THE ORIGINS OF COMPUTER PROGRAMMING

Brian Randell

Abstract

This paper describes some of the early developments which can now be viewed as constituting steps towards the development of program control, and of the modern concept of a stored program. In particular, it discusses early automatic devices, Babbage's contributions set against a background of the technology of his day, the contributions of some of his direct successors, and the genesis of the stored-program idea.

Keywords

Automata, Charles Babbage, History of Programming, Sequence Control, Stored Program Concept.

INTRODUCTION

This paper concerns the pre-history of programming, and some of the major developments that occurred en route to the development of the stored-program concept, but does not attempt to provide a complete history of the subject. Indeed, given the difficulty of assessing the ways and extent to which any given achievement affected later developments, this account will be more in the nature of a partial chronology than a history. However, I have tried to avoid the standard pitfall of a chronology – that of becoming mainly a catalogue of claimed "firsts". Such identifications are often misleading and controversial; in any case, with enough qualifications, almost anything can be so categorized.

Another pitfall I have attempted to avoid is that of giving just a "Whig interpretation" of history. Quoting [11]: "Whig historians produced chronicles of the heroes of the past, whose achievements were celebrated because they did well on a scale of values determined by the degree of accord with a present state of scientific knowledge and belief. . . . Historians of science since the 1950s have generally abandoned [this approach to history because] they have come to see the advantages of studying the scientific thought of the past in the direct terms of the problems and intellectual currents of the time under which any work was done, rather than merely 'grading' it in a schoolmasterish way in terms of its degree of accord with the present."

Thus, though this brief account has no pretensions to advancing the state of historical investigations into the origins of programming, it does aim to provide at least some brief explanations of the nature and extent of the intellectual and technical achievements that were involved in a few selected developments. However, it is important to realize that many of these particular developments have been selected more because I personally find them interesting from our current perspective than because of any contemporary importance or subsequent influence that I might believe or hope they have had.

EARLY AUTOMATA

One of the difficulties of discussing the historical origins of a subject is to decide where to begin. Charles Babbage's ideas relating to what we now know as programming significantly surpassed what had gone before, and like virtually all of his work on computing were hardly to be matched, leave alone surpassed, for a century afterwards. Thus, regardless of whether or not they significantly influenced the modern development of the subject, they could make a very appropriate starting point for this account. However, this paper takes a much earlier era as its starting point, since the great degree of innovativeness Babbage demonstrated cannot be adequately appreciated without some knowledge of the state of the "relevant arts" when he started his work, especially from an era in which computers have become so ubiquitous.

A most important art in this regard was that of means for specifying a sequence of choices amongst a set of possible machine actions in such a way that the machine can carry out the sequence completely automatically. There were of course other specialized arts that Babbage needed for his Analytical Engine (quite apart from such general facilities as tools for the accurate machining of mechanical components). These included means of storing large quantities of retrievable and changeable numerical and logical data, and of means for performing arithmetic operations mechanically. And in fact technologies for these latter arts were nothing like as well established by the 1830s, as were pegged cylinders and Jacquard cards, the two technologies that Babbage planned to use as means for automatic sequence control for his Analytical Engine. But it is just these two technologies that will be taken as the initial point of departure, for an account which deliberately confines itself closely just to programming-related issues.

The pegged cylinder, still used in music boxes on sale today, though probably not for much longer, can (at least with hindsight), be traced back to the time of Heron of ria [4; Alexand31]. In about 100 AD he described mechanisms involving the winding of a rope to and fro over the surface of a cylinder, from peg to peg, in such a way that when the cylinder was turned the rope wound and unwound irregularly, causing various other devices to perform a small sequence of actions. By this means several apparently miraculous effects were achieved, such as temple doors that apparently opened themselves and religious effigies that moved uncannily.

FIGURE 1

Figure 1: The Moving Temple of Bacchus, by Heron of Alexandria, showing a weight resting on a layer of millet or mustard seed. This weight descended slowly as the seed escaped from the upper compartment, and caused the rope to be unwound from the cylinder.

FIGURE 2

Figure 2: The forerunner of the pegged cylinder used by Heron of Alexandria. The rope contained eyelets which fitted over pegs protruding from the cylinder, and was fixed to the cylinder with wax. Thus as it was slowly pulled off the cylinder, the cylinder rotated first one way, then another, at various speeds.

