BIOGRAPHICAL
MEMOIRS

David John Wheeler. 9 February 1927 –– 13 December 2004: Elected FRS 1981

Martin Campbell-Kelly

Biogr. Mems Fell. R. Soc. 2006 **52**, 437-453, published 1 December 2006

DAVID JOHN WHEELER

9 February 1927 — 13 December 2004

Biogr. Mems Fell. R. Soc. **52**, 437–453 (2006)

D. J. Wheeler

DAVID JOHN WHEELER

9 February 1927 — 13 December 2004

Elected FRS 1981

BY MARTIN CAMPBELL-KELLY

Department of Computer Science, University of Warwick, Coventry CV5 6FQ, UK

David Wheeler's scientific career was unusual. He was not the architect of a grand project, nor did he become a senior scientific administrator. Instead, his career had an unusually level trajectory. His individual style of working barely changed in half a century, from research student to professor emeritus. Apart from a few sabbatical leaves, his whole working life took place in the Mathematical (later Computer) Laboratory at Cambridge University.

Wheeler's *forte* was problem solving, usually for others. His first task, when he was a 21-year-old research student, was to investigate the problem of programming for the newly invented digital computer. For the last decade of his life he worked on algorithms, ranging from data encryption to the elimination of spam email. In between he worked on programming systems, hardware design, computer architecture, and numerical algorithms, to all of which he made important contributions.

Although Wheeler was a computer scientist all his working life, he was a mathematics wrangler and this defined his working style. However mundane or utilitarian the problem to be solved, his solution always bore the hallmark of his mathematical training. His best work had the elegance and economy of a theorem in mathematics, with not a redundant symbol nor an unnecessary line of code. His tangible life's work amounts to a collection of small works of genius, any one of which would have been enough to establish a normal reputation.

Wheeler loved the process of problem solving, creating computer artefacts and algorithms for others to use—he liked to think this was an inheritance from his forebears, who were engineering tool makers. He had a gentle, unassuming demeanour. He interacted with research students and colleagues on equal terms, and developed with them some extraordinarily fruitful collaborations. What Wheeler did not enjoy was writing academic papers, and his published *oeuvre* is surprisingly thin. His working style was to solve a problem and then move onto the next. He had no taste for administration, although on the few occasions it fell on him, his humanity served him well. He served on few national committees, and bureaucratic activities gave him no pleasure, other than social. For all these reasons, Wheeler's was not a meteoric

doi:10.1098/rsbm.2006.0030

career, and success came rather late—an *ad hominem* chair at the age of 50 years and election to the Fellowship of the Royal Society four years later.

Early life

David Wheeler was born in Birmingham on 9 February 1927, the second of three children of Arthur William Wheeler and Marjorie (*née* Gudgeon). Arthur Wheeler was an engineer and proprietor of a small shopfitting and repair firm. The family lived in Small Heath, Birmingham, where Wheeler attended the local primary school. In 1938 he won a scholarship to the King Edward VI Grammar School, Camphill, Birmingham (a non-fee-paying grammar school supported by an educational foundation). In 1940 the school was evacuated first to Warwick and then to Lichfield. These disruptions undermined Wheeler's general education—he never became a confident writer, and he did not get the opportunity to study Latin until he reached the sixth form, when it did not come easily.

In 1942, the family moved for the duration to Tunstall, Stoke-on-Trent, where Wheeler's father made tools for aircraft production. Wheeler completed his sixth-form education at Hanley High School without further interruption. He shone at mathematics and physics, and his parents entered him for Cambridge University at the urging of his physics master, Roy Major (later principal of Alsager Teacher Training College). In 1945 he won a minor entrance scholarship and a state scholarship to Trinity College, where he studied mathematics and graduated BA in 1948.

The Mathematical Laboratory and the EDSAC

In November 1946, while still an undergraduate, Wheeler attended the inaugural lecture of Professor Douglas Rayner Hartree FRS, the newly appointed Plumian Professor of Mathematical Physics (Hartree 1947). Hartree had recently returned from the USA, where he had used the ENIAC, the first electronic computer designed for mathematical work, which was built at the Moore School of Electrical Engineering, University of Pennsylvania. Hartree was convinced that electronic computers would revolutionize science and mathematical computation. The lecture was a turning point for Wheeler, from which he never looked back.

Shortly after Hartree's lecture, Wheeler found his way to the University Mathematical Laboratory. There an electronic computer known as the EDSAC (Electronic Delay Storage Automatic Calculator) was under construction by a team led by Maurice V. Wilkes (FRS 1956). The enthusiastic Wheeler was pressed into service as a wireman. The EDSAC was one of the first computer projects to be based on a new type of machine, known as a 'stored-program computer', designed by John von Neumann and others at the Moore School in 1945. The stored-program computer was the blueprint on which almost all subsequent computers have been based.

The Mathematical Laboratory had been established under the direction of Professor Sir John Lennard-Jones FRS in 1937, with Wilkes appointed a University Demonstrator in charge of day-to-day running (Wilkes 1985). The Laboratory's activities were put on hold during the war. In 1945 Wilkes returned from war service and was formally appointed director of the Laboratory the next year. Following a visit to the Moore School in summer 1946, Wilkes initiated the EDSAC project. When the EDSAC sprang into life in spring 1949, it was the first practical machine of the new type to offer a regular computing service and it established the Laboratory as a world centre for computing research.

