploit some potential attributes of nanoelectronics are likely to have impact in systems designed with both MOSFET and unconventional nanodevices.
- *Dominance of Interconnect.* The impact of interconnect on system performance will continue to get worse, as resistance increases and capacitance stays the same (to a first order). RC and RLC delay in nanoscale interconnect will present tremendous design challenges. Quantum conductivity issues may be important.

It was felt that the first likely commercial use of non MOSFET devices would use silicon and MOSFETs as a foundation to build upon. I.e. Hybrid technology systems are likely to be prevalent for a significant period of time, even after the scaling of MOSFETs slow down or stop. Future design systems and design concepts should recognize this and not treat the design with nanoFET and non-silicon devices as being separate.

3.2 Abstractions

The committee identified a number of layers of abstraction likely to be useful to researchers in this field. Below, we list those layers, briefly define it, and list research problems that need to be solved by Computer Scientists and Engineers. Each section finishes with an example of sample research challenges deserving solution.

3.2.1 Devices

The device as a lumped circuit element approximating a distributed element has been an extremely useful abstraction and is likely to continue as such.

Tools and methods applicable here provide the interface between the physical device technology, and the circuit level. For example, an input might be a synthesized molecule, defined with its contacts, and an output might be a Spice model deck. Another example, would be a tool and methodology that allows a device designer to tune a device towards a specific set of requirements by performing an objective-driven optimization. The tools at this level of abstraction are directly analogous to Technology CAD tools used today for traditional solid state devices (e.g. Silvaco).

A number of unique and difficult research problems need to be solved in order to deliver this type of tool. These include the following:

- *Efficient and Precise Quantum Solver.* Quantum calculations are used to model potential devices and predict their circuit level quantities. However, the algorithms are extremely slow, hindering progress in this area. New numerical methods are needed to solve quantum level calculations faster by several orders of magnitude.
- *Device Abstractions.* Device need & functionality needs to be thought of in different ways than in traditional MOSFETs. e.g. Separate out Switching, Input/Output Isolation, Restoration of logic level (gain), and Wiring/Contact functions in device concepts, e.g. as separate devices. This permits more flexibility when trying to conceptualize useful devices. A similar issue is related to how to handle the new classes of devices in circuit simulators such as Spice. For example, handling devices with abrupt state changes.
- *Programmable devices.* Programmable devices are likely to be more important in future applications than have previously been the case. An example would be a nanoscale fuse/anti-fuse -like device that can be programmed into either conductive or insulating states through the application of voltage pulses at its terminals. Attention should be focused on desirable device characteristics.

Researchers interested in targeting this area, should leverage work already occurring in this community, including that available through the Nanocomputer Network (NCN [**?**]) and the Nanohub [**?**].

One research challenge would be to produce verifiable parameters for a Spice model from a molecule description. Another would be to determine what the most suitable Spice-type models are, and to modify the Spice engine to handle new devices types, especially multi-state devices.

3.2.2 Device Integration

Device integration refers to the process of assembling nanodevices. The processes could vary from photolithography to thermodynamically-directed assembly.

Modeling of interconnect structures and processes will be very different in a number of nanotechnologies than it is today. For example, the dynamics of molecular and nano-particle self assembly. Another example, would be the modeling of contact structures at the atomic level, as will be necessary to understand the interaction between Fermi levels and the molecular orbitals.

There seems to be a need for tools that can assist nanotechnologists in assembling molecules. Such tools will include an optimization and selection tool that takes as input the characteristics of the molecules and the substrate and the assembly objectives and determine the procedure that has to be used to perform the assembly. For example, it could recommend the solvent that has to be used, temperature, and other environment condition. This could potentially save a lot in terms of money and the effort put into performing experiments. Effective visualization could be a highly effective component of such a suite of tools.

One research challenge is to develop tools that can efficiently model the self-assembly of large number of components.

3.2.3 Circuits

This abstraction specifies the circuits designed to perform logic functions, whether at the gate level (e.g. as in an ASIC) or the block level (e.g. an ALU).

