Computational Star Formation
Proceedings IAU Symposium No. 270, 2011
J. Alves, B.G. Elmegreen, J. M. Girart & V. Trimble, eds.

doi:10.1017/S174392131100072X

Future Trends in Computing

Bruce G. Elmegreen[1]

[1]IBM Research Division, T.J. Watson Research Center,
1101 Kitchawan Road, Yorktown Hts., NY 10598 USA
email: bge@us.ibm.com

Abstract. According to a Top500.org compilation, large computer systems have been doubling in sustained speed every 1.14 years for the last 17 years. If this rapid growth continues, we will have computers by 2020 that can execute an Exaflop (10^{18}) per second. Storage is also improving in cost and density at an exponential rate. Several innovations that will accompany this growth are reviewed here, including shrinkage of basic circuit components on Silicon, three-dimensional integration, and Phase Change Memory. Further growth will require new technologies, most notably those surrounding the basic building block of computers, the Field Effect Transistor. Implications of these changes for the types of problems that can be solved are briefly discussed.

Keywords. methods: n-body simulations, methods: numerical

1. Introduction

We can make progress in numerical star formation only as fast as we make progress in its three main components: our understanding of the important physical processes, our ability to program these processes for accurate simulations in a computer, and the capabilities of the computers that are used. This talk describes the expected progress of computer capabilities in the coming decade.

Our first consideration should be the scale of computations that are being done today. The recent SPH simulation of star formation in a cluster by Bate (2009) took $\sim 10^{17}$ floating point instructions and resulted in 2 TeraBytes of movie-format output. The AMR simulations by Kritsuk *et al.* (2007) and Norman *et al.* (2009) used $\sim 10^{17}$ flops and generated ∼20 TBytes of data. On the fastest machine in the world, which can run a well-tuned problem at ∼1 Petaflop (10^{12} floating point operations per second), a simulation with 10^{17} flops would take 28 hours. In fact these simulations were run on smaller machines for longer times. Generally we choose to run problems that finish in a reasonable time, such as a day or a week or a month, depending on how important and unique the problem is. Computer centers rarely give their full machine power to a single user. If these two things remain true, the time we are willing to wait for results and our willingness to share, then the scale of our computation will progress with the overall power of our computers.

Figure 1 shows the sustained speed of the fastest computers in the world over the last 17 years (middle squares), and the sustained speed of the 500th-fastest computers (bottom squares) (reference: Top500.org). The sums of all computers in the Top500.org lists are the top squares. All three speeds increase exponentially with time, and they follow each other well. It takes about 7 years for a system speed that is number one in the world to be bypassed by increasingly faster new computers until it is number 500 in the world. The whole field of computation scales approximately in proportion to the fastest computers.

Scaling for an individual user is determined also by the availability of funds. If a person accustomed to 1% of the time on a public system were suddenly given 100% of the time

