Computing: 50 Years On

Mathai Joseph*

From a TIFR Alumni Association Public Lecture
27 July 2007

Introduction

We may each remember a year for different things: some personal, some related to our work and interests and some for the unusual events that took place. 1957 is probably like any other year in this respect. For many scientists, the launch of the first Sputnik from Baikonur in the then-USSR will be remembered as the event which transformed the race to space and was the stimulus for the man-on-the-moon program that followed in the United States. Physicists may remember the year for the award of the Nobel Prize to Yann Chen Ning and Lee Tsung-Dao, both from China and working respectively at the Institute of Advanced Study, Princeton, and Columbia, for their work on parity laws. We in India may recall that it marked the first full year of operation of the Apsara nuclear reactor at Trombay. And, chess-lovers will certainly remember the year for the remarkable feat by Bobby Fischer in becoming the US chess champion at the age of 14!

There were landmarks in computing in 1957. By that time, around 1000 computers had been sold worldwide (none of them in India) and new computer series (with solid-state technology) were announced by the big manufacturers such as IBM, Univac, NCR and Siemens. Development of the integrated circuit was nearing completion in the laboratories of Jack Kilby at Texas Instruments and Robert Noyce at Fairchild. The Fortran programming language was formally published by a team led by John Backus at IBM, marking the start of a great era of programming language development. And Simon, Newell and Shaw devised their General Purpose Solver (GPS) which used *Means-Ends* analysis to solve some classical problems and joined John McCarthy's pioneering work as another milestone in the development of artificial intelligence.

Computing was not cheap in those days and strenuous efforts went to making the most effective use of the available computers. Large computers cost in the region of $500,000 and were beyond the reach of most universities, even in the United States. It was only when a few well-known US universities were gifted with Univac-1 machines that they were able to venture into large-scale computing. Among them was the University of Pennsylvania, famous for the construction of ENIAC (1943-46; the first large-scale computer, used extensively by the US Army) and EDVAC (1949; the first stored program computer in the United States). The Univac-1 was a decimal machine made with 1000 vacuum tubes and 1000 6-bit words of acoustic delay line storage – a powerful machine for its time but with rapidly aging technology.

* Advisor, Tata Consultancy Services, 1 Mangaldas Road, Pune 411 001, India.

TIFRAC

This was the context in which India took the first steps towards digital computing with the design and construction of the TIFR *A*utomatic *C*alculator, TIFRAC. A pilot study from 1954-56 was used to try out different electronic designs using Indian components and the team, led by R. Narasimhan, was ready to embark on building a large digital computer. When completed in 1960, TIFRAC had 2700 vacuum tubes, 1700 germanium diodes and, with a great deal of foresight, 2048 words of the then relatively new ferrite core memory. TIFRAC also resulted in the first attempts at system programming in India. In terms of hardware technology, TIFRAC was ahead of the Univac-1.

The design and construction of TIFRAC was a remarkable feat for Indian electronics engineering. It also provided an introduction to computing for many Indian scientists. Publications from that period show just how much the calculations made possible through the use of TIFRAC contributed to the development of science in TIFR (e.g. S.K. Bhattacharya, S.K. Mitra, Beta-Gamma directional correlation in Eu^{154}, *Phys. Rev*. **126**, 1154-1159, 1962) and in other Indian institutions like Benaras Hindu University (M.L. Rustgi, S.N. Mukherjee, Octupole deformation in even-even medium mass nuclei, *Phys. Rev.* **131**, 2615-2616, 1963).

So by 1960, thanks to TIFRAC, Indian scientists were just a few years behind their counterparts in leading institutions in other countries in terms of access to scientific computing. Of course, TIFRAC had to serve the whole of the small but widely dispersed Indian scientific community, while scientists in the West and Japan had computing facilities nearer to hand.

Computing Without Computers?

Algorithms have been of mathematical interest from at least the time of the Greeks (e.g. Euclid's algorithm for finding the Greatest Common Divisor of a set of numbers). And interest in complexity can be found from the 19th Century, for example Lamé's Theorem (1845) proving an upper bound for the number of steps taken by Euclid's algorithm. An honours list of mathematicians had looked at problems in complexity, from Cantor and Hilbert to Post, Church, and Gödel.

