Charles Babbage's Analytical Engine, 1838

ALLAN G. BROMLEY

Charles Babbage commenced work on the design of the Analytical Engine in 1834, following the collapse of the project to build the Difference Engine. His ideas evolved rapidly, and by 1838, most of the important concepts used in his later designs were established. This paper introduces the design of the Analytical Engine as it stood in early 1838, concentrating on the overall functional organization of the mill (or central processing portion) and the methods generally used for the basic arithmetic operations of multiplication, division, and signed addition. The paper describes the working of the mechanisms that Babbage devised for storing, transferring, and adding numbers and how they were organized together by the microprogrammed *control system; the paper also introduces the facilities provided for user-level programming. The intention of the paper is to show that an automatic computing machine could be built using mechanical devices and that Babbage's designs provide both an effective set of basic mechanisms and a workable organization of a complete machine.*

[*Editor's Note: This article originally appeared in the* Annals*, vol. 4, no. 3, 1982.*]

Introduction

Charles Babbage commenced work on the design of the Analytical Engine shortly after the collapse in 1833 of the 10-year project to build the Difference Engine. He was 42 years old at the time.

Progress on the Analytical Engine went rapidly. The first notes appeared in mid-1834, and by mid-1836, a workable design had evolved. A major revision of the design took place in late 1837. For the next decade, work on the Analytical Engine consisted largely of refinement and elaboration of the basic design of 1837–1838. During this period, Babbage appears to have made no attempt to construct the Analytical Engine, but preferred the unfettered intellectual exploration of the concepts he was evolving.

After 1849, Babbage ceased designing calculating devices. He returned to the subject in about 1856, at the age of 64, probably inspired by the completion of the Scheutz Difference Engine, which had been developed from a description of his own earlier Difference Engine. In this second phase of work on the Analytical Engine, Babbage concentrated on methods by which it might be built at a reasonable price and explored both precision-stamping and pressure-die-casting techniques. A substantial model was under construction at the time of his death in 1871 at the age of almost 80.

Very little in the way of a general introduction to the Analytical Engine was written in Babbage's lifetime. At the end of 1837, Babbage wrote a general description, but it was not published until recent times.[8] In 1840, Babbage journeyed to Turin, Italy, and presented a series of seminars to a group of distinguished Italian scientists with the intention that one of them would publish an account of his ideas. Such a description was published by L.F. Menabrea and subsequently translated from French into English by Ada Lovelace in 1843.[2] The translation has extensive notes, written under Babbage's supervision, that give an excellent account of his understanding of the mechanization of computational processes and the mathematical powers of the machine. Unfortunately, none of the contemporary accounts gives any useful technical detail on the mechanism or design of the Analytical Engine, and it is impossible to assess from those accounts its merits as a computing machine.

Babbage did, however, keep extensive private notes on the design of the mechanism of the Analytical Engine. The notes, now in the collections of the Science Museum in London, comprise:

- about 300 sheets of engineering drawings, typically two by three feet;
- 600–700 *notations*—abstract representations of the mechanism, such as timing diagrams, logic diagrams, flow charts, state-transition diagrams, and walk-throughs of microprograms, all typically two by three feet or larger of close handwriting; and
- 6,000–7,000 pages of notebooks.

The total amount of material is daunting, and it suffers the disadvantage that it is largely very detailed and technical in nature and does not provide much overview of the design. In particular, matters such as the intended user's instruction set are difficult to determine, although there are several dozen sample programs prepared between 1837 and 1840 (all, incidentally, substantially predating the preparation of Lovelace's notes, which incorporate several of them).

This paper gives an account of the design of the Analytical Engine as it stood in 1838. It was this basic design that Babbage refined and elaborated in the period up to 1848. The description is based on a close examination of the papers in the Science Museum but has been simplified in many respects, particularly with regard to details of the mechanism. The intention of this paper is to show that an automatic computing machine could be built using mechanical devices and that Babbage's designs provide both an effective set of basic mechanisms and a workable organization of a complete machine. The paper is also intended to serve as an introduction to more detailed analyses of Babbage's designs that are in preparation.

The paper starts with a discussion of mechanical devices for the storage, transfer, and addition of numbers. The arrangement of these basic parts into a block-diagram-level description of the organization of the machine is presented. The algorithms for multiplication, division, and signed addition are then described along with the mechanism by which the control of these was microprogrammed. Finally, the user programming of the Analytical Engine is examined as it was understood at this stage of the design.

Three other persons have studied the Babbage papers in the Science Museum collection. Bruce Collier[5] has provided an excellent overview of Babbage's work on automatic calculating devices. His study, based largely on the notebooks and on the correspondence in the British Library and the Buxton papers at Oxford University, is primarily concerned with the intellectual history of Babbage's ideas and, in particular, with the transition from the special-purpose Difference Engine to the general-purpose Analytical Engine. Maurice Wilkes[9] has studied the notebooks to achieve a more detailed understanding of some aspects of the user-programming facilities of the Analytical Engine than is provided in the present paper. Anthony Hyman[7] has also provided an overview of Babbage's calculating machines in the context of a biographical study of Babbage's wide-ranging intellectual achievements.

The present paper is the first one based primarily on a detailed study of Babbage's notations and is, therefore, the first to provide a detailed account of the organization of the machine and, in particular, of the sophistication of Babbage's microprogramming ideas. It discusses only one model out of the complete evolutionary range of Babbage's designs (though one I believe to be in most respects typical), and many of the conclusions must therefore be regarded as tentative, pending the extension of this study to other designs.

In almost all its facets, Babbage's work on the Analytical Engine was completely original—the only antecedent was his own Difference Engine. Only in the adoption of the barrel, from automata and music boxes, and punched cards, from the Jacquard loom, is there any significant borrowing from an existing technology, but Babbage's substantial elaboration of the capabilities of these, as well as his specific application of them, is again quite original.

Overview

The Analytical Engine was a decimal machine that used a sign-and-magnitude representation for numbers in the store. Babbage's decisions to adopt decimal and sign-and-magnitude number representation (both of which are now uncommon) were arrived at after careful deliberation.

Modern computers are largely binary, because it is much easier to design a two-state electronic circuit than one with more states, such as 10 for a decimal-number representation. No such consideration applies to mechanical apparatus. A wheel rotating on a shaft is a particularly easy element to construct and can as readily have 10 or 100 positions as two. Babbage considered number bases over this range with respect to speed and quantity of apparatus required and found no significant reason to depart from the decimal system that made input/output and the examination of internal states much easier. While electronic computers are built, conceptually at least, with very low-level logic devices, so that addition is assembled from NAND functions, the components of the Analytical Engine are generally at a higher level and correspond more directly with the function they perform. The decision to adopt a decimal representation is, therefore, a consequence of a characteristic difference between electronic and mechanical devices, although there are elements in the Analytical Engine whose functions are essentially of a logic nature.

A sign-and-magnitude representation is used for numbers for the same reasons of ease of input/output and examination of internal states that caused Babbage to adopt a decimal system—and also because it simplifies multiplication and division operations. The Difference Engine used a 10s-complement representation for negative numbers, and the Analytical Engine uses a variety of complement and recoded representations to simplify its internal operations.

In marked contrast to the Difference Engine, the Analytical Engine makes a clear distinction between the *store*, in which operands and results are held between operations, and the *mill*, to which they are brought to perform arithmetic operations. The store and the mill correspond closely to the memory and central processing unit of a modern computer.

In both the store and the mill, the digits of numbers are represented by the positions of wheels (*figure wheels*) rotating about vertical *axes*. A collection of figure wheels on an *axis* corresponds to a register in a modern computer. The Analytical Engine is built from a series of horizontal plates separating the figure wheels on the various axes. The space between two plates is called a *cage*. The bottom cage holds the units figure wheels on each axis, the next above the 10s, then 100s, and so on. In general, digit transfers take place simultaneously in all cages, so the Analytical Engine is a digitwise parallel machine.

All numbers in the Analytical Engine are of 40 digits, and there are 40 cages and 40 figure wheels on each axis. This large number was possibly chosen to simplify scaling problems in the absence of a floating-point number system. Since the plates dividing the cages were about three inches apart, a figure axis would be about 10 feet high. Allowing for the control mechanism underneath, the Analytical Engine would have stood about 15 feet high. The mill would have been about six feet in diameter, and the store would have run 10–20 feet to one side. The Analytical Engine would therefore have been about the size and weight of a small railway locomotive.

The internal operations of the Analytical Engine, such as multiplication and division, were controlled by *barrels* to which were fixed studs, something like in a music box or barrel organ. The barrels acted to effect control in a manner closely analogous to modern microprogram control; they were just like microprogram stores. The person programming the Analytical Engine would use punched cards that initiated sequences of micro-operations. Babbage had a clear understanding of such hierarchies of control.

It is tempting in describing the Analytical Engine to use modern terms, such as *register* and *microprogram* in place of Babbage's *axes* and *barrels*. There is some risk in doing so, because Babbage's elements differed in detail from their modern equivalents and had, in consequence, different functional characteristics. For example, two sets of figure wheels on an axis usually shared a common reading and transfer mechanism. An axis could therefore store more than one number, unlike a modern register. All transfers involved a destructive readout, again unlike a modern register, so two sets of figure wheels were necessary if an axis was required to give off and retain a number. I will use Babbage's terminology when describing details of his mechanisms, but use the modern terms when emphasizing the functional analogy with modern machines. This method will provide a painless introduction to Babbage's terminology, the use of which will be essential in more detailed analyses.

Throughout this paper, the mechanisms of the Analytical Engine are described by simplified figures that avoid much of the detail of Babbage's designs. The figures have been developed for a pedagogic purpose, because most computing people cannot be expected to be familiar with and fluent in reading detailed mechanical-engineering drawings. Thus, for example, pictorial isometric drawings of mechanisms are used in place of Babbage's plan and elevation drawings; and the mechanisms have been simplified and rearranged to show only those features essential to the discussion in this paper. If the figures suggest gross design flaws or evident physical impossibilities, the fault is mine, not Babbage's. An examination of his original drawings is essential for a critical appraisal of his designs. Where practicable, the simplified figures have been labeled to correspond to Babbage's drawings.

Storage and Transfer of Numbers

A typical figure wheel and its axis are shown in Fig. 1. The figure wheel can be rotated by means of the gear teeth to stand in any one of 10 positions; the digit then stored is shown by the figure opposite the index mark carried on the framework. In practice, the figure wheel is brought correctly to position and secured against the possibility of subsequently being jarred out of place by locking arms that engage locking teeth on the figure wheel. These arrangements are not shown in the figure but are essential to the reliable operation of the machine.

A finger b protrudes to the inside of the figure wheel opposite the zero digit position. To read out the number stored on a figure wheel, the figure axis is raised until a finger a on it is at the same level as the finger b of the figure wheel. The axis is then rotated through nine digit positions. At some point during this rotation, the fingers a and b will come into contact, and the figure wheel will be rotated until it finally comes to stand at the zero position. In the process, the figure wheel will move through exactly as many digit positions as the value of the digit initially stored on it; this movement can be conveyed via the gear teeth of the figure wheel to some other part of the Analytical Engine.

