REVIEW

doi:10.1038/nature13570

Limits on fundamental limits to computation

Igor L. Markov[1]*

An indispensable part of our personal and working lives, computing has also become essential to industries and governments. Steady improvements in computer hardware have been supported by periodic doubling of transistor densities in integrated circuits over the past fifty years. Such Moore scaling now requires ever-increasing efforts, stimulating research in alternative hardware and stirring controversy. To help evaluate emerging technologies and increase our understanding of integrated-circuit scaling, here I review fundamental limits to computation in the areas of manufacturing, energy, physical space, design and verification effort, and algorithms. To outline what is achievable in principle and in practice, I recapitulate how some limits were circumvented, and compare loose and tight limits. Engineering difficulties encountered by emerging technologies may indicate yet unknown limits.

Emerging technologies for computing promise to outperform conventional integrated circuits in computation bandwidth or speed, power consumption, manufacturing cost, or form factor[1,2]. However, razor-sharp focus on any one nascent technology and its benefits sometimes neglects serious limitations or discounts ongoing improvements in established approaches. To foster a richer context for evaluating emerging technologies, here I review limiting factors and the salient trends in computing that determine what is achievable in principle and in practice. Several fundamental limits remain substantially loose, possibly indicating viable opportunities for emerging technologies. To clarify this uncertainty, I examine the limits on fundamental limits.

Universal and general-purpose computers

If we view clocks and watches as early computers, it is easy to see the importance of long-running calculations that can be repeated with high accuracy by mass-produced devices. The significance of programmable digital computers became clear at least 200 years ago, as illustrated by Jacquard looms in textile manufacturing. However, the existence of universal computers that can efficiently simulate (almost) all other computing devices—analogue or digital—was only articulated in the 1930s by Church and Turing (Turing excluded quantum physics when considering universality)[3]. Efficiency was studied from a theoretical perspective at first, but strong demand in military applications in the 1940s led Turing and von Neumann to develop detailed hardware architectures for universal computers—Turing's design (Pilot ACE) was more efficient, but von Neumann's was easier to program. The stored-program architecture made universal computers practical in the sense that a single computer design could be effective in many diverse applications if supplied with appropriate software. Such practical universality thrives (1) in economies of scale in computer hardware and (2) among extensive software stacks. Not surprisingly, the most sophisticated and commercially successful computer designs and components, such as Intel and IBM central processing units (CPUs), were based on the von Neumann paradigm. The numerous uses and large markets of general-purpose chips, as well as the exact reproducibility of their results, justify the enormous capital investment in the design, verification and manufacturing of leading-edge integrated circuits. Today general-purpose CPUs power cloud server-farms and displace specialized (but still universal) mainframe processors in many supercomputers. Emerging universal computers based on field-programmable gate-arrays and general-purpose graphics processing units outperform CPUs in some cases, but their efficiencies remain complementary to those of CPUs. The success of deterministic general-purpose computing is manifest in the convergence of diverse functionalities in portable, inexpensive smartphones. After steady improvement, general-purpose computing displaced entire industries (newspapers, photography, and so on) and launched new applications (video conferencing, GPS navigation, online shopping, networked entertainment, and so on)[4]. Application-specific integrated circuits streamline input–output and networking, or optimize functionalities previously performed by general-purpose hardware. They speed up biomolecular simulation 100-fold[5,6] and improve the efficiency of video decoding 500-fold[7], but they require design efforts with a keen understanding of specific computations, impose high costs and financial risks, need markets where general-purpose computers lag behind, and often cannot adapt to new algorithms. Recent techniques for customizable domain-specific computing[8] offer better tradeoffs, while many applications favour the combination of general-purpose hardware and domain-specific software, including specialized programming languages[9,10] such as Erlang, which was used to implement the popular Whatsapp instant messenger.

Limits as aids to evaluating emerging technologies

Without sufficient history, we cannot extrapolate scaling laws for emerging technologies, yet expectations run high. For example, new proposals for analogue processors appear frequently (as illustrated by adiabatic quantum computers), but fail to address concerns about analogue computing, such as its limitations on scale, reliability, and long-running error-free computation. General-purpose computers meet these requirements with digital integrated circuits and now command the electronics market. In comparison, quantum computers—both digital and analogue—hold promise only in niche applications and do not offer faster general-purpose computing because they are no faster for sorting and other specific tasks[11–13]. In exaggerating the engineering impact of quantum computers, the popular press has missed this important point. But in scientific research, attempts to build quantum computers may help in simulating quantum-chemical phenomena and reveal new fundamental limits. The sections 'Asymptotic space-time limits' and 'Conclusions' below discuss the limits on emerging technologies.

Technology extrapolation versus fundamental limits

The scaling of commercial computing hardware regularly runs into formidable obstacles[1,2], but near-term technological advances often circumvent

[1]EECS Department, The University of Michigan, Ann Arbor, Michigan 48109-2121, USA. *Present address: Google Inc., 1600 Amphitheatre Parkway, Mountain View, California 94043, USA.

Table 1 | Some of the known limits to computation

Limits	Engineering	Design and validation	Energy, time	Space, time	Information, complexity
Fundamental	Abbe (diffraction); Amdahl; Gustafson	Error-correction and dense codes; fault-tolerance thresholds	Einstein ($E = mc^2$); Heisenberg ($\Delta E \Delta t$); Landauer ($kT\ln 2$); Bremermann; adiabatic theorems	Speed of light; Planck scale; Bekenstein; Fisher ($T(n)^{1/(d+1)}$)	Shannon channel capacity; Holevo bound; NC, NP, #P; decidability
Material	Dielectric constant; carrier mobility; surface morphology; fabrication-related	Analytical and numerical modelling	Conductivity; permittivity; bandgap; heat flow	Propagation speed; atomic spacing; no gravitational collapse	Information transfer between carriers
Device	Gate dielectric; channel charge control; leakage; latency; cross-talk; ageing	Compact modelling; parameter selection	CMOS; quantum; charge-centric; signal-to-noise ratio; energy conversion	Interfaces and contacts; entropy density; entropy flow; size and delay variation; universality	
Circuit	Delay; inductance; thermal-related; yield; reliability; input–output	Interconnect; test; validation	Dark, darker, dim and grey silicon; interconnect; cooling efficiency; power density; power supply; two or three dimensions		Circuit complexity bounds
System and software	Specification; implementation; validation; cost		Synchronization; physical integration; parallelism; *ab initio* limits (Lloyd)		The 'consistency, availability, partitioning tolerance' (CAP) theorem

Summary of material from refs 5, 13–15, 17, 18, 22, 23, 26, 31, 39, 41, 42, 46, 48–50, 53, 54, 57–60, 62, 63, 65, 74–76, 78, 87, 96, 98 and 99.

them. The ITRS[14] keeps track of such obstacles and possible solutions with a focus on frequently revised consensus estimates. For example, consensus estimates initially predicted 10-GHz CPUs for the 45-nm technology node[15], versus the 3–4-GHz range seen in practice. In 2004, the unrelated Quantum Information Science and Technology Roadmap[16] forecast 50 'digital' physical qubits by 2012. Such optimism arose by assuming technological solutions long before they were developed and validated, and by overlooking important limits. The authors of refs 17 and 18 classify the limits to devices and interconnects as fundamental, material, device, circuit, and system limits. These categories define the rows of Table 1, and the columns reflect the sections of this Review in which I examine the impact of specific limits on feasible computing technologies, looking for 'tight' limits, which obstruct the long-term improvement of key parameters.