Yet, to create the fundamental basis of computer science before the existence of any computers took a truly remarkable man, Alan Turing. A lot of Turing's early work can be traced back to the major mathematical problems of the time. However, his abstract constructions of computing machines were entirely unique. Turing invented the concept of the universal computing machine, defined computability, proved basic decidability results and worked on a wide range of areas such as formal program verification, artificial intelligence, chess programs, neural nets and (what he is often best remembered for in the public imagination) statistical methods for code breaking.

(Turing led the way in other areas too: he was a world-class long distance runner and completed a full marathon just 11 seconds slower than the Olympic winner of the time.)

It is for his work on what we now call Turing machines that he is best remembered. A Turing machine is the simplest possible abstract computation device and Turing showed that any problem that is computable is solvable on it. So deciding whether a problem is computable or not depends on whether there is an algorithm for a Turing machine to solve the problem. Not all problems can be so simply categorized: there are some problems that are undecidable. For example, given a program and an input to the program, will the program eventually stop when it is given that input? This *Halting Problem* is a classical example of an undecidable problem.

Given the breadth of his abilities and interests, it is surprising that there actually were areas in which Turing did *not* make contributions. It appears that he did not consider the speed of execution of a Turing machine, or that an infinitely fast Turing machine could 'solve' the Halting Problem. He also did not make a breakthrough in complexity theory, despite the long interest of mathematicians in complexity.

Computing in the 1960's

Across the developed world, computing picked up momentum in the 1960's. IBM announced the first large series of computers, the 360 series, intended to span all applications, from small business requirements to large-scale scientific computing. The machines initially used 'hybrid' logic modules but by the mid-1960's all models used integrated circuits. The IBM 360 Series found widespread commercial and educational use, especially because there was close integration between the system and the wide range of application software, and backed by the largest disc memories of the time.

Similar but smaller series of computers were being manufactured by other companies in the US and Europe. But it was also the period when new kinds of computing systems were being introduced by companies like Digital Equipment Corporation: these included the mini-computer, for use by a single programmer or for a single application, and time-sharing systems that allowed tens and sometimes hundreds of people to access a computer system simultaneously. There was talk of computing being provided as a utility, in the same way that power and water were. There was even a suggestion (fortunately ignored) that countries like India could avoid buying and maintaining expensive computers and rely entirely on using telecommunication links to access large computers installed in the United States!

Work had started on the design of the hardware and the software for large parallel computers, notably at the University of Illinois where Illiac IV was being developed. (Illiac IV was later to achieve a speed of 200 million operations a second, far beyond the capabilities of the fastest machines of the time.)

In India …

Commercial computing started in India in the early 1960's when an IBM 1401 was installed at the new Backbay office of Esso Standard Eastern. This was the first of a set of similar computers installed in Indian companies over the next 10-12 years.

By 1962, the first Indian transistorized computer, ISUJU, was designed and built by the Indian Statistical Institute and Jadavpur University. Shortly after that, large scientific computers were installed: a CDC 3600 at TIFR and, in 1967, an IBM 7044 at the Indian Institute of Technology in Kanpur which took over from the earlier IBM 1620 (all with discrete components, not integrated circuits). In 1968, the first Indian software and management consultancy was created: Tata Consultancy Services can rightly claim that from the time of their creation they have remained the largest Indian software consultancy.

Within a few years, it became clear that the design and construction of modern computers had moved beyond the capabilities of universities and research institutes. There were still a few examples of special-purpose computers being built (the TIFR Oldap was an example) but those activities slowly died out.

Computer Science in the 1960's

This was an enormously formative time for computer science throughout the world. After Turing's foundational work in the 1930's, the pressures of the war had moved attention towards the design and use of computers for solving large numerical problems, especially because of the demands from the military and the funding available for such applications.

By the 1960's, a great deal of attention was directed towards artificial intelligence, once again because of military interest. Areas like automated language translation (predictably, given the preoccupations of the Cold War, from Russian to English), automated speech recognition and war-gaming gained prominence. Considerable effort went into these areas but more theoretical work also prospered, especially in automated theorem proving. Programs to play chess occupied the middle ground and Mikhail Botvinnik, the 1963 World champion, predicted that "a Russian computer chess program will beat the world champion" (presumably not him).