In October 1948, Wheeler became one of the first two research students in the Laboratory, with Wilkes as his supervisor. Wilkes was then 35 years old, half a generation older than Wheeler and far more worldly, with several years of war service and university politics behind him. Moreover, their temperaments were entirely different: Wilkes was a scientific administrator first, and a researcher second; Wheeler was a researcher, first and second. For more than 30 years their lives would be ineluctably entwined in the Laboratory. They developed a relationship based on deep mutual respect that fell short of intimacy.

When Wheeler began his studentship in October 1948, the EDSAC was still under construction, although its specification was largely determined. At that time, no one had given any thought to the practical problems of getting a program inside a computer. Computers only understood binary codes, whereas humans need to write programs at a higher, symbolic level. Wilkes's insight was to realize that the translation between the external symbolic form and the internal binary form could be done by the computer itself. This was the task that he assigned to Wheeler as his research topic.

EDSAC users would be able to write program instructions in a simple form using code letters and decimal numbers—for example an instruction such as 'add the number in long storage location 123' would be written as 'A 123 L'. Wheeler's first translation program for converting symbolic code into binary consisted of just 32 lines of code. These 'Initial Orders' were permanently wired into the computer's memory, so that programs could be punched on teleprinter tape and inserted directly into the machine (1)*. The Initial Orders were in use when the machine performed its first fully automatic computations on 6 May 1949.

Throughout its history, software has been an extraordinarily difficult technology to tame—and we are still a long way from taming it. As soon as the EDSAC programming seemed to be under control with Wheeler's Initial Orders, another difficulty surfaced. It was proving extremely difficult to write even simple programs correctly, and even more difficult to debug them. Suddenly, Wheeler's research topic became a real challenge.

One way to reduce the number of errors in a program was to use a library of 'subroutines'—small, pre-written programs that could be incorporated in a user's program (2). Wheeler invented a technique known as the 'closed subroutine' that made the process of incorporating subroutines in user programs very flexible and powerful. A library of several dozen subroutines was created, each subroutine consisting of a paper tape kept in a set of steel filing drawers; users would mechanically copy subroutines into their programs. This was a very characteristic activity in the Laboratory for many years. Wheeler wrote much of the subroutine library, and every subroutine he wrote was a masterpiece in miniature. Wheeler incorporated closed subroutines, and much more, into a new set of Initial Orders that were wired into the machine in September 1950. It was a *tour de force* of programming—just 42 instructions—that established his reputation and led (with a little coaxing from Wilkes) to one of his few world-class publications (3). Libraries of closed subroutines are still the basis of virtually all programming systems up to the present day. In 2003 Wheeler was inducted into the hall of fame of the Computer History Museum, Mountain View, California, not least for this fundamental contribution to computing.

* Numbers in this form refer to the bibliography at the end of the text.

In 1951 Wilkes, Wheeler and another Laboratory research student, Stanley Gill, became the authors of the classic *The preparation of programs for an electronic digital computer* (4). Usually known as 'Wilkes, Wheeler & Gill', or simply 'WWG', it was the first textbook on programming and it rushed in to fill the vacuum of programming knowledge. The ideas it contained served as the basis for 'assembly systems' in the first generation of computers worldwide, until programming languages such as Fortran and Algol took over in the early 1960s. Indeed, a curious outcome of Wheeler's masterly programming system was that the Laboratory felt little need to develop high-level programming languages—it was not until 1960 that David Hartley, then a research student, wrote an 'autocode' for the EDSAC's successor, EDSAC 2.

As well as creating the programming system for the EDSAC, Wheeler also wrote programs on behalf of some users and helped others. For example, he assisted the formidable R. A. (later Sir Ronald) Fisher FRS in computing a statistical table published in *Biometrics* in December 1950 (Fisher 1950)—this was valuable, not least because Fisher held some sway in university affairs that impinged on the funding of the Laboratory. Around the same time, Wheeler worked with the mathematician J. C. P. Miller, a university lecturer on the Laboratory staff, to compute what was briefly the highest known prime number (5).

After submitting his thesis in summer 1951, Wheeler was elected to a research fellowship at Trinity College. Before taking up this appointment he served, with the benefit of a Fulbright Travel Grant, as a visiting assistant professor at the University of Illinois. There he did more of the same, producing a programming environment for the ILLIAC computer similar to that at Cambridge, and also working with users (6, 11). It was journeyman programming work of the very highest quality.

EDSAC 2

For the first 20 years of university computing, up to the mid-1960s, it was possible for a leading department to build its own computer from the ground up. Such a project provided a research focus for the whole department—in hardware, software and applications—as well as supplying computing power to the university. In the UK, this was the strategy adopted by the two leading departments, at Cambridge and Manchester.

Surprisingly, building a computer was not necessarily more expensive than buying one, if one ignored staff costs. EDSAC and its successor both probably cost less than £50 000. Building one's own computer meant that it was possible to catch the wave of breaking technologies in the initial design. Also, it was possible to upgrade a self-built machine as new technologies emerged. When Wheeler returned from the University of Illinois in 1953, his first job was to incorporate an index register in the EDSAC. This new invention simplified programming, although retrofitting it to the EDSAC called for some engineering subtlety to maintain backward compatibility, so that all the existing software still worked.

