The Technology Lane on the Road to a Zettaflops

Abstract

Given that viable petaflops-level machines are now within sight, the time has come to begin considering how we can reach the next performance milestones. This work spells out the upper limits of performance that currently understood technology might achieve. Our "design target" is a zettaflops - or 10^{21} flops). We explore whether or not a classical approach has any chance of achieving this target of one million petaflops, and find that area and power might very well be insurmountable obstacles even in the absolute best case. We also explore whether or not non-silicon, but still classical, nano-scale devices might move us any closer. In a best case scenario, while significant improvements over silicon are at least possible, a zettaflops still appears improbably ambitious. Finally, we explore the capabilities of reversible logic as it may present one of the few plausible inroads into the monumental power problem that appears before us. Based on these studies, we suggest a forward path for future technology development, and couple it with needed work in architecture and algorithms.

1. Introduction

The future of high performance computing, even as it is energized by the momentum of near-term commercial semiconductor advances, will be challenged by barriers in device physics, fabrication processes, heat, failures, concurrency, cost, and complexity. If we are to move beyond a few tens of Petaflops, and eventually deliver Exaflops of computational capacity to scientists, national defense, and society as a whole, we must understand the roadblocks sure to occur at the hands of Moore's Law. Developing innovative strategies to overcome them at all levels - technology, architecture, and software - is a must. Research in fundamental physics, device fabrication, models of parallel computation and programming, parallel algorithms, system structures and architectures, and user environments must all be redirected if we are to reach an exaflops, let alone a zettaflops. This work attempts to identify what alternative paths and technology research investments should be considered if we are to reach these desired levels of performance.

It has taken more than a decade to move from a teraflops to a petaflops. The teraflops milestone was first surpassed in the early 1990s, shortly after which the start of a long dialog of what it what take to reach 1000x that level began. Initial discussions evolved to a now famous workshop [St95], at least one major project (HTMT: Hybrid Technology Multi-Threaded Computing), and finally a major U.S. Government initiative (HPCS:[1] High Performance Computing Systems). Several major vendors now plan peta-scale systems by 2010.

This paper does not lay out a roadmap to reach the exaflops milestone. Rather, our focus in on an "ultimate limit" of computation: a million petaflops. Zettaflops do not just represent another design point, but quite possibly a bounding limit to what may be possible with currently conceived technologies.

This paper is the first of several that will try to coalesce what is required to achieve a zettaflops. Its primary emphasis is to define the fundamental limits imposed by technology on zettaflops computation. Section 2 reviews the key issues involved in modern HPC systems. Section 3 baselines the twin capacity requirements of storage and computation rate using silicon technology projected out through 2020. The projections of Section 4 then show that, in the BEST case, the silicon for a silicon-based supercomputer will probably require a minimum area somewhere between the size of Vatican City and Manhattan, and would dissipate over 2000 times more power than a present day HPC machine.

Section 5 then explores the idea that there is at least the potential that some emerging nanotechnologies might provide a minimum level of suitability. One such technology for which at least notional estimates can be made is Quantum-dot Cellular Automata (QCA).

Finally, Section 6 considers reversible logic as it

1. http://www.highproductivity.org/

may be one of the few candidates that can improve power dissipation by the orders of magnitude that appear necessary. Section 7 then suggests a forward plan for determining a viable technology approach - and how that might affect algorithms, applications, and programming models.

Future papers will address applications and algorithms and their unique needs at these performance levels, architectures that may be needed to bridge the gap from technology to delivered performance, and system and software issues that may come into play.

2. Key Problems and Issues

Sustained growth in delivered HPC performance within practical constraints is increasingly challenged by a range of factors that, if not addressed, will cause an age of stagnation in computing (comparable to that experienced by the commercial aviation industry). As we focus on a gain of a million-fold in real performance, it may be necessary to undertake the greatest revolution in means and methods seen in more than 30 years with the advent of the original microprocessors.

As discussed in the two recent workshops[1], the overriding problems that must be resolved in order ton continue progress in delivered performance are:

1.Power consumption (energy per operation),

2.Limited feature size headroom as we converge to the nanoscale,

3.Latency for global interaction measured in local clock cycles,

4.Execution and program parallelism to handle the resulting extremes of physical concurrency,

5.Bandwidth - including both system-wide bandwidth and memory access bandwidth - often referred to as the "memory wall" or the "von Neumann bottleneck,"

6.The overhead of managing fine grain parallel resources and concurrent actions.

Power consumption is clearly already a concern and is already causing significant changes to commercial hardware structures (in terms of active power management, multi-core structures, etc.). Both dynamic switching and leakage currents are contributing to this challenge with the latter approaching dominance. Power budget projections for large server farms indicate that we are rapidly reaching the point where the per year costs of electricity rivals the cost of the hardware itself [Ba05]. This alone may doom any system requiring more than the few megawatts that we handle today.

Even if improved fabrication processes and technologies provide short term solutions, energy dissipation in the form of Landauer's Limit imposes an ultimate bound for conventional processing methods and architectural structures. More than any single obstacle, power consumption may constrain systems in both the short term and long term - and may dictate a fundamental change in the basic model we use for logic circuits. Operations per watt, rather than operations per second will be the paramount concern.

Continually declining feature size (especially in DRAM) is the basis of Moore's Law, and is now challenged both by lithographic technology and impending near atomic-scale dimensions (especially the transistor gate insulating layer that makes FET transistors work). This may result in the end of Moore's Law, at least in the classical sense. With 45 nm announced for implementation of 2010 microprocessors, if the oft cited factor of 4 in density continued every three years, then 1 nm feature size would be reached shortly after 2020. In reality the roadmap now projects only a doubling of density every three years, with a termination point of about 14 nm. However, even if silicon doesn't reach the atomic scale, several emerging nanotechnologies appear to be feasible in that regime. However, deep sub nanometer devices seem beyond the scope of any of them. This means that without some radical alternative to classical digital logic as a computing model (such as quantum computing) we will probably be faced with some relatively hard limit on circuit sizes with two decades.