This form of storage exhibits a destructive readout; following the read operation, all figure wheels will stand at zero, irrespective of the digit originally stored, and the number originally stored is lost. If it is desired not to lose the number, then it must be stored, as it is read, on another set of figure wheels. For this purpose, each figure axis of the mill of the Analytical Engine is provided with two figure wheels in each cage. Frequently, as a number is *given off* by one set of figure wheels, it is stored on the other set on the same axis and, hence, later given off back to the original set. The two sets of figure wheels on one axis may, however, be used independently if so desired. In Babbage's terminology, a number is given off by a set of figure wheels as they are *reduced to zero*.

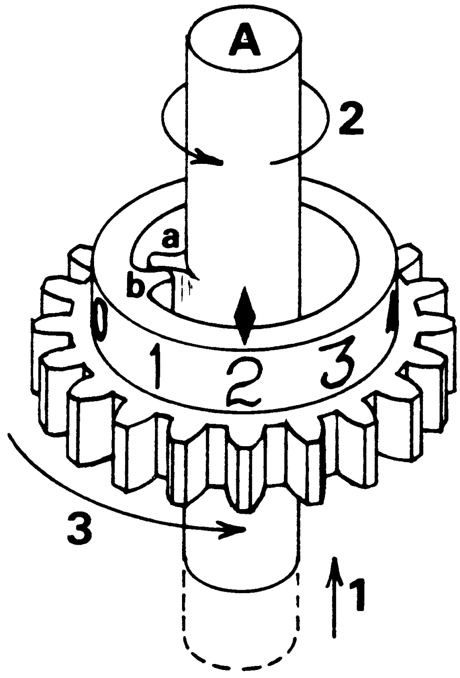


Fig. 1. Figure wheel typical of the store and the mill figure axes. The wheel may stand in one of 10 positions to store the digit shown by the index mark. The number may be read by raising the axis until its finger is level with that on the inside of the figure wheel and rotating it through nine digit positions. The figure wheel will come to stand at the zero position, and a movement proportional to the digit originally stored will be given to the remainder of the mechanism by the gear teeth. The process is termed *giving off*.

Based on a plate in *Babbage's Calculating Engines*.[2]

The basic process of addition is shown in Fig. 2. A digit is given off by the figure wheel on the right, on axis A, and transferred by the intermediate gears G and J to the figure wheel on axis 'A. The directions of motion are so judged that as the figure wheel on A is reduced to zero by moving in the subtractive sense, the figure wheel on 'A moves in the additive sense. If, before the transfer, the figure wheel on 'A stood at zero, it will come to stand at the same digit as was originally stored on A. If, however, the figure wheel on 'A initially stood at a nonzero digit, it will finally come to stand at the sum of that digit and the digit given off by A. The same process occurs simultaneously in all cages, so we have digitwise parallel addition.

Before we examine the possibility of a carry generated in the addition process, we will consider the arrangement of the transfer gearing in more detail. Fig. 3 shows two cages from the Analytical Engine; there are two sets of figure wheels on each of the figure axes A and 'A. A number can be given off by either set of figure wheels on A by raising or lowering the axis so that the internal fingers of the appropriate set of figure wheels are engaged by the fingers on the axis. The *transfer pinions* are loose on axis G but can be raised or lowered with it to engage with either set of figure wheels of A. The transfer pinions on G are, however, always engaged with the *long pinions* on axis S, so the long pinions will receive the motion from whichever set of figure wheels on A gives off a number. This motion is transferred to the figure wheels on 'A by the long pinions on L and the transfer pinions on J as shown. The net result is digit transfers identical to those described in Fig. 2 from either set of figure wheels on A to either set on 'A.

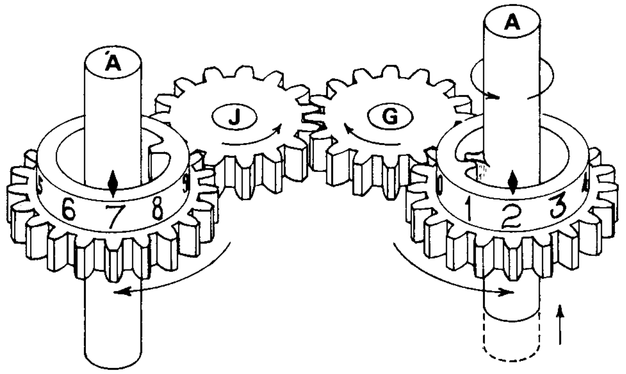


Fig. 2. Basic process of addition. A digit is given off by the figure wheel on the right and received by the figure wheel on the left. If the receiving figure wheel is not originally at zero, it will finally come to stand at the sum of its original value and the digit received. This process occurs simultaneously for all digits of numbers.

Based on a plate in *Babbage's Calculating Engines*.[2]

are also commonly used as the takeoff point for transfers to and from other parts of the machine.

With such long trains of gearing between the source and destination of motion, substantial amounts of lost motion and backlash can arise because of the necessary looseness of the gearing. To prevent the lost motion from disturbing the correct functioning of the machine, locking arms act on the figure wheels and some of the intermediate gearing to bring them correctly into place. These *lockings* are applied in sequence from the source to the destination of each transfer. The effect is similar to providing an amplifying element in each gate of an electronic machine, as is universally done, to make output logic levels conform to the same electrical specifications as the inputs. The lockings are essential to the correct action of the Analytical Engine and are demonstrative of the thoroughness of Babbage's designs and his inventive genius. (The Scheutz Difference Engine was not very reliable in practice and soon fell into disuse, largely because it lacked any such provision as lockings. The idea had first appeared in Babbage's own designs for his earlier Difference Engine.)

The long pinions on S can be raised with their axis so that each will engage the long pinion L in the cage above, as shown in Fig. 4. A digit given off by the units figure wheel of A will now be added to the chosen 10s figure wheel of 'A, and every digit given off will similarly be stepped up one cage. This stepping corresponds to a multiplication by 10 and is the decimal equivalent of a shift in a binary computer. Since mechanical gearing is reversible, a number can be given off by 'A and stepped down, or divided by 10, before being received by A. These stepping functions are extensively used in multiplication and division operations.

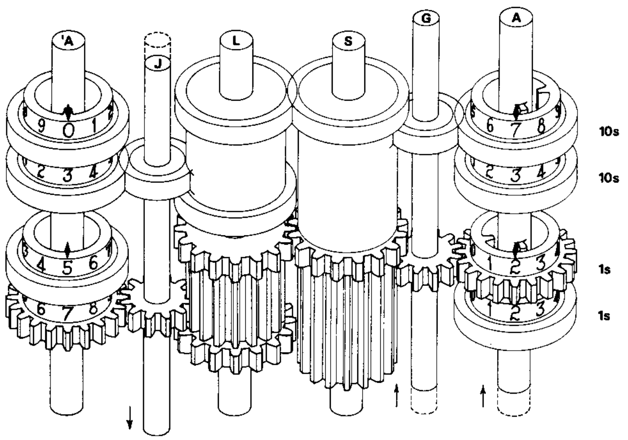


Fig. 3. Practical realization of the method of digitwise parallel addition. Two figure wheels are shown in each cage, so two numbers can be stored on each figure axis. In practice, the mechanism is arranged in a circle so that the axes A and 'A coincide. A number given off may therefore be received on the alternate set of figure wheels of the same axis.

Based on a plate in *Babbage's Calculating Engines*.[2]

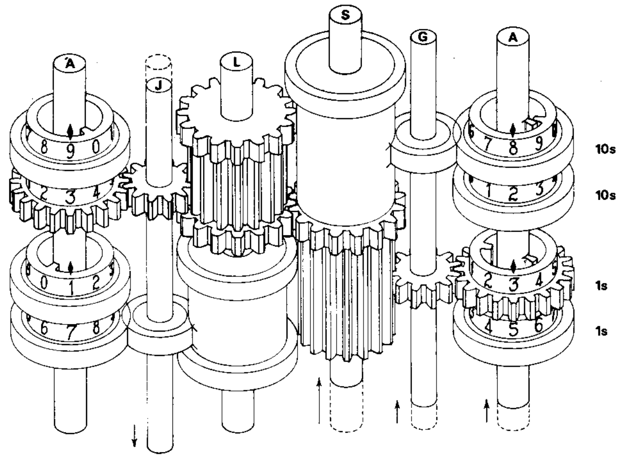


Fig. 4. The process of stepping, or multiplication, and division by 10. The *long pinions* S are raised with their axis to engage the long pinions L in the cage above. A number given off will therefore be stepped up a cage, or multiplied by 10. A transfer in the reverse direction will divide by 10.

Based on a plate in *Babbage's Calculating Engines*.[2]

In practice, digits transferred can be received by figure wheels on the same axis as those that give off by arranging the axes A, G, S, L, J, and 'A not in a line as shown but in a circle, so that axis 'A is the same axis as A. Such a group of axes, labeled as in Fig. 3, will be found to occur frequently in the general arrangement of the Analytical Engine discussed below. The long pinions S and L

Anticipating Carry

The mechanisms described so far show how numbers can be stored, given off, stepped, and transferred between axes of the Analytical Engine. A digitwise parallel form of addition is a natural result of these mechanisms, but a proper arithmetic addition requires a method of performing a carry propagation. The carry mechanism is relatively complex and was the initial incentive for Babbage to separate the store and the mill, with the arithmetic mechanisms concentrated in the latter.

The carry mechanism is based around a special carry axis F of figure wheels similar to those previously described. Some components of the mechanism are shown in Fig. 5, and they are elaborated in Fig. 6 and Fig. 7. The carry figure wheels have a lever arm f attached to them that is placed so that when the figure wheel moves from nine to zero in the additive direction, arm f rotates a piece pivoted on axis E by acting on lever e attached to it (as shown with dashed lines in Fig. 5). This *carry warning*, when in the rotated position, indicates that a carry has been generated by the movement of the carry figure wheel past nine and must be applied to the figure wheel in the cage above.

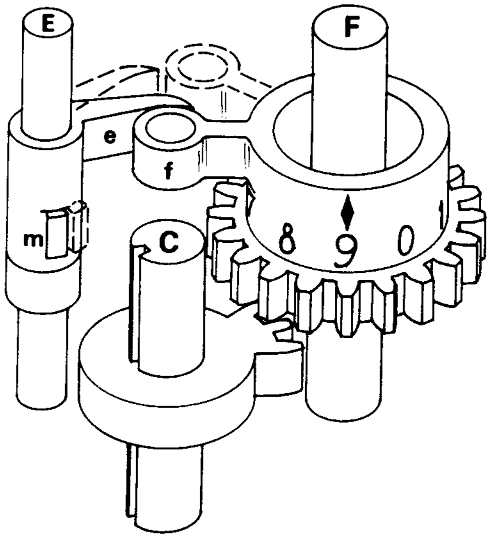


Fig. 5. Elements of the carry mechanism. When the carry figure wheel moves past nine to zero, arm f acts on arm e to rotate the *carry warning,* indicating that a carry is necessary into the cage above. A carry from the cage below will raise the *carry sector* on axis C into gear with the figure wheel and advance it by one digit position.