Engineering obstacles

Engineering obstacles limit specific technologies and choices. For example, a key bottleneck today is integrated circuit manufacture, which packs billions of transistors and wires in several square centimetres of silicon, with astronomically low defect rates. Layers of material are deposited on silicon and patterned with lasers, fabricating all circuit components simultaneously. Precision optics and photochemical processes ensure accuracy.

Limits on manufacturing

No account of limits to computing is complete without the Abbe diffraction limit: light with wavelength λ, traversing a medium with refractive index η, and converging to a spot with angle θ (perhaps focused by a lens) creates a spot with diameter $d = \lambda/\mathrm{NA}$, where $\mathrm{NA} = \eta\sin\theta$ is the numerical aperture. NA reaches 1.4 for modern optics, so it would seem that semiconductor manufacturing is limited to feature sizes of $\lambda/2.8$. Hence, argon-fluoride lasers with a wavelength of 193 nm should not support photolithographic manufacturing of transistors with 65-nm features. Yet these lasers can support subwavelength lithography even for the 45-nm to 14-nm technology nodes if asymmetric illumination and computational lithography are used[19]. In these techniques, one starts with optical masks that look like the intended image, but when the image gets blurry, the masks are altered by gently shifting the edges to improve the image, possibly eventually giving up the semblance between the original mask and the final image. Clearly, some limits are formulated to be broken! Ten years ago, researchers demonstrated the patterning of nanomaterials by live viruses[20]. Known virions exceed 20 nm in diameter, whereas subwavelength lithography using a 193-nm ArF laser was recently extended to 14-nm semiconductor manufacturing[14]. Hence, viruses and microorganisms are no longer at the forefront of semiconductor manufacturing. Extreme ultraviolet (X-ray) lasers have been energy-limited, but are improving. Their use requires changing the optics from refractive to reflective. Additional progress in multiple patterning and directed self-assembly promises to support photolithography beyond the 10-nm technology node.

Limits on individual interconnects

Despite the doubling of transistor density with Moore's law[21], semiconductor integrated circuits would not work without fast and dense interconnects. Copper wires can be either fast or dense, but not both at the same time—a smaller cross-section increases electrical resistance, while greater height or width increase parasitic capacitance with neighbouring wires (wire delay grows with the product of resistance and capacitance, RC). As pointed out in 1995 by an Intel researcher, on-chip interconnect scaling has become the real limiter of high-performance integrated circuits[22]. The scaling of interconnect is also moderated by electron scattering against rough edges of metallic wires[18], which is inevitable with atomic-scale wires. Hence, integrated circuit interconnect stacks have evolved[15,23] from four equal-pitch layers in 2000 to 16 layers with some wires up to 32 times thicker than others (as in Fig. 3) including a large amount of dense (thin) wiring and fast (thick) wires used for global on-chip communication (Fig. 3). Aluminium and copper remain unrivalled for conventional interconnects and can be combined in short wires[98]; carbon-nanotube and spintronic interconnects are also evaluated in ref. 98. Photonic waveguides and radio frequency links offer alternative integrated circuit interconnect[24,25], but still obey fundamental limits derived from Maxwell's equations, such as the maximum propagation speed of electromagnetic waves[18]. The number of input–output links can only grow with the perimeter or surface area of a chip, whereas chip capacity grows with area or volume, respectively.

Limits on conventional transistors

Transistors are limited by their tiniest feature—the width of the gate dielectric—which recently reached the size of several atoms (Fig. 1), creating problems: (1) a few missing atoms can alter transistor performance, (2) manufacturing variation makes all the transistors slightly different (Fig. 2), (3) electric current tends to leak through thin narrow dielectrics[17]. Therefore, transistors are redesigned with wider dielectric layers[26] that surround a fin shape (Fig. 4). Such configurations improve the control of the electric field, reduce current densities and leakage, and diminish process variations. Each field effect transistor (FET) can use several fins, extending transistor scaling by several generations. Semiconductor manufacturers adopted such FinFETs for upcoming technology nodes. Going a step further, in tunnelling transistors[27], a gate wraps around the channel to control the tunnelling rate.

Limits on design effort

In the 1980s, Mead and Conway formalized integrated circuit design using a regular grid, enabling automated layout through algorithms. But the

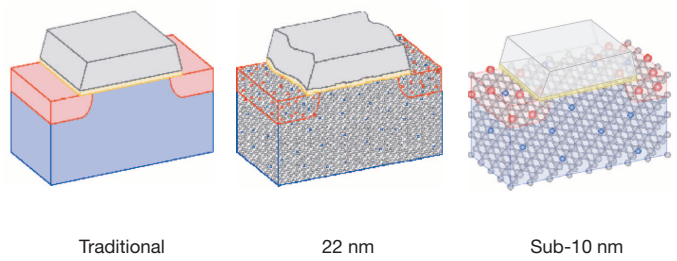

Figure 1 | As a metal oxide–semiconductor field effect transistor (MOSFET) shrinks, the gate dielectric (yellow) thickness approaches several atoms (0.5 nm at the 22-nm technology node). Atomic spacing limits the device density to one device per nanometre, even for radical devices. For advanced transistors, grey spheres indicate silicon atoms, while red and blue spheres indicate dopant atoms (intentional impurities that alter electrical properties). Image redrawn from figure 1 of http://cnx.org/content/m32874/latest/, with permission from Gold Standard Simulations.

resulting optimization problems remain difficult to solve, and heuristics are only good enough for practical use. Besides frequent algorithmic improvements, each technology generation alters circuit physics and requires new computer-aided design software. The cost of design has doubled in a few years, becoming prohibitive for integrated circuits with limited market penetration[14]. Emerging technologies, such as FinFETs and high-κ dielectrics (κ is the dielectric constant), circumvent known obstacles using forms of design optimization. Therefore, reasonably tight limits should account for potential future optimizations. Low-level technology enhancements, no matter how powerful, are often viewed as one-off improvements, in contrast to architectural redesigns that affect many processor generations. Between technology enhancements and architectural redesigns are global and local optimizations that alter the 'texture' of integrated circuit design, such as logic restructuring, gate sizing and device parameter selection. Moore's law promises higher transistor densities, but some transistors are designed to be 32 times larger than others. Large gates consume greater power to drive long interconnects at acceptable speed and satisfy performance constraints. Minimizing circuit area and power, subject to timing constraints (by configuring each logic gate to a certain size, threshold voltage, and so on), is a difficult but increasingly important optimization with a large parameter space. A recent convex optimization method[28] saved 30% power in Intel chips, and the impact of such improvements grows with circuit size. Many aspects of integrated circuit design are being improved, continually raising the bar for technologies that compete with complementary metal-oxide–semiconductors (CMOSs).

