

## **UNIT 2     A HISTORICAL OVERVIEW OF THE COMPUTER**

### **CONTENTS**

- 1.0    Introduction
- 2.0    Objectives
- 3.0    Main Content
  - 3.1    A Brief History of Computer Technology
  - 3.2    First Generation Electronic Computers (1937-1953)
  - 3.3    Second Generation (1954-1972)
  - 3.4    Third Generation (1903-1972)
  - 3.5    Fourth Generation (1972-1984)
  - 3.6    Fifth Generation (1984-1990)
  - 3.7    Sixth Generation (1990-Date)
- 4.0    Conclusion
- 5.0    Summary
- 6.0    Tutor-Marked Assignment
- 7.0    References/Further Reading

### **1.0     INTRODUCTION**

The computer as we know it today has evolved over the ages. An attempt is made in this unit to present in chronological order the various landmarks and milestones in the development of the computer. Based on the milestone achievement of each era, the computer evolution is categorised into generations. The generational classification, however, is not rigid as we may find one generation eating into the next.

### **2.0     OBJECTIVES**

At the end of this unit you should be able to:

- explain the processes leading to the emergence of the modern computer
- predict the direction of research in computer technology in the near future.

### **3.0     MAIN CONTENT**

#### **3.1     A Brief History of Computer Technology**

A complete history of computing would include a multitude of diverse devices such as the ancient Chinese abacus, the Jacquard loom (1805) and Charles Babbage's "analytical engine" (1834). It would also include a discussion of mechanical, analog and digital computing architectures. As late as the 1960s, mechanical devices, such as the Merchant calculator, still found widespread application in science and engineering.

During the early days of electronic computing devices, there was much discussion about the relative merits of analog vs. digital computers. In fact, as late as the 1960s, analog computers were routinely used to solve systems of finite difference equations arising in oil reservoir modeling. In the end, digital computing devices proved to have the power, economics and scalability necessary to deal with large scale computations. Digital computers now dominate the computing world in all areas ranging from the hand calculator to the supercomputer and are pervasive throughout society. Therefore, this brief sketch of the development of scientific computing is limited to the area of digital, electronic computers.

The evolution of digital computing is often divided into generations. Each generation is characterised by dramatic improvements over the previous generation in the technology used to build computers, the internal organisation of computer systems, and programming languages. Although not usually associated with computer generations, there has been a steady improvement in algorithms, including algorithms used in computational science. The following history has been organised using these widely recognized generations as mileposts.

### **3.2 First Generation Electronic Computers (1937 – 1953)**

Three machines have been promoted at various times as the first electronic computers. These machines used electronic switches, in the form of vacuum tubes, instead of electromechanical relays. In principle the electronic switches were more reliable, since they would have no moving parts that would wear out, but technology was still new at that time and the tubes were comparable to relays in reliability. Electronic components had one major benefit, however: they could “open” and “close” about 1,000 times faster than mechanical switches.

The earliest attempt to build an electronic computer was by J. V. Atanasoff, a professor of physics and mathematics at Iowa State, in 1937. Atanasoff set out to build a machine that would help his graduate students solve systems of partial differential equations. By 1941, he and graduate student Clifford Berry had succeeded in building a machine that could solve 29 simultaneous equations with 29 unknowns. However, the machine was not programmable, and was more of an electronic calculator.

A second early electronic machine was Colossus, designed by Alan Turing for the British military in 1943. This machine played an important role in breaking codes used by the German army in World War II. Turing’s main contribution to the field of computer science was the idea of the Turing Machine, a mathematical formalism widely used in the study of computable functions. The existence of Colossus was kept secret until long after the war ended, and the credit due to Turing and his colleagues for designing one of the first working electronic computers was slow in coming.

The first general purposes programmable electronic computer was the Electronic Numerical Integrator and Computer (ENIAC), built by J.