Based on a plate in *Babbage's Calculating Engines*.[2]

After addition to the carry figure wheels is complete, axis E, on which the carry warnings pivot, is raised. The lug m of any carry warning that has been rotated to indicate a carry will engage a similar lug n on the corresponding *carry piece* on axis W (Fig. 6) and cause it to be lifted, too. Each carry piece has two fingers that project through the framing plates into the cage above. The finger nearest the front in the figure bears on the underside of a *carry sector* and, when lifted, causes the gear teeth of the carry sector to engage those of the figure wheel that are to receive the carry. The carry sectors are then all rotated through one digit position by their axis C, to which they are splined, as shown in Fig. 5. If the carry sector has been raised into gear with the figure wheel, it will also move forward through one digit position to effect the carry propagation.

If, say, the 10s figure wheel stood at nine before it received a carry from the units cage, then the carry propagation should generate a new carry into the 100s cage. This carry could in turn generate a new carry into the 1,000s cage if the 100s figure wheel stood at nine, and so on. In the worst case, this carry propagation could continue through all 40 digit cages of a number. The sequential carry propagation suggested by this description will be exceedingly slow. If a carry could be propagated through a cage in the same time that a figure wheel rotates through one digit position (an unlikely possibility given the complex motions of the mechanism), the complete chain of carry propagation could, in the worst case, take five times as long as the complete digitwise parallel addition of two numbers. Babbage overcame this difficulty with his *anticipating carry* mechanism—an invention of which he was justifiably most proud.

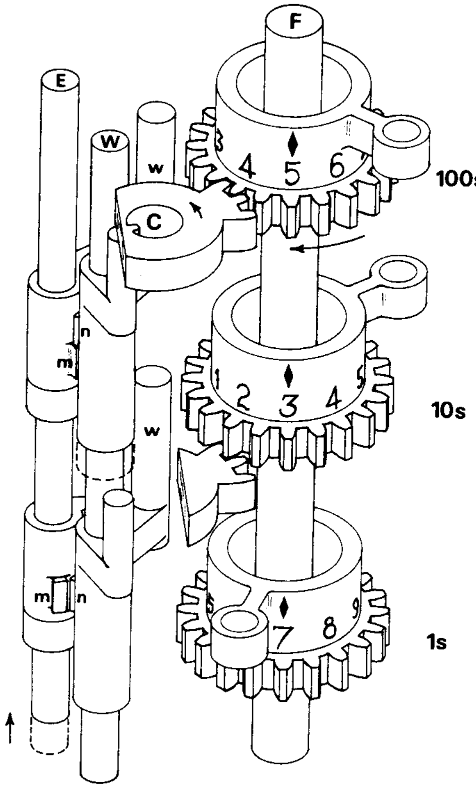


Fig. 6. Carry mechanism. The carry warning is raised with its axis E and lifts the carry sector in the cage above into gear with its figure wheel. In the figure, there is a carry from the 10s into the 100s cage, but none from the units to the 10s.

Based on a plate in *Babbage's Calculating Engines*.[2]

The arm f attached to the carry figure wheel is drilled through with a vertical hole in which is carried a loose slug 'w (Fig. 7). When the figure wheel stands at nine, slug 'w is interposed between the bottom of the rear finger w of the carry piece and the top of the corresponding finger in the cage below. Suppose now that a carry is generated from the cage below. The carry piece will be lifted by the carry warning to indicate the carry. The rear finger w lifts the loose slug 'w and, hence, the carry piece in this cage at the same time to indicate a carry into the next higher cage. The carry-out is thus generated directly, without waiting for the figure wheel to turn past nine to zero as a result of the carry-in. By similar means, a carry can propagate through any number of successive cages without requiring any additional time. The carry sectors will all be put into gear, and the figure wheels will all advance through one digit position simultaneously.

Babbage called carry piece w the *fixed wire* and the loose slug 'w in the arm of the figure wheel the *movable wire*—a *wire* in this context meaning a stiff piece of metal, unlike the modern usage. When a figure wheel stands at nine, he described the movable wire as *completing a chain* with the fixed wires—essentially a logical AND function.

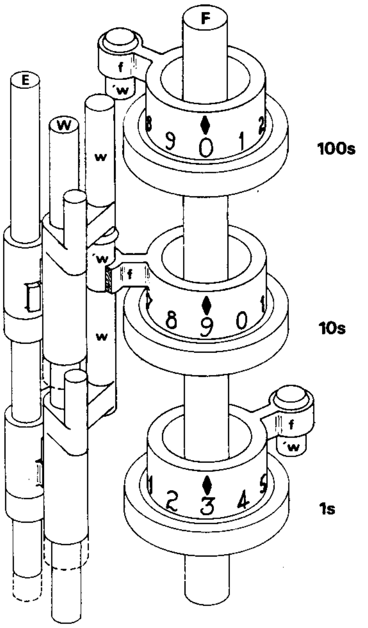


Fig. 7. Anticipating carry. The movable wire 'w carried in the arm of the figure wheel can interpose between the fixed wires w to propagate a carry through several cages. In the figure, a carry is generated in the units cage. Because the 10s figure wheel stands at nine, a carry is propagated to the 100s cage. The 100s figure wheel does not stand at nine and therefore breaks the chain of carry propagation.

Based on a plate in *Babbage's Calculating Engines*.[2]

In practice, the carry mechanism is more complex than shown. For example, the carry warning and fixed wires must be lowered before the figure wheels are moved to assimilate the carry, or the fixed wire will foul the movable wire of any figure wheel that advances from eight to nine. Some mechanism is required, therefore, to keep the carry sector engaged with the figure wheel once it has been raised. Further difficulties raised by subtraction are overcome, in part, by recoding the digits of the carry figure wheels during subtraction so that the movable wires provide borrow propagation by zeros instead of carry propagation by nines. One consequence is that addition and subtraction operations cannot be freely intermixed in the Analytical Engine.

The anticipating carry does not correspond closely with any common modern discrete logic binary carry mechanism. Carry propagate and generate terms are used in each digit position (cage), so the mechanism is unlike a ripple carry, in which a carry-in must be assimilated to form each digit sum before the carry-out is generated. Neither is it a carry look-ahead mechanism, since the carry indication is passed sequentially through the gating (the movable wire) in each digit position. Its speed derives from the fact that the entire column of wires can be lifted in the same time that a figure wheel can be moved through one unit. In effect, the gating for carry propagation provided by the movable wires has a much shorter time delay than movement of the figure wheels. The closest modern analogies are the carry propagation by relay contacts in the Harvard Mark I or by pass transistors in Very Large-Scale Integration design. Both are logically equivalent to the anticipating carry.

Could the Analytical Engine Have Been Built?

In many respects, the anticipating carry mechanism appears to make the most severe technological demands, so here is a convenient place to pause and ask whether the Analytical Engine could have been built with the technology at Babbage's disposal. It is worth noting that in 1838, only one calculating machine had been brought to successful, though small-scale commercial manufacture: the arithmometer of Thomas de Colmar. It provided addition and subtraction to a single accumulator as a direct operation and used a ripple carry propagation. Multiplication and division could be performed by repeated addition and subtraction, but all operations were directly manually controlled.

Consider first the question of machining tolerances in the anticipating carry mechanism. In the worst case, a carry generated in the units cage might have to raise into gear a carry sector in the 40th cage by a chain of 80 fixed and movable wires. A consistent error of only 3/1,000 inch in machining the wires to length would amount to 1/4 inch in the entire column. This tolerance is probably satisfactory in practice, for the carry sectors are held in gear with the figure wheels by tripping a counterweight that could make up the lost motion. Strictly speaking, however, what needs to be controlled is the difference in height between the column of wires and the figure cages, which are, in effect, a column of plates and spacer pieces. We would therefore need a machining tolerance of only 1/1,000 to 2/1,000 inch in the thickness of the plates, the length of the spacers, and the length of the wires. Such tolerances were achievable at the time, as shown by a recent examination of loose parts intended for the Difference Engine. (I wish to thank Michael Wright of the Science Museum for his guidance and assistance in these measurements. His expert knowledge of 19th-century machining is a continuing inspiration.) They have been machined (not hand-finished) consistently with one another to just such an accuracy. It is harder to determine the absolute accuracy of the parts without knowledge of the engineers' working standards, but the fit could be assured by working to mutually consistent gauges. These tolerances would also have had to be maintained as the wires became worn by use.

Babbage realized that the total mass of the wires and carry sectors that had to be lifted through 0.65 inch in 0.15 second for a carry from the units to the 40th cage was around 50 pounds. He proposed to lift a weight during the remainder of the cycle that could then be dropped to provide the force to lift the mechanism and thus relieve the peak load on the main driveshaft. An examination of the drawings shows that this load is all carried by lug m of the carry warning. In Babbage's design, the lug has a cross-section of about 1/10 square inch in the shear plane and, if made of brass, would fail with a load of about 1,800 pounds. A safety factor of at least 20 is thus built into the design.

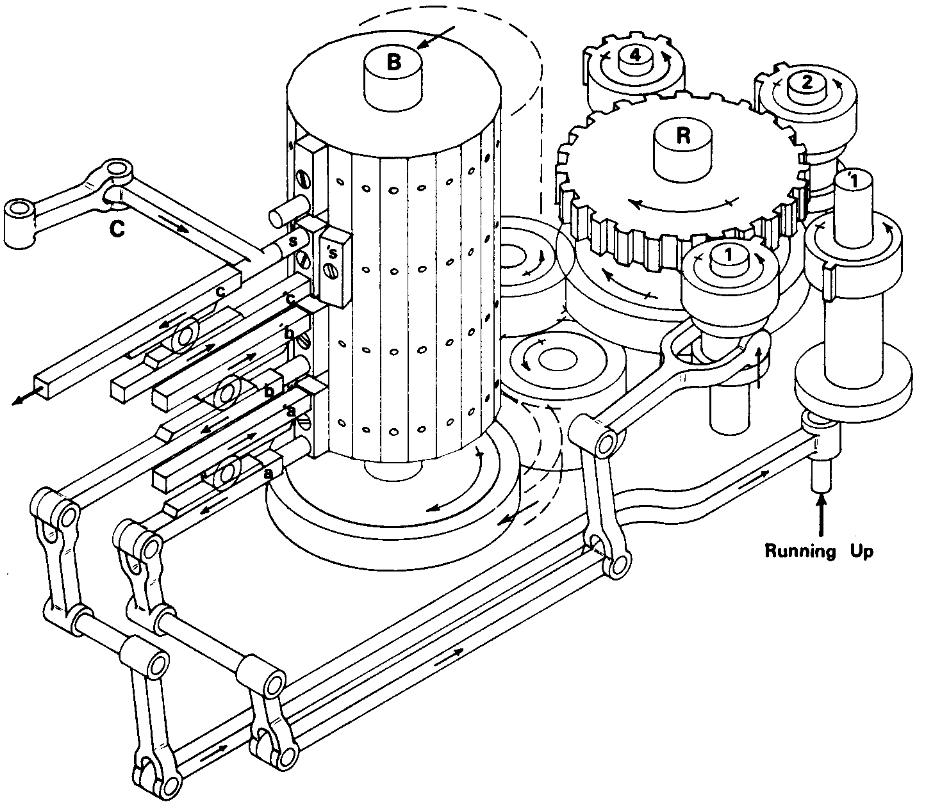


Fig. 8. Barrel and its reducing apparatus. A microprogram word is represented by a vertical row of studs screwed to the barrel. These act on the ends of the control levers when the barrel moves sideways. The *reducing sectors* of one, two, and four teeth advance the barrel over the corresponding number of verticals, and several may act in combination at one time. In the figure, reducing sector 1 is put into gear directly by order of the barrel via control lever a. Reducing sector '1 is put into gear by a *running up* from the carry apparatus if enabled to do so by the barrel and lever b. The effect is a conditional transfer. A *conditional arm* is sensed by control lever c to provide an action conditional on a previous event.