Latency is already a major source of performance degradation. Architecture has been "charged" with

1. *Frontiers of Extreme Computing*, October 23-27, 2005, Santa Cruz, CA and *The Path to Extreme Supercomputing*, October 12, 2004, Santa Fe, NM. These workshops covered not just technology, but architectures, applications, programming models, and software. A web site (www.zettaflops.org) has been established as a repository for such discussions.

hiding local latency, while hiding global latency has become the task of the programmer (i.e. with manual data partitioning and resource allocation). Today, multiple CPU clock cycles are required even across a chip. Access to local main memory (including DRAM cycle time) is measured in many hundreds of clock cycles. Round trip remote access request times for HPC machines can take thousands of cycles. In spite of progress in networking technology, increases in clock rate may cause delays to approach one million cycles in the future worst cases.

At one time, all gates on a chip could be reached within one clock cycle. If we define the metric *t* to be the ratio of the total number of possible gates on a chip to the number of gates through which a signal may propagate round trip in a single clock cycle time, at that time *t* used to equal to 1 (when most of the authors were graduate students). Today, technology estimates put *t* between 30 and 40. For very high clock rate technologies such as superconducting RSFQ devices, *t* is approximately 1000. Estimates for *t* at the nano-scale range from 100,000 to 1,000,000.

The net effect of increasing *t* is that the degree of parallelism necessary to take advantage of future nanoscale systems (which may comprise a million chips) could exceed a trillion-way. The ability to expose sufficient application parallelism through algorithmic and programming techniques, and to create the hardware organization and mechanisms to actually execute it, is a major problem and a grand challenge to achieving a zettaflop.

Contention for access to shared resources, in particular memory banks or remote compute elements, may dominate overall system scalability. Neither network technology nor chip memory bandwidth has grown at the same rate as the processor execution rate or the data access demands. With the success of Moore's Law, the amount of data per memory chip is growing such that it takes an increasing number of chip access cycles to touch all bytes per chip at least once. This imposes a fundamental bound on system scalability and is a significant contributor to the single digit performance efficiencies exhibited by many large scale applications today.

Efficiency, and ultimately the total effective scale possible, are determined by the performance not only the actual work (i.e. floating point operations) but also by the overhead mechanisms that manage both hardware parallelism and software concurrency. Today, this is done with message passing and barrier synchronization which is coarse grained and fails to expose fine grain parallelism. This work will show that it is clear that fine grain parallelism will be essential to achieving increasing levels of performance as we reach the nanoscale. This is especially true when one considers future workloads that manipulate very large scale directed graph data structures - and may dominate many classes of scientific, knowledge based, and national security problems of the next decade. It can be shown that the overhead to manage parallel work must, on average, be less than the work itself. Otherwise, the scalability of performance will be bounded, independent of the total amount of hardware resources for fixed sized problems.

In addition to these critical challenges to sustained, continued performance growth, a number of secondary but nonetheless important problems to effective and practical use of such systems must also be considered. We identify these here - and will address them in future papers:

- Reliability both to improve yield of nanoscale devices and to achieve high up time of ultra-scale systems that could suffer MTBFs on the order of seconds without active mechanisms for graceful degradation
- Programming languages, environments, and methodologies for providing simple semantics and syntax for reflecting the computational needs of future large scale applications while exposing the computational properties that must be exploited by future large scale computer architectures and systems
- A new culture embracing required innovation in sponsoring agencies, academic research institutions, and system implementation industry vendors. It has been observed that the architecture research pipeline is essentially empty[19].

3. The Silicon Roadblock

There are two capacity requirements that need to be satisfied if we are to claim even the potential of

Table 1: Basic Silicon Assumptions

Parameter	2004	2006	2008	2010	2012	2014	2016	2018	2020
DRAM Cell Area (sq.um)	0.082	0.041	0.019	0.012	0.008	0.005	0.003	0.002	0.001
Maximum GB per sq.cm	0.142	0.284	0.613	0.970	1.512	2.425	3.881	6.127	9.701
Area of 1 GB (sq. cm)	7.044	3.522	1.632	1.031	0.661	0.412	0.258	0.163	0.103
DRAM Cap storage energy (fj) Est	12.800	9.740	7.380	5.798	3.067	2.474	1.392	0.902	0.909
MPY array area (sq.mm)	0.124	0.066	0.038	0.022	0.014	0.008	0.005	0.004	0.002
MPY array thruput (GHz)	8	10	14	19	26	36	51	69	95
Est. Peak TFps per sq.cm	6.45	15.59	37.33	86.63	186.71	426.21	966.55	1968.89	4475.21
MPY array mwatts at Max Clock	0.1550	0.0642	0.0268	0.0115	0.0054	0.0023	0.0010	0.0005	0.0002
Est. Energy per flop (nj)	1.800	1.032	0.669	0.525	0.375	0.314	0.214	0.150	0.125
Eqvt Watts/sq cm	0.225	0.10047	0.04751	0.02764	0.01433	0.00867	0.00423	0.00217	0.00131

a "zettaflops" as we might define it today: storage and computational (flop) rate. Only when we have what is minimally required to achieve both simultaneously can we then worry about such things as area, power, cost, and programmability. In this paper we deliberately ignore those effects due to architectures, execution models, etc., and focus on simple absolute minimums.

The ITRS Roadmap[1] gives a wealth of data on technology projections through the year 2020 (when physical gate lengths reach 6nm and gate oxides are essentially one atomic layer). We use this to focus on two trends: what is the absolute minimal area of silicon to achieve storage requirements, and what is the absolute minimal area to achieve flops requirements. For this section we assume logic and circuit designs as practiced today.

For storage we use the densest possible one-bit, one transistor, DRAM cell. For now we ignore the (mandatory) overhead of row decoders, column sense amps, redundant row and columns, parity, error detection and correction bits and logic, column multiplexing, and built-in self test (BIST) engines. Tables 1c and 1d of [SIA05] give the area of such a single cell; we use the reciprocal to compute gigabytes (230 bits) per square cm as a function of time.

For the flops requirement we start with the mantissa portion of a modern floating point unit (FPU) as implemented in a leading edge high performance logic technology ([Bell05]: 0.124 sq. mm in a 90nm technology. The clock rate of up to 8GHz represents the fastest known multiplier rate, and we equate this to 8 gigaflops per second. We chose to baseline the area of just the multiplier array over a complete FPU because it is the major area of an FPU, and if one averages adds (which would take far less area than this) with multiplies (which take somewhat more), we would get an equivalent area somewhat akin to this. Also, as with the DRAM we ignore absolutely everything else: registers, instruction processors, interconnect, control, etc.

