PROSPECTS OF THE NEW CENTURY

Past and future of computing science

MIN Yinghua (闵应骅)

(Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China)

Received February 16, 2000; revised March 20, 2000

Abstract The invention of computer and development of computing technology are great historical events in the twentieth century, which impact on society and the daily life of the human beings drastically. The development of computing science and technology has been one of the primary forces driving people to enter the so-called information age. This paper attempts to present some viewpoints on the past and future of computing science, system-on-chip, software and man-machine interaction, with emphasis on computing science.

Keywords: computing science, system-on-chip, software, man-machine interaction.

When the next century is coming the retrospect and prospect of development of computing science and computer technology become a hot topic of great interest[1, 2]. In fact, the invention of computer was a great historical event in the twentieth century. In 1941, Zuse completed Z3, the first fully functional program-controlled electromechanical digital computer. In 1945, Eckert and Mauchly signed a contract to build the EDVAC (electronic discrete variable automatic computer); and von Neumann introduced the concept of stored program in a June 30 draft report on the EDVAC design. In March 1951, the first Univac 1 was delivered to the US Census Bureau, which could be considered the first commercial computer. Nowadays computers can be found everywhere, and come to all walks of production and life. The information industry has exceeded automobile industry, and becomes the largest industry. Information, as a resource, is described, stored, and transformed by computers. But on the other hand, it has been widely recognized from the experience of environment protection that the human beings will hurt themselves when they do whatever they want without any restriction. Therefore, it is natural for people to be concerned with the past and future of computing science. However, just as Hamming pointed out, it requires a brave person, or a fool to attempt a detailed and accurate prediction for the next century of computing[3]. In fact, nobody could predict what would happen to computers fifty years hence.

This paper neglects technical details, but presents some comments on computing science in some critical sub-areas, including system chips, software, and man-machine interaction. Some viewpoints may be quite different from the public. Any comments and points are welcome.

* Project supported by the National Natural Science Foundation of China (Grant No. 69733010).

1 Computing science

1.1 Computing as a discipline

Computing science is defined as the systematic study of algorithmic processes that describe and transform information: their theory, analysis, design, efficiency, implementation and application[4]. Some people call it computer science, and some other people even do not agree with the viewpoint that it is an independent discipline. In fact, it is not appropriate to call it computer science because a computer is a kind of machine. It is built based on some scientific principles, but it itself cannot be a discipline. We do not have radio science, television science, bridge science or building science. They are based on radio electronics, electrical engineering or mechanics respectively, while the research and development of computers are based on computing science.

In 1937, Turing's paper "On computable numbers" presented the concept of the Turing machine, which was considered a completely general model for abstract computing. After that, many computing models were presented, but they were equivalent to the Turing machine, even they could be very different in form. Corresponding to von Neumann's single instruction sequential stored program computers, multiple instruction and multiple data flow computers appeared. We can see many variations in computer architecture, but essentially they are serial machines. Does a parallel machine mean many serial machines working simultaneously? My answer is no. It is commonly used to calculate the speed of a parallel computer by multiplying the number of serial computers by the speed of every serial computer. Unfortunately users cannot get a 100% computing capability. For a single computing problem, it is necessary that those serial computers have to communicate with each other, and that the parallel machine has to do something serially. The problem is that computing science does not give a truly parallel computing model to essentially improve and generalize the Turing machine model. Computing science does not outdo the logical model to consider computing performance in the computing model. It is a challenging problem how long the Turing machine model or the von Neumann model can survive.

1.2 Digital and analog

Digitalization is developing extremely rapidly nowadays, especially in China. It brings us a lot of convenience indeed. However, there are two problems neglected in digitalization. First, we are living in an analog world. Most of the quantities in the real world are analog signals. We need to transform analog signals into digital ones for processing, and transform the digital output into analog. Digital signals can only be used during information processing. The additional work restricts its usage greatly. The second problem is the discontinuity of digital computers, as Dijkstra pointed[5]. In the real world, a vast majority of our mechanisms are viewed as analogue devices whose behavior is, in a large range, a continuous function of all parameters involved. But in the discrete world of computing, changes of a single bit can have the most drastic consequences. There is no meaningful metric in which "small" changes and "small" effects go hand in hand, and there never will be. This discontinuity is just the essential reason for the unreliability of digital systems. A bridge cannot crash down due to a loose bolt. But this kind of potential risk always exists for digital computers. The introduction of Boolean process[6] is an attempt to fill and level up the gap.