Based on the drawing BAB.[A]35.

Analyses such as these lead me to believe that the Analytical Engine could have been built with the technology at Babbage's disposal, although the work would undoubtedly have been demanding and expensive. It is interesting to speculate whether the Analytical Engine, had it been built, would have had a stimulating effect on the British machine-tool industry, like the Difference Engine had. (Joseph Whitworth, the preeminent machine-tool designer of the mid-19th century, had been earlier employed by Babbage's engineer, Joseph Clement, on the construction of the Difference Engine. Babbage reports that it was commonly said at a later date that "Babbage made Clement, and Clement made Whitworth." Lord Rosse's presidential address to the Royal Society in 1854, as recorded in its proceedings, confirms that Babbage's work had been a major direct stimulus on the British machine-tool industry.)

Microprogram Control

With the mechanisms so far described to store, transfer, step, and add numbers, it should clearly be possible to assemble a computing device of some power. An automatic computing device requires, in addition, some control mechanism by means of which these elements can be coordinated to work together. This was done in the Analytical Engine by a technique that we would now recognize as microprogramming. We will see in subsequent sections something of the versatility with which Babbage applied the technique. For the present, we concentrate on the methods by which Babbage could control:

- components of the machine by information read from a store,
- how a sequence of operations could be effected, and
- how the sequence could be altered in response to conditionally occurring events in the machine.

The detailed action of the Analytical Engine during each step of a complex operation, such as multiplication or division, is determined by a *barrel*, to the surface of which studs can be screwed in any predetermined combination. At the start of each cycle of operation of the mill, the barrel *advances* by moving sideways parallel to its axis, as shown from the dashed lines in Fig. 8, so that the studs bear against the ends of levers. These levers put into gear the mechanisms for the various functions desired in that cycle. In any one cycle, the studs in a single vertical line parallel to the axis of the barrel act to control the machine. The barrel can be thought of as a microprogram store, and a single vertical row of studs can be thought of as a word of that store. Babbage used the word *vertical* in the sense both of a word of the microprogram store and of a combination of micro-operations of the Analytical

Engine caused by the action of that word of the store.

In practice, each control lever consists of a pair, such as a and 'a, geared together so that they can be put positively into one of two positions by the barrel. The appropriate piece of mechanism is therefore put positively in or out of action. The barrel can carry blank studs, such as 's, that act on neither lever. The blank studs do not specify a "don't-care" condition, but are an indication that the existing state of action or inaction should be continued by the mechanism for another cycle.

The control levers do not directly produce actions on the mechanism of the Analytical Engine; instead, each lever connects the mechanism to a source of power. Thus, a control lever whose purpose is to raise an axis will not itself raise it, but will put into gear with the main driveshaft a cam that raises the axis. The control lever therefore does not have to exert any substantial direct force, and the construction of the barrel is greatly simplified.

In practice, a barrel might act on 50, 100, or more control levers and have 50 to 100 verticals. Fig. 8 gives some impression of the complexity of the associated systems of levers, cranks, pivots, and sliders. There was nothing of the ease of backplane wiring for the distribution of control information in the Analytical Engine.

A sequence of micro-operations is effected by arranging for a number of different verticals to act in succession. The movement of the barrel from one vertical to another is produced by the *reducing apparatus* shown on the right in Fig. 8. When the barrel retreats to its unadvanced position, shown as dashed lines in the figure, it is geared up with axis R. If the gear on axis R is moved one or more positions, the barrel will move over a similar number of verticals. In general, the barrel orders its own advance via several of the control levers. Examples of mechanisms for controlling such advances are shown in Fig. 8.

Control lever a acts to lift into gear with the reducing apparatus a sector, 1, with only a single tooth. During the cycle, this sector is caused to rotate, by a mechanism not shown, and in the process advances the reducing apparatus and, hence, the barrel through one position. Similar control levers can put into gear sectors 2 and 4 of two and four teeth, respectively. The rotation of sectors 1, 2, and 4 is so phased that their teeth act at different times. By the barrel acting on a suitable combination of the control levers, a selection of the sectors can be put into gear to move the barrel over from one to seven verticals. Connection between the reducing apparatus and the barrel is by means of reversing gears controlled by a control lever. The barrel can thus order a transfer to another vertical up to seven positions either forward or backward relative to the present one.

In practice, there can be more reducing sectors than are shown here, and the transfer can be over a slightly larger range than the seven verticals in this example. The transfer is always relative to the present vertical, however, and generally over only a portion of the total number of verticals of the barrel. Every vertical orders the transfer to the next vertical in the sequence. If no transfer is specified, the vertical will be repeated indefinitely unless interrupted by a conditional transfer.

Conditional transfers can take place in response to conditional events within the mill of the Analytical Engine. A common source of these is a carry propagation beyond the highest cage of the carry apparatus. Such an event is called a *running up* and, if 10s-complement arithmetic is being performed, signifies a change in sign of an arithmetic result. (There is generally no overflow indication in the Analytical Engine.) A running up can occur in similar circumstances in connection with the specialized counting apparatus of the machine.

Fig. 8 shows, associated with lever b, one mechanism for a conditional transfer. The control lever places a loose slug under sector '1. If a running up occurs later in the cycle, as a result of an addition ordered by other control levers, the slug and, hence, the sector will be raised into gear with the reducing apparatus and cause it to move through one position. If no running up occurs, or if the control lever is not put in, nothing happens. The effect, therefore, is that the normal unconditional transfer ordered by the barrel can conditionally be extended to move over one additional vertical. We finally arrive at one of two different verticals, according to whether the running up has occurred or not. Note that with this mechanism, the s conditional transfer responds to an event taking place later in the same cycle during which the vertical acts.

A conditional response to a preexisting condition is shown by control lever c. The control lever is too short to be acted on directly by the studs on the barrel unless *conditional arm* C has been interposed. Here, the control lever is shown as affecting some part of the mill, but it could as easily act to put a reducing sector into gear. The conditional arm must be in place before the barrel advances, but the response to a conditional arm can be delayed over several cycles by the use of blank studs on the barrel. There is, of course, no reason why both forms of conditional mechanism cannot act during a single cycle of the Analytical Engine, and multiway branches are quite possible.

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 the design of the Analytical Engine. It is a concept of the most profound intellectual importance.

The Timing Cycle of the Analytical Engine

The operation of the barrel and its synchronization with the remainder of the mechanism are well-illustrated by considering the timing of the cycle of operations. Each cycle involves reading a vertical of the barrel and carrying out the functions so ordered. The cycle of operation is counted in terms of units. A unit is the time for a figure wheel to move through a single digit position. Babbage took this unit as about 1/7 second (corresponding to a speed of 10 feet per minute at the periphery of a figure wheel four inches in diameter), although the design of the machine is kinematic in the sense that it could have been run at any lower speed.

A short cycle of 15 units is employed for a simple transfer from one set of figure wheels to another. A long cycle of 20 units is employed when it is necessary to perform an addition involving both a transfer and a carry propagation. The appropriate length of cycle is ordered by the barrel. A simple number transfer takes about 2.4 seconds, and a full addition takes about three seconds.

Fig. 9 is a timing diagram, similar in style to some of Babbage's own notations, and illustrates the events in the basic timing cycles. The extent of an arrow indicates the time in the cycle during which an action takes place; a dashed arrow indicates that the action might occupy only part of the available time interval. The direction of the arrow indicates, relative to a conventional set of axes, the direction of the motion; circular motions are indicated by an arc attached to the arrow's tail. A single arrowhead indicates the basic action; double arrowheads indicate a return to the inactive position (many of these movements have been omitted from the figure for simplicity).

A cycle commences with the advance of the barrel in units one and two to act on the control levers. The barrels retreat into gear with the reducing apparatus in units three and four. The reducing apparatus can move the barrel to a new vertical during the remainder of the cycle. In the short cycle, the available time limits the rotation of the barrel to within 11 verticals of the one that has just acted. A greater transfer is possible in the long cycle, but some of the units will be consumed by provision for conditional transfers as a result of a running up and so forth.

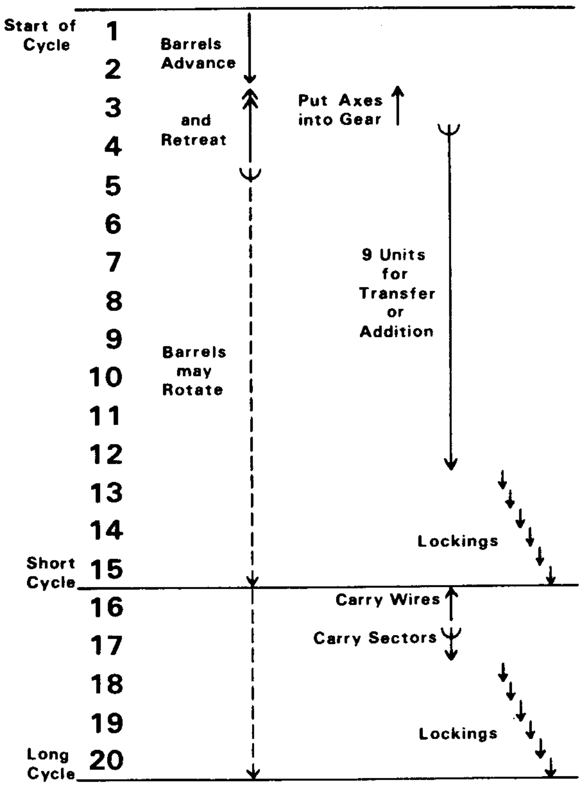


Fig. 9. Timing diagram illustrating the events in the basic timing cycle. This figure is similar in style to Babbage's notation of timing.

Based on the notation BAB.[F]77.

The axes of the mill, transfer pinions, and so forth are put into gear in unit three as soon as the advance of the barrel is complete. The digitwise parallel transfer of numbers then takes place during units four to 12. The lockings are applied during units 13 to 15. A locking is applied generally to every second axis in a train of gearing, and each takes only half a unit, so it is possible for a transfer to extend through a train of up to 12 stages of gearing.

In the case of a long cycle, the carry wires are lifted by the carry warnings during unit 16, and the carry sectors are rotated to perform the carry propagation during unit 17. In general, a sum is formed not only on the figure wheels of the carry apparatus but also on other figure axes of the mill to save the time subsequently required to transfer a sum from the carry figure wheels to the place where it is to be stored. Some of the gearing of the additive transfer will therefore remain in gear during the carry propagation to convey the carry to the accumulating figure wheels. After this transfer, it is again necessary to apply the lockings, during units 18 to 20.

The Functional Organization of the Analytical Engine

We now turn to an examination of how the functional components are organized into the complete Analytical Engine. The discussion is based on Babbage's general plan 25, which he had lithographed and was distributed in his own lifetime. The plan is shown in Fig. 10, and various features of it are highlighted in subsequent figures. Besides showing the mechanical arrangement of the Analytical Engine, the plan serves as a functional block diagram of the design.