Using the scaling numbers from the roadmap, we adjust the potential area of this unit as time goes forward. Using the relative improvement in intrinsic transistor delay allows us to scale the maximum rate at which the unit can deliver flops. The ratio of flops by area gives a measure of gigaflops per second (GF) per sq. cm as a function of time.

For power we take the numbers from the baseline multiplier as the average energy for a flop and adjust for projected roadmap voltage and frequency, assuming a constant capacitance per unit area of silicon. We again ignore any energy required to move operands to/from the FPU, any control logic surrounding the FPU, or any static leakage loss. For DRAM we assume energy is dissipated only when a cell is charged (a "1" written) and when it is read (a "1" is read and a recharge is necessary). We assume 50% "1"s and "0"s. Also we assume the energy wasted in charging a capacitor to "1" equals the stored energy (0.5CV2). Again, the complete dynamics of a DRAM macro, including decoder, full row access, sense amp, bit line, refresh due to leakage, and column muxing have all been ignored. Table 1 summarizes the key numbers.

4. Lower Bound for a Conventional Silicon Zetta

As just mentioned, to achieve a zettaflops, we must

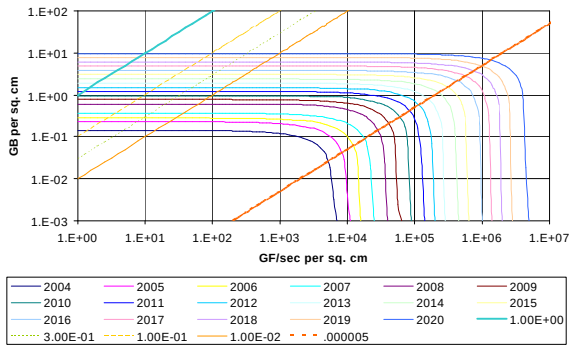

Figure 1: Silicon's Maximum Capacities vs Time

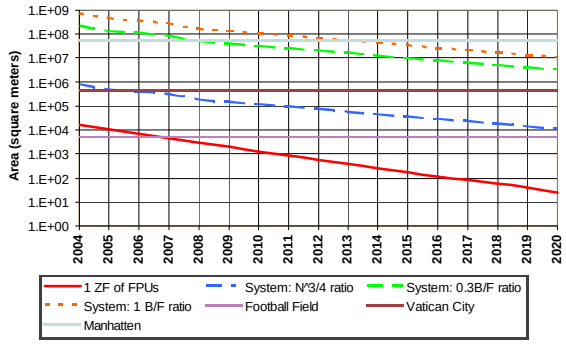

Figure 2: Absolute Minimum Area of Different Zetta Systems in Silicon

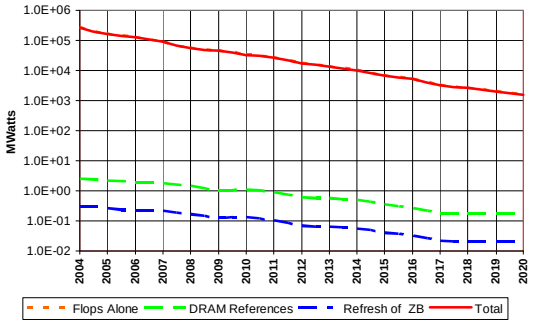

Figure 3: Minimum Power Dissipation for a Silicon Zetta System

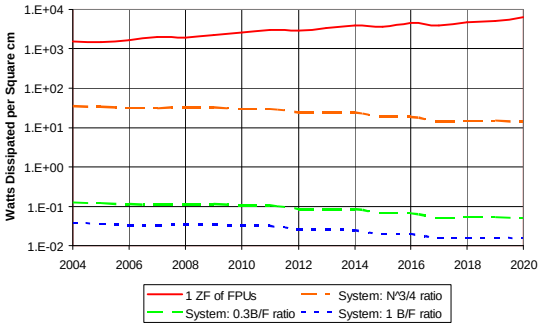

Figure 4: Average Power Density for Different Zetta Systems

also consider storage. Since the birth of the supercomputer there has been a debate as to how much storage is "enough." One rule of thumb is one byte per flops (1 ZB for a ZF); current practice in the ASCI class machines is around 0.3 bytes per flops (0.3 ZB for a ZF). Another relationship that first surfaced at the first Petaflops Workshop[35] was that gigabytes should equal at least sustained gigaflops to the ¾ power (reflecting 3D + time simulations where only the 3D values needed to be kept). This leads to 1 Exabyte for a zettaflops.

Following [18] Figure 1 summarizes the data from Table 1 into one graph. The y-axis is in GB per sq. mm; the x-axis is in GF per second per sq. mm. Each knee-shaped curve corresponds to one technology year, and assumes 1 square mm of silicon can be arbitrarily divided into DRAM or FPUs: 0% FPUs and 100% DRAM cells on the y-axis, and 100% FPUs and 0% DRAM cells on the x-axis. We then superimpose lines of constant GB per GF. Where they intersect the knee curves corresponds to the point in each year's technology where an average square mm of silicon has exactly the right proportion of memory and FPUs to match this ratio. Figure 2 then uses this data to compute the minimal area in silicon for the three ratios as a function of time. Several reference areas (a football field, the area of Vatican City, and the area of Manhattan) are included.

Figure 3 plots minimum power consumption assuming a sustained zettaflops engages all FPUs 100% of the time, and that each flop requires 2 reads and one write of 64 bits from/to some DRAM. This is independent of the storage to flops ratio. Figure 4 then divides this power by the area to obtain power density. As a reference, 100 W/cm^2 is the limit of today's microprocessors.

Finally, Figure 5 computes the average "diameter" of a silicon zetta system, in units of the FPU clock rate, assuming the silicon is packaged in a circle.