Completing new integrated circuit designs, optimizing them and verifying them requires great effort and continuing innovation; for example, the lack of scalable design automation is a limiting factor for analogue

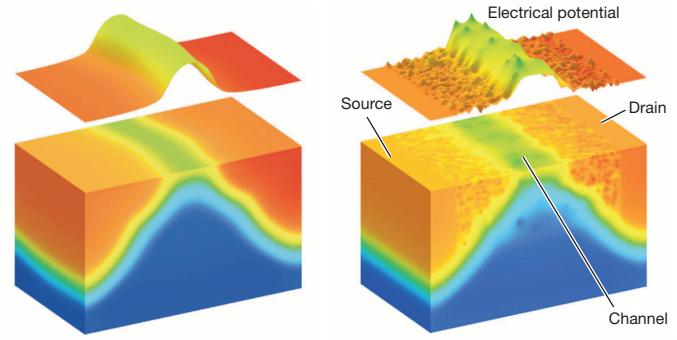

Figure 2 | As a MOSFET transistor shrinks, the shape of its electric field departs from basic rectilinear models, and the level curves become disconnected. Atomic-level manufacturing variations, especially for dopant atoms, start affecting device parameters, making each transistor slightly different[96,97]. Image redrawn from figure 'DOTS and LINES' of ref. 97, with permission from Gold Standard Simulations.

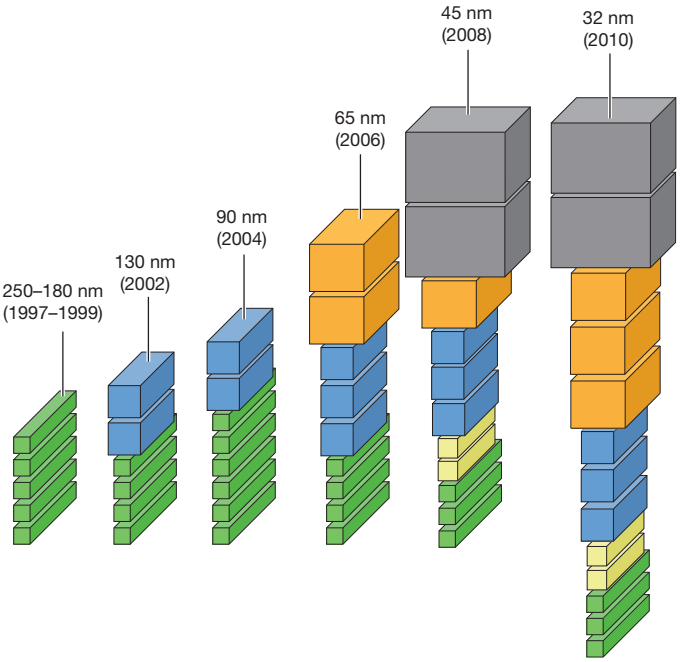

Figure 3 | The evolution of metallic wire stacks from 1997 to 2010. Stacks are ordered by the designation of the semiconductor technology node. Image redrawn from a presentation image by C. Alpert of IBM Research, with permission.

integrated circuits[29,30]. In 1999, bottom-up analysis of digital integrated circuit technologies[15,31] outlined design scaling up to self-contained modules with 50,000 standard cells (each cell contains one to three logic gates), but further scaling was limited by long-range interconnect. In 2010, physical separation of modules became less critical, as large-scale placement optimizations, implemented as software tools, assumed greater responsibility for integrated circuit layout and can now intersperse components of nearby modules[32,33]. In a general trend, powerful design automation[34] frees circuit engineers to focus on microarchitecture[33], but increasingly relies on algorithmic optimization. Until recently, this strategy suffered significant losses in performance[35] and power[36] compared to ideal designs, but has now become both successful and indispensable owing to the rapidly increasing complexity of digital and mixed-signal electronic systems. Hardware and software must now be co-designed and co-verified, with software improving at a faster rate. Platform-based design combines high-level design abstractions with the effective re-use of components and functionalities in engineered systems[37]. Customizable domain-specific computing[8] and domain-specific programming languages[9,10] offload specialization to software running on re-usable hardware platforms.

Energy–time limits

In predicting the main obstacles to improving modern electronics, the 2013 edition of the International Technology Roadmap for Semiconductors (ITRS) highlights the management of system power and energy as the main challenge[14]. The faster the computation, the more energy it consumes, but actual power–performance tradeoffs depend on the physical scale. While the ITRS, by its charter, focuses on near-term projections and integrated circuit design techniques, fundamental limits reflect available energy resources, properties of the physical space, power-dissipation constraints, and energy waste.

Reversibility

A 1961 result by Landauer[38] shows that erasing one bit of information entails an energy loss that $\geq kT\ln 2$ (the thermodynamic threshold), where k is the Boltzmann constant and T is the temperature in Kelvin. This principle was validated empirically in 2012 (ref. 39) and seems to motivate reversible computing[40], where all input information is preserved, incurring additional costs. Formally speaking, zero-energy computation is prohibited by

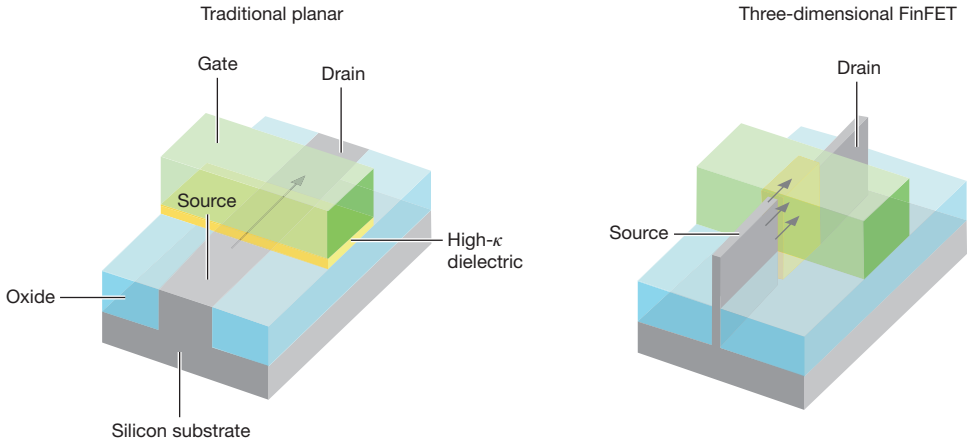

Figure 4 | FinFET transistors possess a much wider gate dielectric layer (surrounding the fin shape) than do MOSFET transistors and can use multiple fins.

the energy–time form of the Heisenberg uncertainty principle ($\Delta t \Delta E \geq h/2$): faster computation requires greater energy[41,42]. However, recent work in applied superconductivity[43] demonstrates "highly exotic" physically reversible circuits operating at 4°K with energy dissipation below the thermodynamic threshold. They apparently fail to scale to large sizes, run into other limits, and remain no more practical than 'mainstream' superconducting circuits and refrigerated low-power CMOS circuits. Technologies that implement quantum circuits[44] can approximate reversible Boolean computing, but currently do not scale to large sizes, are energy-inefficient at the system level, rely on fragile components, and require heavy fault-tolerance overheads[13]. Conventional integrated circuits also do not help to obtain energy savings from reversible computing because they dissipate 30%–60% of all energy in (reversible) wires and repeaters[23]. At room temperature, Landauer's limit amounts to 2.85×10^{-21} J—a very small fraction of the total, given that modern integrated circuits dissipate 0.1–100 W and contain $<10^9$ logic gates. With the increasing dominance of interconnect (see section 'Asymptotic space-time limits'), more energy is spent on communication than on computation. Logically reversible computing is important for reasons other than energy reduction—in cryptography, quantum information processing, and so on[45].

Power constraints and CPUs

The end of CPU frequency scaling. In 2004, Intel abruptly cancelled a 4-GHz CPU project because its high power density required awkward cooling technologies. Other CPU manufacturers kept clock frequencies in the 1–6-GHz range, but also resorted to multicore CPUs[46]. Since dynamic circuit power grows with clock frequency and supply voltage squared[47], energy can be saved by distributing work among slower, lower-voltage parallel CPU cores if the parallelization overhead is small.