Wilkes had anticipated the need for a successor to the EDSAC as soon as it was running a regular service in 1950. The EDSAC 2 project was led by William Renwick on the engineering side and Wheeler on programming—although Wheeler's role necessarily crossed the threshold into many hardware aspects.

The great contribution that EDSAC 2 was to make to computer science, and to the computer industry, was the technique of microprogramming. Wilkes first described this invention in a paper at the Manchester University computer conference in July 1951 (Wilkes 1951). Microprogramming was Wilkes's most important contribution to computer science, although—as with the EDSAC programming system—it was Wheeler who took the basic idea and brought it to life.

Microprogram control brought a great simplification to computer design. In its simplest terms, the idea was to build a computer with a very simple, primitive instruction set (which therefore resulted in a fast and relatively cheap design). Then, real instructions, such as 'multiply', would be implemented by a small microprogram—in the case of multiply this would consist of a sequence of shift and add instructions.

The project began with the construction of a small prototype computer with a 64-word microprogram store to test the feasibility of microprogramming. Known as EDSAC 1.5, the machine in fact did useful work (12). Among its users was Joyce Blackler, then a research student studying stellar equilibrium. She and Wheeler were married in 1957.

By now promoted an assistant director of research, Wheeler had the somewhat awesome task of designing the instruction set of EDSAC 2, implementing the microprogram and writing the entire programming system. He was, of course, supported by other members of the Laboratory, particularly in the commissioning of the read only memory by Peter Swinnerton-Dyer (then a senior assistant-in-research in the Laboratory, who subsequently became a professor of pure mathematics, and later a vice-chancellor of the university as well as being elected FRS in 1967 and awarded a knighthood in 1987). Leading such a project was an opportunity given to few. In Wheeler's own words, it was 'a chance to make the order code and assembly code and other machine features a coherent whole.' In November 1957, Wilkes, Renwick and Wheeler jointly read a paper describing the microprogram unit to a meeting of the Institution of Electrical Engineers (IEE) Measurement and Control Section. The published paper (7) won the 1957 IEE Mather Premium.

Wheeler loved designing and writing the EDSAC 2 programming system; it was perhaps the single body of work of which he was most proud. It was full of inventive ideas, such as 'single pass' assembly and hashing techniques that are now part of the bedrock of computing technique. But Wheeler could rarely be induced to publish his work. His view was that one would inevitably stumble across or invent such ideas—he had no interest in claiming credit for them and no interest in publishing them. Fortunately (and of course Wheeler was well aware of this), his ideas were picked up by his graduate students and colleagues, spread in the Cambridge milieu, and thence to the rest of the world. Wheeler's code bristled with inventiveness and his subroutines for the EDSAC were widely studied—for example, Donald Knuth noted in *The art of computer programming* (Knuth 1969, p. 456) that subroutine V-1 was perhaps the first practical use of a linear list.

The order code for the EDSAC 2 did not use mnemonics (such as 'A' for Add), but instead decimal codes (in fact '12' for add). Wheeler reasoned that if an order code was systematically designed—and his was, to be sure—it was easier to remember numeric codes than quixotic mnemonics. This view very probably informed the designers of other contemporary British machines such as the Pegasus, the Mercury and the Atlas, which all adopted a similar programming style. The entire programming system was contained in a 768-word 'fixed store'—not just the assembly program, but also common subroutines and debugging aids. This enabled EDSAC 2 to load programs in a few seconds, in comparison with the several minutes it could take on commercial machines. EDSAC 2 ran from early 1958 until November 1965; the original EDSAC was shut down in July 1958.

Wheeler made a second visit to the University of Illinois in 1959, as an associate professor, where he designed the arithmetic unit for the ILLIAC 2. Back at Cambridge, he worked on enhancements to the subroutine library and the EDSAC 2 itself. Among the former was a fast Fourier transform technique, a significant research contribution that, typically, went unpublished. The subroutines were used particularly by the university's radio astronomers and crystallographers. Wheeler had assisted the radio astronomers since the early 1950s, but the improved computing speeds of the late 1950s opened up new opportunities. The Cambridge astronomer Sir Martin Ryle FRS noted in his 1974 Nobel Lecture (Ryle 1974, p. 193):

> [T]he completion of the much faster EDSAC II, and the development by Dr. David Wheeler of the Mathematical Laboratory of the fast fourier transform (incidentally some six years before these methods came into general use) made possible the efficient reduction of the 7.9 m and 1.7 m surveys, and also enabled trials of the 1.7 m earth-rotation synthesis to be made in 1961.

Wheeler incorporated a magnetic tape storage unit into EDSAC 2—always part of the original specification, but one that was not undertaken until the machine had been operational for two or three years. Although program 'interrupts' had not yet been invented, the system allowed for autonomous tape positioning using a microcode trap. (Interrupts were added as a late feature to EDSAC 2, though without an operating system they were not much used.) In 1962 a massive 16 Kword core store was added to the machine, at which point it became one of the most capable computers in the country. In these machine improvements, Wheeler had to exercise real engineering judgement to minimize disruption and ensure the backward compatibility of people's existing programs. By the time the Laboratory moved on to its next project, the Titan in 1965, Wheeler was a consummate hardware and software engineer—capabilities rarely found in the same individual, and perhaps uniquely to such a high level.