Several conclusions are readily apparent: First, in terms of area, a silicon zetta system is huge and dominated by memory; even if the FPU area is increased by a factor of 400 (20X to get to a typical

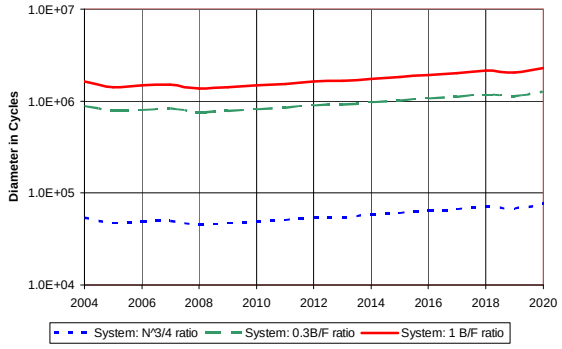

Figure 5: Diameter of Silicon Zetta Systems in Machine Cycles

CPU core today, and 20X for an inefficiency ratio) the smallest possible memory systems still dwarfs the processing logic. Second, this trend is completely reversed when one considers power. The 1000 megawatt projection for 2020 dwarves any supercomputing center today by at least two orders of magnitude. Additionally, given that virtually all this power is in the logic, it would not be unbelievable to assume that when the rest of conventional CPUs are considered, this number could increase by another 100X. One potential positive is that average power density should not be a problem unless we put all the FPUs on the same dies (as we do today). Finally, a system diameter in the millions of cycles makes it unlikely that "business as usual" programming models have any hope of keeping the FPUs fed with data. A more thorough analysis is hardly necessary. A "business as usual" silicon case in already completely unrealistic.

5. A QCA Zetta

Quantum dot Cellular Automata (QCA) (not analogous to quantum computing) is a nanotechnology using a binary representation of information, but replaces the current switch of transistors with a cell with four quantum dots, and arranged in a square having a bistable charge configuration of two electrons[23], [24], [37], [38]. One configuration of charge represents a binary "1," the other a "0," but no current flows into or out of the cell. In today's transistor paradigm, the current from one device charges a gate on the next device (and the interconnect between them) and thus turns it on or off. In the QCA paradigm, the Coulombic forces from the charge configuration of one device alters the charge configuration of the next device, without current flow between the two. This basic device-device interaction is sufficient to allow the computation of any Boolean function [37], [24]. If a clocking potential is added which modulates the effective energy barrier between charge configurations, the logic computation can be driven in a pipeline-wave fashion across a circuit [31], and general-purpose computing becomes possible with very low power dissipation[36].

QCA devices (and circuits) have actually been made with metal dots [2], [3], [20], [21] and with magnetic regions[9], [10], but can also be made from a single chemical molecule. Dots within a molecular QCA cell [25] are simply redox centers (areas of the molecule which can accept (be reduced) or donate (be oxidized) an electron without breaking the chemical bonds that hold the molecule together). Such dots have a very large single-particle energy level spacing and a high Coulomb cost for adding an additional charge. At the molecular level both of the two effects are strong.

The role of the dot in molecular QCA is played by molecular redox centers with the bridging ligands between redox centers providing the tunneling barrier. Upper limits on the switching speed are determined by the tunneling time through the bridging ligands. The effective barrier can be chemically varied to make the transfer time as long as seconds, or as fast as motion within a single extended orbital. Electron transfer rates are known to extend to the THz regime [4], [30]. Recent work [29], [26], [27], [16], [32], [34], [7], [28], [33] has established important milestones for molecular QCA using mixed-valence metal complexes. Fundamental requirements for molecular QCA include: stable complexes with stable mixed-valence states functionalized for substrate binding that can be switched by an electric field (movement of the mobile electron from one dot to the other). In addition, the most efficient building block for QCA circuits is a square of four metal complexes containing two mobile electrons. Both of these necessary requirements have now been demonstrated with molecules synthesized for just this purpose.

Molecular QCA devices can be clocked without the need to make separate clock connections to

each individual molecule [7]. Buried clocking wires can be used to form a patterned time-varying inhomogeneous perpendicular electric field at the molecular QCA plane, which acts as a clocking signal. Shifted sinusoidal clocking phases applied to successive wires results in a continuously varying distributed clock signal that smoothly sweeps information along the QCA circuit, as in a shift register. Adjacent molecules see clocking signals that are only fractionally out-of-phase with one another. This makes the transitions all the more smooth and adiabatic, lowering the energy dissipated as heat.

We now repeat the analysis of Section 4 for molecular QCA. Several designs were considered for the multiplier: a design based on all of the theoretical constructs of the technology (called the "theory" design point and based on [37] and [24]), as well as designs that was more "fabrication friendly" (called the "1x" design), and a fabrication friendly and defect tolerant version with 3 cell wide wires for redundancy (the "3x" design). The core parts of each design were studied at the physical level with both a simulator that solves the time independent Schrödinger equation, and a simulator that does a statistical mechanical analysis of a system of QCA devices, and reports the probability that it will settle in the desired ground state.

Because the clock structure just discussed results in an inherent degree of pipelining, our molecular QCA zettaflop will be a function of throughput. The theoretical maximum of the number of operations per multiplier is just equivalent to the QCA clock rate. In this discussion, we consider four clock rates: 1 THz as well as 1, 10, and 100 GHz clock rates. Thus the number of multiplier arrays equals 10^{21} flops divided by the clock rate.

As with silicon, we have designed a 54x54 array multiplier based on the 1x and 3x design points. It uses just a few fundamental conventional circuit elements (majority gates and inverters) that have been shown to function individually with two different physical simulators. The number of devices per multiplier and the corresponding area assuming 1nm center-to-center cell spacing are reported in Table 2 . (Note that our designs were

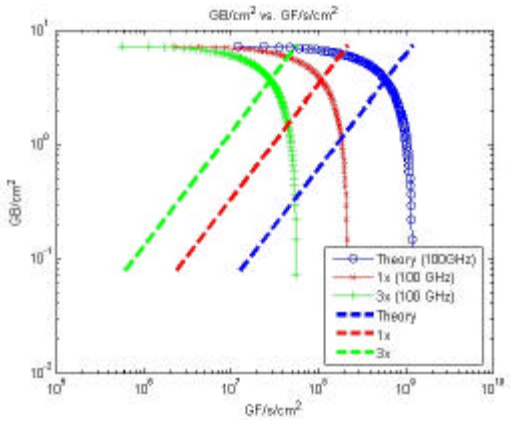

Figure 6: QCAs Maximum Capacities (Compare to Figure 1)

NOT optimized for area).