The technique of using the pegs themselves directly, rather than such a rope, as the means of causing other mechanisms to go through complicated series of repetitive motions can be traced back as far as the thirteenth century, if not beyond. For example, Arabic drawings [16]. dating from this era show mechanisms such as one used in a model boat, powered by water pressure from a tank above the deck, which turned an axle which functioned as a pegged cylinder. The pegs on this cylinder caused little model human figures to move, for example, some so as to row the boat around a pond, and others to bang drums and cymbals - the boat being intended for entertainment purposes at royal drinking parties!

The written descriptions accompanying the drawings are so precise and detailed that there can be little doubt that the device was actually built and used. The flavour of these descriptions can be obtained from the following quotation:

> “The figure of a slave-girl flautist is made from jointed copper. She holds a flute with its end in her mouth. Next to her is a tambourine player, then a harpist, then another tambourine player. ... To the axle [i.e.

cylinder] a short peg is fitted, the end of which, when the [water-driven] wheel rotates, comes down on to the bent-up rod for the [tambourine-player's] hand, and presses it down. So the hand moves up and down. A single peg on the axle is not sufficient, and so two pegs are fitted close to each other opposite this peg, so that the movement of the hand gives two beats and one [beat]."

FIGURE 3

Figure 3: The 13th century model boat of Ibn al-Razzaz al-Jazzari. The pegged cylinder which controlled the actions of the various figures is seen underneath the main water tank, and to the left of the water wheel.

It is also known that similar pegged cylinder mechanisms were in use in Europe a century or so later to control the movements of model figures decorating large church clocks, and the playing of their bells. In most cases each set of pegs around a given circumference of the cylinder simply controlled the occurrences of a given action, such as a particular movement of a marionette, or the sounding of a particular musical note – thus we can view each potential peg position as storing a binary digit. (In some other, later, devices the actual shape or length of the peg was significant, so more information was provided by each peg.) In almost all cases, however, until quite late on, it would seem that such pegged cylinders were regarded as an integral part of the machine they were controlling. The insertion and removal of individual pegs was sometimes facilitated, but the cylinder itself could not usually be readily replaced by another one in order to cause, say, a different tune to be played.

FIGURE 4

Figure 4: Part of the mechanism of the large "Hydraulic Organ" of Salomon de Caus (1576-1626), showing the pegged cylinder which was used to control the operation of the keys of a flute.

VAUCANSON AND JACQUARD

The idea, or certainly the regular practice, of controlling a machine by sequencing information held on some clearly separate medium, so that a large variety of different sequences would typically be prepared away from the machine, over its lifetime, seems to have arisen first in the weaving industry, in fact in the early eighteenth century. In contrast, interchangeable pegged cylinders, and for that matter other interchangeable media such as punched tapes and disks, were not commonly used in automatic musical instruments and other automata until over a century later, though then they became extremely popular – for an account of the subsequent development of musical automata see, for example, [7].

Through the efforts starting in the early 1700s of a small series of French inventors, namely Bouchon, Falcon and Vaucanson, automatic sequencing was applied to silk-weaving so as to produce figured silken cloth [22; 29]. Such sequencing devices finally became commercially successful in the first decade of the nineteenth century, when Jacquard devised a fully automatic draw-loom which used strung-together punched cards (acting essentially as a wide punched paper tape). Each card controlled the selection of warp threads that were to be raised ready for a single passage across the loom of the shuttle carrying the weft thread. The result was to cause a complex pattern to be woven into the cloth. Earlier devices by Bouchon and Falcon also used tape or strung-together cards but were only semi-automated; Vaucanson's fully-automatic drawloom used a perforated cylinder, and hence was suitable only for comparatively simple repetitive patterns. The Jacquard technology spread rapidly – so rapidly in fact as to cause considerable industrial unrest – and thousands of examples of the Jacquard loom, as it came to be known, were in operation by the 1830s, including many in Britain.