The Titan computer

By about 1960, it was clear that the demand for computing in the university was growing so fast that another, more powerful computer would soon be needed. Initially it had been hoped that it would be possible to acquire a Ferranti Atlas computer, a machine based on a design by Manchester University, which was then the world's most powerful computer. However, the cost (said to be over £2 million) far exceeded the university's resources. Instead, Wilkes was able to make a deal with Ferranti, by which the company would supply the hardware for a cut-down Atlas at cost price (of the order of £100 000), to be known as the Titan. In exchange for the Laboratory's undertaking the necessary engineering and software, Ferranti would then be free to use this design for a low-cost production model (which was subsequently marketed as the Atlas II).

During 1961–62, Wheeler acted as 'design authority' for the new machine, which required him to mediate the local design decisions to ensure that all the components of the system worked together and to negotiate with Ferranti engineers. This was a heady responsibility, which called for all his technical and diplomatic skills. Wheeler did all the high-level design work, and much of the low-level hardware design too. He was assisted by Roger Needham (FRS 1985), his first research student, who was appointed to the Laboratory staff at the beginning of 1962. Needham wrote a program, which ran on EDSAC 2, to optimize and print the wiring schedules. This innovation had a considerable long-term influence on Ferranti's design automation processes. On the software side, the operating system team was led by David Barron. Other participants included David Hartley, A. S. (Sandy) Fraser, Barry Landy, and Peter Swinnerton-Dyer. In autumn 1967, Barron was appointed to the founding chair of computer science at Southampton University, and his role was largely assumed by Needham.

While the design of Titan was underway, a new wind had started to blow across the computer landscape—multi-access time-sharing. Wilkes first caught the scent of this new mode of computer use on a visit to Massachusetts Institute of Technology (MIT) in summer 1963, where one of the most important experimental systems, Project MAC, had just begun. In a time-sharing computer, instead of users submitting their program on cards or tape, and returning for the results some hours later, a user sat at a typewriter terminal, entered a program directly into the machine, edited it, executed it, and obtained instant results. This mode of computing was only economically feasible by having many users simultaneously using the computer, so that the machine's facilities were fully exploited by sharing them across heterogeneous demands. Multi-access time-sharing would be the preferred mode of university computing until the advent of personal computers and work stations in the early 1980s.

At the start of the new academic year in September 1963, Wilkes introduced the time-sharing idea to the Laboratory and proposed that the Titan project be realigned. Wheeler and the rest of the Laboratory assented. This change of direction entailed major changes to the hardware and software architectures. Most importantly, it was necessary to design a memory protection system so that it was possible for several user programs to co-exist safely in the memory of the computer. Somewhat out of character—for he was temperamentally inclined to work things out from first principles—Wheeler undertook a major study of contemporary addressing systems, and then designed an elegant and cost-effective scheme. At a lower level in the architecture, it was necessary to equip the computer with a disc store (costing £75 000 for 40 million characters!). It was also necessary to provide a 'multiplexer' to connect some dozens of terminals to the computer. Wilkes had sketched out a preliminary design for a multiplexer, and Wheeler undertook the detailed hardware design. This was an engineering challenge in its own right. A few years later, a multiplexer would have been based on a small dedicated, general purpose computer but in 1963–64 it had to be specially built from 'random logic'—although this was the kind of work in which Wheeler delighted.

In spring 1965, Wheeler visited Los Angeles to attend acceptance tests for the disc store, which was being manufactured by the Data Products Corporation. While in the USA, he visited several of the time-sharing projects that were under development: the JOSS system at the RAND Corporation ('an exquisite example of a limited time-sharing system'), the SDS time-sharing system at Berkeley, MIT's Project MAC, and several others.

Wheeler was promoted to a readership in October 1966. He spent that academic year on a well-earned sabbatical leave at the University of California, Berkeley. Unable to resist the opportunity of doing a job for the second time, only this time properly, he designed a multiplexer for the Berkeley time-sharing system. This enabled hundreds of teletypes to be connected to a CDC 6500 computer. While he was away, in early 1967 the Titan system became available for real users, as opposed to the 'quasi-users who wrote it' (Needham 1969). A year after his return to Cambridge, Wheeler supervised the Laboratory's move to a new building after the death in early 1969 of Eric Mutch, Superintendent of the Computing Service since 1948. Mutch's death came a great shock to everyone, and the readiness and competence with which Wheeler took over was to his lasting credit.

THE CAP COMPUTER

By 1970, the days when the Laboratory could design and build a computer to supply computing power to the whole university were over. The scale of computer development and use, and the diversity of software, had exploded in the 1960s so that only full-scale manufacturers had the resources to develop entire computer systems. In 1970, the Mathematical Laboratory was renamed and reorganized, dividing into two parts, with Wilkes as overall head of department. The Computer Laboratory would continue the roles of teaching and research, while the operational role would become the University Computing Service, with David Hartley as its first director.