1.3 Complexity and modeling

The complexity of problems that computers deal with is increasing constantly. It comes from two aspects. The first comes from computers themselves. In 1950s, computer-processing speed in instructions per second was about 10^3, but it is 10^9 by 2000. An increase of six orders of magnitude in speed is realized. However, users may not feel that fast. Why? A single-user single-task computer does your work solely with all its effort, while multiuser multitask network computers have to take care of the management of all the users and tasks they serve. Your work is arranged and managed under their social governance. A computer becomes a member of a complicated society. It takes time and effort to deal with the society. The other source of complexity comes from the problem to be handled. From the viewpoint of computational complexity, only a few problems are of polynomial complexity. Most of them are NP complete problems. We could not find algorithms of polynomial complexity to solve them. Even if the speed of computers is very high, and memory size is extremely large, the essence of the problem could not be changed. People always try to simplify the problems to be solved, which is the task of modeling. For instance, it is possible to see some crops grow on the screen under some conditions of soil, wet, sunshine, and seed with computer simulation. It seems attractive. But, you should be careful if you want to conclude something based on this simulation, because the simulation is based on some modeling, which can probably reflect the reality, but not necessarily so. It is very hard to describe the impact of soil and other factors on crop growth with a mathematical formula. The essential problem lies on modeling rather than simulation.

The invention and development of computers are great historical events in the twentieth century. Dramatic are their technological progress in speed and storage capability and the versatility of applications. Especially, pipelining and caching can be considered two breakthroughs in computer engineering. But no breakthrough has been reported recently in computing science itself to clarify some essential problems.

2 System chips

During the last five decades, IC technology advanced dramatically. It was the main driving force to accelerate the development of computers. According to the prediction of the Semiconductor Industry Association (SIA), in 2005 a minimum feature size of 0.1 μm of process technology will be used; 200 million transistors or 40 million logic transistors can be integrated on a chip of 520 mm^2. The clock frequency can be as high as 2.0—3.5 GHz; and it can have 4000 I/O connections with 7—8 wiring levels. The supply voltage will be 0.9—1.2 V, supply current, 160 A and power dissipation, 160 W. System chips integrates CPUs, memories, I/O devices, digital-to-analog, analog-to-digital converters, sensors, and even optical and microelectromechanical (MEM) elements on a chip, and thus they are termed system-on-a-chip, system-on-chip or system chip[7].

The above retrospection shows that the development of high-performance, large-scale, and high reliable computers depends on the rapid advance of IC technology. In 1980, Bell Labs believed that UNIX would become the world's dominant operating system. In 1982, Bill Gates thought that a main memory of 640 K would suffice for user workspaces on operating systems for many years to come. Unfortunately, none of these things happened. But, the prediction, named Moore's law given by Intel' s Founder and Chairman, Gordon Moore in 1965, has been valid up to now. Moore's law stipulates the doubling in circuit complexity every 18 months, which is nicely illustrated here by the number of bits per chip of dynamic random-access memory (DRAM). Moore's law says that every three years semiconductor memory capabilities increase fourfold. In fact, the years in which type of chip was first introduced are as follows: 1972(1K), 1975(4K), 1978(16K), ⋯, 1996(64M). This trend is likely to continue until 2010, say 128 megabytes in 2001, and 8 gigabytes in 2010. But, it is not clear how the formula can be derived. Many physicists predicted the limitation of IC technology to demonstrate the limitation of Moore's law. It may be correct theoretically. However, the semiconductor-technology juggernaut shows no signs of weakening as researchers are laboring to create tools and to acquire knowledge needed to build ICs.

Just due to the advance of IC technology, the speed, capability and reliability of computers enhance several orders of magnitude, making it possible to integrate a system-on-a-chip, and/or a parallel computing system with hundreds of microprocessors. Researchers working on computer architecture and organization have bright prospect due to the tremendous support of system sources. In the next century, computer designers perform their creative work on the tremendous stage from algorithm, hardware/software co-design to architecture along the top-down approach. Computer architects may no longer build computers with existing microprocessors, but design their own computers by using the broad capacity the system chips provide. During this process, intellectual property (IP) blocks that already exist will be extensively used. On the other hand, computers are a necessary for the development of IC technology. As early as 1964, IBM developed a CAD (Computer-Aided Design) system. Up to now, Electronic Design Automation (EDA) has become an industry. Computer architects consider how to use hundreds of millions of transistors on a chip efficiently, while computer designers consider hardware-software co-design and logical design and test. Those are big challenges facing computer scientists and engineers. In 1970, the Intel 4004 with a complexity of about 1 000 transistors required 4 designers to work over a period of 9 months. In 1982, the Intel 80286 with a complexity of about 3 million transistors required a total of 20 designers. In 1992, the Intel Pentium with about 3 million transistors required 100 designers. It is forecasted that circuits will have a complexity of about 150 million transistors in 2001, and require perhaps 1 000 designers. The figures show that design productivity is a problem to be taken into account. Furthermore, since transistors in a chip increases dramatically in number, the size is tending to its smallest limit, and the operating frequency approaches the range of wireless frequency, problems related to circuit distributed parameters have to be considered. A 500 MHz microprocessor works with a period of 2 ns. Including the pulse rising and falling times, and attenuating surge, the signal would be stable on logic 1 or 0 in a very short time, even no instant. It is very difficult for us to observe the waveforms. In this case, design, simulation and test can no longer be performed only at logic level. In addition, a system-on-a-chip should contain some analog circuitry, even sensor. We need an efficient and systematic approach to dealing with the new design problems for system chips. Facing the bright future of computing, computing scientists will play a major role in the next century.