Viewed from a current perspective, Jacquard looms are also of interest as marking the first time that automatic sequence control was used for serious commercial purposes, as opposed to being used for "merely" impressing or entertaining, or even frightening, people. However this view of their relative importance, and hence implication concerning how they were appreciated at the time, owes overmuch to hindsight and twentieth century values. Indeed, it seems clear that for centuries what many people, even serious philosophers, found most fascinating was the idea that human-like mechanical figures could be made to perform intricate life-like movements completely automatically [10].

For example, in the mid-18th century Vaucanson seems to have been much more famous for constructing and exhibiting such mechanical automata, which included a life-like human figure that played a real flute, and a very realistic duck, than for the very significant contributions he made to what later became known as the Jacquard technology [12]. He incidentally provides the first known link between the topics of automatic weaving and of mechanical automata. And Babbage himself gives the impression that visitors to his famous soirées tended to be more interested in the mechanical automaton that he had acquired, which he called his "silver lady", than in his work on machines which were intended to be used for automating the production and printing various practically-useful mathematical tables.

FIGURE 5

Figure 5: The famous mechanical automata constructed and exhibited by Jacques Vaucanson in the mid-18th century - the very lifelike Flute Player, Duck and Drummer

Jacquard technology enabled much longer action sequences to be specified than did pegged cylinders, and hence could be used to control the weaving of extremely complex patterns – one famous early example being a woven silk portrait of Jacquard, fine enough to be taken for a print made from a steel engraving, whose weaving involved no less than 20,000 cards. (Babbage is known to have been the proud possessor of one of these portraits, and to have presented another to the Grand Duke of Tuscany [3].) Moreover the sets of cards were manifestly physically separate from the machine they controlled, and the need for their production, on a grand scale, gave rise to a whole range of skills and tools.

Designs were normally drawn out on squared paper, from the successive lines of which the holes to be punched in a series of cards could be determined directly. Machines were soon introduced to aid the correct punching of cards from such designs, and also the making of multiple copies of a given card, and of a duplicate of a card sequence [22]. Incidentally, a splendid set of such machines is to be found, along with historic looms by Vaucanson, Jacquard and others, in the Musée des Techniques of the Conservatoire des Arts et Métiers, in Paris.

FIGURE 6

Figure 6: Devices used in early Jacquard-card making

However, so direct was the step of transcribing a drawing in order to produce a card sequence that it would be misleading to regard Jacquard cards and their preparation as involving any real form of programming. Card strings could be tied into a simple single loop, so as to control the weaving of a repeated pattern. But each of the set of possible actions that could be requested always had exactly the same effect, and thus each card caused the machine to do exactly the same thing each time it was read; and the sequence of such actions was totally immutable.

It was in fact to be Babbage who, in connection with his plans for using both punched cards and pegged cylinders, was for the first time to get past, indeed far past, these restrictions. As a result he got very near, if not quite, to the ideas of (machine-level) programming and of microprogramming as they are now understood.

CHARLES BABBAGE

Charles Babbage first became interested in automatic calculation in 1821, when he started his work on difference engines [15; 32]. However his ideas soon progressed far beyond that of a special-purpose calculating machine – indeed almost as soon as he started work on his first full size Difference Engine he became dissatisfied with its limitations. In particular, he wished to avoid the need to have the highest order of difference constant, in order to be able to use the machine directly for transcendental as well as algebraic functions.

In 1834 Babbage started active work on these matters, and on problems such as division and the need to speed up the part of the addition mechanism which dealt with the assimilation of carry digits. He developed several very ingenious methods of carry assimilation, but the time savings so obtained would have been at the cost of a considerable amount of complex machinery. This led Babbage to realize the advantages of having a single centralized arithmetic mechanism, the "mill", separate from the "figure axes", i.e. columns of geared wheels which acted merely as storage locations rather than as accumulators, each with their own adding mechanisms, as in his difference engines.

The complexity of his new "Analytical Engine" and in particular its mill, was such that Babbage sought a method of simplifying and making explicit the required sequencing of the activities of the various component mechanisms, for example in carrying out each individual addition, multiplication, etc. He made what was perhaps at the time a fairly natural choice, namely a pegged cylinder, for this sequencing requirement.