The major axes (registers) and transfer paths (data buses) are shown in Fig. 11. On the right is the store comprising a group of figure axes ranged alongside a set of racks, or toothed bars, one in each cage. Each store axis has two sets of figure wheels and can thus store two numbers. The number of axes intended is not clear from the figure (which can extend farther to the right), but we can assume 50 from the layout of the variable cards. The store would thus hold 100 numbers, which is more than adequate for any of Babbage's programs. Transfer gears enable any set of figure wheels to receive a number from the racks or give off a number to the racks. Numbers given off by the store are received by *ingress axis* I of the mill. Numbers to be placed in the store are given off by the *egress axis* "A. The ingress and egress axes act as memory buffer registers.

The axes of the mill are arranged around a set of *central wheels*, one in each cage, to which the figure wheels of the axes can be connected by transfer gears.

These central wheels act as a data bus interconnecting the various registers of the mill. Aside from the ingress and egress axes I and "A, the axes of the mill include the *head* and *tail* axes A and 'A, used as accumulators, and nine *table axes* T_1 to T_9, used in multiplication and division.

Two sets of figure wheels are on each of the mill axes, and associated with each is a set of transfer and long pinions (G, S, L, J) similar to those in Fig. 3 and Fig. 4. The pinions provide for transfers between the two sets of figure wheels and steppings, as we have described. Associated with each of the table axes is an additional axis (such as $'S_2$ near T_2) that connects the top and bottom cages to make the stepping equivalent to a cyclic shift. The head and tail axes A and 'A are interconnected, so that the stepping provides a double-length shift, and a triple-length shift is possible between A, 'A, and "A.

Three sets of anticipating carry apparatus (F, 'F, and "F) are provided. Normally, F and 'F are associated with the head and tail axes A and 'A, respectively, to provide a double-length accumulator for use in multiplication. However, both F and 'F are associated with A in signed addition operations.

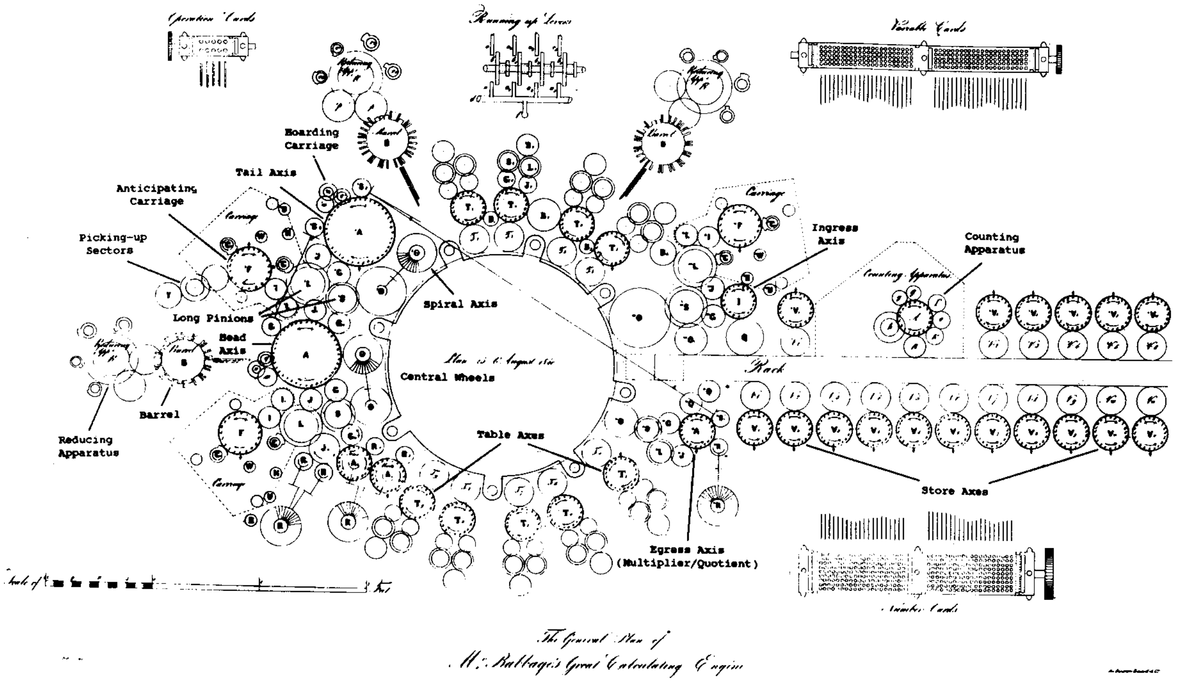


Fig. 10. Plan of the general arrangement of the Analytical Engine. It also serves as a functional block diagram, as explained in the text and in Fig. 11.
From Babbage.[2] The design is detailed by the notations BAB.[F]74-107.

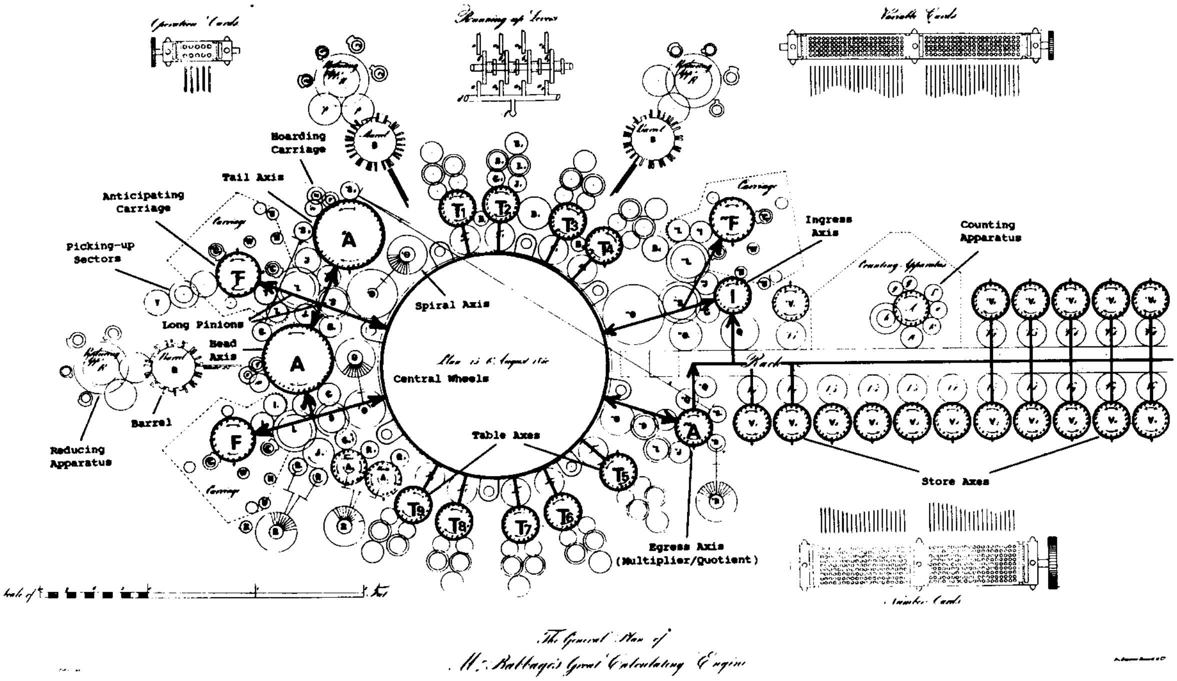


Fig. 11. Major registers and data paths of the Analytical Engine. The store axes are arranged along the racks to the right and communicate with the mill via the *ingress* and *egress* axes, which act as buffer registers. The mill is arranged around the *central wheels*, which act as an internal data bus servicing accumulators A and 'A and table axes T_1 to T_9 used in multiplication and division. F, 'F, and "F are the figure axes of the three sets of anticipating carry apparatus.

After the preliminary stages of a multiplication or division operation, neither the ingress axis I nor the carry apparatus "F is required, and there is no demand for memory transfers. This apparatus can be used to perform simple addition operations concurrently with the execution of the multiplication or division operation in the mill. If the Analytical Engine were used for computing tables, new coefficients could thus be computed in parallel by difference techniques with no additional time required in the mill. Such concurrent computations would use 10s-complement arithmetic and store numbers in this form. (Sign-and-magnitude addition would be possible with just the apparatus shown, but it would be complex and slow.) Unfortunately, I have little additional information at present on these concurrent operations.

Fig. 11 shows three barrels and associated reducing apparatus. If these barrels always move in synchronization, they are logically equivalent to a single larger barrel. The separation will reduce the height of each barrel to more manageable proportions and allow a simpler and more direct connection of the control levers to the apparatus they control—both important practical considerations. Babbage found significant advantages in allowing the barrels to step between their verticals independently of one another, however, as we will see in the discussion of signed addition.

As with so many other facets of the design of the Analytical Engine, there is an unmistakably modern feel about the organization of the store and mill. It is lost to some extent in the following description of the arithmetic operations of the mill, where the characteristics of the sign-and-magnitude decimal arithmetic play a dominant role. The feeling of familiarity will be retained, however, by anyone who compares Babbage's algorithms with those of decimal machines, such as the Harvard Mark I or the ENIAC.

Multiplication

It may seem surprising to discuss the operation of multiplication in the Analytical Engine before discussing addition and subtraction. Signed addition is logically the most complex operation in the machine, however, and the control techniques used to implement it are by far the most sophisticated. This sequence also parallels the order in which Babbage developed the microprogram sequences. Although multiplication uses a substantial amount of apparatus, it is logically quite straightforward.

Multiplication commences by fetching the operands from the store via ingress axis I to head and tail axes A and 'A. The operand with the lesser number of digits (the smaller in absolute magnitude) is used as the multiplier and is transferred to "A. The other operand is used as the multiplicand. The sign of a number is represented by a figure wheel in the 41st cage of every store and figure axis. If the digit stored on the sign figure wheel is even (zero, two, four, six, or eight), the number is taken as positive, otherwise as negative. As the operands are fetched from the store, the signs are stripped and added to give the sign of the product. Before the multiplication commences, the multiples from one to nine of the multiplicand are made by repeated addition and stored on table axes T_1 to T_9.

The multiplication commences from the least significant digit of the multiplier. A digit of the multiplier selects one of the multiples of the multiplicand on the table axes as a partial product. The partial product is given off to the central wheels (as shown in Fig. 12) and added to the double-length product on the head (most significant half) and tail (least significant half) axes using the anticipating carry apparatus F and 'F. The multiplier, on "A, is stepped down to select the next multiplier digit, and the table axes (including the one giving off) are all stepped up to preserve the correct alignment of the partial products with the accumulating sum. Significant digits stepped off the top of the table axes are returned to the bottom by the cyclic nature of the shift. A special apparatus, controlled by the *spiral axes*, controls the transfer pinions linking the central wheels to the head and tail axes to ensure that the correct digits of each partial product are added onto the head and tail axes. The multiplication process is illustrated by the example in Table 1.