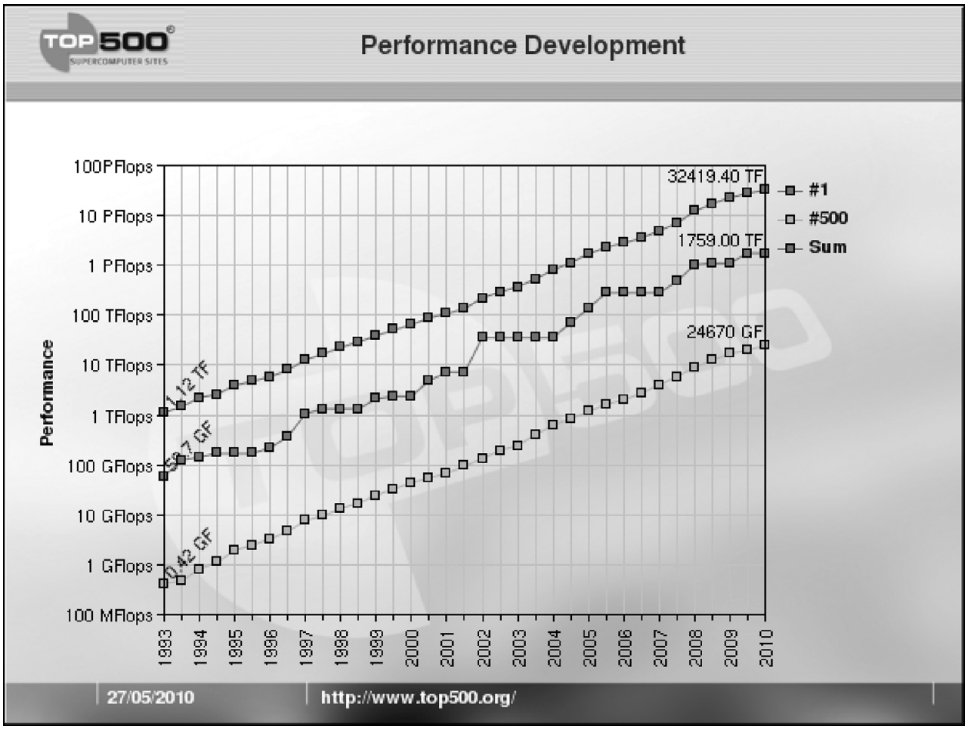

Figure 1. The sustained speed in floating point operations per second on the software package LINPACK of the Top 500 supercomputers in the world as a function of the year. The fastest computer is indicated by the middle curve and the sum of all is the top curve. From the URL http://www.top500.org/lists/2010/06/performance_development. This trend suggests an Exaflop computer (10^{18} flops per second) might be achieved by the year 2020.

on a dedicated machine of the same total size, then the jump in productivity would be a factor of 100 – much greater than the scaling factor of the technology. This boost from machine ownership is one of the strong points of dedicated hardware like GRAPE or GPU clusters. Generally the user base for these systems is much smaller than for all-purpose hardware on a campus or government network, so in addition to the speed factor from the pipelining or multicore technologies, respectively, there is also a speed factor from dedicated use. Even specialized hardware should scale with the technology, however, just like the number 1 computer.

This concept of scaling assumes that the user can program his or her algorithm to a reasonably high efficiency on any of these machines. That may not be true for all algorithms and systems. Since 2004, increasing peak speed has been the result of increasing parallelism with an approximately constant clock speed. The clock speed has been constant to keep the power consumption about constant, given that the voltage of Field Effect Transistors (FETs), has reached its minimum useful value. Power consumption scales with the square of the voltage multiplied by the clock frequency.

Increasing parallelism means that there are more separate arithmetic units, or cores, in a processor chip, and also more processor chips in a system. For example, the three fastest computers in the world, all Peta-flop scale, contain about 10^5 cores each. To make use of all this capacity, a problem has to be very finely divided into many separate computations, one set for each core. Thus scaling to larger-size problems, more grid points, for example, or more particles, is relatively easy as computers grow with increasing parallelism, but

https://doi.org/10.1017/S174392131100072X Published online by Cambridge University Press

scaling to shorter run times for the same size simulation is more difficult. Adding more physics to the same-size problem and expecting the same run time in a larger system is also difficult. If a simulation requires a very long relative run time today, such as thousands of crossing times, and is impractical to run because of that, then it may be impractical to run for a long time in the future too, if technology scaling is only by increased parallelism. This is a very different situation than what we had 20 years ago, when clock speed was increasing at an exponential rate and all problems were run on single cores. The current era of increasing parallelism, rather than increasing clock speed, is a major challenge for algorithmic development and programmers.