But this trend began to change with new foundational work on the theory of algorithms and complexity theory. Turing's work had showed how to determine which problems were solvable and which were not, which were decidable, and which were not. There remained the basic question of whether all computable problems can be solved equally easily. This was the question addressed by Hartmanis and Stearns in their now-famous paper of 1965 which built on earlier work by Yamada and by Cobham. They were able to define an abstract quantification of the time and space of a computation, e.g. as the number of steps needed by a Turing machine to solve the problem and the amount of storage needed. Their work laid the foundation for modern complexity theory[†].

For computer science, the work of Hartmanis and Stearns led to the definition of the classes of P and NP problems and the still incomplete classification of the polynomial hierarchy. The work was key to defining the fundamental problem of whether P=NP, the first solution for which will receive a prize of $1 million from the Clay Mathematics Institute.

Program verification also had its modern roots in the 1960's through the work of Floyd (1967) who showed how to prove properties of programs and Hoare (1969) who defined an axiomatic basis for proving program properties. Independently, Dijkstra was leading efforts for formal program derivation, going from the desired goals to the programs that would achieve those goals. (Dijkstra described himself as a 'mathematical engineer'). The later work of Dahl, Dijkstra and Hoare on what was called 'structured programming'

[†] Juris Hartmanis lectured at TIFR in 1966 on switching theory and notes of his lectures appeared as a volume in the Lecture Notes in Physics. However, the lectures do not seem to have led to any further research in the area in TIFR.

marked the start of interest in formal programming methodology which has since influenced programming substantially.

In India …

It took many years before the new face of computer science was reflected in work done in India. There was the long-held belief that the only 'fundamental' research in computer science was in selected areas of artificial intelligence and that the main scientific applications were to solve ever more complex numerical problems. Anything else was ignored and the enormously exciting developments of the time in computer science received no attention.

Whatever be the reasons for this skewed perception, there is no argument about the severely debilitating effect of the lack of computers in India. Until the mid-1970's, there were fewer than 100 computers in the whole country and most of them were used for commercial applications in companies. Except in very few institutions, students learnt programming as a paper-and-pencil exercise and only later, and very briefly, were they able to actually execute a program through a few batch runs on a computer. How could modern computer science have grown in this inhospitable environment? It was like doing natural science without any basic experimental facilities!

The result was that the years from 1960-1975 were a dull period for computer science in India. Today, it is hard to locate any significant computer science publications from work done in India at that time.

1970's and beyond in India

By the mid-1970's, the availability of computers in India improved. At the same time, work started in some new areas as more computer scientists returned to India. Groups grew at TIFR, the Indian Institute of Science, Bangalore, the five Indian Institutes of Technology, Jadavpur University, and the Birla Institute of Technology and Science, for example. A large group with interest in programming formed at the National Centre for Software Development and Computing Techniques (NCSDCT) at TIFR.

The manufacture of computers started in India. The Electronics Corporation of India started producing, albeit in very small numbers, mini-computer class machines. Commercial computers were assembled by IBM in Mumbai and ICL in Pune. These were all machines that needed to be installed in dedicated, air-conditioned facilities, out of the reach of most of the people who wrote programs.

One major limiting constraint to growing computer technology was the lack of availability of components from Indian sources. Integrated circuits were not manufactured in India (a situation that has not improved much even today) and imports of so-called high-technology components were fraught with difficulty and delay. This led to the questionable decision to limit computer design to what was possible using Indian components. While computing technology in the rest of the world moved on to faster and more compact medium-scale and large-scale integrated circuits, computers were designed and built in India using discrete components. Bucking technology, even if purporting to support nationalism, is always futile and, as the subsequent history showed, this case was no exception.

1980’s and Onwards, in India and Elsewhere

There were two events that make the early 1980’s memorable. The first was the launch of the Space Shuttle which promised to change the way near-space would be used for travel and scientific experiments, and possibly as a staging post for longer journeys. The second was a purely commercial event: the launch of the IBM PC.