Presper Eckert and John V. Mauchly at the University of Pennsylvania. Work began in 1943, funded by the Army Ordinance Department, which needed a way to compute ballistics during World War II. The machine wasn't completed until 1945, but then it was used extensively for calculations during the design of the hydrogen bomb. By the time it was decommissioned in 1955 it had been used for research on the design of wind tunnels, random number generators, and weather prediction. Eckert, Mauchly, and John Von Neumann, a consultant to the ENIAC project, began work on a new machine before ENIAC was finished. The main contribution of EDVAC, their new project, was the notion of a stored program.

There is some controversy over who deserves the credit for this idea, but no one knows how important the idea was to the future of general purpose computers. ENIAC was controlled by a set of external switches and dials; to change the program required physically altering the settings on these controls. These controls also limited the speed of the internal electronic operations. Through the use of a memory that was large enough to hold both instructions and data, and using the program stored in memory to control the order of arithmetic operations, EDVAC was able to run orders of magnitude faster than ENIAC. By storing instructions in the same medium as data, designers could concentrate on improving the internal structure of the machine without worrying about matching it to the speed of an external control.

Regardless of who deserves the credit for the stored program idea, the EDVAC project is significant as an example of the power of interdisciplinary projects that characterise modern computational science. By recognising that functions, in the form of a sequence of instructions for a computer, can be encoded as numbers, the EDVAC group knew the instructions could be stored in the computer's memory along with numerical data. The notion of using numbers to represent functions was a key step used by Goedel in his incompleteness theorem in 1937, work which Von Neumann, as a logician, was quite familiar with. Von Neumann's background in logic, combined with Eckert and Mauchly's electrical engineering skills, formed a very powerful interdisciplinary team.

Software technology during this period was very primitive. The first programs were written out in machine code, i.e. programmers directly wrote down the numbers that corresponded to the instructions they wanted to store in memory. By the 1950s programmers were using a symbolic notation, known as assembly language, then hand-translating the symbolic notation into machine code. Later programs known as assemblers performed the translation task.

As primitive as they were, these first electronic machines were quite useful in applied science and engineering. Atanasoff estimated that it would take eight hours to solve a set of equations with eight unknowns using a Marchant calculator, and 381 hours to solve 29 equations for 29 unknowns. The Atanasoff-Berry computer was able to complete the task in under an hour. The first problem run on the ENIAC, a numerical simulation

used in the design of the hydrogen bomb, required 20 seconds, as opposed to forty hours using mechanical calculators. Eckert and Mauchly later developed what was arguably the first commercially successful computer, the UNIVAC; in 1952, 45 minutes after the polls closed and with 7% of the vote counted, UNIVAC predicted Eisenhower would defeat Stevenson with 438 electoral votes (he ended up with 442).

### **3.3 Second Generation (1954 – 1962)**

The second generation saw several important developments at all levels of computer system design, from the technology used to build the basic circuits to the programming languages used to write scientific applications.

Electronic switches in this era were based on discrete diode and transistor technology with a switching time of approximately 0.3 microseconds. The first machines to be built with this technology include TRADIC at Bell Laboratories in 1954 and TX-0 at MIT's Lincoln Laboratory. Memory technology was based on magnetic cores which could be accessed in random order, as opposed to mercury delay lines, in which data was stored as an acoustic wave that passed sequentially through the medium and could be accessed only when the data moved by the I/O interface.

Important innovations in computer architecture included index registers for controlling loops and floating point units for calculations based on real numbers. Prior to this accessing successive elements in an array was quite tedious and often involved writing self-modifying codes

(programs which modified themselves as they ran; at the time viewed as a powerful application of the principle that programs and data were fundamentally the same, this practice is now frowned upon as extremely hard to debug and is impossible in most high level languages). Floating point operations were performed by libraries of software routines in early computers, but were done in hardware in second generation machines.

During this second generation many high level programming languages were introduced, including FORTRAN (1956), ALGOL (1958), and COBOL (1959). Important commercial machines of this era include the IBM 704 and 7094. The latter introduced I/O processors for better throughput between I/O devices and main memory.