TABLE 1
EXAMPLE OF METHOD USED FOR MULTIPLICATION

Multiplier Digit	**Head**	**Tail**	**Central Wheels**	**T_1**	**T_2**	**T_3**
 2	000	000 650	 650|	325|	650|	975|
 3	000 9	650 75	 75|9	25|3	50|6	75|9
 1	010 32	400 5	 5|32	5|32	0|65	5|97
	042	900		|325	|650	|975

Note. In this example, 325 is multiplied by 132. The multiples of the multiplicand are first constructed on the table axes. Each multiplier digit selects a partial product from the table axes and adds it to the total on the head and tail axes. The table axes are cyclically shifted at each step. The central wheels distribute the digits of the selected partial product to the head and tail axes

At the end of the multiplication, the product is stepped (shifted) to align its decimal point with that of the operands and is then given off to the store via egress axis "A. Provision is made for high-precision multilength multiplications to be compounded from single-length products.

The time taken for multiplication depends on the number of digits in the multiplier and the amount of stepping required to align the decimal point of the product. About four minutes would be required in the worst case of 40-digit operands, but about two minutes might be regarded as more typical. Square root was intended to be performed in the Analytical Engine by using an iterative formula employing multiplication. To speed the early steps of such a process, a special form of *approximative multiplication* is provided, in which only the most significant few digits of the multiplier are used.

The development of the multiplication algorithm played an important role in the history of the Analytical Engine, because it led Babbage to the invention of the anticipating carry to speed the addition of the partial products. The complexity of the anticipating carry apparatus led him, in turn, to the clear separation of the functions of the mill and store, a concept that did not clearly emerge in modern computers until the work of John von Neumann. (In Babbage's Difference Engine, each storage axis had associated with it a ripple carry mechanism, so that it became an accumulator in a similar manner to the organization of the Harvard Mark I and the ENIAC.)

Division

Division is a little more complex than multiplication but follows generally similar principles. The multiples from one to nine of the divisor are first made by repeated addition and stored on the table axes. In the process, the two most significant digits of each multiple are stored on special figure wheels on each table axis for later use in determining trial quotient digits.

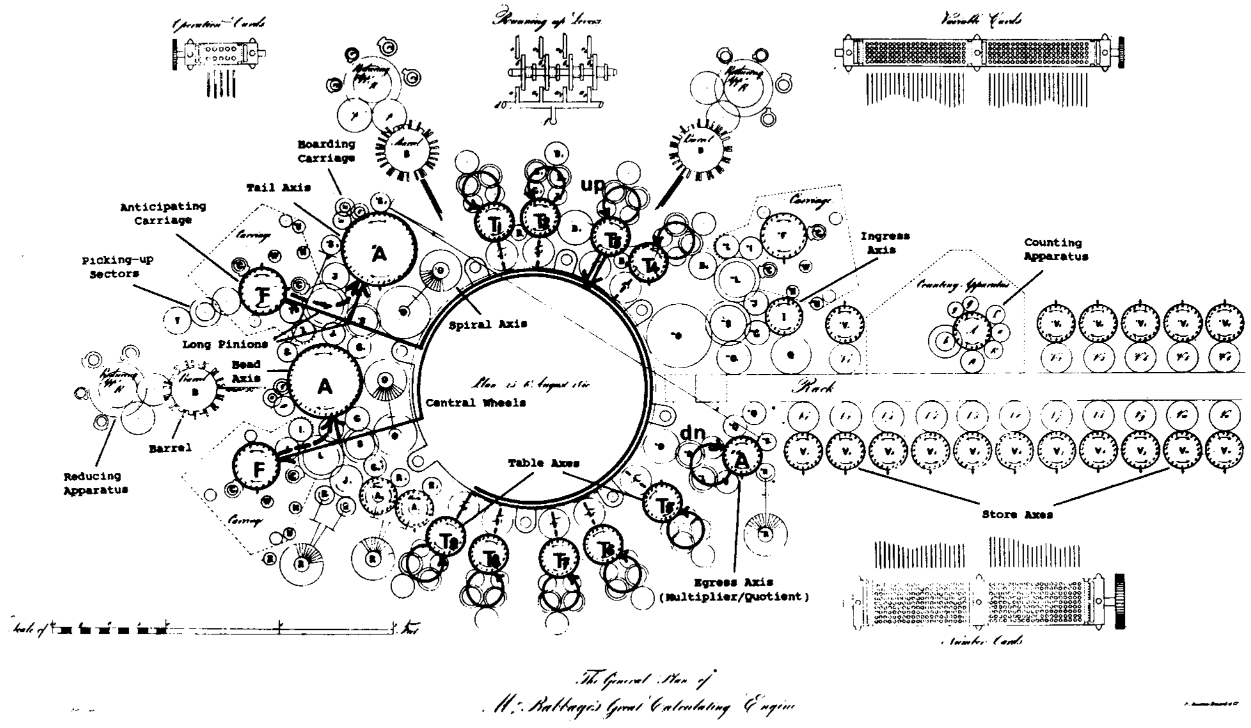


Fig. 12. Actions during the main loop of multiplication. A partial product, selected by a digit of the multiplier, is given off by a table axis (in this case T3) and added to the product on the head and tail axes A and 'A. Simultaneously, the multiplier is stepped down on "A to select the next partial product, and the table axes are all stepped up to maintain the correct alignment with the product.

The dividend is stepped up or down on the head and tail axes to align its most significant digit with the most significant digit of the divisor. Stepping is done, where possible, concurrently with making the table of multiples of the divisor. (The decision on how to step the dividend involves a six-way branch in a single microinstruction, which gives some indication of the flexibility of the reducing apparatus associated with the barrels.)

In each step of the division, the two most significant digits of the remainder are compared with the two most significant digits of each multiple of the divisor. This comparison yields a correct estimate of the next quotient digit except in the case in which the two most significant digits of the remainder equal those of one of the multiples. In this case, the trial quotient digit might possibly be one too large. The multiple of the divisor indicated by the trial quotient digit is selected from the table axes and subtracted from the remainder. If the new remainder is negative, the trial quotient digit was one too large, and we add back the divisor to generate the correct remainder. The quotient is assembled digit by digit on "A and is finally given off back to the store.

Division is somewhat slower than multiplication, because the selection and subtraction steps are not overlapped. Four minutes can be taken as a representative division time. As with multiplication, an approximative form of division, which generates only the first few significant digits of the quotient, is provided for use with iterative formulas.

The development of division also played an important role in the history of the Analytical Engine. An examination of the tentative processes involved in the determination of the quotient digits first led Babbage to a clear understanding of conditional processes conditioned by events occurring during the calculation. As we saw, these conditional processes are clearly embodied in the reducing apparatus of the barrels.

Babbage continued to exert considerable energy to improving the division process. Fig. 10 shows a later design than that just described. Table making is shortened by first stepping the divisor to form 10 times it and then subtracting the divisor to form the ninth, eighth, etc. multiples concurrently with the formation of the smaller multiples by addition. The same process can be applied in multiplication, of course. Babbage also found a method of overlapping a trial subtraction in division with the selection of the next quotient digit and of avoiding the difficulties created by choosing a trial quotient digit too large.

Signed Addition and Subtraction

The use of a sign-and-magnitude representation for numbers is particularly convenient in multiplication and division, because it makes possible a very simple determination of the sign of the result. Addition and subtraction, however, are both complicated by the use of a sign-and-magnitude representation. The Analytical Engine uses a 10s-complement representation internally for addition and subtraction, and conversions to and from this representation are performed as operands are read and as results are stored.

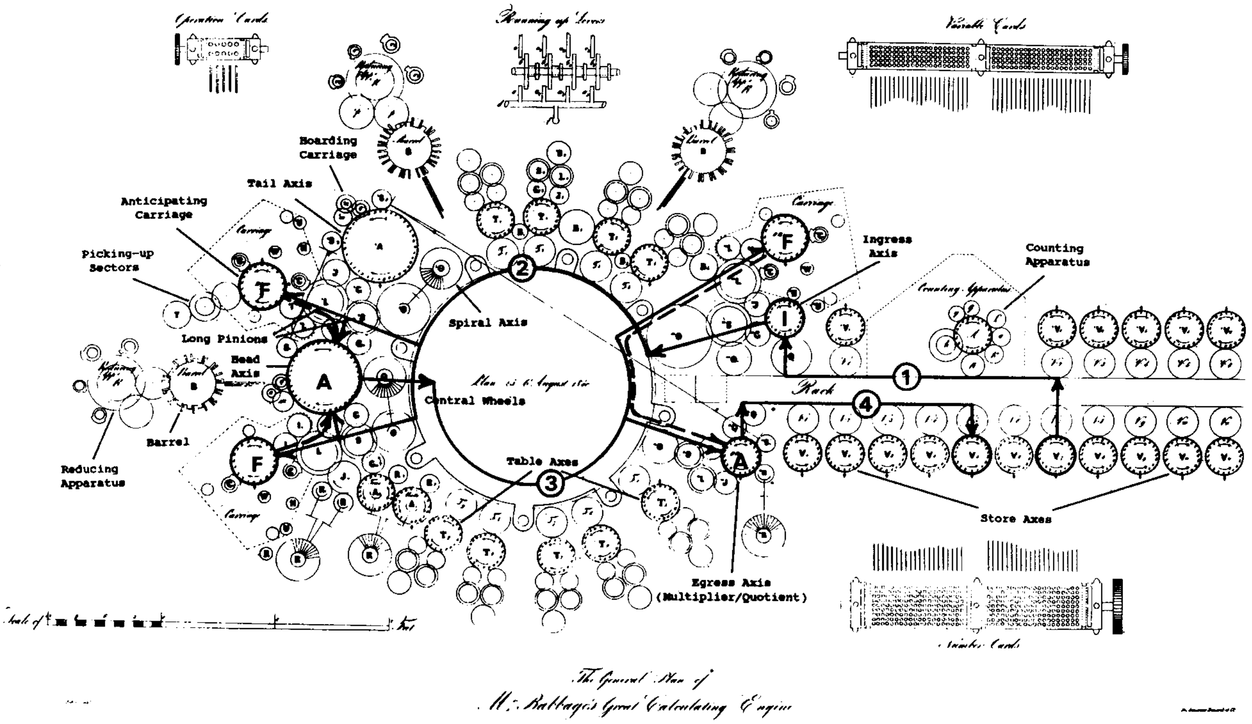


Fig. 13. Stages in the pipeline of signed addition. An operand is fetched from the store to the ingress axis I, added to or subtracted from the total on A, the result converted into sign-and-magnitude representation on egress axis "A, and returned to store, in four separate steps that can be overlapped with one another for different operands.

Another complication was introduced by the fact that the Analytical Engine did not provide residual storage between operations. Every result was returned to the store and, if required as an operand, was again fetched from the store for the next operation. The head and tail axes were not used, as are the registers in a modern computer, to hold the result of one operation as an operand for the next. A string of addition operations would therefore be quite slow. Babbage avoided this difficulty by making a string of additions and subtractions the basic additive operation. If desired, only the final sum would be returned to store, but any or all of the partial sums could also be stored. All the partial sums of a series could be formed and stored as a new series by a single operation. To achieve this end, addition is pipelined so that several operands are at different stages of processing at one instant.