The full sophistication of Babbage's designs has only really become clear in recent years through Alan Bromley's detailed studies of Babbage's drawings and notations [5; 6]. These studies have revealed that Babbage had fully detailed, and essentially workable, designs for mechanizing various highly complex algorithms. These algorithms were to be executed under the control of a pegged cylinder that Babbage envisaged as providing 100 or more pegs, or "studs", in each of 50-100 "verticals", i.e. lines of stud positions in a vertical line parallel to the axis of the barrel. Thus, in modern terminology, the barrel acted as a microprogram store with 50-100 words, each of 100 or more bits. (The use of modern terminology in describing historical devices is often rather misleading, but in this case seems fully justified, so great are the conceptual similarities involved.)

Complex sequencing possibilities were allowed for by the fact that the barrel could be made to rotate a small number positions either forward or backward, or alternatively retain the same position, after a given vertical had been acted upon. Moreover, such movements of the barrel could be made conditional on the current state of the machine, and for example could depend on whether a particular arithmetic value had changed sign. To quote Bromley: "The whole concept of a conditional sequence of actions in a machine, and in particular of a conditional dependence on the outcome of previous actions of the machine, is original to Babbage and to the design of the Analytical Engine. It is a concept of the most profound importance." (This point will be returned to later.)

By such means, Babbage planned to control the execution of what were in many cases highly parallel algorithms. He worked out these algorithms with the aid of a number of different graphical notations that he himself had invented, and which functioned effectively as what we would term timing diagrams, logic diagrams, state-transition diagrams, and micro-program walkthroughs. In fact the logical sophistication of these algorithms, and indeed of the overall design of the Engine and its "microprogramming", exceeded that of many of the first generation of electronic computers [32].

Babbage had very early on decided that his machine should be of wide utility. Initially he also planned to use a pegged cylinder (with removable pegs) for controlling the sequence of major operations executed, and the choice of operands to be used, but very soon decided to use Jacquard cards instead – for what it seems very fair to describe as the programmed control of his machine. He took advantage of the fact that these cards were strung together to plan on the provision of means for "backing-up" (i.e. reversing through) a controlled number of cards, so as to be able to have cycles of operations, and to provide for alternative sequences to be executed. In so doing, he developed a very full understanding of the conceptual significance of his planned use of Jacquard cards (for what he termed "formulae"). In particular he realized that by virtue of the unbounded number of cards that could be used to control the machine, the ease with which complicated conditional branches could be built from a sequence of simple ones, and the fact that automatic card input and output, and multiple precision arithmetic were to be provided, Babbage stated that [2]:

> ". . . it appears that the whole of the conditions which enable a *finite* machine to make calculations of *unlimited* extent are fulfilled in the Analytical Engine. . . . I have converted the infinity of space, which was required by the conditions of the problem, into the infinity of time."

He found the concept of conditional branching particularly fascinating from a philosophical point of view. Indeed, in his book "The Ninth Bridgewater Treatise" (an unsolicited contribution to a series of theological texts) he devoted an extended section to a discussion of how a machine (or the Universe), if controlled by a program that used conditional branching, could exhibit surprising, even apparently miraculous, behaviour [1]. Moreover in an intended introduction to his machine, unpublished in his lifetime, Babbage included a very perceptive discussion of the problems of designing efficient programs, and of what could be done to prevent or tolerate faults in the machinery of the Analytical Engine itself, in the input data provided for it, and in the programs that were used to control it [2]. With regard to this latter point he wrote:

> "It must, however be observed, that if care is demanded from the attendants for the insertion of numbers which are changed at every new calculation of a formula, any neglect would be absolutely unpardonable in combining the proper cards in proper order, for the much more important purpose of constructing the formula itself, the arrangement of whose cards is never changed at any after time. . . . When the formula to be computed is very complicated, it may be algebraically arranged for computation in two or more distinct ways, and two or more sets of cards may be made. If the same constants are now employed with each set, and if under these circumstances the results agree, we may then be quite sure of the accuracy of them all."

Babbage did not, as far as is now known, work out the details of his program control to the same level of detail as his microprogram control. The justification for this comment is that the very large set of technical drawings and notebooks that Babbage left behind him, and that now constitute the most prized possession of the London Science Museum, contain far less information about the use of the Jacquard mechanism than on the use of the barrels. However from the descriptions he did leave, a number of his design decisions, and omissions, seem rather strange to modern readers. For example, he planned to use two separate strings of cards to control the Analytical Engine. The "operation cards" controlled the sequence of operations to be performed, the "variable cards" identified the storage locations which were to be used to provide the operands for these operations and to receive their results – and the means provided for the two strings of cards to be moved forwards or backwards seem to be quite independent of each other.