The second generation also saw the first two supercomputers designed specifically for numeric processing in scientific applications. The term "supercomputer" is generally reserved for a machine that is an order of magnitude more powerful than other machines of its era. Two machines of the 1950s deserve this title. The Livermore Atomic Research Computer (LARC) and the IBM 7030 (aka Stretch) were early examples of machines that overlapped memory operations with processor operations and had primitive forms of parallel processing.

### **3.4 Third Generation (1963 – 1972)**

The third generation brought huge gains in computational power. Innovations in this era include the use of integrated circuits, or ICs (semiconductor devices with several transistors built into one physical component), semiconductor memories starting to be used instead of magnetic cores, microprogramming as a technique for efficiently designing complex processors, the coming of age of pipelining and other forms of parallel processing, and the introduction of operating systems and time-sharing.

The first ICs were based on small-scale integration (SSI) circuits, which had around 10 devices per circuit (or “chip”), and evolved to the use of medium-scale integrated (MSI) circuits, which had up to 100 devices per chip. Multilayered printed circuits were developed and core memory was replaced by faster, solid state memories. Computer designers began to take advantage of parallelism by using multiple functional units, overlapping CPU and I/O operations, and pipelining (internal parallelism) in both the instruction stream and the data stream. In 1964, Seymour Cray developed the CDC 6600, which was the first architecture to use functional parallelism. By using 10 separate functional units that could operate simultaneously and 32 independent memory banks, the CDC 6600 was able to attain a computation rate of 1 million floating point operations per second (1 Mflops). Five years later CDC released the 7600, also developed by Seymour Cray. The CDC 7600, with its pipelined functional units, is considered to be the first vector processor and was capable of executing at 10 Mflops. The IBM 360/91, released during the same period, was roughly twice as fast as the CDC 6600. It employed instruction look ahead, separate floating point and integer functional units and pipelined instruction stream. The IBM 360-195 was comparable to the CDC 7600, deriving much of its performance from a very fast cache memory. The SOLOMON computer, developed by Westinghouse Corporation, and the ILLIAC IV, jointly developed by Burroughs, the Department of Defence and the University of Illinois, was representative of the first parallel computers. The Texas Instrument Advanced Scientific Computer (TI-ASC) and the STAR-100 of CDC were pipelined vector processors that demonstrated the viability of that design and set the standards for subsequent vector processors.

Early in this third generation, Cambridge and the University of London cooperated in the development of CPL (Combined Programming Language, 1963). CPL was, according to its authors, an attempt to capture only the important features of the complicated and sophisticated ALGOL. However, the ALGOL CPL was large with many features that were hard to learn. In an attempt at further simplification, Martin Richards of Cambridge developed a subset of CPL called BCPL (Basic Computer Programming Language, 1967).

### **3.5 Fourth Generation (1972 – 1984)**

The next generation of computer systems saw the use of large scale integration (LSI –1000 devices per chip) and very large scale integration (VLSI –100,000 devices per chip)

in the construction of computing elements. At this scale entire processors will fit onto a single chip, and for simple systems the entire computer (processor, main memory, and I/O controllers) can fit on one chip. Gate delays dropped to about 1 ns per gate.

Semiconductor memories replaced core memories as the main memory in most systems; until this time the use of semiconductor memory in most systems was limited to registers and cache. During this period, high speed vector processors, such as the CRAY 1, CRAY X-MP and CYBER 205 dominated the high performance computing scene. Computers with large main memory, such as the CRAY 2, began to emerge. A variety of parallel architectures began to appear; however, during this period the parallel computing efforts were of a mostly experimental nature and most computational science was carried out on vector processors. Microcomputers and workstations were introduced and saw wide use as alternatives to time-shared mainframe computers.