In the first step of an addition (Fig. 13), an operand is transferred from the store via the racks to ingress axis I. Babbage associated two signs with each operand. The *algebraic sign* is the sign associated with a variable in a mathematical formula and is specified by an addition or subtraction being ordered by the cards. The *accidental sign* is the sign that the number is found to have, as a result of some previous operation, when it is fetched from the store. The algebraic and accidental signs are combined when the operand is fetched to the ingress axis and used to determine whether the magnitude of the number must be added to or subtracted from the total.

In step two of an addition, the operand is transferred from ingress axis I, via the central wheels, and added to or subtracted from head axis A. A technical difficulty arises because addition and subtraction cannot be mixed in the anticipating carry apparatus. To avoid this difficulty, both sets of carry apparatus F and 'F are geared to the figure wheels of A in such a way that the nines of the figure wheels of 'F and A correspond to the zeros of F. The carry apparatus 'F is used when adding an operand, and the wires provide carry propagation by nines. Carry apparatus F is used when subtracting, and the wires provide borrow propagation by zeros. The sign of the sum is maintained by the running up acting on a sign figure wheel in a 41st cage. An even digit on this sign wheel signifies a positive number; an odd digit signifies a negative number. There is no provision for detection of arithmetic overflow.

If a partial sum is to be stored, it is transferred, in a third step, from head axis A, via the central wheels, to egress axis "A. If the sum is negative, it is transferred in the subtractive sense, and carry apparatus "F is used to convert it to the correct sign-and-magnitude form. The fourth step of addition is to transfer the sum from egress axis "A, via the racks, into the store.

The interaction of the four steps of the addition pipeline is shown in Fig. 14 and Fig. 15. These figures, and Fig. 16, are quite unlike any of Babbage's notations at this time. (Deciphering the details of the addition pipeline from the original notations was a most difficult and demanding task.)

Fig. 14 shows a simple case in which five quantities are summed and only their total is given off to the store. The operation forms

$$+ n_1 + n_2 + n_3 - n_4 - n_5$$

where

$$n_1 \ldots n_5 = +7, -8, +5, +6, -3.$$

(The signs + + + – – that occur in the formula are the algebraic signs; the signs + – + + – associated with the values of the operands are the accidental signs.) This operation would take a total of about 28 seconds. Note that as the first operand (+7) is added from the ingress axis to the head axis, the second operand (–8) is fetched from the store to the other set of figure wheels of the ingress axis in the same cycle. Similarly, the fetching of each of the other operands is overlapped with an addition to or subtraction from the head axis.

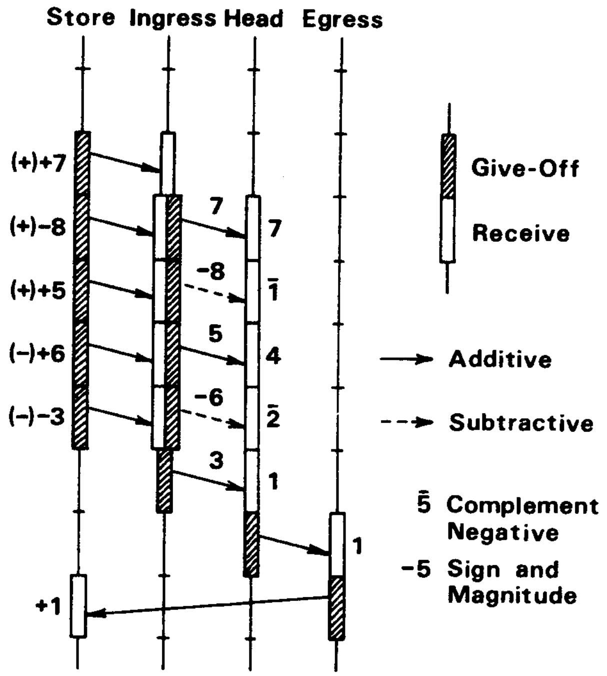


Fig. 14. Overlapping of functions in the pipeline for signed addition. Five operands are accumulated, and only the result is returned to the store.

Fig. 15 shows the same summation of five quantities, but in this instance, every partial sum is given off back to the store. The operation would take about 40 seconds. In this case, we note that the second operand (–8) is fetched from the store to the ingress axis in the same cycle as the first partial sum (7) is transferred from the head axis to the egress axis. The second operand is, in the next cycle, added into the total at the same time the first partial sum is given off to the store. Similar overlapped operations are performed in the remaining cycles.

The signed addition operation is controlled by three barrels that control the actions of the ingress, head, and egress axes, respectively, and the associated apparatus in each case. These axes are the interaction points of the four steps of the addition pipeline.

Babbage made the three barrels sequence through their verticals independently of one another. To understand his reasons, consider first the simpler example in Fig. 14. The second operand (–8) is fetched to the ingress axis in the same cycle as the first operand (+7) is added into the total and hence must be received on the second set of figure wheels of the ingress axis. This second operand is subtracted from the sum in the next cycle, so that the third operand, which is fetched from the store in the same cycle, must be received on the first set of figure wheels of the ingress axis. In general, the operands will be received from the store alternately onto the two sets of figure wheels. Which set of figure wheels is receiving must be remembered by the control apparatus, so that the transfer pinions etc. can be set appropriately. The control apparatus does this most easily and naturally by having two sets of states in the microprogram—that is, two sets of verticals on the barrel.

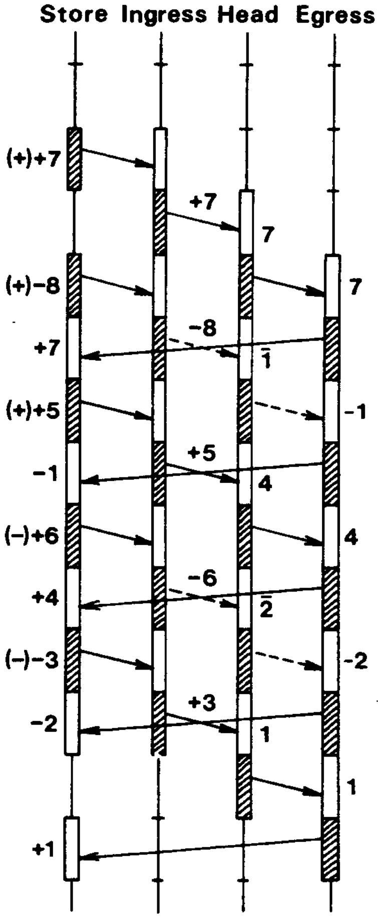


Fig. 15. Overlapping of functions in the pipeline for signed addition. The same five operands are accumulated as in Fig. 14, but each of the five partial sums is returned to store.

A number of similar conditions need to be remembered by the control mechanism. The figure wheels of A are alternated with every partial sum given off to the store. Addition or subtraction of individual operands requires different usages of the carry apparatus F and 'F. A negative partial sum given off requires the use of carry apparatus "F, which a positive partial sum does not. Since these four conditions are independent of one another, they would require a total of 16 complete sets of states in the microprogram or sets of verticals on the barrels. By having the barrels sequence independently of one another, the barrels associated with the ingress and egress axes require only two sets of verticals each, and the barrel associated with the head axis needs only four sets of verticals.

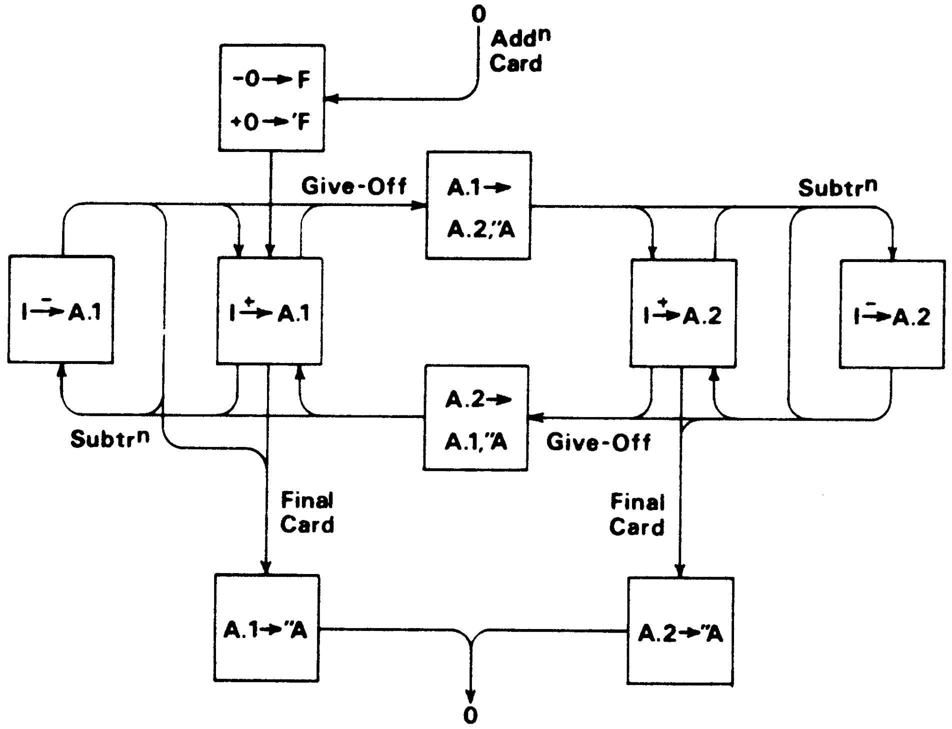


Fig. 16. A state transition diagram for one of the barrels in signed addition. The transitions are between the verticals of the barrel controlling the accumulator A and the anticipating carries F and 'F in signed addition (see Fig. 13). The transitions between the states, shown by the arrows, are generally multiway conditional branches.

By way of example, Fig. 16 shows a modern state-transition diagram of the arrangement of the barrel controlling head axis A and carry apparatus F and 'F. (Babbage adopted similar diagrams only after 1840.) Only nine verticals on the barrel are required, but two-, three-, and four-way conditional branches occur between states. The barrels for the ingress and egress axes are of a similar complexity, and all exhibit a beautifully symmetric disposition of the verticals on the barrel.

Programming the Analytical Engine

Now we turn to an examination of the user-level programming of the Analytical Engine. It should be remarked at the outset that user programming was the least well-developed aspect of the design, and Babbage's work in this regard (at least throughout the design of the Analytical Engine discussed here and until 1840) shows some serious deficiencies. In assessing Babbage's understanding of programming concepts, we should, I think, bear in mind the considerable sophistication of his microprogramming as described in the preceding sections.

Babbage used punched cards to program the Analytical Engine. These cards were strung together by narrow ribbons, after the manner of cards for the Jacquard loom, into strings that are logically more closely analogous to paper tape than to modern punched cards. Therefore, it is possible to move both forward and backward through the strings of cards. In the earliest designs of the Analytical Engine, a barrel, similar to the microprogram barrels, was used in place of the punched cards, and the existence of a hierarchy of levels of control was strikingly evident. Punched cards were substituted to overcome the evident inflexibility of the barrel. Babbage realized, however, that a program on punched cards could be of unlimited extent and that by giving the Analytical Engine the power to read and punch numbers on other cards, he had created a universal calculating machine of effectively unlimited capacity. A deficiency of resources in the machine could be overcome by the expenditure of greater execution time.