Although the Laboratory could no longer resource the development of a major computer system, it fulfilled its research mission by exploring emerging technologies. The Laboratory's two most important projects at that time were the CAP computer and the Cambridge Ring.

The CAP computer project (not an acronym) ran from 1970 to 1979. In the 1960s, as the technology of multi-access time-sharing computers had matured, the idea of capability-based computers came into prominence. Multi-access computers posed technical problems that did not arise in computers that executed one program at a time or that did not need to provide non-stop operation. For example, in a multi-access computer it was necessary to prevent one program from encroaching on another when they shared the same memory space, or obtaining information to which it had no access rights. The capability concept, originated by Dennis and van Horn at MIT in 1966, addressed these problems (Levy 1984). It was a technique that used 'tickets' or 'tokens' to enable programs to access information owned by other entities under carefully controlled circumstances. In the late 1960s a number of academic and industrial research laboratories had begun to develop software-based capability systems, although these turned out to be impractically slow. Cambridge decided it would therefore develop a hardware-based system, which promised a fast, practical implementation.

The CAP computer project was jointly managed by Wheeler and Needham—Wheeler directing the hardware side and Needham the software. During 1970–74, Wheeler designed and commissioned a complete minicomputer system based around a microprogram unit. In 1976 he described his role thus:

> I designed the CAP, a computer designed to test the usefulness of various forms of capability. This is micro programmed, but has many auxiliary features, so that computers similar to the ones envisaged are fast and take a few micro orders to design. As the micro program is in random access store, the computer can be readily changed.

Thus, instead of a single fixed system, Wheeler's design enabled the machine to be rapidly reconfigured to validate different capability schemes. The CAP computer came close to being the first hardware-implemented, capability-based computer. (It was narrowly beaten by the Plessey 250, a communications processor that needed carrier-grade reliability and whose design was directly influenced by Cambridge.) The design and commissioning of a minicomputer on the scale of the CAP, with the resources available to a university department, was a remarkable achievement by Wheeler.

The CAP went into regular service in the Laboratory with an operating system based on a Titan-like machine architecture micro-programmed by Robin Walker; in parallel, others, including Andrew Herbert, explored radically different architectures, taking advantage of the great flexibility of Wheeler's hardware (Wilkes & Needham 1979). Active research on the machine ceased in 1980, after which it was used as a server on the Cambridge Model

Distributed System for several years. The commercial impact of the CAP, however, was small. It was one of several capability research initiatives that never lived up to their original promise in terms of influence, for complex economic and technical reasons. Capability concepts informed subsequent developments, but not with the central position expected at the outset.

The Cambridge Ring

In the mid-1970s, local area networks (LANs) were in the air. A LAN enabled minicomputers, terminals and peripheral devices to be interconnected by digital transmission links at speeds that were far higher than using ordinary telephony. There were at least 20 distinct LAN research initiatives at different universities and research laboratories around the world.

During 1975–77, the Laboratory developed a novel LAN technology that became known as the Cambridge Ring (8). Wilkes had conceived the idea for a ring-type network while visiting the Hasler Company in Berne in January 1974. The Hasler scheme was for a conventional telecommunications application, but Wilkes saw it could be a adapted to a computer network. In a re-run of earlier collaborations, Wheeler took this raw idea and brought it to engineering reality. In the Cambridge design, all the devices on the network were connected into a ring. Data were shuttled round the ring in a small number of 'packets' at 10 Mbit s^{-1}, rather like pass-the-parcel. A device could remove the contents of a packet, or introduce new contents, while it momentarily had possession of the packet.

Wheeler's particular role in the Cambridge Ring was to design the hardware and the 'hardware protocol'. In this he was assisted by his research student Andy Hopper (who became head of the Laboratory in 2004 and was elected FRS in 2006). In typical Wheeler fashion, the logic diagrams for the hardware were sketched in a notebook in his characteristically neat hand. The system design was a demanding challenge of the kind that brought out the best in Wheeler, as it required both a mastery of engineering realities and algorithmic thinking. The design had to guarantee the perfect transmission of a packet of data in a network that was susceptible to breakdown and malfunction. One notable innovation was a monitoring station that kept a log of corrupt packets. This information enabled maintenance staff to locate malfunctioning equipment, and in some cases anticipate operational problems. Wheeler and Hopper also explored operational issues and alternative topologies to the simple ring (9, 10).

The Cambridge Ring served as the underlying infrastructure for the Cambridge Model Distributed System (CMDS) (Wilkes & Needham 1980; Needham & Herbert 1982). This system consisted of a 'processor bank' of single-user microprocessor-based computers, designed as a replacement for a time-sharing service. To support the single-user machines there were file servers, print servers and ancillary services. The CMDS was conceived by Needham (who had succeeded Wilkes as head of the Laboratory in 1980). Needham also designed the high-level software protocols; other major contributors included Andrew Herbert and Martin Richards. A fast version of the ring came on stream in 1982 operating at 100 Mbit s^{-1}, and a 'metropolitan network' developed in collaboration with Olivetti Research, operating at 1 Gbit s^{-1}, was available by 1988.