Table 2: QCA-Based Multiplier Characteristics

Design Point	Approx # of QCA Devices	Approx area per multiplier (in sq. mm)
Theory	10^5	8.33×10^{-6}
1X	6.1×10^6	4.55×10^{-5}
3X	2.7×1^7	1.75×10^{-4}

To repeat the analysis of Section 4, we must also consider memory. [14], [15] reports a QCA-based memory with a maximum storage density of 58Gb/cm^2. Such memories assume that data is kept in recursive arrays of very dense storage rings, where data recirculates constantly. Using this data, the area for a specific memory capacity can be determined directly. Again, no memory addressing, redundancy, or interfacing logic is included.

The QCA analogs of Figure 1 and Figure 2 appears in Figure 6 and Figure 7. Note that while memory densities are similar to those provided by end-of-the-roadmap silicon, QCA's worst case functional unit densities match silicon's best case. As we move toward higher clock rates in QCA, silicon's functional unit densities are bested by as much as three orders of magnitude.

We next consider a molecular QCA zettaflop in terms of power. For this study we use the number of devices required for each 54x54 bit multiply (Table 2), and the energy dissipation per switching event. At room temperature, each irreversible switching operation must dissipate at least ~2.9x10^{-21} J (the oft quoted K_bT ln(2) "Landauer's Limit" [22], [6], [17]). In the context of molecular QCA, the energy dissipation will need to be higher than this. [36] assumes that 100 meV (or 1.6x10^{-20} J) would be dissipated per switching event. Additionally, we note that, to a first approximation, the power dissipated for a molecular QCA zettaflop will be constant - or independent of clock rate. As context, as the clock rate decreases, the number of multipliers required to achieve a zettaflop must increase - in a constant ratio. Figure 8 considers the amount of power dissipated by 54x54 multipliers as a function of the energy dissipated per switching event at three different temperatures. The x-axis is the kT multiplier - N=1 is the theoretical minimum. An N between 70 and 100 is something that might be closer to a practical minimum that has some noise insensitivity.

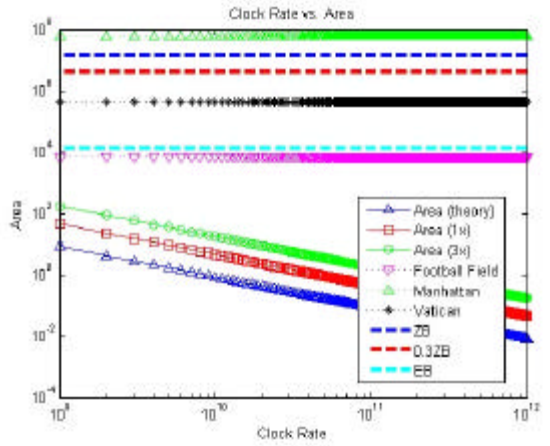

Figure 7: QCA Clock Rate vs Area for Zetta Systems

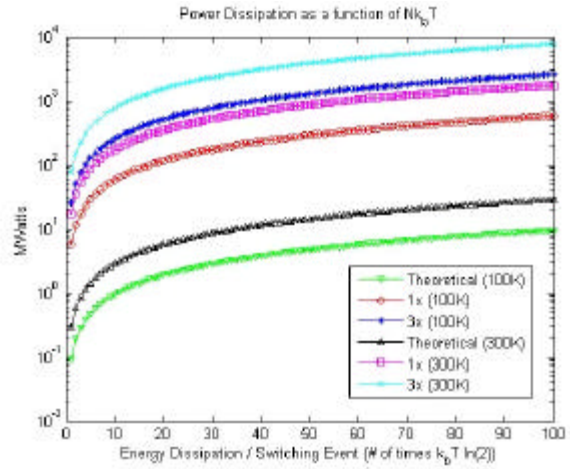

Figure 8: Power Dissipation as a Function of NK_bT

Theoretically, the power dissipation of a QCA zettaflops could be significantly below that of silicon. However, we must consider what redundancy is required to ensure that the computation is reliable in a production system. Otherwise, implementation friendly and redundant designs might better silicon only for values of N that are probably unrealistic. (Further study of this is already underway). Note also that reduced temperatures have the potential for about a 10X savings, not counting the power needed to perform the deeper cooling.

Given that the assumed memory consists of huge numbers of recirculating bits, there is liable to be a significant power consumption needed to keep the bits going. The exact power, and how to minimize it, is an open question for further research.

6. Reversible Logic

There is growing recognition that the end of the CMOS roadmap [1] will be set by a physics principle known as the "thermal limit." However, at least one approach to QCA circuit design has been proposed to use "reversible logic" to circumvent this limit and attain a potential thousandfold performance boost [36]. This thermal limit is not widely known today because it has been overshadowed by more serious transistor limits, but as shown above will become very significant somewhere between petaflops and zettaflops.

Regardless of circuit family, Landauer's Limit bounds the maximum power for a computer of conventional design. It is a basic property of all logic that the strength of the signals (0's and 1's) must be sufficiently stronger than random motion or particles in the environment so that signals can be reliably distinguished from noise. In a device at temperature T (typically 300K), the average amount of thermal energy per sampling time of a logic signal (per unit bandwidth) is k_bT, where k_b is Boltzmann's constant of 1.38 x10^{-23}. Furthermore, the amount of thermal energy is randomly

distributed and has the probability of about e^{-N} of being greater than N times its average value ($>Nk_bT$). To appear reliable in human terms (for example, when viewing an oscilloscope trace), N must exceed 10. To be reliable enough for sustained use in a supercomputer without error correction, N must exceed 100. (see p. 56 of the Emerging Research Devices chapter of [1]). All of today's mainline computer logic has the property that signal energy is turned into heat every time it reaches a gate. Thus, any computer consuming W watts constructed of "today's computer logic" is limited to performing $W/(100k_bT)$ logic operations per second.