As explained by Bromley [5], it is clear that Babbage had what at the time seemed good reasons for the separation he made between operation and variable cards. But the result was that he apparently never arrived at the idea of what we would recognize as instructions, each identifying both an operation and its operands. Similarly, his ideas on card sequencing, and on loop control, do not seem to have been fully worked out – and to the best of present-day knowledge he never planned on providing means for the machine to calculate the address of a variable. Indeed, the various "formulae" he worked out, many of which were included in the annotated translation by Ada Lovelace of an Italian report on Babbage's lectures in Turin on his plans [19], are all really annotated traces rather than what would now be called programs.

Such comments should not be taken as "criticisms" of Babbage's work on programming. Rather, his achievements, and the way they range from detailed engineering design to deep understanding of the conceptual issues and consequences involved, are immensely impressive, especially given the level of knowledge and technology that existed at the time. One can argue about whether he was so far ahead of his time as to be pursuing unreasonable and unrealistic goals, and what impact his efforts and the publicity that surrounded them had subsequently. However, what is clear is that it was, as far as we know, another seventy or so years before anyone else progressed beyond Babbage's programming ideas towards those on which the modern computer is based.

BABBAGE'S EARLY SUCCESSORS

The first of Babbage's successors was Percy Ludgate, an Irish accountant [23]. In 1903, at the age of twenty, he started work on a novel method of performing decimal arithmetic by mechanical means, quite different from any incorporated in any of the fairly considerable variety of desk calculators that by then were on the market. So striking are the differences between Ludgate's plans and Babbage's that there seems little reason to dispute Ludgate's statement that he did not learn of Babbage's work until the later stages of his own. It does however seem likely that Babbage was the eventual inspiration for Ludgate to investigate the provision of a sequence control mechanism for his planned calculating machine.

The advance that Ludgate made was simply that of planning to use a single punched tape to control his machine. To quote his own account:

> "Babbage's Jacquard system and mine differ considerably; for, while Babbage designed two sets of cards – one set to govern the operations, and the other to select the numbers to be operated upon – I use one sheet or roll of perforated paper (which in principle exactly corresponds to a set of Jacquard-cards) to perform both these functions in the order and manner necessary to solve the formula to which the particular paper is assigned." [20]

Each row of perforations across the tape thus constituted a single instruction, as we would now term it. Control transfers would then just involve moving the tape the appropriate number of rows forwards or backwards. Moreover he also envisaged the use of what we would now call subroutines, represented by sequences of similarly-coded rows of perforations around the circumference of special cylinders - one such cylinder was to be provided for performing division, which he did not envisage providing as a built-in operation. There is no evidence that he ever tried to construct his machine, which he apparently worked on alone, and in his spare time. Indeed all that is known of his work comes from the two papers he published [20; 21], the second of which was a survey paper, containing a description of Babbage's machine. This incidentally was apparently one of the two main sources of Howard Aiken's later "discovery" of Babbage.

Perhaps just one other development that occurred within the century following Babbage's invention of the Analytical Engine is worth mentioning here – namely the work of Torres y Quevedo, a renowned Spanish scientist and engineer. Torres did much work on the development of electromechanical digital devices, and selected Babbage's Analytical Engine as an important and interesting challenge to demonstrate their power and utility. In 1914 he published a paper [26] showing how the various components of an Analytical Engine might be built from his technology, in which he described the importance of conditional branching in the following terms:

> "Moreover it is essential – being the chief objective of Automatics – that the automata be capable of *discernment*; that they can at each moment, take account of the information they receive, or *even information that they have received beforehand*, in controlling the required operation. *It is necessary that the automata imitate living beings in regulating their actions according to their inputs, and adapt their conduct to changing circumstances*."

FIGURE 7

Figure 7: A schematic design for an electromechanical calculating device, controlled by the pattern of conducting areas on the surface of a rotating cylinder, and embodying means for conditional branching, devised by Leonardo Torres y Quevedo (1852-1936).