The Cambridge Ring had great commercial potential. It was the subject of the ISO CR82 standard and was manufactured under licence. In the end, however, the 'battle' of the LAN was won by Ethernet, developed at Xerox PARC in Palo Alto (with which the Laboratory enjoyed good relations). The economics of mass production forced a single standard onto the world, and Ethernet is now ubiquitous in offices and homes. Despite this rather disappointing commercial outcome, the Cambridge Ring was one of Wheeler's outstanding technical achievements. Further, the project created a legacy in spin-off companies and technology diffusion that persists in firms such as ARM (formerly Advanced RISC Machines) to the present day.

The Burrows–Wheeler Transform

By the early 1980s, the days of big projects that united the Computing Laboratory's research had ended. After Wilkes's retirement in 1980, subsequent heads of the Laboratory oversaw a more diverse range of activities.

In the later years of his career, Wheeler increasingly turned to algorithms and their implementation. His interest in programming and algorithms had been reawakened during a period spent at Bell Labs in 1978. His algorithmic interests were eclectic and highly original. Invariably they demonstrated the same kind of economy and ingenuity first seen in the EDSAC's Initial Orders in 1949. Two of his algorithmic inventions have earned a permanent place in the canon of computer science: the Burrows–Wheeler Transform for data compression, and the Tiny Encryption Algorithm.

Data compression has always been an important topic in computing and data transmission, but it became much more important with the rise of the Internet and digital media. Because the information people use is highly redundant, data compression techniques can fairly easily reduce the size of files by a factor of 3 or 4, providing a corresponding economy in data transmission and digital storage costs. While data transmission and data storage remain finite resources, data compression will continue to be an important technology.

The algorithm (later known as the Burrows–Wheeler Transform, or BWT) came to Wheeler while he was investigating data compression methods on a visit to Bell Labs in 1978. Rather like his Initial Orders, it is easier to see what the transform does than to see how it does it. As Wheeler noted:

> One of the troubles of bwt is explaining it well. My descriptive powers were inadequate for the task but Roger Needham gave one which seemed acceptable. (I thought it obscured the simple origin!) I think Mike [Burrows] used it as a kind of intelligence test on some of his colleagues.

There are now numerous descriptions of BWT in the pedagogic literature, and they are mostly clear up to the point at which the scaffolding falls away, and then the non-expert is left with more admiration than elucidation.

Wheeler did not do very much with the algorithm for several years, and certainly did not publish it. In 1984, Terry Welch of the Unisys Corporation had published the LZW algorithm, based on earlier work by Lempel and Ziv. LZW became the most popular data compression technique for use with the Internet and in digital imaging, because it gave excellent compression and was very fast. The latter attribute was important for its practical use on relatively low-powered personal computers and digital devices.

Having languished for a decade, Wheeler's algorithm was picked up again by his research student Mike Burrows. The way in which Wheeler had implemented the transform was not particularly fast, and Burrows worked on improving its performance. Burrows recalls with wry amusement that a submission to the Data Compression Conference was rejected. After completing his PhD, Burrows took up a position at the Digital Equipment Corporation's Systems Research Center in Palo Alto. He and Wheeler subsequently spent 'an interactive year improving and testing' the algorithm, which Burrows then published as a research report under their joint names (13). After that the BWT began to get noticed. One reason for its timeliness was that the LZW, which had become very popular, was encumbered by a patent that Unisys had started to enforce (eventually selling some 3000 licences). The BWT algorithm was picked up by Mark Nelson, the author of the *Data compression handbook*. Nelson wrote an article promoting BWT as a public-domain data compression algorithm in *Dr. Jobbs Journal*, a monthly magazine for software practitioners, and also posted an implementation on the Internet. After that, BWT spread like wildfire. Independently, a young British computer scientist Julian Seward created an open-source program bzip (now bzip2), which was incorporated with the Linux operating system in 1996. Wheeler advised Seward on a number of occasions in his kindly, if sometimes incomprehensible, way.

BWT continues to gather momentum. In December 2004, the National Science Foundation's Center for Discrete Mathematics and Theoretical Computer Science at Rutgers University celebrated BWT's 10th anniversary with an international conference of more than 30 speakers, including Burrows. Wheeler was unable to attend although, unusually, he put pen to paper to write a short manuscript history of BWT's development. So far well over 100 academic papers have been written to refine and elaborate BWT for different contexts.

TEA and other algorithms

Around 1990, a computer security group had formed in the Laboratory. Wheeler was an enthusiastic member and always attended the Easter vacation workshops. One of his important contributions was a 'bulk data encryption algorithm', WAKE (14). The algorithm made it possible to encrypt a whole message in a single process. This was suited to the way that protocol designers use and reason about encryption, and Wheeler's work opened up a new research direction.

Designing algorithms is an uncertain business—perhaps a little like writing poems—in that one cannot predict which ones will take the public's fancy. Much to the delight of Wheeler and his collaborator Needham, the public took particularly to another encryption technique, their Tiny Encryption Algorithm. When people asked to use it, Needham would reply: 'There are no restrictions whatsoever on anyone using TEA or any associated algorithms for any purpose at all, commercial or not. Wheeler and I simply put it in the public domain. If it does get used we quite like to be told what for—that's all!'