Dark, darker, dim, grey silicon. A companion trend to Moore's law—the Dennard scaling theory[48]—shows how to keep power consumption of semiconductor integrated circuits constant while increasing their density. But Dennard scaling broke down ten years ago[48]. Extrapolation of semiconductor scaling trends for CMOSs—the dominant semiconductor technology for the past 20 years—shows that the power consumption of transistors available in modern integrated circuits reduces more slowly than their size (which is subject to Moore's law)[49,50]. To ensure acceptable performance characteristics of transistors, chip power density must be limited, and a fraction of transistors must be kept dark at any given time. Modern CPUs have not been able to use all their circuits at once, but this asymptotic effect—termed the "utilization wall"[49]—will soon black out 99% of the chip, hence the term 'dark silicon' and a reasoned reference to the apocalypse[49]. Saving power by slowing CPU cores down is termed 'dim silicon'. Detailed studies of dark silicon[50] show similar results. To this end, executives from Microsoft and IBM have recently proclaimed an end to the era of multicore microprocessors[51]. Two related trends appeared earlier: (1) increasingly large integrated circuit regions remain transistor-free to aid routeing and physical synthesis, to accommodate power-supply networks, and so on[52,53]—we call them 'darker silicon', (2) increasingly many gates do not perform useful computation but reinforce long, weak interconnects[54] or slow down wires that are too short—which I call 'grey silicon'. Today, 50%–80% of all gates in high-performance integrated circuits are repeaters.

Limits for power supply and cooling. Data centres in the USA consumed 2.2% of its total electricity in 2011. Because power plants take time to build, we cannot sustain past trends of doubled power consumption per year. It is possible to improve the efficiency of transmission lines (using high-temperature superconductors[55]) and power conversion in data centres, but the efficiency of on-chip power networks may soon reach 80%–90%, leaving little room for improvement. Modern integrated circuit power management includes clock-network and power gating[46], per-core voltage scaling[56], charge recovery[57] and, in recent processors, a CPU core dedicated to power scheduling. Integrated circuit power consumption depends quadratically on supply voltage, which has decreased steadily for many years, but has recently stabilized at 0.5–2 V (ref. 47). Supply voltage typically exceeds the threshold voltage of FETs by a safety margin that ensures circuit reliability, fast operation and low leakage. Threshold voltage depends on the thickness of the gate dielectric, which reached a practical limit of several atoms (see section 'Engineering obstacles'). Transistors cannot operate with supply voltage below approximately 200 mV (ref. 17)—five times below current practice—and simple circuits reach this limit. With slower operation, near- and sub-threshold circuits may consume a hundred times less energy[58]. Cooling technologies can improve too, but fundamental quantum limits bound the efficiency of heat removal[59–61].

Broader limits

The study in ref. 62 explores a general binary-logic switch model with binary states represented by two quantum wells separated by a potential barrier. Representing information by electric charge requires energy for binary switching and thus limits the logic-switching density, if a significant fraction of the chip can switch simultaneously. To circumvent this limit, one can encode information in spin-states, photon polarizations, super-conducting currents, or magnetic flux, noting that these carriers have already been in commercial use (spin-states are particularly attractive because they promise high-density nonvolatile storage[63]). More powerful limits are based on the amount of material in the Earth's crust (where silicon is the second most common element after oxygen), on atomic spacing (see section 'Engineering obstacles'), radii, energies and bandgaps, as well as the wavelength of the electron. We are currently using only a tiny fraction of the Earth's mass for computing, and yet various limits could be circumvented if new particles are discovered. Beyond atomic physics, some limits rely on basic constants: the speed of light, the gravitational constant,

the quantum (Planck) scale, the Boltzmann constant, and so on. Lloyd[42] and Kraus[64] extend well-known bounds by Bremermann and Bekenstein, and give Moore's law another 150 years and 600 years, respectively. These results are too loose to obstruct the performance of practical computers. In contrast, current consensus estimates from the ITRS[14] give Moore's law only another 10–20 years, due to technological and economic considerations[2].

Asymptotic space–time limits

Engineering limits for deployed technologies can often be circumvented, while first-principles limits on energy and power are loose. Reasonably tight limits are rare.

Limits to parallelism

Suppose we wish to compare a parallel and sequential computer built from the same units, to argue that a new parallel algorithm is many times faster than the best sequential algorithm (the same reasoning applies to logic gates on an integrated circuit). Given N parallel units and an algorithm that runs M times faster on sufficiently large inputs, one can simulate the parallel system on the sequential system by dividing its time between N computational slices. Since this simulation is roughly N times slower, it runs M/N times faster than the original sequential algorithm. If this original sequential algorithm was the fastest possible, we have $M \leq N$. In other words, a fair comparison should not demonstrate a parallel speedup that exceeds the number of processors—a superlinear speedup can indicate an inferior sequential algorithm or the availability of a larger amount of memory to N processors. The bound is reasonably tight in practice for small N and can be violated slightly because N CPUs include more CPU cache, but such violations alone do not justify parallel algorithms—one could instead buy or build one CPU with a larger cache. A linear speedup is optimistically assumed for the parallelizable component in the 1988 Gustafson's law that suggests scaling the number of processors with input size (as illustrated by instantaneous search queries over massive data sets)[5]. Also in 1988, Fisher[65] employed asymptotic runtime estimates instead of numerical limits without considering the parallel and sequential runtime components that were assumed in Amdahl's law[66] and Gustafson's law[5]. Asymptotic estimates neglect leading constants and offer a powerful way to capture nonlinear phenomena occurring at large scale.

Fisher[65] assumes a sequential computation with $T(n)$ elementary steps for input of size n, and limits the performance of its parallel variants that can use an unbounded d-dimensional grid of finite-size computing units (electrical switches on a semiconductor chip, logic gates, CPU cores, and so on) communicating at a finite speed, say, bounded by the speed of light. I highlight only one aspect of this four-page work: the number of steps required by parallel computation grows as the $(d+1)$th root of $T(n)$. This result undermines the N-fold speedup assumed in Gustafson's law for N processors on appropriately sized input data[5]. A speedup from runtime polynomial in n to approximately $\log n$ can be achieved in an abstract model of computation for matrix multiplication and fast Fourier transforms. But not in physical space[65]. Surprising as it may seem, after reviewing many loose limits to computation, we have identified a reasonably tight limit (the impact of input–output, which is a major bottleneck today, is also covered in ref. 65). Indeed, many parallel computations today (excluding multimedia processing and World Wide Web searching) are limited by several forms of communication and synchronization, including network and storage access. The billions of logic gates and memory elements in modern integrated circuits are linked by up to 16 levels of wires (Fig. 3); longer wires are segmented by repeaters. Most of the physical volume and circuit delay are attributed to interconnect[23]. This is relatively new, because gate delays were dominant until 2000 (ref. 14), but wires get slower relative to gates at each new technology node. This uneven scaling has compounded in ways that would have surprised Turing and von Neumann—a single clock cycle is now far too short for a signal to cross the entire chip, and even the distance covered in 200 ps (5 GHz) at light speed is close to the chip size. Yet most electrical engineers and computer scientists are still primarily concerned with gates.