He later constructed a typewriter-controlled calculator, which embodied provisions for such "discernment", to demonstrate the practicality of his electromechanical components [27]. This and the other devices that he built successfully, which included two chess-playing automata, were not really program-controlled devices, though they do provide evidence that should the need have been pressing, Torres could in the 1920s have successfully produced a complete Analytical Engine.

THE STORED PROGRAM CONCEPT

As viewed from a present-day vantage point, perhaps the really crucial machine-level programming concept remaining to be devised was the "stored program concept". There has been much controversy over the credit due for this development, some of which at least is due to lack of agreement as to just what the concept involves.

The machines mentioned so far (and others such as those of Aiken and IBM, of Stibitz and Bell Labs, of Konrad Zuse in Germany, as well as the Colossus series of code-breaking machines and Eckert and Mauchly's ENIAC) were all controlled by a program held on some read-only medium, such as switches, punched cards or tape, which was quite separate from the (writeable) storage device used to hold the information that was being manipulated by the machine. Nevertheless, Babbage at least was aware of the possibility, and the potential importance, of having the Analytical Engine be able to generate and output its own programs, i.e. punched card formulae, from more abstract descriptions of the intended calculations. (This he viewed as a good means of reducing the incidence of errors in lengthy formulae [2].)

The advent of electronics, and the first attempts at building programmable electronic calculating devices in the late 1930s and early 1940s exposed a need for some means of representing programs:

(i) which could cope with the required program sizes,

(ii) whose access speed matched that of the (fully-electronic) operations they were controlling, and

(iii) which allowed adequately fast means of replacing a program whose task had been finished with the next program to be executed.

This problem had not existed with card- or tape-controlled mechanical or electro-mechanical devices, whose calculation speeds were reasonably well-matched to the speed with which the cards or tape that controlled them could be read. And though the system of plugs and cables used for programming the ENIAC [8] was well matched to the calculation speed of its electronics, the plugging task involved in replacing one program by another could take several days. (An essentially similar system used a few years earlier for the Colossus [24] did not take so long, simply because the programs that could be set up were very much shorter and simpler.) This situation led to the realization, probably first in the ENIAC/EDVAC team at the University of Pennsylvania, of the advantages of storing the program within the computer, in a memory that could be read at electronic speeds during program execution [13]. Then the fact that different types of applications had greatly differing relative requirements for instruction and (both variable and constant) data storage soon led to a realization of the practical benefits of using a single store for all three types of information [14].

> "The first type of internal memory is a high-speed memory which has a high-speed input by which the data could be placed rapidly in the memory, and a high speed output from which the data could be taken out rapidly. This first type of memory, which is contained in the accummulators, is rather limited in size since it can hold only two hundred decimal digits and twenty binary digits. The second type of memory, which has a low input speed and a high output speed, is typified by the function tables. The third type of memory, which is again characterized, as was the second type of memory, by having a low input speed and a high output speed, is the memory which is used to control and sequence the arithmetic operations of the ENIAC. [If] we can produce a form of memory cheap enough, and if that memory has the characteristics of type one memory, there is no reason why the same form of memory cannot be used for all three types of memory."

To some commentators this constitutes the essentials of the stored program concept. However, to many, myself included, the concept also has strong connotations of the computer being able to construct, manipulate and then (surpassing the notion that Babbage had arrived at over a century earlier) execute its own programs, all completely automatically. With this latter view the stored program concept becomes an engineering approximation to the theoretical universal automaton that Turing had postulated in his (now) famous 1936 paper [28] – i.e. a machine which is general-purpose in a very fundamental mathematical sense as well as in a very practical sense. Thus, given the practical requirement of replacing the Turing Machine's infinite tape by a sufficiently large store random access store, it is crucial for the computer to be able to calculate the addresses that are used to access the store, rather than only being able to use pre-calculated (i.e. fixed) addresses. This, to my mind, is a crucial characteristic of a modern stored program computer [25].

In fact by these standards the first (1945) design for EDVAC [30] does not qualify as a stored program computer. Although data and instructions were to be held in the same store their representations were quite distinct, and no means were provided for converting data items into instructions. Furthermore, normal arithmetic operations could not be applied to instructions – though the address field of an instruction could be modified.