3 Software

A conference sponsored by the Science Committee of the North Atlantic Treaty Organization

(NATO) addressed the "software crisis" and introduced the term "software engineering" in 1968. Dijkstra introduced the concept of structured programming in the same year[8]. The software industry has grown rapidly since then. Up to now, system software becomes extremely sophisticated, data base system becomes very large, and application softwares are too numerous to count. Different programming languages and methodologies have been presented. But software crisis has not been over. Because, firstly, software complexity requires coordinated work of hundreds of people, and reuse of mature software. The software production heavily depends on the team management, rather than a systematic scientific approach. Secondly, the correctness of such a complicated software cannot be guaranteed. We do not have formal method to guarantee software reliability. The third issue is the transformation of series programs to parallel ones. It is not reasonable to throw away all past hardware and software as garbage. We produce new versions and new programs everyday. Software should have its inheritance. How to solve this problem has been a long-standing argument. Dijkstra thinks that software engineering should be known as "The doomed discipline", and that attempting to solve the software production problem with "programming tools" shows the shallowness of the underlying analogy[5]. He pins his hope on formal methodology, and claims that programming a procedure corresponds to proving a theorem, and designing a digital system corresponds to building a mathematical theory[9]. He also claims that from an operational point of view, a program can be a manipulator of abstract symbols; the designer had better regard the program as a sophisticated formula[9]. But, the variation of software seems to be all-inclusive, and complete formalization would be as difficult as formalization of sociology and economics. A good solution can, however, be obtained with formal method for problems of high computational complexity in some particular domains.

Hamming drew an analogy of novel writing with software writing. In both cases the essence is clear thinking first followed by the straightforward writing of the actual work[3]. To what extent can great writers be taught? Shakespeare did not take creative writing courses, and most people who take creative courses do not end up among the greatest writers. All that programming courses can do, apparently, is to make poor programmers a bit better. There is little evidence that experience with many programming languages and many years of writing programs makes people much better at programming. Meanwhile, nor do we see major amounts of novels copied into other novels. Therefore, he suspected the effectiveness of software reuse. In the past fifty years, it is about a factor of ten easier to write a program, while the speedup of computers is more than a millionfold. For large programs the speedup is less than the small ones, merely suggesting that it takes more time for the human brain to process the data defining writing effort itself. Good programmers outperform poor ones by more than a factor of ten. Automatic programming often results in a longer instruction sequence than programmers write, and slows down the computing efficiency. It is expected in the next century we will be able to solve the software production problem with reliable and efficient formal methodologies for different kinds of software. But for operating systems, it will be as difficult as establishing a satisfactory government for everyone.

4 Man-machine interaction

Intellegentized computers make it possible for computers and people to closer and more friendly. In the past, a computer could only communicate with one person at a time. But, now it is possible for all people to communicate and interact with any computers all over the world. However, as the builder of computers, human beings still consider computer as a tool.

4.1 The term of "electronic brain"

In 1950, IBM and Univac bet that computers would become the engines to run large businesses, hence the news media were calling them "electronic brains"[2]. Up to now, the term is quite popular in commercial usage especially in Asia. But, from the scientific point of view, it is not appropriate to call a computer an electronic brain. Firstly, the intelligence of a computer even today is far inferior to that of human brain. Computing science can get some elicitation from mechanism of human brain. But, many details of the brain are still unknown to human beings themselves. Furthermore, computers cannot realize many known functions of the brain. Secondly, a computer is not necessarily "electronic". It can be optical or quantum, etc. Computing science does not restrict itself to electronic digital computers.