Implications for three-dimensional and other emerging circuits

The promise of three-dimensional integration for improving circuit performance can be undermined by the technical obstructions to its industry adoption. To derive limits on possible improvement, we use the result from ref. 65, which is sensitive to the dimension of the physical space: a sequential computation with $T(n)$ steps requires of the order of $T^{1/3}(n)$ steps in two dimensions and $T^{1/4}(n)$ in three. Letting $t = T^{1/3}(n)$ shows that three-dimensional integration asymptotically reduces t to $t^{3/4}$—a significant but not dramatic speedup. This speedup requires an unbounded number of two-dimensional device layers, otherwise there is no asymptotic speedup[67]. For three-dimensional integrated circuits with two to three layers, the main benefits of three-dimensional integrated circuit integration today are in improving manufacturing yield, improving input–output bandwidth, and combining two-dimensional integrated circuits that are optimized for random logic, dense memory, field-programmable gate-arrays, analogue, microelectromechanical systems and so on. Ultrahigh-density CMOS logic integrated circuits with monolithic three-dimensional integration[68] suffer higher routeing congestion than traditional two-dimensional integrated circuits.

Emerging technologies promise to improve device parameters, but often remain limited by scale, faults, and interconnect. For example, quantum dots enable terahertz switching but hamper nonlocal communication[69]. Carbon nanotube FETs[70] leverage the extraordinary carrier mobility in semiconducting carbon nanotubes to use interconnect more efficiently by improving drive strength, while reducing supply voltage. Emerging interconnects include silicon photonics, demonstrated by Intel in 2013 (ref. 71) and intended as a 100-Gb s^{-1} replacement of copper cables connecting adjacent chips. Silicon photonics promises to reduce power consumption and form factor.

In a different twist, quantum physics alters the nature of communication with Einstein's "spooky action at a distance" facilitated by entanglement[13]. However, the flows of information and entropy are subject to quantum limits[59,60]. Several quantum algorithms run asymptotically faster than the best conventional algorithms[13], but fault-tolerance overhead offsets their potential benefits in practice except for large input sizes, and the empirical evidence of quantum speedups has not been compelling so far[72,73]. Several stages in the development of quantum information processing remain challenging[99], and the surprising difficulty of scaling up reliable quantum computation could stem from limits on communication and entropy[13,59,60]. In contrast, Lloyd[42] notes that individual quantum devices now approach the energy limits for switching, whereas non-quantum devices remain orders of magnitude away. This suggests a possible obstacle to simulating quantum physics on conventional parallel computers (abstract models aside). In terms of computational complexity though, quantum computers cannot attain a significant advantage for many problem types[11–13] and are unlikely to overcome the Fisher limit on parallelism from ref. 65. A similar lack of a consistent general-purpose speedup limits the benefits of several emerging technologies in mature applications that contain diverse algorithmic steps, such as World Wide Web searching and computer-aided design. Accelerating one step usually does not dramatically speed up the entire application, as noted by Amdahl[66] in 1967. Figuratively speaking, the most successful computers are designed for the decathlon rather than for the sprint only.

Complexity-theoretic limits

The previous section, 'Asymptotic space-time limits', enabled tighter limits by neglecting energy and using asymptotic rather than numeric bounds. I now review a more abstract model in order to focus on the impact of scale, and to show how recurring trends quickly overtake one-off device-specific effects. I neglect spatial effects and focus on the nature of computation in an abstract model (used by software engineers) that represents computation by elementary steps with input-independent runtimes. Such limits survive many improvements in computer technologies, and are often stronger for specific problems. For example, the best-known algorithms for multiplying large numbers are only slightly slower than reading the input (an obvious speed limit), but only in the asymptotic sense: for numbers with less than a thousand bits, those algorithms lag behind simpler algorithms

in actual performance. To focus on what matters most, I no longer track the asymptotic worst-case complexity of the best algorithms for a given problem, but merely distinguish polynomial asymptotic growth from exponential.

Limits formulated in such crude terms (unsolvability in polynomial time on any computer) are powerful[74]: the hardness of number-factoring underpins Internet commerce, while the P ≠ NP conjecture explains the lack of satisfactory, scalable solutions to important algorithmic problems, in optimization and verification of integrated circuit designs, for example[75]. (Here P is the class of decision problems that can be solved using simple computational steps whose number grows no faster than a polynomial of the size of input data, and NP is the non-deterministic polynomial class representing those decision problems for which a non-deterministically guessed solution can be reliably checked using a polynomial number of steps.) A similar conjecture, P ≠ NC, seeks to explain why many algorithmic problems that can be solved efficiently have not parallelized efficiently[76]. Most of these limits have not been proved. Some can be circumvented by using radically different physics, for example, quantum computers can solve number factoring in polynomial time (in theory). But quantum computation does not affect P ≠ NP (ref. 77). The lack of proofs, despite heavy empirical evidence, requires faith and is an important limitation of many nonphysical limits to computing. This faith is not universally shared—Knuth (see question 17 in http://www.informit.com/articles/article.aspx?p=2213858) argues that P = NP would not contradict anything we know today. A rare proved result by Turing states that checking whether a given program ever halts is undecidable: no algorithm solves this problem in all cases regardless of runtime. Yet software developers solve this problem during peer code reviews, and so do computer science teachers when grading exams in programming courses.

Worst-case analysis is another limitation of nonphysical limits to computing, but suggests potential gains through approximation and specialization. For some NP-hard optimization problems, such as the Euclidean Travelling Salesman Problem, polynomial-time approximations exist, but in other cases, such as the Maximum Clique problem, accurate approximation is as hard as finding optimal solutions[78]. For some important problems and algorithms, such as the Simplex algorithm for linear programming, few inputs lead to exponential runtime, and minute perturbations reduce runtime to polynomial[79].

Conclusions

The death march of Moore's law[1,2] invites discussions of fundamental limits and alternatives to silicon semiconductors[70]. Near-term constraints (obstacles to performance, power, materials, laser sources, manufacturing technologies and so on) are invariably tied to costs and capital, but are disregarded for the moment as new markets for electronics open up, populations increase, and the world economy grows[2]. Such economic pressures emphasize the value of computational universality and the broad applicability of integrated circuit architectures to solve multiple tasks under conventional environmental conditions. In a likely scenario, only CPUs, graphics processing units, field-programmable gate-arrays and dense memory integrated circuits will remain viable at the end of Moore's law, while specialized circuits will be predominantly manufactured with less advanced technologies for financial reasons. Indeed, memory chips have exemplified Moore scaling because of their simpler structure, modest interconnect, and more controllable manufacturing, but the miniaturization of memory cells is now slowing down[2]. The decelerated scaling of CMOS integrated circuits still outperforms the scaling of the most viable emerging technologies. Empirical scaling laws describing the evolution of computing are well known[80]. In addition to Moore's law, Dennard scaling, Amdahl's law and Gustafson's law (reviewed above), Metcalfe's law[81] states that the value of a computer network, such as the Internet or Facebook, scales as the number of user-to-user connections that can be formed. Grosch's law[82] ties N-fold improvements in computer performance to N^2-fold cost increases (in equivalent units). Applying it in reverse, we can estimate the acceptable performance of cheaper computers. However, such laws only capture ongoing scaling and may not apply in the future.