There will be long debates about which event had a greater scientific impact and which affected more people. But it is hard to deny that the IBM PC, and the path this cleared for other manufacturers’ PC’s to be sold worldwide, changed the way we all view computing. Even the World Wide Web, proposed by Tim Berners-Lee at CERN in 1989, would not have had such a fertile reception without the wide availability of computing at a personal, rather than an institutional level.

In an India that had been starved of computing facilities for so long, the arrival of the PC made computing much more widely accessible. Institutions were the first major buyers of PCs but they found often place on individual desks, rather than only in computer centres. It was much later, in the 1990’s, that PCs started to find a home market in India, following the liberalization of the economy, the rapid growth in telecommunication access in cities and the fall in communication rates. Selling a PC to an individual was still hard but a PC with an Internet connection could offer promises that vendors were quick to exploit. Not coincidentally, this was also the time when Indian software companies started to grow very rapidly.

Computer science in India saw the start of a new era. The FST&TCS conferences started by NCSDCT in 1981 grew in importance and rose to become among the most significant theoretical computer science conferences outside the West. Publishers who were loath to publish the Proceedings of FST&TCS in its early years later vied for the publishing rights. Important text books were written by Indian computer scientists based in India and highly referenced publications started to make their appearance.

More research work was being done in different research groups and began to attract wider attention. Of this, undoubtedly the most important result was the discovery announced in 2002 on primality testing. It had long been suspected that primality testing was in P (and hence potentially of practical importance) but it needed the ingenious algorithm devised by Manindra Agrawal and two undergraduate students at the Indian Institute of Technology in Kanpur to prove this. The result was important enough to bring them high honours: the Clay Research Award (2002), the Fulkerson Prize (2006) and the Gödel Prize (2006).

But Indian computer science was (and still is) very small when compared to the burgeoning Indian computer industry. There were relatively few additions to the number of computer science departments that existed in the 1980’s and very few new staff or students were added to any of them. In comparison, from small beginnings and with undiminished persistence, Indian companies had begun to make themselves known across the world and to demonstrate the effectiveness of outsourcing and off-shoring. If factorization remained out of reach of computer scientists, the Indian software industry had made factoring of IT services into a practical reality that could not be ignored.

Indian IT industry – 1980's onwards

The large Indian IT companies started their existence by forming alliances with major IT players and computer manufacturers, providing them and their customers with staff to meet their growing needs. But this was not seen as an end by any of them and at least one company invested in R&D. In 1981, Tata Consultancy Services created the Tata Research Development and Design Centre, the first R&D centre in the Indian software industry and today still the largest. Some companies chose to follow what they decided were the technology imperatives laid down by companies in the United States but others began to develop technology and build their own software tools to increase productivity.

For many years, outsourcing work was done by staff from Indian software companies working exclusively within the premises of the outsourcing company. That situation began to change by the late-1980s and work was moved to centres in India, each dedicated to a specific client. These overseas development centres, or ODCs, functioned as part of the outsourcing company in all respects but their location. The staff working there became familiar with the work ethos of the outsourcing company and worked in a computing environment that was identical to the one used in that company: the aim was to create a single team divided only by geography and time zone that would take turns in operating through long 15-20 hour working days.

Setting up an ODC marked a level of confidence that led to closer integration between the teams in different locations. Joint work became less a question of 'them' and 'us' and more about assigning responsibilities to the location where they would be carried out best. The work became more successful when the outsourcing companies became more mature in their practices and had a clear idea of how they wished to manage their IT development across the world. The ability to provide services for this kind of work was key to the success of the Indian industry and gave it a level of immunity from even the .COM failures. By 2002, just over a year after the .COM carnage among Silicon Valley companies, an Indian IT company quietly crossed $1 billion in revenues.

From the late 1990s, the Indian IT industry grew at a compounded annual rate of 28%; its share of the Indian GDP rose from 1.9% to 4.1% in 2004. By 2008, it is expected to contribute 7% to the Indian GDP and by 2010 the total revenues could reach $80-100 billion.

These are not small sums of money, and the applications that the Indian IT industry has been responsible for are by no means minor or trivial. Modern software systems represent some of the most complex artifacts ever produced and an increasing number of them now have their origins in India. When deservedly praising the achievements in other fields that we can see, we should not overlook the software that we cannot see but which makes possible the successes in fields as diverse as international financial services and intricate drug discovery systems.