Along with increased parallelism has come a decrease in the memory associated with each processor. Memory includes cache for ready use by the core, and more distant on-chip memory for use several tens of clock cycles away. There is also memory on a typical processor board that is separate from the processor, and several hundred clock cycles away. Clock cycle distance means how long it takes, in step-by-step instruction cycles, to fetch a number from memory and deliver it to the core floating point register where it can be ready for multiplication or addition. If the numbers used in a sequence of calculations are random or dependent on the results of the calculation, i.e., unpredictable for future clock cycles, then the processors will spend much of their time waiting for data. Good programming means high predictability for the numbers fetched from memory. Vectors are highly predictable, for example, because the calculations can be done in the order of the vector index. In a well written program, compilers that convert user language into machine language can hide almost all of the memory latency (clock cycles to memory) inside other useful work that takes place at the same time. Generally a programmer has to iterate with the compiler, trying little tricks like unwrapping do-loops and ordering numbers in memory, in order to coax even the best compilers to compile efficient code.

The trade off between memory space and processing space on a chip depends on the use of that chip. Graphics Processing Units (GPUs), for example, were designed for very rapid graphics processing, which involves algorithms that stream low-bit data from calculations or video into numbers used by a display screen. GPUs need a lot of computation speed but not much memory per processor. If an algorithm matches the balance between the data input and output rate, the memory access rate, and the computation rate in a chip or system, then the calculation can be efficient for that hardware. Otherwise there will be a bottleneck somewhere. Most of the processors and systems that have grown rapidly in the last few years by extreme parallelism have migrated toward low ratios of memory to computation per core, often limiting their use to problems that can be very finely divided.

The trend toward greater parallelism implies that coding for algorithms has to change continuously. Professional programmers and astronomers who develop new algorithms will become an increasingly important part of team research efforts. This suggests a change toward increased specialization within astronomy subfields and increased division between programmers and users. Ten to 20 years ago, a programmer could write a code and have it scale for many years simply by increasing clock speeds, but this era is gone for a while. It may come back before the end of the decade with the advent of storage-class memory, as discussed below, but for a while, programs will have to be rewritten or significantly retuned for each new generation of hardware.

2. Future Speeds and Storage

Figure 1 suggests that with further improvements in system architecture and basic technologies, by 2020, the fastest computer in the world will be ∼1000 times faster than

https://doi.org/10.1017/S174392131100072X Published online by Cambridge University Press

the fastest today. It would run well-tuned jobs at an Exaflop per second. The trend in Figure 1 for the number 1 system is: speed = Petaflop$\times 2^{(Year-2009)/1.14}$. Thus the doubling time is 1.14 year.

The same improvements can be expected for smaller systems too. By 2020, something close to one million dollars (in today's currency) should buy a Petaflop computer with a general purpose design. This would be the scale of a university data center or a small government lab. A lap-top computer for ∼$1000 might run at a Teraflop. There should also be more specialized computers too, like GPUs and GRAPEs, which even today can be purchased with Teraflop speeds for ∼$1000. Special hardware like this might run with Petaflop speeds by 2020 and be available for private use ($1000 price).

Magnetic disk storage is increasing in capacity at an exponential rate too. For a given cost, storage capacity has increased by a factor of ∼2 every year for the last 30 years. This is faster than the increase in computation rate per dollar, which corresponds to a doubling every 1.5 years or so. The rapid growth of storage capabilities is the primary reason for the current "information age," where enormous quantities of data are available to us on the internet. By 2020, a single magnetic disk could hold ∼100 TeraBytes of data (Walter 2005) and a PetaByte could cost only $200 (Komorowski 2009), with the current trends. Exponential growth of storage means that we never have to erase anything (although we should for sanity reasons). With exponential growth, the sum of everything ever stored in a unit of a certain cost equals what can be stored in a fraction of the next generation system for the same cost.

Ray Kurzweil (2001) has considered many of the implications of increasing computer speeds if the trends continue for the next several decades. He thinks we will achieve the Human Brain capacity, which is 10^{16} computations per second, for around $1,000 by the year 2023, and for one cent by 2037. We will also achieve the Human Race capacity (10^{26} cps) for $1,000 around the year 2049, and for one cent around the year 2059. We cannot imagine how the world will differ in this era, but fortunately most graduate students today will still be around to see it.