The roadmapping process represented by the ITRS[14] relies on consensus estimates and works around engineering obstacles. It tracks improvements in materials and tools, collects best practices and outlines promising design strategies. As suggested in refs 17 and 18, it can be enriched by an analysis of limits. I additionally focus on how closely such limits can be approached. Aside from the historical 'wrong turns' mentioned in the 'Engineering obstacles' and 'Energy–time limits' sections above, I uncover interesting effects when examining the tightness of individual limits. Although energy–time limits are most critical in computer design[14,83], space-time limits appear tighter[65] and capture bottlenecks formed by interconnect and communication. They suggest optimizing gate locations and sizes, and placing gates in three dimensions. One can also adapt algorithms to spatial embeddings[84,85] and seek space-time limits. But the gap between current technologies and energy–time limits hints at greater possible rewards. Charge recovery[57], power management[46], voltage scaling[56], and near-threshold computing[58] reduce energy waste. Optimizing algorithms and circuits simultaneously for energy and spatial embedding[86] gives biological systems an edge (from the 'one-dimensional' nematode *Caenorhabditis elegans* with 302 neurons to the three-dimensional human brain with 86 billion neurons)[1]. Yet, using the energy associated with mass (according to Einstein's $E = mc^2$ formula) to compute can truly be a 'nuclear option'—both powerful and controversial. In a well known 1959 talk, which predated Moore's law, Richard Feynman suggested that there was "plenty of room at the bottom," forecasting the miniaturization of electronics. Today, with relatively little physical room left, there is plenty of energy at the bottom. If this energy is tapped for computing, how can the resulting heat be removed? Recycling heat into mass or electricity seems to be ruled out by limits to energy conversion and the acceptable thermal range for modern computers.

Technology-specific limits for modern computers tend to express trade-offs, especially for systems with conflicting performance parameters and properties[87]. Little is known about limits on design technologies. Given that large-scale complex systems are often designed and implemented hierarchically[52] with multiple levels of abstraction, it would be valuable to capture losses incurred at abstraction boundaries (for example, the physical layout and manufacturing considerations required to optimize and build a logic circuit may mean that the logic circuit itself needs to change) and between levels of design hierarchies. It is common to estimate resources required for a subsystem and then to implement the subsystem to satisfy resource budgets. Underestimation is avoided because it leads to failures, but overestimation results in overdesign. Inaccuracies in estimation and physical modelling also lead to losses during optimization, especially in the presence of uncertainty. Clarifying engineering limits gives us the hope of circumventing them.

Technology-agnostic limits appear to be simple and have had significant effects in practice; for example, Aaronson explains why NP-hardness is unlikely to be circumvented through physics[77]. Limits to parallel computation became prominent after CPU speed levelled off ten years ago. These limits suggest that it will be helpful to use the following: faster interconnect[18], local computation that reduces communication[88], time-division multiplexing of logic[89], architectural and algorithmic techniques[90], and applications altered to embrace parallelism[5]. Gustafson advocates a 'natural selection': the survival of the applications that are fittest for parallelism. In another twist, the performance and power consumption of industry-scale distributed systems is often described by probability distributions, rather than single numbers[91,92], making it harder even to formulate appropriate limits. We also cannot yet formulate fundamental limits related to the complexity of the software-development effort, the efficiency of CPU caches[93], and the computational requirements of incremental functional verification, but we have noticed that many known limits are either loose or can be circumvented, leading to secondary limits. For example, the P ≠ NP limit is worded in terms of worst-case rather than average-case performance, and has not been proved despite much empirical evidence. Researchers have ruled out entire categories of proof techniques as insufficient to complete such a proof[75,94]. They may be esoteric, but such tertiary limits can be effective in practice—in August 2010, they helped researchers quickly invalidate Vinay Deolalikar's highly technical attempt at proving

P ≠ NP. On the other hand, the correctness of lengthy proofs for some key results could not be established with an acceptable level of certainty by reviewers, prompting efforts towards verifying mathematics by computation[95].

In summary, I have reviewed what is known about limits to computation, including existential challenges arising in the sciences, optimization challenges arising in engineering, and the current state of the art. These categories are closely linked during rapid technology development. When a specific limit is approached and obstructs progress, understanding its assumptions is a key to circumventing it. Some limits are hopelessly loose and can be ignored, while other limits remain conjectural and are based on empirical evidence only; these may be very difficult to establish rigorously. Such limits on limits to computation deserve further study.

Received 21 December 2013; accepted 5 June 2014.