4.2 Artificial intelligence

In 1956, McCarthy and Minsky chaired a meeting, and developed the concept of artificial intelligence (AI)[1]. McCarthy formed MIT's Artificial Intelligence Department in 1957, and developed LISP for AI applications in 1959[1]. Colmerauer at the University of Marseille developed PROLOG, which popularized the key logic programming concepts. In 1982, Japan launched its "fifth generation" computer project focusing on AI. But, it has not been clear what AI expected to do. The AI projects in the US were soon contracted. But in China, almost all computer departments at universities have done research on AI since the late 1980s. The defeat of the fifth generation computer project in Japan reminds people to think what human beings want computers to do. Either computers do whatever people cannot do, and do not want to do, or computers are built as smart as people, and capable of doing anything as human beings. If it is the latter, people are seeking for trouble. Human beings are good at many things like walking, seeing, sensing, and imprecise reasoning, but are not good at others such as arithmetic. The differences between humans and machines are just mutually complementary, even while the interaction between them are still remarkably poor[10]. Today, computer visualization, pattern recognition, and intelligence computing are the efforts for improving human-machine interaction. Computer applications are changing from transforming data to directly interacting with people. Miniaturization of components has enabled systems to be wearable and nearly invisible, and then individuals can move about and interact freely, supported by their personal information domain. It is even possible to make neural interfaces using biological signals to control a computer to link the brain and the machine.

4.3 Network

In 1968, the Rand Corp. presented a concept of decentralized communications network to ARPA, and in 1971, Tomlinson sent the first network e-mail message. By the end of this century, the network covering everywhere in the world integrating computer network with communication technologies is considered a remarkable progress in civilizations. It provides not only interface between humans and heterogeneous, multitask computer systems all over the world, but also a close link among human beings. The information service industry will promote economics in the twenty-first century. Information networks, the transport part of that industry, are emerging as critical infrastructure for commerce, like ships and railways in the nineteenth century and automobiles and airplanes in the twentieth century. The World Wide Web (WWW) is the universe of global network-accessible information. It is an abstract space within which people can interact through inter-linked pages of text, images, and animations with sounds, videos, and three-dimensional worlds in spite of incompatibility of computer systems. There is no doubt that it will greatly impact on the society and economics. This ideality is attractive, and rapid developments have been achieved. But, from technological point of view there are still lots of difficulties, and from the political and economical point of view, its full implementation would not be easy. We are far away from the realization of cosmopolitanism.

5 Concluding remarks

The invention and development of computers are great historical events in the twentieth century. Several orders of magnitude of enhancement in speed, storage space, computing capability, reliability, and miniaturization of computers have been achieved in the area of computer engineering, which are out of predictions. Unfortunately, from the viewpoint of scientific development, no astonishing breakthrough has ever been reported. Many new problems in computer technology are under investigation. Hopefully, computing scientists will advance the computing science largely, especially in the areas of system chips, software, and man-machine interaction in the next century.

In the new century, as the largest developing country in the world, China needs more achievements in the core technology of computing science and computer technology, rather than only in computer applications. Freedom in research is necessary, although big scientific engineering projects should be well scheduled under the control of the government.

References

1 IEEE Computer Society, 50 years of computing, *Computer*, *Innovation Technology for Computer Professionals*, 1996, 29 (10): 24.

2 Denning, P. J., Metcalfe, R. M., *Beyond Calculation: The Next Fifty Years of Computing*, New York: Springer-Verlag, 1997.

3 Hamming, R. W., How to think about trends, in *Beyond Calculation: The Next Fifty Years of Computing*, New York: Springer-Verlag, 1997, 65—74.

4 Denning, P., Computing as a discipline, *Communications of ACM*, 1989, 32(1): 9.

5 Dijkstra, E. W., On the cruelty of really teaching computing science, *Communications of ACM*, 1989, 32(12): 1 398.

6 Min, Y., Li, Z., Zhao, Z., Boolean process, *Science in China*, Series E, 1997, 40(3): 250.

7 Geppert, L., Special report: semiconductors, the 1000-million transistor IC, *IEEE Spectrum*, 1999, 36(7): 22.

8 Dijkstra, E. W., Goto statement considered harmful, *Communications of ACM*, May 1968.

9 Dijkstra, E. W., The tide, not the waves, in *Beyond Calculation: The Next Fifty Years of Computing*, New York: Springer-Verlag, 1997, 59—64.

10 Norman, D., Why it's good that computers don't work like the brain, in *Beyond Calculation: The Next Fifty Years of Computing*, New York: Springer-Verlag, 1997, 105—116.