A large supercomputer today delivers around W=500KW to the active components. At 300K and the $100k_bT$ point, this corresponds to about 1024 logic operations per second. If we ignore all architectural overhead for instruction decoding, memory, pipelining, etc., and just divide 1024 by the approximate 10,000 logic operations per floating point operation, we would get a maximum possible supercomputer performance of 100 exaflops for this same power budget, regardless of the technology (silicon or QCA).

The results discussed in both Section 4 and Section 5 assumed "classical" heat-dissipating families of logic. However, there is an alternative formulation of logic known as "reversible logic" [6], [13] that passes information from input to output without turning signals into heat. Reversible logic circuits can be implemented in various technologies - including CMOS and QCA. A glance at [36] shows that QCA can operate both "irreversibly" and "reversibly" with a 1000-fold power savings for reversible operation. In fact, according to the "physics of computation," there is no lower limit on the energy dissipation per logic operation [6].

We note that this is significantly different from the adiabatic and clock power recovery styles of logic used in several research projects where "reuse" of energy is the goal.

While the statement above is correct, it has an unpleasant consequence: in order that gates do not turn signals into heat, gates must (1) have the same number of outputs as inputs and (2) their function must not destroy information. Of the three gates in the ubiquitous AND-OR-NOT logic family, the AND and OR gates have fewer outputs than inputs and also destroy the state of their inputs (for 3 of 4 input combinations). Thus AND and OR gates are required by the laws of physics to generate Nk_BT heat per operation (clock cycle).

However, there are universal logic families that are reversible and need not generate heat. For example, the Toffoli and Fredkin gates [13] are each universal and reversible. A Fredkin gate has three inputs and three outputs with functionality as follows: if the first input is a 1, the other inputs are swapped and become outputs, otherwise they are not. Reversible logic gates of this type can be combined into familiar structures, including microprocessors [39].

While reversible logic gates are in some ways superior to irreversible ones, they are virtually unknown to practicing engineers and certainly not supported by industrial-strength design tools. Thus, there is a substantial non-technical barrier to the use of reversible logic.

The distinction between reversible and conventional (irreversible) operation is subtle enough that QCA-based logic, for example, can be evaluated in both modes, providing a comparative example:

1. QCA can be used to create conventional AND-OR-NOT logic. The AND and OR gates will turn signal energy into heat just like CMOS. While there can be debate about whether QCA or CMOS will be better, their performance potentials are similar as discussed above. Figure 9 (taken from [36]) illustrates conventional operating modes at the top of the gray region.
2. A designer can choose to use QCA cells only in reversible logic designs, such as with "cascade" clocking or as parts of reversible logic circuits (like Toffoli gates). If this is the case, no signal energy will be turned into heat and overall dissipation will be much less. The bottom of the shaded region in Figure 9 shows these losses are 2-3 orders of magnitude lower power dissipation the same clock rate.

A zettaflops is above the thermal limit for conven-

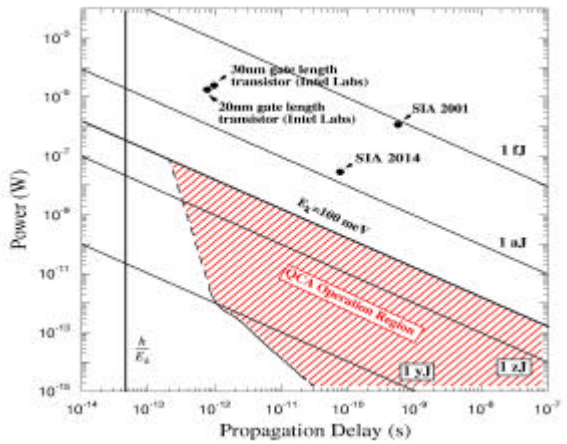

Figure 9: QCA Operating Regions

tional (irreversible) computer logic - at least without heroic power levels. While reversible logic is obscure today, we may have to master it before zettaflops computers will be possible.

7. Planning a Forward Path

The field of high performance computing has entered a period of significant uncertainty, with an end to Moore's Law for conventional semiconductor technology, and the confidence of continued exponential growth of performance through conventional approaches dimming beyond a few years into the future. As Section 4 demonstrated, designs with conventional silicon are dominated in area (and thus component cost) by memory, but more importantly have power budgets that are totally beyond feasibility - even in a "best case".

However, new technologies are emerging that could sustain the growth down to nanometer scale for another decade or longer. As Section 5 demonstrated, at least one of these has at least the potential for a zettaflops with a "conventional" logic style. While power may be reduced, especially if run at low temperatures, it is still excessive, and the cost of memory (in terms of area) is still a real concern.

In both cases sheer area concerns mean that huge amounts of parallelism must be efficiently employed if we are to get any sort of reasonable efficiency out of such designs.

Finally, yet another consideration, reliability, is liable to act as unwanted positive feedback to make all of the above estimates much worse in real life. To make such massive systems run long enough to solve real problems will require redundancy in many forms: at the circuit level to make the devices less susceptible to manufacturing faults, at the subsystem level to introduce redundancy and error correction to overcome both faults and the "kT" noise, and at the system level to provide checkpointing resources (especially more memory) to prevent loss of computation when a fault does occur. All of these require (significantly) more devices that add area, and more importantly, additional power.

Ultimately, atomic scale and fundamental power limits will demand a radical departure from conventional logic practices. Energy per operation must become absolutely the first design metric. Unless radically different models of computation such as quantum computing become feasible for a very broad spectrum of problems, this requires novel techniques such as reversible logic to be proven and then adopted - and soon. While the focus of such techniques must first address the huge FPU power, very quickly thereafter a new look at power in storage is needed.

In all stages of this evolution, a renewed and aggressive investment in all aspects of supercomputing research (technology, architecture, and software) is required if we are to continue to use supercomputing to advance in opportunity and capability. Planning and undertaking the path forward is essential now if the innovative methods are to be in place and sufficiently mature in time to fill the critical gaps identified as semiconductor feature size shrinks. Described here is a three stage plan of development and research to establish this path forward.