This is a difficult topic to define a tight research agenda on, due to the rapidly evolving nature of the underlying technologies. Right now, two major integration approaches are being developed. One common approach is to build regular grid structures such as memories and crossbars connected with two-terminal devices. Another common approach is to totally rely on self assembly (e.g. the Nanocell [**?**] and Virus scaffold [**?**]). These will have a more random nature and will lead to other circuit styles. This is area is expected to evolve rapidly and tremendously. It is important to recognize that it might be possible to map a complex logic function into a crossbar or self assembled structure. In that case, this level of abstraction is really combined with the next and solve together as one problem.

One research challenge would be to come up with a library of functional units for a reasonable fabrication process.

3.2.4 Computer Organization Design

The traditional abstractions used to design and organize computing systems may need to change in nanocomputing systems. This could lead to new computing models or may instead lead to different methods of organizing computer systems within traditional models. The change in the ratio between memory density and logic density indicates that a further investigation of processor-in-memory (PIM) systems. For example, the previous barriers to a wide acceptance of PIM was partly in response to the differences between DRAM processes and logic processes. With nanocomputing, on the other hand, it is likely that memory and logic will both be developed with the same technology.

The change in the device to wire ratio will also have a significant impact computer organizations. For example, organizations that favor local interconnect and highly localized memory distribution are likely to be needed. Many local wires, as opposed to a few global wires, is also going to be important to reduce the impact of defects and faults. Additional architectural changes will be needed to reduce the impact of defects and faults. For example, reconfigurable fabrics support defect tolerance in two ways. First, reconfigurability can be used to reduce the cost of testing. Second, once defects are detected, they can be avoided, by reconfiguring around them.

ISA no longer serves our purpose. It was designed for humans. Instead we need an interface to the architecture which exposes the necessary resources such that compiler can work with it. **TODO(seth):** (Include sense that ISA is too rigid, does not address communication or reliability, and we need to expose something more suitable for compiler)

3.2.5 compiling to hardware

TODO(seth): include info about certifying circuits & scalability

3.3 Cross Cutting Research Agendas

There are a number of research issues which cut across multiple layers of abstraction and which are fundamentally important to future progress in this area. These issues include the following:

3.3.1 Test, Defect Tolerance, and Fault Tolerance

This area is considered by many to be the most important area for consideration by Computer Scientists and Engineers. High levels of both permanent and transient faults are expected in viable nanotechnology systems, including to some extent CMOS ones. Traditional techniques for fault tolerance, error correction, error detection and diagnosis assume much lower levels of faults and defects. For example, modular redundancy (MR) replicates circuits and then reports as the correct answer, the one that a majority of the circuits report. In order to reduce the area-delay penalty that arises from the voting logic used in MR one would like to replicate large circuits. However, MR only works if a majority of the circuits report the correct value. When the fault rate is high, one will need to reduce the size of the circuits voting, otherwise there may never be a majority of voters that agree. Thus, MR alone is unlikely to be viable. We conclude that a combination of techniques will be needed, in other words, no single solution will be able to economically provide the fault tolerance necessary and fault tolerance will have to be handled at multiple levels of the hierarchy.

Fault tolerance at the system level will come from a combination of techniques at many levels. Device designers can begin by developing devices which are resistant to single-event upsets. At the circuit level, circuits employing built in redundancy can further increase fault tolerance. Information could be transmitted between blocks using a noise resistant encoding. Functional blocks could be designed which operate on an alternative codespace, providing built in fault tolerance. Organizations which emphasize local interconnect need to be explored. Architectures could be designed which build in checking functions and at an even higher level, rollback and recovery could be incorporated into the operating system. Intense research needs to be carried out at all of these levels. To pick one example, a fruitful area of research may be to apply the many years of expertise developed in coding theory to computation.

Defect tolerance has it own set of challenges. It has been argued that reconfigurable architectures are naturally defect tolerant. Such architectures can essentially be programmed to test themselves. The key to the success of the self-testing methods is that its time complexity should be independent of the expected defect density of the devices. Developing such fast methods is an important research area. Furthermore, the maps that represent the defects need to be constructed so that they do not themselves require megabytes of memory. Then, CAD tools, e.g., circuit synthesis and place-and-route tools, need to be developed which can implement circuits that avoid the defects of the device.