Developments in software include very high level languages such as FP (functional programming) and Prolog (programming in logic). These languages tend to use a *declarative* programming style as opposed to the *imperative* style of Pascal, C, FORTRAN, et al. In a declarative style, a programmer gives a mathematical specification of what should be computed, leaving many details of how it should be computed to the compiler and/or runtime system. These languages are not yet in wide use, but are very promising as notations for programs that will run on massively parallel computers (systems with over 1,000 processors). Compilers for established languages started to use sophisticated optimisation techniques to improve codes, and compilers for vector processors were able to vectorise simple loops (turn loops into single instructions that would initiate an operation over an entire vector).

Two important events marked the early part of the third generation: the development of the C programming language and the UNIX operating system, both at Bell Labs. In 1972, Dennis Ritchie, seeking to meet the design goals of CPL and generalise Thompson's B, developed the C language. Thompson and Ritchie then used C to write a version of UNIX for the DEC PDP-11. This C-based UNIX was soon ported to many different computers, relieving users from having to learn a new operating system each time they change computer hardware. UNIX or a derivative of UNIX is now a de facto standard on virtually every computer system.

An important event in the development of computational science was the publication of the Lax report. In 1982, the US Department of Defence (DOD) and National Science Foundation (NSF) sponsored a panel on Large Scale Computing in Science and Engineering, chaired by Peter D. Lax. The Lax Report stated that aggressive and focused foreign initiatives in high performance computing, especially in Japan, were in sharp contrast to the absence of coordinated national attention in the United States. The report noted that university researchers had inadequate access to high performance computers. One of the first and most visible of the responses to the Lax report was the establishment of the NSF supercomputing centres. Phase I on this NSF program was designed to encourage the use of high performance computing at American universities

by making cycles and training on three (and later six) existing supercomputers immediately available. Following this Phase I stage, in 1984 – 1985 the NSF provided funding for the establishment of five Phase II supercomputing centres.

The Phase II centres, located in San Diego (San Diego Supercomputing Centre); Illinois (National Centre for Supercomputing Applications); Pittsburgh (Pittsburgh Supercomputing Center); Cornell (Cornell Theory Centre); and Princeton (John Von Neumann Centre), have been extremely successful at providing computing time on supercomputers to the academic community. In addition they have provided many valuable training programmes and have developed several software packages that are available free of charge. These Phase II centres continue to augment the substantial high performance computing efforts at the National Laboratories, especially the Department of Energy (DOE) and NASA sites.

### **3.6 Fifth Generation (1984 – 1990)**

The development of the next generation of computer systems is characterised mainly by the acceptance of parallel processing. Until this time, parallelism was limited to pipelining and vector processing, or at most to a few processors sharing jobs. The fifth generation saw the introduction of machines with hundreds of processors that could all be working on different parts of a single program. The scale of integration in semiconductors continued at an incredible pace, so that by 1990 it was possible to build chips with a million components – and semiconductor memories became standard on all computers.

Other new developments were the widespread use of computer networks and the increasing use of single-user workstations. Prior to 1985, large scale parallel processing was viewed as a research goal, but two systems introduced around this time are typical of the first commercial products to be based on parallel processing. The Sequent Balance 8000 connected up to 20 processors to a single shared memory module (but each processor had its own local cache). The machine was designed to compete with the DEC VAX-780 as a general purpose Unix system, with each processor working on a different user's job. However, Sequent provided a library of subroutines that would allow programmers to write programs that would use more than one processor, and the machine was widely used to explore parallel algorithms and programming techniques.

The Intel iPSC-1, nicknamed “the hypercube”, took a different approach. Instead of using one memory module, Intel connected each processor to its own memory and used a network interface to connect processors. This distributed memory architecture meant memory was no longer a bottleneck and large systems (using more processors) could be built. The largest iPSC-1 had 128 processors. Toward the end of this period, a third type of parallel processor was introduced to the market. In this style of machine, known as a *data-parallel* or SIMD, there are several thousand very simple processors. All processors work under the direction of a single control unit; i.e. if the control unit says

“add a to b” then all processors find their local copy of a and add it to their local copy of b. Machines in this class include the Connection Machine from Thinking Machines, Inc., and the MP-1 from MasPar, Inc.