3. Technology Improvements

Scaling for the next 10 years is not as difficult to imagine as scaling for the next 40 years because the next decade is part of the planned technology road map for many computer and chip companies. Basically, it will result from continued shrinkage of each component size, allowing more and more to fit on a single chip (increased parallelism and increased functionality), combined with a few important one-shot improvements that provide a new boost every few years. Currently, the best available technology has a smallest design scale of 45 nanometers, which means that 200 Billion design structures can be put on a 2-centimeter-square chip. For example, the most complex chips today contain around 2 Billion transistors, each of which has considerable internal structure that has to be designed on the surface of a Silicon crystal. In a few years, the target size for a minimum scale will be 32 nm, in around 5 years it will be 22 nm, and by 2017 or so, it could be 15 nm. Chip complexity and component count in two dimensions increases as the inverse square of this minimum size.

3.1. *Packaging*

Further gains are being made with new technologies in packaging. Currently, with conventional packaging, individual Silicon chips are attached to printed circuit cards with a density for communication between the chip and the card of several thousand input/output terminals per cm^2. Multiple chips can be packaged into a single module with

https://doi.org/10.1017/S174392131100072X Published online by Cambridge University Press

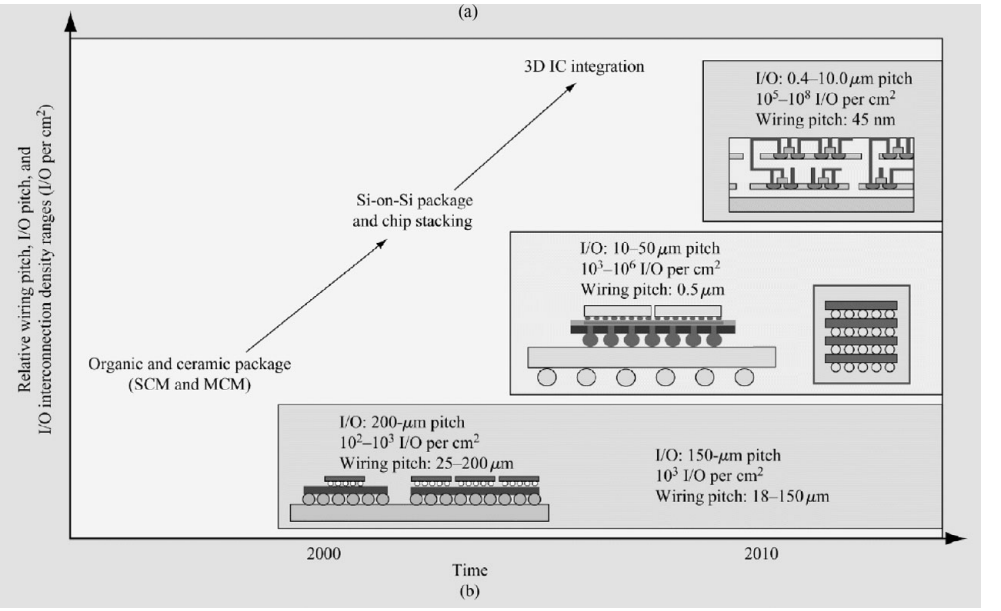

Figure 2. Roadmap of packaging showing progress toward more 3D integration and higher IO density (Knickerbocker *et al.* 2008). The abbreviations are: SCM, single-chip module; MCM, multichip module.

communication bandwidth like this from each chip to an underlying Silicon carrier layer, and slightly lower bandwidth between the Silicon carrier and a substrate. Newer technology can stack Silicon layers vertically using "through Silicon vias," which are thin metal rods that carry currents vertically from one layer to another. Vertical stacking is important because all of the functional parts of Silicon are close to the surface, built up in layers with masking, using vapor and other deposition techniques, and etching or other selective removal techniques. Stacking of two layers doubles the density of useful components in a device. A timeline for packaging improvements is shown in Figure 2.