7.1. Near Term (first 5 years)

- (Highest priority) While the theoretical underpinnings for alternative logic systems such as reversible logic are reasonably well-established, a great deal of work is needed to understand precisely how those techniques can be implemented in known device technologies (both conventional silicon and newer nanotechnologies such as QCA). We need to determine the practical effects on real area and

power consumption. Metrics such as energy per operations and operations per second per unit area must be computed to the point where realistic projections of minimum system size and power can be made and compared to "business as usual." Similar studies must look at not just logic but storage as well. New techniques borrowed from silicon such as sleep modes must be integrated with alternative logic and storage styles to help address inevitable static and residual energy losses.

- Undertake basic research in a number of enabling device technologies including MRAM, nanotubes, QCA, RSFQ and others to establish likely candidates for future commercial components to replace - or at least supplement - semiconductors, regardless of logic approach. Emphasis should be on reducing energy per operation, sustaining energy per bit of storage, practical levels of inherent reliability, and area per bit and flops.
- Devise new classes of computer architectures that address the critical challenges of growing latency, contention (inadequate bandwidth), overhead, and starvation (lack of parallelism and load balancing). Simulate and prototype these. In particular, alternative architectures will become important that blend "memory" and "computational logic" in alternative fashions that may reduce significantly many of the overheads that we ignored.
- Begin to revisit the whole question of the relationship between device, circuit, and system reliability - with power and area as the driving concern. Old techniques such as ECC must be recycled in novel ways such as increasing refresh times in the presence of kT noise.
- Expand theoretical studies of checkpointing to increase our understanding of exactly how, when, and where should data be checkpointed during a computation. Again, a tight integration with minimum energy concerns and alternative logic models is important.
- Extend existing programming languages and tools to incorporate the semantics of fine grain parallelism and dynamic resource management to make parallel programming more productive and drive next generation system and processor architectures.
- Establish an aggressive and integrated multi-discipline research program in high performance computing in advanced computer technologies, advanced parallel computer architecture, advanced parallel programming languages, tools, and methods, and advanced system software, and fault tolerance.

7.2. Medium Term (2010 to 2015)

- Development of revised design tools that are compatible with alternative logic and technology choices, and that elevate energy consumption to the same priority that timing is today.
- Demonstration of practical circuits in both silicon and nanotechnologies that do in fact offer significant changes in the energy per operation metrics.
- Development of alternative storage implementations that offer better density and closer coupling to processing than we have today.
- Undertake the design, simulation, and initial prototyping of advanced parallel computer architecture employing innovative structures and mechanisms. These should include intrinsically fault resilient structures.
- Implement research prototypes of new programming models, languages, and software systems that may be needed to match the emerging architectures.

7.3. Long Term (2015 to 2025)

- Begin designs of prototype systems based on these techniques.
- Determine where the balance point for cost lies for mixed technology solutions in terms of performance, size, and power.
- Such complex hybrid system structures that will dominate computing beyond the end of the next decade will demand a new software framework and intelligent tools to adapt the complex and disparate physical resources to the needs of unprecedented scale multi-scale multi-physics applications
- Investigate radical solutions to long-term extreme mass storage technologies for reliable yotta scale (1,000 ZB) data archiving.

Finally, to put our work in proper perspective, we

note that supercomputers capable of a megaflops were viable in the 1970s, a gigaflops in the early 1980s, a teraflops in ~1994, and we are currently on the verge of a petaflops. One might reasonably conclude that three orders of magnitude of performance have been achieved approximately every 12 years. What this means is that a zettaflops is only one half of a professional lifetime away - but the technology to achieve it appears to be much farther away. In short, this paper (while too late!) brings to life important challenges that must be addressed by the next generation of researchers, and how research efforts perhaps ought to be redirected to solve them in a timely fashion.

8. Acknowledgements

This material is based in part upon work supported by the Defense Advanced Research Projects Agency (DARPA) under its Contract No. NBCH3039003, in part by NSF under grant CCR-0210153, and in part by Sandia National Labs under grant 478022.

9. References

[1] International Technology Roadmap for Semiconductors, 2005 Ed., http://public.itrs.net.

[2] I. Amlani, A. O. Orlov, G. L. Snider, C. S. Lent, G. H. Bernstein, "Demonstration of a six-dot quantum cellular automata system," Applied Physics Letters 72 (17), April 27 1998, pp. 2179-2181.

[3] I. Amlani, A.O. Orlov, G. Toth, C. S. Lent, G.H. Bernstein, and G.L. Snider, Digital Logic Gate Using Quantum-Dot Cellular Automata, Science 284 (5412), April 9 1999, pp. 289-291.

[4] P. F. Barbara, T.J. Meyer, and M.A. Ratner, "Contemporary Issues in Electron Transfer Research," J. Phys. Chem. 100, 1996, pp. 13148-13168,.

[5] L. Barroso, "The Price of Performance," ACM Queue, vol. 3, no. 7, Sept. 2005.

[6] C. H. Bennett, "The thermodynamics of computation-a review," International Journal of Theoretical Physics, Volume 21, Issue 12, Dec 1982, pp. 905 - 940.

[7] E.P. Blair and C.S. Lent, "An architecture for molecular computing using quantum-dot cellular automata nanotechnology," IEEE-NANO 2003. Third IEEE Conference on Nanotechnology, vol. 2, 2003, pp 402-405.

[8] W. Belluomini, et al, "An 8 GHz Floating Point Multiply," ISSCC, Feb. 6-10, 2005, San Francisco, pp. 302-303.

[9] G.H. Bernstein, A. Imre, V. Metlushko, A. Orlov, L. Zhou, L. Ji, G. Csaba, and W. Porod, "Magnetic QCA systems," Microelectronics Journal, 36, 2005, pp. 619-624.

[10] R.P. Cowburn and M.E. Welland, "Room Temperature Magnetic Quantum

[11] Cellular Automata," Science, Vol. 287, Issue 5457, 1466-1468, Feb. 2000

[12] E. P. DeBenedictus, "Reversible Logic for Supercomputing," 1st Int. Workshop on Reversible Computing, May 4-7, 2005, Ischia, Italy.

[13] E. Fredkin, T. Toffoli, "Conservative logic, International Journal of Theoretical Physics, "Volume 21, Issue 3-4, Apr 1982, Pages 219 - 253.