A research challenge in this area would be to find a linear-time algorithm for detecting faults in a reconfigurable architecture with an expected defect density of 10%. Furthermore, the algorithm should create a defect map which is no more than 0.01% the size of the target being tested.

3.3.2 Alternative Computing Models

The traditional computing models may not be the most effective at harnessing the power of nanocomputing. The massive number of devices available and the ability to integrate memory and logic may allow the realization of computing models that up til now have only had great theoretical potential. Additionally some alternative computing models are inherently fault tolerant in that do not require a series of precise state transitions to perform a computing task. Some example models that might be investigated are: Neural Networks, Cellular Automata, Data flow, stochastic computing (e.g., fuzzy logic, stochastic automata), reconfigurable computing, analog computing, and mead-nueromorphic-physics-computation.

Another non-standard approach would be to move away from binary representations represented by current or voltage. In particular, ultra low power computing could benefit enormously from such an advance. Examples that exist include quantum computing, cellular automata based on electric dipoles, and neural networks. Some of these approaches require advances in nanotechnology in that otherwise they could not operate at room temperature.

A research challenge in this area is to demonstrate an alternative computing model which outperforms for more than one benchmark[1], at least some metrics (cost, area, power, performance, reliable), traditional computing structures. **TODO(jose):** For example, show that a content addressable memory using a neural network-based model.

3.3.3 Intelligent Nanoscale Sensors

Nanotechnology is likely to revolutionize sensor capabilities. Potential examples include airborne particulate sensors that outperform a dog's nose and large area imagers, including optical and medical ones. Given new sensor capabilities, new problems will arise in signal conditioning, sensor fusion, sensor-computer interconnection, etc. For example, will a wall sized imager benefit from local (nanoelectronic) computation at each pixel or group of pixels, so as to avoid highly complex I/O? Will new nanoelectronic device elements be needed for local low-noise signal conditioning? Much of the research in this area will be driven by the emerging sensor capabilities, the likely signal transduction mechanisms, and the application needs.

TODO(seth): Make sure we capture discussion here.

3.3.4 Design Automation

Design automation traditionally relies on solving well defined design problems such as synthesis, place and route, verification, test insertion, etc. Each of these problems generally has a well defined set of objectives, constraints and inputs and outputs. However, even in the conventional design community, new paradigms are emerging that cut through several of these layers simultaneously. Much of this is being driven by the increasingly difficult nature of achieving the goal of performance convergence (meeting the clock frequency target) with the traditional hierarchy. Examples include combined physical and logical synthesis, interconnect-driven floor planning etc.

It is likely that design automation for nanoelectronics will have to include solutions that cut across several layers of abstraction simultaneously. However, the likely goals and metrics for the NanoDA are going to be very different. For example, how should total design synthesis result in a structure that maximizes the performance per unit area in a highly defective unit, including concepts such as reconfiguration etc.

The current embodiment of CAD tools will not successfully scale to handle systems with billions of components. Furthermore they are not focused on defect or fault tolerance. While verification, and other areas in CAD are important

[1] We recognize that there is a huge need for nanocomputing benchmarks to be defined and developed.

and research must continue in those areas, with respect to nanoscale computing we outline the following two (of many) research challenges.

As we have made clear throughout this document, the underlying technology for nanocomputing is changing rapidly. However, CAD development can happen if a reasonable set of abstractions for circuit and devices are developed. For example, a general abstraction could be to parameterize functionality by the area it would take to implement it. The interconnect could be modeled by a parameter for the density of wiring tracks available. The area that a given function takes could be a function of the device technology and the defect distribution. This level of abstraction will enable both CAD tool developers and device engineers to progress simultaneously.

- **place-and-route** Scalable, defect-aware Place-and-route tools. These tools may be required at both design time and run-time. At design time, place-and-route systems need to scale such that placing and routing systems with 10^x components can be performed in y seconds. We envision three scenarios ranging from defect-free targets to targets with unknown defects. In between are reconfigurable targets with known defect maps.