Input/output densities are currently increasing to several 10's of thousands per cm^2. This can be done, for example, with a 2D array of microsolder bumps 25 μm in size and 50μm apart. In the next 5 years, three-dimensional Silicon integration and packaging should become even more advanced, with IO densities per chip in the millions per cm^2. Problems with cooling the middle components, alignment, and assembly will have to be overcome. At the end of this decade, 3D integration should be able to combine processors, memory, accelerators, and dense, probably optical, IO, into complex structures, enabling greater functionality, increased proximity of memory to processors, and greater IO per compute cycle.

Packaging of boards into racks and racks into systems is improving too. The current family of IBM BlueGene computers, which ranked number 1 in the world from 2004 to 2007 in terms of speed on the benchmark software LINPACK, has a 3D torus design with copper cables for all communications, board-to-board and rack-to-rack. The number 1 ranked computer in 2008 and 2009, the IBM Roadrunner at Los Alamos, has copper cables between the boards inside a rack and optical fiber connections between the racks.

3.2. *New Memory Technology*

One of the oldest bottlenecks in a computer is the memory bandwidth, which is the rate at which the processor can put and retrieve data to the memory. Current chips

https://doi.org/10.1017/S174392131100072X Published online by Cambridge University Press

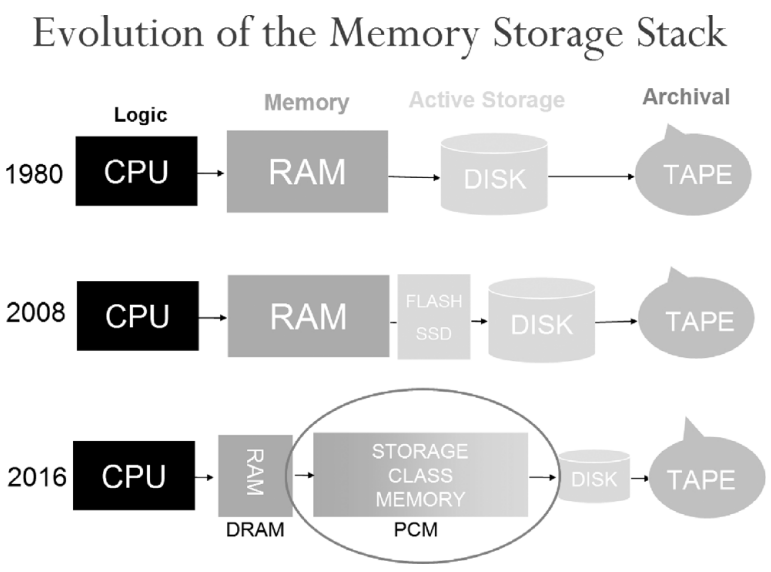

Figure 3. Schematic showing progress toward increased accessibility of memory to the central processing unit (CPU; from Freitas *et al.* 2009, with thanks to Dr. KK Rao, IBM Almaden). Random access memory (RAM) is the first step. Recently, Storage Class Memory has been inserted between the RAM and the Disk to increase the speed of high volume data access. This intermediate step should grow over time until it contains much of the memory currently in RAM and Disk.

have memory inside them ("Embedded DRAM"), which helps by decreasing the distance the current pulses have to travel, but most of the memory is still on the board that contains the chip. Standard memory is DRAM, which stands for Dynamic Random Access Memory. DRAM is very simple, consisting of a Field Effect Transistor (FET) and a capacitor. The capacitor can be a trench in the Silicon that holds electrons on the inner surface, or it can be a stack of plates above the FET gate. The value of a bit (0 or 1) depends on the level of charge on the capacitor. Charging the capacitor is done by passing a current through the FET, which acts as a switch (http://en.wikipedia.org/wiki/Dynamic_random_access_memory). This simple design was invented by Robert Dennard at IBM in 1968 and has been at the core of computer memory since 1970 when Intel released the 1103 chip (http://inventors.about.com/od/rstartinventions/a/Ram.htm). The advantage of DRAM is its simplicity, high density, and scalability with new generations of technology. The disadvantage of DRAM is that the electrons leak out of the capacitor and so all the memory units have to be recharged according to their state every 64 millisecond or so. This takes power – far too much power for continued scaling into the coming decade. DRAM also loses its state when the computer is turned off.