The Indian IT industry has changed the whole nature of software development, from using single teams in one location to dividing work across teams in multiple locations bound together by a well-defined software process. This was a necessity for successful outsourcing work but the quality of what resulted is best assessed in terms of the growth of the companies. It is no accident that most of the highest certified IT companies in the world are in India.

Price of Success

The success of the Indian IT industry made it a magnet for employment. The demand it created led to an educational shift towards computing and today an increasing number of young people look for careers centred on IT. There has consequently been a big growth in IT-related educational courses but unfortunately this has not been matched by a growth in the number or quality of the teaching staff.

With almost no more research happening in computer science departments than a decade ago, very few research students and a worrying trend towards a further fall in educational standards, there is a great deal about computer science and engineering education that needs to be corrected. In this, the Indian IT industry needs to play a much larger and more active role. It is the major employer of these students and it will be the most affected if educational standards do not start to rise. The industry needs to work with a much larger research base than is available today: there are fewer PhD students in computer science in the whole of India than in an average university in the United States! Just as the energy companies of the 1970s began to worry about the depletion in natural sources of energy, Indian IT companies need to consider how they will continue to grow, and grow effectively, unless their sources of trained staff are augmented and the quality of education improved.

The demand from the Indian software industry has created an unfortunate bias in the spread of science and technology courses. Given the uncertainty of finding gainful employment in science and many branches of engineering, students have flocked towards computer science and IT courses. This is also not a trend that should continue. But rather than decrying the software industry despite its successes, the more important task is to find other areas of science and technology that can aspire for the same success.

But is it Science?

Computing has often been misunderstood by non-practitioners in India. In the early years, natural scientists treated it as another form of electronic instrumentation, as complex and as necessary as, say, an oscilloscope and merely another tool for scientific work. Software for commercial applications was derided as no more than electronic clerical work. That view has been heightened today because of the feeling that the indisputable success of the industry and the well-paid jobs it offers must represent a passing phase before an inevitable collapse. Yet, the industry continues to prosper, year after year …

Is computer science really a ‘science’? Is it a technology? Or it is engineering? These are valid questions because computing technology usually runs well ahead of computer science. But computing technology and computer science have complementary roles: the technology generates the ‘phenomena’ that are observed and studied, creates the need for new theory and provides the environment for validating theory. Computer science can also proceed independently, as Turing’s work showed, but thrives with a closer association with practice. Computer technology and science prosper when they work together and will wither if they are kept apart for too long.

A traditional science relies on observations of natural phenomena at different levels. Computer science is not a natural science in those terms but it is unquestionably a mathematical science. Like any science, it has theories and it makes predictions; but the phenomena it observes are from the world of technology. Computer science is related to information technology as closely as any science to its application technology. One should not be confused for the other and neither tells the whole story.

Finally …

The success of the Indian IT industry has set a challenge: can Indian computer science make as big an impact? This is not a frivolous question because unless industry and science are more closely matched in the future, they will both be confronted with barriers that could become insurmountable.

The successes of today suggest that a far better future awaits them both, provided the industry and its educational and research counterpart work as closely together as theory and practice in any science. There is a lot that both have to discover.

Acknowledgements

The occasion for this talk is the birthday of Mr. J.R.D. Tata on 29th July. His influence on Indian industry and Indian science was so outstanding and so large that it is hard to envisage where they would stand today without his discerning support at the early and critical periods of their existence. It is a great honour to be able to give this talk on such a notable occasion.

I would like to thank the TIFR Alumni Association for inviting me to give this talk and giving me the reason to collect my thoughts on computing in India over the last half century.

I have relied on a number of sources for the information I have presented here because the story starts long before I started my research career at TIFR and spans a far wider canvas than I have had the privilege to be part of. I have made liberal use of that wonderful and beguiling tool, hindsight, and it is possible that my account refers to information that was not easily available at the time the events took place.

Finally, I would like to thank former colleagues and present friends at TIFR, at Warwick University and at the Tata Research Development and Design Centre for all I have learnt from them over the years. Some of them were gracious enough to comment on earlier versions of this material. None of them is responsible for the views I have expressed or for any errors that remain in the text.