  - **defect-free:** In this case, the challenge is to build tools which can scale to large designs.
  - **defect-mapped targets:** In this case, the target is a reconfigurable device which has been defect mapped. The place-and-route tool will have to map the circuit onto the target device avoiding the defects. This needs to be done without reducing the utilization of the defect-free components by more than 90%.
  - **defect probabilities:** In the final case, the place-and-route tool is given a probability distribution of the potential defects in the target process. It must correctly map a defect-tolerant circuit to the target regardless of the physical distribution of the defects such that it works with high probability. Of course, this requires the development of a defect-tolerant circuit design methodology.

  We realize that the entire CAD process is complicated by the expectation that designs will have to work in and adapt to the presence of defects or multiple task requirements. Furthermore, optimizing designs completely at design time may limit the scalability of the design process. This probably leads to a system solution that requires both static and dynamic components. For example, dynamic place-and-route could be used to: 1) finalize the place-and-route when the circuit is loaded, 2) optimize the global layout in response to changing load or other situations in the field, and 3) can reroute the circuit to respond to permanent failures in the field.

- **Timing simulation** Nanocomputing will present additional timing challenges over traditional CMOS. For example, circuits may have to be more resilient to changes in signaling paths and wire delays. Furthermore, at the nanoscale power becomes a driving concern and global clock distribution may be hard. The development of tools to support alternative design strategies will be essential.

  One alternative example, inspired by biological models, is spike-based signaling. In this approach, logic levels are not used to transmit information, instead it is transmitted by the timing of spikes in the signal. (This model is already being considered in hybrid digital/analog designs, however, no CAD tool research has been performed in this area.)

4 Challenges Problems

In this section we outline some describe some challenge problems which we hope will motivate researchers. We begin by outlining some application areas which would be enabled by nanocomputing. Then, we specify some research problems which need to be solved in order to successfully harness the power of nanocomputing.

4.1 Systems

Nanocomputing, as we define it here, relates to computing devices built from nanoscale components. These components may be defined lithographically, e.g., CMOS with a half-pitch of 22nm, or they may be synthesized in a beaker and assembled using thermodynamically directed assembly, e.g., molecular computing. In either case nanocomputing systems will achieve densities of 10^12 grids/cm^2, with the potential of achieving more than 10^13 grids/cm^2. Thus, independent of the technology we are concerned with computing systems which have high-density of devices.

Building systems at the atomic scale will be entail serious challenges. We outline here a progression of challenges in system design for nanocomputing. In the near term, nanoscale memory is the first and easiest goal to achieve. The longest term goal would be a high-performance low-cost nanoscale processor that could interface with to real world at the atomic scale.

There are many metrics which can be used to characterize the following challenges. We suggest that to meet the challenge at least one metric needs to outperform the ITRS prediction by at least one order of magnitude.. Example metrics are cost, energy/op, ops/cm^2, reliability, etc.