This inadequacy, as we would of course view it, was not present in Turing's proposed design for an Automatic Computing Engine (ACE), which slightly post-dated the EDVAC report, and was also very quickly remedied in the EDVAC design [9]. In both cases, however, what now seem very awkward techniques of program self-modification were needed in order to make the machine calculate the addresses of variables – since neither the idea of index registers (B-lines as they were to be called, at Manchester, where they were invented [17; 18]) or of indirection had yet arisen. However, once all these aspects of the stored program concept had been provided, and although there was a huge space of possible instruction formats and sets still to explore and exploit, (machine-level) programming essentially as we know it now had arrived. Needless to say, the immense importance of this “event” has only become evident with the benefit of hindsight.

ACKNOWLEDGEMENTS

This paper is based on one that was commissioned originally for a conference organized by the IEE to mark the bi-centenary of the birth of Charles Babbage – a conference which unfortunately never took place. The present version of the paper has benefitted significantly from the advice obtained from the referees and the editor of this Special Issue of the Annals.

Figures 1 and 2 first appeared in reference [4], and are reprinted by permission of R. Oldenburg Verlag GmbH, publishers of Elektronische Rechenanlagen, in volume 15 of which they appeared in 1973. Figure 3 is from reference [16] and is reprinted by permission of Kluwer Academic Publishers. Figures 4 and 5 are from reference [10], and are reprinted by permission of the publishers, Éditions du Griffon. Figure 6 is reprinted with permission of the Smithsonian Museum. Figure 7 is from reference [26] and is reprinted by permission of Real Academio de Ciencias, Madrid.

REFERENCES

1. C. Babbage. *The Ninth Bridgewater Treatise: A Fragment,* London, John Murray, 1837.

2. C. Babbage. *On the Mathematical Powers of the Calculating Engine,* Unpublished Manuscript, 1837. (First published in *Origins of Digital Computers: Selected Papers* (ed. B. Randell) Springer-Verlag, 1973.)

3. C. Babbage. *Passages From the Life of a Philosopher,* London, Longman, Green, Longman and Roberts (Reprinted 1969 by Augustus S. Kelley, New York), 1864.

4. T. Beck. *Beiträge zur Geschichte des Maschinenbaues,* Berlin, Springer, 1900.

5. A.G. Bromley, "Charles Babbage's Analytical Engine, 1983," *Annals Hist. Comput.*, vol. 4, no. 3, pp.196-217, 1982.

6. A.G. Bromley, "The Evolution of Babbage's Calculating Engines," *Annals Hist. Comput.*, vol. 9, no. 2, pp.113-136, 1987.

7. A. Buchner. *Mechanical Musical Instruments,* London, Batchworth Press, 1950.

8. A.W. Burks and A.R. Burks, "The ENIAC: First General Purpose Electronic Computer," *Annals of the History of Computing*, vol. 3, no. 4, pp.310-389, 1981.

9. B.E. Carpenter and R.W. Doran, "The Other Turing Machine," *Comp. J.*, vol. 20, no. 3, pp.269-279, 1977.

10. A. Chapuis and E. Droz. *Automata: A historical and technological study,* London, V.T. Batsford, 1958.

11. I.B. Cohen, "Babbage and Aiken: With Notes on Henry Babbage's Gift to Harvard, and to Other Institutions, of a Portion of His Father's Difference Engine," *Annals Hist. Comput.*, vol. 10, no. 3, pp.171-193, 1988.

12. A. Doyon and L. Liagre. *Jacques Vaucanson: Méchanicien de Génie,* Paris, Presses Universitaires de France, 1966.

13. J.P. Eckert. *Disclosure of a Magnetic Calculating Machine,* Unpublished Typescript, 1945. (Reprinted in J.P. Eckert. "The ENIAC," in *A History of Computing in the Twentieth Century,* pp. 525-539, New York, Academic Press, 1980.)

14. J.P. Eckert. "A Preview of a Digital Computing Machine," in *Theory and Techniques for the Design of Electronic Digital Computers, ed. C. C. Chambers*, pp.10.1-10.26, Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, 1947.