There are several options for "non-volatile" memory, which is the designation for memory that retains its state without power, although it still takes power for changing states. Flash is non-volatile memory used in cameras, cell-phones and in some large-scale computer memories, but Flash is slow and the number of writes before degradation is small, $\sim 10^5$ (compared to DRAM, which is $\sim 10^{15}$).

A promising new type of memory that recently came to market is Phase-Change Memory (PCM), which uses the crystalline structure of Germanium-Antimony-Tellurium compounds, or other chalcogenide glasses, to store information. In these compounds, the crystalline phase passes electricity and light, while the amorphous phase has high

https://doi.org/10.1017/S174392131100072X Published online by Cambridge University Press

electrical resistance and is opaque. It has been used in re-writable Compact Disks and DVDs since the mid-1990's. The original patent for these materials was granted to Stanford R.Ovshinsky in 1961 (http://en.wikipedia.org/wiki/Stanford_R._Ovshinsky). In September 2009, Samsung announced a 512 Mbit memory chip based on PCM and in April 2010, Numonyx BV announced a 128 Mbit memory chip.

Compared to Silicon DRAM, PCM has about the same read/write time, bandwidth, and power, but 100 times the density and is non-volatile. Compared to Flash memory, PCM has the same power, but PCM is 1000 times faster and PCM can write more before degradation (10^5 for Flash, $10^8 - 10^{12}$ times for PCRAM, 10^{12} for disks, and 10^{15} for DRAM). Compared to Disks, PCM is 10^5 times faster, 1% of the power, and has about the same number of writes before degradation. With this potential, PCM could replace disks as a storage medium, greatly increasing the speed of data storage and greatly decreasing the power consumption. With greater speed to stored data, somewhat approaching the speed to DRAM, it might be possible to return to systems with enormous memory capacity, even virtual memory or shared memory among processors. Figure 3 shows the possible evolution of logic, memory, active storage, and archival storage over the next few years. Storage class memory, based on PCM or other non-volatile types, would allow less DRAM and less disk storage, and act as an intermediate step replacing some of the function of both.

3.3. *New Transistor Technology*

The metal - oxide - semiconductor field-effect transistor (MOSFET) is the basis for much of computer technology including logic and memory. It was proposed in 1925 by Julius Edgar Lilienfeld. Today, the metal contact at the gate has been replaced by polycrystalline silicon, but newer, smaller technologies are sometimes returning to metal.

The FET has source and drain electrodes at two ends of a doped semiconductor layer like Silicon, and a gate electrode in the middle. A voltage between the gate and the substrate beneath the Silicon layer induces a field in the Silicon that opens a channel for current to flow from the source to the drain. Unlike bipolar transistors that pass a current through the gate, the gate of an FET acts like a capacitor and changes only the internal field structure. This saves power.

A problem with FETs at very small scales is that the ultrathin ($\sim$2 nm) insulator between the gate electrode and the doped Silicon passes a small current by quantum mechanical tunneling. There is also a small leakage current between the source and the drain. These currents are a net energy loss and a source of chip heating. The insulator has been improved considerably over the last few years from the most common Silicon Dioxide to new materials with high dielectric constants. A high dielectric constant allows the gate to maintain a high capacitance as the area gets smaller, without requiring the capacitor to get any thinner, thereby avoiding serious tunneling losses. Since 2007, an important material for FETs with high dielectric constant contains Hafnium.