- **Memory:** The challenge is to build a nanoscale memory which can be integrated with existing electronics. This memory should have a realized bit-density of at least 10^{11} bits/cm^2 and should be addressable from the micronscale. It needs to be tolerant of both defects and faults, providing 7 nines of availability (i.e., correct 99.99999% of the time). There are several other metrics that can be used to describe memories, e.g., volatility, power, cost/bit, bandwidth, and latency. To meet the grand challenge nanoscale memories will need to outperform current systems on at least one of these axis by at least an order of magnitude. For example, nanoscale memory may be, non-volatile and more than two-orders of magnitude cheaper than todays disks, but have poor bandwidth and they will still enable significant new applications.
- **Embedded Controller:** Embedded controllers are particularly sensitive to the following five metrics: power (in watts), price (in dollars), volume (in cm^3), performance (in ops/sec), and reliability. The challenge is to build an embedded controller (e.g., an ARM) that that improves at least one of the previous measures by more than an order of magnitude over the ITRS road map. This requires both logic and memory to be in built from nanoscale components.
- **Network Router:** A nanoscale network router has the additional challenge of incorporating high bandwidth I/O at the nanoscale. Furthermore, this requires that the nanoscale interconnect, logic, and, memory to have low latencies.
- **General purpose computer:** The challenge for nanoscale general purpose (GP) computers is twofold. First, it must integrate the technologies of nanoscale memories, logic, and I/O in an organization that balances bandwidth and latency. For example, the attributes of nanoscale memories and interconnect may significantly change the memory hierarchy. Second, the complexity of such a system far exceeds either of the previous three systems.
- **Intelligent NEMS:** The challenge is to devise new ways to integrate the inherently analog world that NEMS interact with with nanoscale computing. Meeting this challenge enables in-situ computing. In-situ computing is computing at the point of sensing and actuation. It enables new applications when it is not possible to bring the information at the point of sensing out to a remote processor. The important metrics for this challenge are logic and memory density, cost, and power. Performance, in terms of ops/sec, is not a key metric. On the other hand, it will be important for the individual systems to communicate and coordinate with each other.
- **Intelligent sub-micron Sensor & actuator** The challenge is to integrate computing and sensing & actuation into a self-contained system. The primary difference between this challenge and the previous one is that the resulting systems act at or below the micron scale, as opposed to acting as an ensemble on bulk materials. The key will be to have sufficient logic in a small enough area to enable the applications of interest. In other words, these systems will have to be self-contained. The important metrics for this challenge are similar to the previous one, but also include reliability.

4.2 Applications

Nanocomputing systems can succeed in one of many ways. We divide the systems into three categories based on whether they outperform the ITRS metrics primarily in terms of performance, cost, or size. Each category will enable different applications.

We can roughly define three different classes of nanocomputing systems based on their overall performance, cost, and size:

- **high-performance computing** If nanocomputing reaches the above density levels and maintains the kind of scaling we expect with today's silicon this will enable supercomputing applications of unheralded price/performance. **TODO(?)** finish this up

- **high-density inexpensive computing** In this scenario we reach a high device density, but the performance may not scale as CMOS scaled. However, due to self-assembly these systems will be very inexpensive. This leads to a new set of applications: e.g., ubiquitous computing ... **TODO(?)** finish this up
- **Micron-scale in-situ Computing** Ability to meld with real world artifacts at the sub-micron scale.

In this report we do not address the first class of nanocomputers since this is simply an extension of current scaling laws. Instead we focus on some of the applications that can be engendered by inexpensive high-density or atomic scale computing systems.

4.2.1 Inexpensive High-density

Inexpensive high-density computing supports previously inaccessible applications on the road towards truly ubiquitous computing. Examples of these applications include:

- Intelligent sensors which can be “painted” onto surfaces.
- Adaptable camouflage.
- Eye-glasses capable face-recognition.

These applications will require nanocomputing systems which are capable of ad hoc networking, power scavenging, distributed resource management, fault tolerance at the system level. Furthermore, programming such systems, with potentially billions of processors, will require new programming models and/or machine learning techniques to produce coordinated global behavior from local specification.

4.2.2 Micron-scale in-situ Computing

Nanoelectronics-based intelligent sensors provide a unique opportunity to interface with the physical world at the submicron scale and below. This is particularly true in the case when it is not possible to move the information from the site to or from a remote processor. For example, this will be particularly useful for interacting with biological systems. While progress needs to be made in many different areas of nanoscience, nanocomputing will, in concert with sensors and actuators, enable interesting applications at this level.

The very nature of some of the technologies underlying nanocomputing supports interactions at the micron-scale. For example, molecular computing uses the properties of molecules to perform computation. Molecular sensors can be incorporated into molecular circuits increasing the efficiency of signal transduction, possible reducing power requirements and decreasing response time.

Body-implantable devices are already available for drug delivery, electrical stimulation, imaging of internal organs and other internal medical purposes. Adding intelligence to these devices enables embeddable disease-specific doctors which perform in-situ diagnosis and treatment. This can only be done if computational resources are integrated with the sensors and actuators.