[14] S. E. Frost, A. F. Rodrigues, A. W. Janiszewski, R. T. Rausch, and P. M. Kogge "Memory in Motion: A Study of Storage Structures in QCA." 1st Workshop on Non-Silicon Computation (NSC-1), held in conjunction with 8th Int. Symp. on High Performance Computer Architecture (HPCA-8), Boston, MA. Feb. 3, 2002

[15] S.E. Frost, "Memory Architectures for Quantum-dot Cellular Automata," Master's Thesis, University of Notre Dame, April 2005.

[16] J. Jiao , Long, G. J., Grandjean, F., Beatty, A. M., Fehlner, T. P., "Building Blocks for the Molecular Expression of Quantum Cellular Automata. Isolation and Characterization of a Covalently Bonded Square Array of Two Ferrocenium and Two Ferrocene Complexes," J. Am. Chem. Soc., 2003. 125: p. 7522

[17] R.W. Keyes and R. Landauer, "Minimal Energy Dissipation in Logic," IBM J. Res. Dev. 14, 1970, pp. 152-157.

[18] P. M. Kogge, T. Sunaga, E. Retter, et al, "Combined DRAM and Logic Chip for Massively Parallel Applications," 16th IEEE Conf. on Advanced Research in VLSI, Raleigh, NC, March 1995, pp. 4-16.

[19] P. M. Kogge, "Long Term Trends in Com-

puter Architecture Research Funding as Seen thru ISCA," invited talk, CRA Workshop on Funding in Computer Architecture, Aptos, CA,: Dec. 4-7, 2005 and CSE Dept., Univ. of Notre Dame, Technical Report TR 2005-16.

[20] R. K. Kummamuru, A. O. Orlov, R. Ramasubramaniam, C. S. Lent, G. H. Bernstein, and G. L. Snider, "Operation of a Quantum-Dot Cellular Automata (QCA) Shift Register and Analysis of Errors," IEEE Transactions on Electron Devices, Vol. 50, 2003, pp. 1906-1913.

[21] R. K. Kummamuru, J. Timler, G. Toth, C. S. Lent, R. Ramasubramaniam, A. O. Orlov, G. H. Bernstein, G. L. Snider, Power gain in a quantum-dot cellular automata latch, Applied Physics Letters 81 (7): 1332-1334 Aug 12, 2002

[22] R. Landauer, "Irreversibility and heat generation in the computing process," IBM J. Res. Dev. 5, 183, 1961.

[23] C. S. Lent, P. D. Tougaw, W. Porod, and G. H. Bernstein, "Quantum Cellular Automata", Nanotechnology 4, 49, 1993.

[24] C. S. Lent, P. D. Tougaw, "A device architecture for computing with quantum dots," Proceedings of the IEEE 85 (4): 541-557 Apr 1997.

[25] C. S. Lent, "Bypassing the Transistor Paradigm," Science 228(5471): 1597-1598 2000.

[26] Z. Li, A.M. Beatty, and T.P. Fehlner, "Molecular QCA Cells. 1. Structure and Functionalization of an Unsymmetrical Dinuclear Mixed-Valence Complex for Surface Binding." Inorg. Chem., 42: 2003, p. 5707

[27] Z. Li, and T.P. Fehlner, "Molecular QCA Cells. 2. Electrochemical Characterization of an Unsymmetrical Dinuclear Mixed-Valence Complex Bound to a Au Surface by an Organic Linker," Inorg. Chem., 42: 2003: p. 5715.

[28] Y. Lu and C.S. Lent, "Theoretical Study of Molecular Quantum-dot Cellular Automata," J. Computational Electronics 4, 115-118, 2005.

[29] M. Manimarian, G.L. Snider, C.S. Lent, V. Sarveswaran, M. Lieberman, Z. Li, and T.P. Fehlner, "Scanning tunneling microscopy and spectroscopy investigations of QCA molecules," Ultramicroscopy 97, 55-63, 2003.

[30] Y. Nagasawa, Y. Ando, D. Kataoka, H. Matsuda, H. Miyasaka, and T. Okada, "Ultrafast Excited State Deactivation of Triphenylmethane Dyes," J. Phys. Chem. 106, 2024-2035 2002.

[31] M. T. Niemier and P. M. Kogge, "Exploring and Exploiting Wire-Level Pipelining in Emerging Technologies," Int. Symp. of Computer Architecture (ISCA), Sweden, July 2001, pp. 166-177.

[32] H. Qi, S. Sharma, Z. Li, G.L. Snider, A.O. Orlov, C.S. Lent, and T.P. Fehlner, "Molecular quantum cellular automata cells. Electric field driven switching of a silicon surface bound array of vertically oriented two-dot molecular quantum cellular automata," J. Am. Chem. Soc.125, 15250-15259, 2003

[33] H. Qi, A. Gupta, B. C. Noll, G. L. Snider, Y. Lu, C. Lent, and T. P. Fehlner, "Dependence of Field Switched Ordered Arrays of Dinuclear Mixed-Valence Complexes on the Distance between the Redox Centers and the Size of the Counterions," J. Am. Chem. Soc.127, 15218-15227, 2005.

[34] K. Sarveswaran, P. Huber, M.Lieberman, C. Russo, C.S. Lent, "Nanometer scale rafts built from DNA tiles," IEEE-NANO 2003 Third IEEE Conference on Nanotechnology Volume 1, 12-14 Aug. 2003 pp. 417 - 420, 2003.

[35] T. P. Sterling, P. Messina, P. H. Smith, Enabling Technologies for Petaflops Computing, MIT Press, 1995

[36] J. Timler, C. S. Lent, "Power gain and dissipation in quantum-dot cellular automata," Journal of Applied Physics 91 (2): 823-831 Jan. 15, 2002.

[37] P. D. Tougaw, C. S. Lent, "Logical devices implemented using quantum cellular automata," Journal of Applied Physics 75(3): 1818-25 Feb 1 1994.

[38] P. D. Tougaw, C. S. Lent, "Dynamic behavior of quantum cellular automata," Journal of Applied Physics 80(8): 4722-4736 Oct. 15 1996

[39] C. Vieri, Reversible Computer Engineering and Architecture, Ph. D. Thesis, Massachusetts Institute of Technology, June 1999.