Scientific computing in this period was still dominated by vector processing. Most manufacturers of vector processors introduced parallel models, but there were very few (two to eight) processors in these parallel machines. In the area of computer networking, both wide area network (WAN) and local area network (LAN) technology developed at a rapid pace, stimulating a transition from the traditional mainframe computing environment towards a distributed computing environment in which each user has their own workstation for relatively simple tasks (editing and compiling programs, reading mail) but sharing large, expensive resources such as file servers and supercomputers. RISC technology (a style of internal organisation of the CPU) and plummeting costs for RAM brought tremendous gains in computational power of relatively low cost workstations and servers. This period also saw a marked increase in both the quality and quantity of scientific visualisation.

### **3.7 Sixth Generation (1990 to date)**

Transitions between generations in computer technology are hard to define, especially as they are taking place. Some changes, such as the switch from vacuum tubes to transistors, are immediately apparent as fundamental changes, but others are clear only in retrospect. Many of the developments in computer systems since 1990 reflect gradual improvements over established systems, and thus it is hard to claim they represent a transition to a new “generation”, but other developments will prove to be significant changes.

In this section, we offer some assessments about recent developments and current trends that we think will have a significant impact on computational science.

This generation is beginning with many gains in parallel computing, both in the hardware area and in improved understanding of how to develop algorithms to exploit diverse, massively parallel architectures. Parallel systems now compete with vector processors in terms of total computing power and, most especially, parallel systems to dominate the future.

Combinations of parallel/vector architectures are well established, and one corporation (Fujitsu) has announced plans to build a system with over 200 of its high and vector processors. Manufacturers have set themselves the goal of achieving teraflops ( $10^{12}$  arithmetic operations per second) performance by the middle of the decade, and it is clear this will be obtained only by a system with a thousand processors or more. Workstation technology has continued to improve, with processor designs now using a combination of RISC, pipelining, and parallel processing. As a result it is now possible to procure a desktop workstation that has the same overall computing power (100 megaflops) as



fourth generation supercomputers. This development has sparked an interest in heterogeneous computing: a program started on one workstation can find idle workstations elsewhere in the local network to run parallel subtasks.

One of the most dramatic changes in the sixth generation is the explosive growth of wide area networking. Network bandwidth has expanded tremendously in the last few years and will continue to improve for the next several years. T1 transmission rates are now standard for regional networks, and the national “backbone” that interconnects regional networks uses T3. Networking technology is becoming more widespread than its original strong base in universities and government laboratories as it is rapidly finding application in K-12 education, community networks and private industry. A little over a decade after the warning voiced in the Lax report, the future of a strong computational science infrastructure is bright.

#### **4.0 CONCLUSION**

The development of the computer spans through many generations with each generation chronicling the landmark achievements of the period.

#### **5.0 SUMMARY**

This unit teaches that the development of the computer has spanned through six generations.

#### **6.0 TUTOR-MARKED ASSIGNMENT**

1. Outline the major landmarks of the fourth and fifth generations of computers.
2. Explain what is meant by stored program computer architecture.

#### **7.0 REFERENCES/FURTHER READING**

Akinyokun, O.C. (1999). *Principles and Practice of Computing Technology*. Ibadan: International Publishers Limited.

Balogun, V.F., Daramola, O.A., Obe, O.O., Ojokoh, B.A. and Oluwadare S.A., (2006). *Introduction to Computing: A Practical Approach*. Akure: Tom-Ray Publications.

Larry Long (1984). *Introduction to Computers and Information Processing*. New Jersey: Prentice-Hall Inc.

Tunji & Dokun (1993). *Data Processing, Principles and Concepts*. Lagos: Informatics Books.