4.3 Research Challenges

In order to create nanocomputing systems and enable the above applications there are many significant research challenges. Here we describe some of the research challenges in terms of the goals they must meet. We formulate these challenge in an almost technology independent manner. The challenges are engendered by the massive quantity of devices that will need to be integrated and the size scale of the individual devices.

5 Promoting Interaction

Progress in nanocomputing requires interdisciplinary research teams. This is even more true of the applications driven by nanocomputing. We therefore suggest that NSF do more than just encourage multidisciplinary teams. In fact, the consensus of the workshop is that NSF has the unique ability to provide a forcing function which will cause multidisciplinary teams to be formed.

We recommend:

- Require a computer scientist or electrical engineer on the team for proposals concerned with devices and materials for electronic nanotechnology.
- The NSEC request for proposals could focus on applications involving nanocomputing.
- A prize/award that is awarded for meeting particular goals with respect to nanocomputing. One example goals might be the construction of the first sixteen state FSM that fits within one square micron.

Following such recommendations would hopefully have the side effect of bridging the communication gap between the physical scientists and computer engineers. Computer scientist and engineers clearly need to play a key role in the development of successful nanocomputing. However, they cannot do this without a deep understanding of the basic technology, which will only come from working closely with nanotechnologists.

6 Teaching

Nanotechnology is an area in which many computer science and engineering students have little formal preparation, or, in many cases motivation to learn. However, it is increasingly clear that nanotechnology will strongly impact the careers of many of our Ph.D. graduates and is likely to be relevant to a lot of our BS and MS graduates.

It is clear that Universities should have mechanisms to encourage Ph.D. student, even those not doing nanotechnology research, to learn about this area. There are different avenues to achieving this goal, ranging from the design of courses specifically structured for their needs, to recommending that they take a small range of chemistry, biochemistry or physics courses. Focused courses in electronic nanotechnology, generally taught by CSE faculty, are likely to become more important as this area grows.

At this stage, no suitable texts, nor much in the way of structured class materials is available. NSF sponsorship of the generation of such materials, and the conduct of ”educating the educator” type of workshops in this area are strongly encouraged. A direct analogy can be drawn with previous similar NSF efforts in microelectronics design education.

In the longer term, the increasing impact of nanotechnology on mainstream computer design, will lead to a need for our Undergraduates to have a better founding in the basics. This might lead to a re-balancing of the curriculum towards chemistry and physics. However, students will need to be motivated in these courses, by thinking of system that they can build 20 years from now. A simple example would be to take the Robocup example, used at CMU, and change the vision system to a sensor requiring a nanotechnology.

7 Conclusions

Evolving nanodevices and nano-integration technology are going to greatly change how computers are designed, built and used over the next two decades. This report discusses the nature of these changes and the research challenges that arise from them. The solution of several of these challenges will require radically different architectures, methods and tools than those likely to emerge from the current CMOS-driven research agenda. It is important and imperative that Computer Scientists and Engineers establish relationships with nanoscientists and work towards solving these problems in a timely fashion.

A Attendees

The workshop participants included:

- Kamal Abdali, NSF

- Sankar Basu, NSF
- Kia Bazargan, University of Minnesota
- Shawn Blanton, Carnegie Mellon University
- George Bourianoff, Intel Corporation
- Jose Fortes, University of Florida
- Paul Franzon, North Carolina State University
- Seth Goldstein, Carnegie Mellon University
- Konstantin Likharev, Sonybrook University
- David Lilja, University of Minnesota
- Vijaykrishnan Narayanan, Pennsylvania State University
- John Randall, Zyvex, Inc.
- Jorge Seminario, University of South Carolina
- Peter Varman, NSF
- Wayne Wolf, Princeton University

B Procedures

The workshop was hosted at Carnegie Mellon University on October 17^{th} and 18^{th}. Before attending the workshop attendees submitted a one page position statement. On the first day talks were presented along with several breakout sessions. During the breakout sesssions this document began to take shape. On the 18th we finished a draft of this document.