In one respect, the arrangement of Babbage's cards appears very strange to a modern reader. An operation to be performed, such as a multiplication, and the variables in the store to be used as operands and to store the result were not only specified on separate cards; these cards were in separate strings that were able to be sequenced independently of one another. Babbage called the two strings of cards the *operation cards* and the *variable cards*. Separate counting apparatuses were associated with the strings of operation and variable cards, and the cards could be ordered to move forward and reverse (and hence to loop) independently of one another by an amount controlled by the counting apparatus.

Babbage had good reasons to back his decision to separate the operation and variable cards. His earlier mathematical work had led him to a serious study of mathematical notation,[6] and he emphasized the distinction in kind between the symbols for operations and the symbols for the quantities on which they acted. Babbage saw the operation and variable cards as representative of these two classes of symbols and therefore saw their separation as entailed by his general considerations of mathematical notation. This opinion may have been reinforced by the observation that cycles in the operation cards were easily found, while cycles in the variable cards were very hard to find. In modern practice, we would make a distinction between the name of a variable (which is what a variable card really is) and the value it contains, rather than between the operation and the name of the variable. I am not convinced that Babbage had clearly resolved even the representational difficulties that his separation of operation and variable cards implies (for example, with respect to giving off partial sums in signed addition), let alone the broader programming implications.

Some two dozen programs for the Analytical Engine exist that are dated between 1837 and 1840. Strictly speaking, they are not programs in the modern sense, but are walk-throughs of particular execution instances of programs. In consequence, the mechanism by which the sequencing of operations is obtained is obscure. Indeed, the "user instruction set" of the Analytical Engine seems nowhere to be clearly stated.

One group of programs deals with the tabulation of polynomials by difference techniques. These programs appear not to use the signed addition operation but just the 10s-complement addition provided by ingress axis I and carry apparatus "F. The Analytical Engine could certainly perform the calculations for which the Difference Engine was intended, but to greater precision and using polynomials of higher degree.

A second group of programs deals with the tabulation of iterative formulas of varying complexity. Frequently, these programs involve the tabulation of a polynomial by difference techniques as a subfunction. The most complex program of this style is that for the Bernoulli numbers included in Lovelace's notes to Menabrea's paper on the Analytical Engine. This program requires a nested-loop structure with a varying range for the inner loop.

A third group of programs deals with the solution of simultaneous equations by variants of Gaussian elimination. Part of the group explores various methods of doing the row reductions, with a view to minimizing the computing time. The remainder of the programs explores ways in which the coefficients of the equations can be arranged in the store. Babbage is clearly seeking simple looping structures for the operation and variable cards. The cycle of operation cards is easily found; it is of similar complexity to that required for the Bernoulli-numbers program. In the last of these programs, the variable cards are listed. Beside groups of them are notations, such as: "like D," "somewhat like B," and "not identical with 'C." At the end, the notation "Not similar" signifies that Babbage had not found the loop structure he sought.

With hindsight, we can note that in the Analytical Engine (at least until 1840), Babbage did not possess the variable-address concept; that is, there was no mechanism by which the machine could, as a result of a calculation, specify a particular variable in the store to be used as the operand for an instruction. Such a mechanism, used as an array or vector index as in a modern computer, would have provided an immediate solution to the difficulties in the Gaussian-elimination program. Babbage's store has no structure accessible to the programmer; the variable cards simply nominate specific store axes in total isolation from one another.

One program in which Babbage comes to grips with structuring his data space forms the product of two polynomials by accumulating the cross-products of the coefficients to give the coefficients of the product polynomial. For this program, the Analytical Engine is provided with two mechanisms for reading number cards, each with a separate sequencing mechanism. One set of coefficients is placed on each reading mechanism; these are moved independently to select the operands for the cross-products. This artifice is scarcely a general solution to the variable-address difficulty, although the program in this instance was structured well enough to allow the generation of any number of coefficients of the product polynomial without program change.

To put Babbage's difficulty into perspective, it is instructive to remark that neither the Harvard Mark I nor the ENIAC possessed the variable-address concept, and both would have suffered the same difficulty as the Analytical Engine in performing Gaussian elimination. Both did, however, possess table-lookup mechanisms that could have been used in programs analogous to Babbage's for forming the product of two polynomials. The first workable solution came with von Neumann's proposal for the EDVAC, in which the program was in the same memory as the data, and a computed address could be substituted in the address field of any instruction. In von Neumann's hands, the stored-program concept seems to have amounted to nothing more than a particular solution (and a nasty one at that) to the variable-address problem. The more general conception made possible in a stored-program machine of a program-building instruction sequence, as in an assembler or compiler, did not come until later. It is evident that the difficulties that troubled Babbage also troubled the 20th-century inventors of computers.

On the User Instruction Set

The reader may have detected a certain hesitation in the preceding section. It is difficult to write with authority on the programming and user-level facilities of the Analytical Engine. Most of Babbage's papers deal with the mechanism of the component parts of the machine and, hence, with what can be termed the microprogramming level; only in rare instances do the papers appear to limit the facilities that might have been provided at the programming level.

Let me take one extreme example. Fig. 10 shows a *counting apparatus* connected to the store racks. It consisted of several sets of two- and three-digit counters mounted above one another on a single set of axes. Among many other purposes, some of the counting apparatus was used for calculating the stepping required to position the decimal point correctly in the result of multiplication and division. For this purpose, it is clearly necessary to know where the decimal point is supposed to be located. In Babbage's 1837 paper,[8] the decimal point is fixed for the duration of any one program and specified by a single large hand wheel. By mid-1838, it is manipulated in decimal form and is apparently obtained from special figure wheels in the mill via the store racks. It could have been fetched from the store and could indeed have been fetched along with each operand read from the store with only trivial changes to the mechanism. Nothing in the mechanism would have precluded Babbage's developing a floating-point number system. I am confident that he did not do so (at least until 1840), because, of course, it would have altered the microprograms for multiplication, division, and (especially) signed addition.

As a more modest example, could the "fixed" decimal-point indication have been altered in the store somewhere to achieve a dynamic rescaling of problems? I simply do not know, although with 40-digit numbers, there cannot have been much demand for it. Could the limited-precision forms of multiplication and division, intended for iterative approximations to square root etc., have been generally used for low-precision calculations in the midst of high-precision ones? This is an eminently practical suggestion with significant speed implications, but, again, I simply do not know enough of how this facility was to be used by the programmer to do more than guess at the answer.

A more modern example may help to clarify the idea that a machine can reach a threshold level of complexity (or richness), after which significant functional extensions can be achieved with negligible additions to the mechanism. The ENIAC was not designed as a stored-program machine. Only relatively minor additions were necessary, however, to simulate a stored-program machine effectively with instructions taken from a function-table unit.

Conclusion

Babbage's Analytical Engine is remarkably sophisticated, far more so than has previously been supposed, as is most evident on examination of the design at the level of the microprogramming of the operations of multiplication, division, and signed addition.

The descriptions of the Analytical Engine provided in this paper are introductory in two senses. First, the paper is based on a detailed analysis of less than 10 percent of the material in the Science Museum collection and only a cursory examination of the remainder. Doubtless, further detailed examination will reveal the design of the Analytical Engine to have been even richer than I have suggested, but such an examination will take several years. Second, this paper is introductory in the sense that it provides only a greatly simplified description of the designs on which it is based.

The study of Babbage's Analytical Engine provides a rare opportunity in the history of technology. The design of the Analytical Engine was carried to a considerable state of perfection, but the design was effectively completely independent of modern developments. Babbage's papers were not subjected to any detailed examination until 1969 and, hence, cannot have influenced the design of modern computers in more than the most superficial manner.

In this circumstance, it is remarkable that Babbage's design and many details of it seem so thoroughly modern, especially in view of the enormous differences in the technology used. Indeed, I am bothered that the Analytical Engine is too much like a modern computer. Do we infer that a computer can only be built in fundamentally one sort of way? Or have we allowed ourselves to be backed into a corner in using only one computational style? Perhaps, in the long run, questions such as these will prove to be important fundamental issues that the study of the history of computing should discuss.

Acknowledgments

The work of studying Babbage's original papers and deciphering the design of the Analytical Engine was undertaken while I was on sabbatical leave from the University of Sydney and a Visiting Research Fellow at the Science Museum, London. I wish to thank the Science Museum—its director and staff—for the hospitality extended to me and for continuing encouragement in this study. The work is now supported by a grant from the Australian Research Grants Committee, which is gratefully acknowledged.

References and Notes

I have not provided detailed references to the collection of Babbage papers at the Science Museum, London. They did not seem appropriate in an introductory paper and would, in any case, have been difficult to provide, because the designs presented here are considerable simplifications of the originals; many details and some salient points have been glossed over. A detailed catalog of the Science Museum papers and a technical history of all of Babbage's designs are currently in preparation. The list of references is simply a first-level guide to the principal sources of the one design discussed in this paper. (Some items in the figures are referenced by the catalog numbers that have been assigned to them.)

References

[1] C. Babbage, "On the Mathematical Powers of the Calculating Engine, 1837," B. Randell, ed., *The Origins of Digital Computers*. Berlin: Springer-Verlag, 1973.

[2] H.P. Babbage, *Babbage's Calculating Engines*. London: E. and F.N. Spon, 1889. (The Menabrea/Lovelace paper appears in *Annals*, vol. 4, no. 3, July 1982.)

[3] A.G. Bromley, "The Mechanism of Charles Babbage's Analytical Engine circa 1838," Univ. of Sydney, Basser Department of Computer Science Technical Report no. 166, 39 pp., 1980.

[4] A.G. Bromley, "The Functional Algorithms and Control of Charles Babbage's Analytical Engine circa 1838," Univ. of Sydney, Basser Department of Computer Science Technical Report no. 167, 41 pp., 1980.

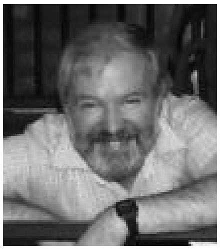
[5] B. Collier, "The Little Engines That Could've: The Calculating Machines of Charles Babbage," PhD thesis, Harvard Univ., 1970.

[6] J.M. Dubbey, *The Mathematical Work of Charles Babbage*. Cambridge, England: Cambridge Univ. Press, 1978.

[7] A. Hyman, *Charles Babbage: A Biography*. Oxford, England: Oxford Univ. Press, 1982.

[8] B. Randell, ed., *The Origins of Digital Computers*. Berlin: Springer-Verlag, 1973.

[9] M.V. Wilkes, "Babbage as a Computer Pioneer," *Proc. Babbage Memorial Meeting*, British Computer Society, London, 1971. (Also in *Historia Mathematica*, 1977, pp. 415-440.)



Allan G. Bromley received a BSc in physics in 1967 and a PhD in theoretical astrophysics in 1971 from the University of Sydney. He joined the university's Basser Department of Computer Science in 1972 and is now an associate professor. In 1979, on leave at the Science Museum, London, he first became seriously interested in the work of Charles Babbage and has since also studied mechanical analog computing devices and the intellectual threads that led to the emergence of digital computers in the 1940s.

The author can be contacted at
Basser Department of Computer Science
Madsen Building F09
University of Sydney
Sydney, NSW 2006, Australia
e-mail: allan@cs.usyd.edu.au