Higher performance FETs could come from Silicon nanowires, which are fairly easy to grow and have properties that can be used in transistors (Schmidt *et al.* 2006; Yoon *et al.* 2006), and Carbon nanotubes, which are also commonly available and can make transistors (Martel *et al.* 2001; Chen *et al.* 2008). Assembly of these nanoscale objects into useful circuitry is a challenge. An interesting new technique is to use lithography and etching to pattern artificial DNA nanostructures on a Silicon surface, which can then bind Carbon nanotubes and Silicon nanowires into useful shapes (Kershner *et al.* 2009). Commercial products with these and other novel technologies are not likely to be seen for at least 10 years.

https://doi.org/10.1017/S174392131100072X Published online by Cambridge University Press

A bigger problem for FETs is the inability to reduce the supply voltage, V. Recall that FET power depends on $V^2 f$ for clock frequency f, and that the power dissipation limit has already been reached for today's V and f; higher f requires lower V. Theis (2010) reviews two emerging concepts for FET-like switches that might operate at lower voltage. One is the Tunnel FET (Banerjee *et al.* 1987; Appenzeller *et al.* 2004), in which the source-to-drain current depends sensitively on tunneling through a barrier that is regulated by small changes in gate voltage. So far this has too small an on-state current for use in common devices. Another uses a layer of ferroelectric material in the gate dielectric that can switch polarization abruptly with small changes in the gate voltage (Salahuddin & Datta 2008). The energy barrier for the source-to-drain current depends on the polarization of this ferroelectric, so the current is controlled by the gate again. Unfortunately, ferroelectric switching is too slow for useful devices at the present time. Nevertheless, these and other emerging concepts suggest that today's limits on clock frequency may be temporary.

4. Conclusions

Exponential growth of computer capabilities should continue into the near future, driven by the needs of scientists, financiers, industries, and the general public, but the rate of growth will depend on the state of the world economy and the ability of research, development, and manufacturing labs to overcome technology challenges. Exponential growth implies that all that computers have ever done throughout history can be repeated in the next e-folding time. The same would be true for astronomical observations, considering that detectors are following the same technology curve. One wonders whether our ability to understand the results of our computations and observations can keep up with such a rapid pace in technology development.

Acknowledgements: Helpful comments and information were provided by Drs. Thomas Theis and KK Rao, of IBM Research in Yorktown, NY, and Almaden, CA, respectively.

References

Appenzeller, J., Lin, Y.-M., Knoch, J., & Avouris, Ph., 2004, *Phys. Rev. Lett.*, 93, 19
Banerjee, S., Richardson, W., Coleman, J., & Chatterjee, A. 1987, *IEEE Electron Device Lett.*, 8, 347
Bate, M.R. 2009, *MNRAS*, 392, 590
Chen, Z. *et al.*, 2008, *IEEE EDL*, 29, 183
Freitas, R., Wilcke, W., Kurdi, B., & Burr, G. 2009, FAST 2009 Tutorial T3, http://www.usenix.org/events/fast09/tutorials/T3.pdf
Kershner, R.J., *et al.* 2009, *Nature Nanotechnology*, 4, 557
Knickerbocker, J.U. *et al.* 2008, *IBM J. Res.& Dev.*, 52, no. 6, p 553
Komorowski, M. 2009, http://www.mkomo.com/cost-per-gigabyte
Kritsuk, A. G. Norman, M. L., Padoan, P., & Wagner, R. 2007, *ApJ*, 665, 416
Kurzweil, R. 2001, Lifeboat Foundation Special Report, Law of Accelerating Returns, http://lifeboat.com/ex/law.of.accelerating.returns
Martel, R., Wong, H.-S.P., Chan, K., & Avouris, Ph. 2001, *IEDM Tech. Dig.*, 159
Norman, M. L., Paschos, P. & Harkness, R. 2009, *J. Phys., Conf. Ser.*, 180 012021
Salahuddin, S. & Datta, S. 2008, *Nano Lett.*, 8, 405
Schmidt, V., Riel, H., Senz, S., Karg, S., Riess, W., & Gösele, U. 2006, *Small*, 2, 85
Theis, T. N. 2010, *Science*, 327, 1600
Walter, C. 2005, *Scientific American*, August Issue
Yoon, C., *et al.* 2006, *Nanotech. Materials Devices Conference*, IEEE, 1, 424