1. Cavin, R. K., Lugli, P. & Zhirnov, V. V. Science and engineering beyond Moore's law. *Proc. IEEE* **100,** 1720–1749 (2012).
   This paper reviews the historical effects and benefits of Moore's law, discusses challenges to further growth, and offers several strategies to maintain progress in electronics.
2. Chien, A. A. & Karamcheti, V. Moore's law: the first ending and a new beginning. *IEEE Computer* **46,** 48–53 (2013).
3. Herken, R. (ed.) *The Universal Turing Machine: A Half-Century Survey* 2nd edn (Springer, 2013).
4. Andreesen, M. Why software is eating the world. *The Wall Street Journal* http://online.wsj.com/news/articles/SB10001424053111903480904576512250915629460(11 (August 2011).
5. Padua, D. A. (ed.) *Encyclopedia of Parallel Computing* (Springer, 2011).
6. Shaw, D. E. Anton: a special-purpose machine that achieves a hundred-fold speedup in biomolecular simulations. In *Proc. Int. Symp. on High Performance Distributed Computing* 129–130 (IEEE, 2013).
7. Hameed, R. *et al.* Understanding sources of inefficiency in general-purpose chips. *Commun. ACM* **54,** 85–93 (2011).
8. Cong, J., Reinman, G., Bui, A. T. & Sarkar, V. Customizable domain-specific computing. *IEEE Des. Test Comput.* **28,** 6–15 (2011).
9. Mernik, M., Heering, J. & Sloane, A. M. When and how to develop domain-specific languages. *ACM Comput. Surv.* **37,** 316–344 (2005).
10. Olukotun, K. Beyond parallel programming with domain specific languages. In *Proc. Symp. on Principles and Practice of Parallel Programming* 179180 (ACM, 2014).
11. Aaronson, S. & Shi, Y. Quantum lower bounds for the collision and the element distinctness problems. *J. ACM* **51,** 595–605 (2004).
12. Jain, R., Ji, Z., Upadhyay, S. & Watrous, J. QIP = PSPACE. *Commun. ACM* **53,** 102–109 (2010).
13. Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information* (Cambridge Univ. Press, 2011).
14. International Technology Roadmap for Semiconductors (ITRS). http://www.itrs.net/ (2013).
    Documents available at this website describe in detail the current state of the art in integrated circuits and near-term milestones.
15. Sylvester, D. & Keutzer, K. A global wiring paradigm for deep submicron design. *IEEE Trans. CAD* **19,** 242–252 (2000).
16. *A Quantum Information Science and Technology Roadmap* Los Alamos Technical Report LA-UR-04–1778, http://qist.lanl.gov (2004).
17. Meindl, J. Low power microelectronics: retrospective and prospect. *Proc. IEEE* **83,** 619–635 (1995).
18. Davis, J. A. *et al.* Interconnect limits on gigascale integration (GSI) in the 21st Century. *Proc. IEEE* **89,** 305–324 (2001).
    This paper discusses physical limits to scaling interconnects in integrated circuits, classifying them into fundamental, material, device, circuit and system limits.
19. Ma, X. & Arce, G. R. *Computational Lithography* (Wiley, 2011).
20. Mazzola, L. Commercializing nanotechnology. *Nature Biotechnol.* **21,** 1137–1143 (2003).
21. Moore, G. E. Cramming more components onto integrated circuits. *Electronics* **38,** 1–4 (1965).
22. Bohr, M. Interconnect scaling—the real limiter to high performance ULSI. In *Proc. Int. Elec. Device Meeting* 241–244 (IEEE, 1995).
    This paper explains why wires, rather than gates, have become the main limiter to the performance of ultra-large integrated circuits.
23. Shelar, R. & Patyra, M. Impact of local interconnects on timing and power in a high performance microprocessor. *IEEE Trans. CAD* **32,** 1623–1627 (2013).
24. Almeida, V. R., Barrios, C. A., Panepucci, R. R. & Lipson, M. All-optical control of light on a silicon chip. *Nature* **431,** 1081–1084 (2004).
25. Chang, M.-C. F., Roychowdhury, V. P., Zhang, L., Shin, H. & Qian, Y. RF/wireless interconnect for inter-and intra-chip communications. *Proc. IEEE* **89,** 455–466 (2002).
26. Hisamoto, D. *et al.* FinFET—a self-aligned double-gate MOSFET scalable to 20 nm. *IEEE Trans. Electron. Dev.* **47,** 2320–2325 (2002).
27. Seabaugh, A. The tunneling transistor. *IEEE Spectrum* http://spectrum.ieee.org/semiconductors/devices/the-tunneling-transistor (2013).
28. Ozdal, M. M., Burns, S. M. & Hu, J. Algorithms for gate sizing and device parameter selection for high-performance designs. *IEEE Trans. CAD* **31,** 1558–1571 (2012).
29. Rutenbar, R. A. Design automation for analog: the next generation of tool challenges. In *Proc. Int. Conf. Computer-Aided Design of Integrated Circuits* 458–460 (IEEE, 2006).
30. Rutenbar, R. A. Analog layout synthesis: what's missing? In *Proc. Int. Symp. Physical Design of Integrated Circuits* 43 (ACM, 2010).
31. Ho, R., Mai, K., Kapadia, H. & Horowitz, M. Interconnect scaling implications for CAD. In *Proc. Int. Conf. Computer-Aided Design of Integrated Circuits* 425–429 (IEEE, 1999).
32. Markov, I. L., Hu, J. & Kim, M.-C. Progress and challenges in VLSI placement research. In *Proc. Int. Conf. Computer-Aided Design of Integrated Circuits* 275–282 (IEEE, 2012).
33. Puri, R. Opportunities and challenges for high-performance CPU designs and design automation. In *Proc. Int. Symp. Physical Design of Integrated Circuits* 179 (ACM, 2013).
34. Lavagno, L., Martin, G. & Scheffer, L. *Electronic Design Automation for Integrated Circuits Handbook* (CRC Press, 2006).
35. Chinnery, D. G. & Keutzer, K. *Closing the Gap Between ASIC and Custom: Tools And Techniques For High-Performance ASIC Design* (Springer, 2004).
36. Chinnery, D. G. & Keutzer, K. *Closing the Power Gap between ASIC and Custom: Tools and Techniques for Low Power Design* (Springer, 2007).
37. Sangiovanni-Vincentelli, A. L., Carloni, L. P., De Bernardinis, F. & Sgroi, M. Benefits and challenges for platform-based design. In *Proc. Design Automation Conf.* 409–414 (ACM, 2004).
38. Landauer, R. Irreversibility and heat generation in the computing process. *IBM J. Res. Develop.* **5,** 183–191 (1961).
39. Bérut, A. *et al.* Experimental verification of Landauer's principle linking information and thermodynamics. *Nature* **483,** 187–189 (2012).
40. Bennett, C. H. & Landauer, R. The fundamental limits of computation. *Sci. Am.* **253,** 48–56 (1985).
41. Aharonov, Y. & Bohm, D. Time in the quantum theory and the uncertainty relation for time and energy. *Phys. Rev.* **122,** 1649–1658 (1961).
42. Lloyd, S. Ultimate physical limits on computation. *Nature* **406,** 1047–1054 (2000).
    This paper derives several *ab initio* limits to computation, points out that modern quantum devices operate close to their energy-efficiency limits, but concludes that resulting large-scale limits are very loose.
43. Ren, J. & Semenov, V. K. Progress with physically and logically reversible superconducting digital circuits. *IEEE Trans. Appl. Supercond.* **21,** 780–786 (2011).
44. Monroe, C. *et al.* Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects. *Phys. Rev. A* **89,** 022317 (2014).
45. Saeedi, M. & Markov, I. L. Synthesis and optimization of reversible circuits — a survey. *ACM Comput. Surv.* **45** (2)**,** 21 (2013).
46. Borkar, S. Thousand-core chips: a technology perspective. In *Proc. Design Automation Conf.* 746–749 (ACM, 2007).
    This paper describes Intel's thousand-core CPU architecture with an emphasis on fine-grain power management, memory bandwidth, on-die networks, and system resiliency.
47. Rabaey, J. M., Chandrakasan, A. & Nikolic, B. *Digital Integrated Circuits A Design Perspective* (Pearson Education, 2003).
48. Bohr, M. A 30 year retrospective on Dennard's MOSFET scaling paper. *IEEE Solid-State Circ. Soc. Newsl.* **12,** 11–13 (2007).
    This paper reviews power scaling of integrated circuits, which held for 30 years, but has now broken down.
49. Taylor, M. B. Is dark silicon useful? harnessing the four horsemen of the coming dark silicon apocalypse. In *Proc. Design Automation Conf.* 1131–1136 (ACM, 2012).
50. Esmaeilzadeh, H., Blem, E. R., St.-Amant, R., Sankaralingam, K. & Burger, D. Power challenges may end the multicore era. *Commun. ACM* **56,** 93–102 (2013).
51. Yeraswork, Z. 3D stacks and security key for IBM in server market. *EE Times* http://www.eetimes.com/document.asp?doc_id=1320403 (17 December 2013).
52. Caldwell, A. E., Kahng, A. B. & Markov, I. L. Hierarchical whitespace allocation in top-down placement. *IEEE Trans. Computer-Aided Design Integrated Circ.* **22,** 716–724 (2003).
53. Adya, S. N., Markov, I. L. & Villarrubia, P. G. On whitespace and stability in physical synthesis. *Integration VLSI J.* **39,** 340–362 (2006).
54. Saxena, P., Menezes, N., Cocchini, P. & Kirkpatrick, D. Repeater scaling and its impact on CAD. *IEEE Trans. Computer-Aided Design Integrated Circ.* **23,** 451–463 (2004).
55. Oestergaard, J., Okholm, J., Lomholt, K. & Toennesen, G. Energy losses of superconducting power transmission cables in the grid. *IEEE Trans. Appl. Supercond.* **11,** 2375 (2001).
56. Pinckney, N. R. *et al.* Limits of parallelism and boosting in dim silicon. *IEEE Micro* **33,** 30–37 (2013).
57. Kim, S., Ziesler, C. H. & Papaefthymiou, M. C. Charge-recovery computing on silicon. *IEEE Trans. Comput.* **54,** 651–659 (2005).
58. Dreslinski, R. G., Wieckowski, M., Blaauw, D., Sylvester, D. & Mudge, T. Near-threshold computing: reclaiming Moore's law through energy efficient integrated circuits. *Proc. IEEE* **98,** 253–266 (2010).
59. Pendry, J. B. Quantum limits to the flow of information and entropy. *J. Phys. Math. Gen.* **16,** 2161–2171 (1983).
60. Blencowe, M. P. & Vitelli, V. Universal quantum limits on single-channel information, entropy, and heat flow. *Phys. Rev. A* **62,** 052104 (2000).
61. Whitney, R. S. Most efficient quantum thermoelectric at finite power output. *Phys. Rev. Lett.* **112,** 130601 (2014).
62. Zhirnov, V. V., Cavin, R. K., Hutchby, J. A. & Bourianoff, G. I. Limits to binary logic switch scaling—a Gedanken model. *Proc. IEEE* **91,** 1934–1939 (2003).