In 1992, when Wheeler and Needham began their collaboration, the most popular commercial encryption products were based on the US Data Encryption Standard (DES), developed by IBM in the early 1970s. However, DES had a number of disadvantages. First, its key is rather short, which means that a DES-encrypted message may be broken by an opponent with access to a lot of computing power. Second, it was complicated to implement, especially if a fast implementation was required. Third, most countries had export controls on cryptographic algorithms, and the USA applied these to software as well as hardware. Wheeler, and Needham especially, believed that encryption was a basic human right. Their idea was to develop a very good encryption algorithm that was so small and simple that just about anyone who could program a computer would be able to use it. And once it was in the public domain no one would be able to suppress it. The algorithm that Wheeler eventually wrote consisted of just eight lines of code (15). The program was so short that it could be written on the back of an envelope if necessary, and then nothing could prevent its diffusion.

The TEA algorithm attracted a lot of attention. It was fast, very easy to implement on any computer, and gave excellent security. It is now part of the world of free software and continues to be used in commercial products.

Emeritus

TEA was about the last algorithm Wheeler wrote before he reached the university's statutory retiring age of 67. In October 1994 he became professor emeritus. Freed of the responsibility of supervising research students and teaching commitments, his life otherwise continued much the same. He came into the Laboratory most days, usually cycling from the home he and Joyce had built to their own design and shared since 1965—and where they had raised their three children, all long flown the nest.

Wheeler spent his days in the Laboratory, despite failing eyesight, working on algorithms, attending and giving seminars, and talking to people about technical issues or gently socializing. Much of his work remained academic and utilitarian, but there was also a playful aspect. For example, he devised a personal spam-email filter that applied his data compression algorithm to incoming email; it turned out that spam email compressed differently from normal messages. Another project explored the use of probability theory to eliminate small financial payments. He explained the idea with a nice example of a hypothetical soft-drink dispensing machine. This machine accepted one-pound coins only, but if the real cost of the soft-drink was 60 pence, then 40% of the time the consumer's money would be refunded. On average, no one would be any the worse off, and the coin-receiving mechanism would be simpler and cheaper. He then elaborated this simple idea into a serious proposal for an Internet payment mechanism, which attracted considerable attention (16).

Assessment

On 13 December 2004 Wheeler cycled to work on a cold day and succumbed to a heart attack from which he never recovered. Many years earlier, in 1976, when he was halfway through his working life, he wrote: 'Thus it appears I have spent 30 years in helping people make better use of computers, by improving programming systems, hardware design, and occasionally numerical methods.' With less emphasis on hardware and more on algorithms, this would serve as a fair description of the second half of Wheeler's working life too.

Wheeler's life's work amounts to a collection of small works of genius. As a PhD student in 1949 he created a tiny programming system that went on to shape the computing environment worldwide in the 1950s; and to a degree it still does. His work on machine design in the 1960s and 1970s was more ephemeral, largely because the technology moved on. That, sadly, is the fate of much research, although this is not something that distressed Wheeler. His later work on algorithms will be more long lasting. To be the co-author of two widely used algorithms—BWT and TEA—is exceptional, and they are likely to remain useful for many years.

Wheeler brought a unique working style to the Laboratory, which has been described as a combination of the fundamental and the compact. Roger Needham recalled that, as a research student, he always valued talking to Wheeler. Wheeler might not know anything about the particular problem, but by the time Needham had finished discussing it with him he had learned a great deal. The quality of the Laboratory's research students gave Wheeler endless satisfaction.

One of the paradoxes of Wheeler's work is that he lacked a definite research agenda or a long-term vision. He worked across the spectrum of computer science—from hardware through software to algorithms, refusing to specialize. He saw himself as a tool builder: he delighted in solving a problem and then moving on to the next. If this was a fault, he was in good company. It was exactly this tendency to flit from topic to topic that worried the philosopher Sir Isaiah Berlin, who fretted that his work lacked coherence, distinction, or originality; he once remarked 'I am an intellectual taxi; people flag me down and give me destinations and I go off' (Crowder 2004, p. 9). For Wheeler, who enjoyed the process of research as much as the results, this would be an equally apt characterization.

Biographical summary

Academic degrees and appointments

1948	BA, Cambridge University
1951	PhD, Cambridge University
1951–57	Research Fellow, Trinity College, Cambridge University
1951–53	Visiting Assistant Professor, University of Illinois
1956–66	Assistant Director of Research, Mathematical Laboratory, Cambridge University
1959	Visiting Associate Professor, University of Illinois (April to September)
1965–94	Fellow, Darwin College, Cambridge University, thereafter emeritus
1966–77	Reader in Computer Science, Cambridge University
1966–67	Visiting Professor, University of California, Berkeley
1977–94	Professor of Computer Science, Cambridge University, thereafter emeritus
1981	Visiting Professor, University of Sydney (August to December)

Honours and industrial appointments

1957	(With M. V. Wilkes and W. Renwick) IEE Mather Premium
1978, 1983	Research Consultant, Bell Labs, Murray Hill, NJ
1985	IEEE Pioneer Award
1990	Research Consultant, Digital Equipment Corp., Palo Alto, CA
2003	Hall of Fellows, Computer History Museum, Mountain View, CA

References to other authors

Crowder, G. 2004 *Isaiah Berlin: liberty, pluralism and liberalism.* London: Polity Press.