15. A. Hyman. *Charles Babbage: Pioneer of the Computer,* Oxford, Oxford Univ. Press, 1982.

16. Ibn al-Razzaz al-Jazzari. *The Book of Knowledge of Ingenious Mechanical Devices (Trans. and ed. by D.R. Hill),* Dordrecht, D. Reidel, 1974.

17. S.H. Lavington, "A Golden Era Rich in Talent," *Computing*, 11 Aug.1977.

18. S.H. Lavington. "Computer Development at Manchester University," in *A History of Computing in the Twentieth Century, ed. N. Metropolis, J. Howlett and G. C. Rota*, pp.433-443, Academic Press, New York, 1980.

19. A.A. Lovelace, "Sketch of the Analytical Engine invented by Charles Babbage, by L.F. Menabrea, Officer of the Military Engineers, with notes upon the memoir by the Translator," *Taylor's Scientific Memoirs*, vol. 3, no. Article 29, pp.666-731, 1843. (Reprinted in *Origins of Digital Computers: Selected Papers* (ed. B. Randell) Springer-Verlag, 1973.)

20. P.E. Ludgate, "On a Proposed Analytical Machine," *Sci. Proc. Roy. Dublin Soc.*, vol. 12, no. 9, pp.77-91, 1909. (Reprinted in *Origins of Digital Computers: Selected Papers* (ed. B. Randell) Springer-Verlag, 1973.)

21. P.E. Ludgate. "Automatic Calculating Engines," in *Napier Tercentenary Celebration: Handbook of the Exhibition, ed. E. M. Horsburgh*, pp.124-127, Edinburgh, Royal Society of Edinburgh, 1914. (Also published as *Modern Instruments and Methods of Calculation: A Handbook of the Napier Tercentenary Celebration Exhibition*, G. Bell and Sons, London, 1914.)

22. E.A. Posselt. *The Jacquard Machine Analyzed and Explained: with an Appendix on the preparation of Jacquard cards, and practical hints to learners of Jacquard designing,* Philadelphia, PA, Pennsylvania Museum and School of Industrial Art, 1887.

23. B. Randell, "Ludgate's Analytical Machine of 1909," *Computer J.*, vol. 14, no. 3, pp.317-326, 1971.

24. B. Randell. "The Colossus," in *A History of Computing in the Twentieth Century, ed. N. Metropolis, J. Howlett and G. C. Rota*, pp.47-92, Academic Press, New York, 1980.

25. B. Randell, "Comment (on The ENIAC: The First General-Purpose Electronic Computer, by A. W. Burks and A. R. Burks)," *Annals of the History of Computing*, vol. 3, no. 4, pp.396-397, 1981.

26. L. Torres y Quevedo, "Essais sur l'Automatique. Sa définition. Étendue theorique de ses Applications.," *Revue de l'Academie des Sciences de Madrid*, 1914. (Reprinted in *Origins of Digital Computers: Selected Papers* (ed. B. Randell) Springer-Verlag, 1973.)

27. L. Torres y Quevedo, “Arithmomètre Électroméchanique,” *Bulletin de la Soc. d’Encouragement pour l’Industrie Nationale*, vol. 119,1920.

28. A.M. Turing, “On Computable Numbers, with an Application to the Entscheidungsproblem,” *Proc. London Math. Soc. (2)*, vol. 42, pp.230-267, 1936.

29. A.P. Usher. *A History of Mechanical Inventions (2nd rev. ed.),* Cambridge, MA, Harvard Univ. Press, 1954.

30. J. von Neumann. *First Draft of a Report on the EDVAC. Contract No. W-670-ORD-4926,* Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, PA, 1945. (Extracts reprinted in *Origins of Digital Computers: Selected Papers* (ed. B. Randell) Springer-Verlag, 1973.)

31. W. Walther, “Die Gespeircherten Programme des Heron von Alexandria: Heron’s of Alexandria stored programs,” *Elektronische Rechenanlagen*, vol. 15, no. 3, pp.113-118, 1973.

32. M.V. Wilkes. “Babbage as a Computer Pioneer,” in *Report of Proceedings, Babbage Memorial Meeting, London, 18 Oct. 1971,* London, British Computer Society, 1971.