63. Wolf, S. A. *et al.* Spintronics: a spin-based electronics vision for the future. *Science* **294,** 1488–1495 (2001).
64. Krauss, L. M. & Starkman, G. D. Universal limits on computation. Preprint at http://arxiv.org/abs/astro-ph/0404510 (2004).
65. Fisher, D. Your favorite parallel algorithms might not be as fast as you think. *IEEE Trans. Comput.* **37,** 211–213 (1988).
66. Amdahl, G. M. Computer architecture and Amdahl's law. *IEEE Computer* **46,** 38–46 (2013).
67. Mak, W.-K. & Chu, C. Rethinking the wirelength benefit of 3-D integration. *IEEE Trans. VLSI Syst.* **20,** 2346–2351 (2012).
68. Lee, Y.-J., Morrow, P. & Lim, S. K. Ultra high density logic designs using transistor-level monolithic 3D integration. In *Proc. Int. Conf. Computer-Aided Design of Integrated Circuits* 539–546 (IEEE, 2012).
69. Sherwin, M. S., Imamoglu, A. & Montroy, Th. Quantum computation with quantum dots and terahertz cavity quantum electrodynamics. *Phys. Rev. A* **60,** 3508 (1999).
70. Shulaker, M. *et al.* Carbon nanotube computer. *Nature* **501,** 526–530 (2013).
71. Simonite, T. Intel's laser chips could make data centers run better. *MIT Technol. Rev.* (4 September 2013).
72. Rønnow, T. F. *et al.* Defining and detecting quantum speedup. *Science* **20,** 1330–1331 (2014).
This paper shows how to define and measure quantum speedup, while avoiding pitfalls and overly optimistic results—an empirical study with a D-Wave 2 chip with up to 503 qubits finds no convincing evidence of speed-up.
73. Shin, S. W., Smith, G., Smolin, J. A. & Vazirani, U. How 'quantum' is the D-Wave machine? Preprint at http://arxiv.org/abs/1401.7087 (2014).
74. Sipser, M. *Introduction to the Theory of Computation* 3rd edn (Cengage Learning, 2012).
75. Fortnow, L. The status of the P versus NP problem. *Commun. ACM* **52,** 78–86 (2009).
76. Markov, I. L. Know your limits: a review of 'limits to parallel computation: P-completeness theory'. *IEEE Design Test* **30,** 78–83 (2013).
77. Aaronson, S. Guest column: NP-complete problems and physical reality. *SIGACT (ACM Special Interest Group on Algorithms and Computation Theory) News* **36,** 30–52 (2005).
This paper explores the possibility of efficiently solving NP-complete problems using analog, adiabatic and quantum computing, protein folding and soap bubbles, as well as other proposals for physics-based computing— it concludes that this is unlikely, but suggests other benefits of studying such proposals.
78. Vazirani, V. *Approximation Algorithms* (Springer, 2002).
79. Spielman, D. & Teng, S.-H. Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time. In *Proc. Symp. Theory of Computing* 296305 (ACM, 2001).
80. Getov, V. Computing laws: origins, standing, and impact. *IEEE Computer* **46,** 24–25 (2013).
81. Metcalfe, B. Metcalfe's law after 40 years of ethernet. *IEEE Computer* **46,** 26–31 (2013).
82. Ryan, P. S., Falvey, S. & Merchant, R. When the cloud goes local: the global problem with data localization. *IEEE Computer* **46,** 54–59 (2013).
83. Wenisch, T. F. & Buyuktosunoglu, A. Energy-aware computing. *IEEE Micro* **32,** 6–8 (2012).
84. Bachrach, J. & Beal, J. Developing spatial computers. Technical Report MITCSAIL-TR-2007–017 (MIT, 2007).
85. Rosenbaum, D. Optimal quantum circuits for nearest-neighbor architectures. Preprint at http://arxiv.org/abs/1205.0036; in *8th Conf. on the Theory of Quantum Computation, Communication and Cryptography* 294–307 (Schloss Dagstuhl—Leibniz-Zentrum fuer Informatik, 2012).
86. Patil, D., Azizi, O., Horowitz, M., Ho, R. & Ananthraman, R. Robust energy-efficient adder topologies. In *Proc. IEEE Symp. on Computer Arithmetic* 16–28 (IEEE, 2007).
87. Brewer, E. CAP twelve years later: how the 'rules' have changed. *IEEE Computer* **45,** 23–29 (2012).
88. Demmel, J. Communication-avoiding algorithms for linear algebra and beyond. In *Proc. Int. Parallel and Distributed Processing Symp.* 585 (IEEE, 2013).
89. Halfill, T. R. Tabula's time machine. *Microprocessor Report* (29 March 2010).
90. Dror, R. O. *et al.* Overcoming communication latency barriers in massively parallel scientific computation. *IEEE Micro* **31,** 8–19 (2011).
91. Dean, J. & Barroso, L. A. The tail at scale. *Commun. ACM* **56,** 74–80 (2013).
92. Barroso, L. A., Clidaras, J. & Hölzle, U. *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines* 2nd edn (Synthesis Lectures on Computer Architecture, Morgan & Claypool, 2013).
93. Balasubramonian, R., Jouppi, N. P. & Muralimanohar, N. *Multi-Core Cache Hierarchies* (Synthesis Lectures on Computer Architecture, Morgan & Claypool, 2011).
94. Aaronson, S. & Wigderson, A. Algebrization: a new barrier in complexity theory. *ACM Trans. Complexity Theory* **1** (1), http://www.scottaaronson.com/papers/alg.pdf (2009).
95. Avigad, J. & Harrison, J. Formally verified mathematics. *Commun. ACM* **57,** 66–75 (2014).
96. Asenov, A. Random dopant induced threshold voltage lowering and fluctuations in sub-0.1 m MOSFETs: a 3-D ''atomistic'' simulation study. *IEEE Trans. Electron. Dev.* **45,** 2505–2513 (1998).
97. Miranda, M. The threat of semiconductor variability. *IEEE Spectrum* http://spectrum.ieee.org/semiconductors/design/the-threat-of-semiconductor-variability (2012).
98. Naeemi, A. *et al.* BEOL scaling limits and next generation technology prospects. *Proc. Design Automation Conf.* 1–6 (ACM, 2014).
99. Devoret, M. H. & Schoelkopf, R. J. Superconducting circuits for quantum information: an outlook. *Science* **339,** 1169–1173 (2013).

Acknowledgements This work was supported in part by the Semiconductor Research Corporation (SRC) Task 2264.001 (funded by Intel and IBM), a US Airforce Research Laboratory award (FA8750-11-2-0043), and a US National Science Foundation (NSF) award (1162087).

Author Information Reprints and permissions information is available at www.nature.com/reprints. The authors declare no competing financial interests. Readers are welcome to comment on the online version of the paper. Correspondence and requests for materials should be addressed to the author (imarkov@eecs.umich.edu).