Fisher, R. A. 1950 Gene frequencies in a cline determined by selection and diffusion. *Biometrics* **6**, 353–361.

Hartree, D. R. 1947 *Calculating machines, recent and prospective developments and their impact on mathematical physics. Inaugural Lecture.* Cambridge University Press.

Knuth, D. E. 1969 *The art of computer programming*, vol. 1. Reading, MA: Addison-Wesley.

Levy, H. M. 1984 *Capability-based computer systems*. Bedford, MA: Digital Press.

Needham, R. M. 1969 Consoles in the cloisters. *Datamation* (January), pp. 59–60.

Needham, R. M. & Herbert, A. J. 1982 *The Cambridge distributed computing system*. London: Addison-Wesley.

Ryle, M. 1974 Radio telescopes of large resolving power. Nobel lecture, 14 December 1974. http://nobelprize.org/physics/laureates/1974/ryle-lecture.pdf.

Wilkes, M. V. 1951 The best way to design an automatic calculating machine. In *Proceedings of Manchester University Computer Inaugural Conference, July 1951*, pp. 16–18. Manchester: Computer Laboratory, Manchester University.

Wilkes, M. V. 1985 *Memoirs of a computer pioneer*. Cambridge, MA: MIT Press.

Wilkes, M. V. & Needham, R. M. 1979 *The Cambridge CAP computer and its operating system*. New York: North Holland.

Wilkes, M. V. & Needham, R. M. 1980 The Cambridge Model Distributed System. *ACM Operating Syst. Rev.* **14**, 21–29.

Acknowledgements

I am most grateful to Joyce Wheeler, who gave me access to David Wheeler's personal papers and supplied me with many insights, both personal and academic. Karen Spärck-Jones eased my access to the Computer Laboratory's staff and library, and provided a holistic view of the Laboratory, which shows in the final result. Professor Sir Maurice Wilkes FRS, the senior member of the Laboratory, granted me a personal interview and a helpful written commentary. Several of Wheeler's colleagues and former colleagues read early drafts of the memoir and answered my emailed enquiries: Ross Anderson, David Barron, Mike Burrows, David Hartley, Andrew Herbert and Andy Hopper. Julian Seward provided information on open-source data compression. Anne Locker of IET Archives and Nicholas Cutler of the Computer Laboratory library provided me with research information.

The frontispiece photograph was taken in 1981 by Godfrey Argent, and is reproduced with permission.

Bibliography

The following publications are those referred to directly in the text. A full bibliography is available as electronic supplementary material at http://dx.doi.org/10.1098/rsbm.2006.0030 or via http://www.journals.royalsoc.ac.uk.

(1) 1949 Planning the use of a paper library. In *Report of a conference on high-speed automatic calculating machines, 22–25 June*, pp. 36–40. Cambridge: University of Cambridge Mathematical Laboratory.

(2) 1950 The use of sub-routines in programmes. In *Proc. ACM Nat. Conf., Pittsburg, May*, pp. 235–236. New York, NY: ACM Press.

(3) Programme organization and initial orders for the EDSAC. *Proc. R. Soc. A* **202**, 573–589.

(4) 1951 (With M. V. Wilkes & S. Gill) *The preparation of programs for an electronic digital computer*. Cambridge, MA: Addison-Wesley.

(5) (With J. C. P. Miller) Large prime numbers. *Nature* **168**, 1951.

(6) 1953 (With J. E. Robertson) Diagnostic programs for the Illiac. *Proc. IRE* **41**, 1320–1325.

(7) 1958 (With M. V. Wilkes & W. Renwick) The design of the control unit of an electronic digital computer. *Proc. IEE* **105B**, 121–135.

(8) 1979 (With M. V. Wilkes) The Cambridge digital communications ring. In *Local area communications network symposium, May, Boston* (ed. R. Rosenthal & N. B. Meisner). Washington, DC: National Bureau of Standards.

(9) (With A. Hopper) Maintenance of ring communication systems. *IEEE Trans. Commun.* **COM-27**, 760–761.

(10) (With A. Hopper) Binary routing networks. *IEEE Trans. Comput.* **C-28**, 699–703.

(11) 1990 Programmed computing at the Universities of Cambridge and Illinois in the early fifties. In *A History of Scientific Computing*. (ed. S. G. Nash), pp. 53–58. New York: ACM Press.

(12) 1992 The EDSAC programming system. *IEEE Ann. Hist. Comput.* **14** (4), 34–40.

(13) 1994 (With M. Burrows) *A Block-Sorting Lossless Data Compression Algorithm*. Palo Alto, CA: Systems Research Center, Digital Equipment Corp.

(14) A bulk data encryption algorithm. In *Fast Software Encryption 1993* (*Lecture notes in computer science* **809**) (ed. R. Anderson), pp. 127–134. Springer-Verlag.

(15) (With R. M. Needham) *Two cryptographic notes*. Technical Report 355, Computer Laboratory, Cambridge University.

(16) 1996 Transactions using bets. In *Security protocols* (*Lecture notes in computer science* **1189**) (ed. M. Lomas), pp. 89–92. Berlin: Springer.