

Chapter 12

Foundations (Boole and Babbage)

Key Topics

Boolean Algebra
Switching Circuits
Difference Engine
Analytic Engine
Lady Ada Lovelace

12.1 Introduction

This chapter considers the work of George Boole and Charles Babbage who are considered grandfathers of computing. George Boole was a nineteenth-century English mathematician who made contributions to logic, probability theory and the differential and integral calculus. His calculus of logic (Boolean logic) acts as the foundation of all modern digital computers.

Charles Babbage was a nineteenth-century scientist and inventor who did pioneering work on calculating machines. He invented the difference engine (a sophisticated calculator that could be used for the production of mathematical tables), and he also designed the analytic engine (the world's first mechanical computer). The design of the analytic engine included a processor, memory and a way to input information and output results. However, it was never built in Babbage's lifetime.

Babbage intended that the program be stored on read-only memory using punch cards, and that input and output for the analytic engine be carried out using punch cards. He intended that the machine would be able to store numbers and intermediate results in memory where they could then be processed. This machine would have employed features subsequently used in modern computers such as

sequential control, branching and looping. He even intended that the machine would be capable of parallel processing where several calculations could be performed at once. It would have been the first mechanical device to be Turing complete.

Lady Ada Lovelace became familiar with Babbage's ideas on the analytic engine at a dinner party, and she was fascinated by the idea of such a machine. She wrote what is considered the first computer program and may be considered the world's first computer programmer.

12.2 George Boole

George Boole was born in Lincoln, England, in 1815. His father (a cobbler who was interested in mathematics and optical instruments) taught him mathematics and showed him how to make optical instruments. Boole inherited his father's interest in knowledge and was self-taught in mathematics and Greek. He was taught Latin by a tutor. Boole taught in various schools near Lincoln and developed his mathematical knowledge by working his way through Newton's *Principia*, as well as applying himself to the work of mathematicians such as Laplace and Lagrange (Fig. 12.1).

He published regular papers from his early twenties onwards, and these included contributions to probability theory, differential equations and finite differences. He developed Boolean algebra which is the foundation for modern computing, and he is considered (along with Babbage) to be one of the grandfathers of computing. His work was theoretical; he was not an engineer and never actually built a computer or calculating machine. However, Boole's symbolic logic provided the perfect mathematical model for switching theory and for the design of digital circuits. Claude Shannon recognised its applicability and applied it successfully to switching circuits.

Boole was interested in formulating a calculus of reasoning, and he published a pamphlet titled 'Mathematical Analysis of Logic' in 1847 [Boo:48]. This article developed novel ideas on a logical method, and he argued that logic should be considered as a separate branch of mathematics, rather than being considered

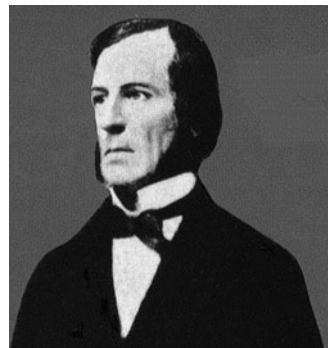


Fig. 12.1 George Boole

a part of philosophy. Boole argued that there are mathematical laws to express the operation of reasoning in the human mind, and he showed how Aristotle's syllogistic logic could be rendered as algebraic equations. He corresponded regularly on logic with the British mathematician, Augustus De Morgan¹ on logic.

Boole had no formal university qualification, and he had difficulty in obtaining a university position. However, his publications were recognised as excellent,² and he was awarded the position as the first professor of mathematics at the newly founded Queens College Cork,³ Ireland, in 1849.

Boole's influential paper on a calculus of logic introduced two quantities 0 and 1. He used the quantity 1 to represent the universe of thinkable objects (i.e. the universal set), with the quantity 0 representing the absence of any objects (i.e. the empty set). He then employed symbols such as x , y , z , etc. to represent collections or classes of objects given by the meaning attached to adjectives and nouns. Next, he introduced three operators (+, − and \times) that combined classes of objects.

For example, the expression xy (i.e. x multiplied by y or $x \times y$) combines the two classes x , y to form the new class xy (i.e. the class whose objects satisfy the two meanings represented by the classes x and y). Similarly, the expression $x + y$ combines the two classes x , y to form the new class $x + y$ (that satisfies either the meaning represented by class x or by class y). The expression $x - y$ combines the two classes x , y to form the new class $x - y$. This represents the class that satisfies the meaning represented by class x but not class y . The expression $(1 - x)$ represents objects that do not have the attribute that represents class x .

Thus, if $x = \text{black}$ and $y = \text{sheep}$, then xy represents the class of black sheep. Similarly, $(1 - x)$ would represent the class obtained by the operation of selecting all things in the world except black things, $x(1 - y)$ represents the class of all things that are black but not sheep and $(1 - x)(1 - y)$ would give us all things that are neither sheep nor black.

He showed that these symbols obeyed a rich collection of algebraic laws and could be added, multiplied, etc. in a manner that is similar to real numbers. He showed how these symbols could be used to reduce propositions to equations, and algebraic rules could be used to solve the equations. The algebraic rules satisfied by his system included:

-
- | | |
|--------------------------------|---------------------|
| 1. $x + 0 = x$ | (Additive identity) |
| 2. $x + (y + z) = (x + y) + z$ | (Associativity) |
| 3. $x + y = y + x$ | (Commutativity) |

¹ De Morgan was a nineteenth-century British mathematician based at University College London. De Morgan's laws in set theory and logic state that $(A \cap B)^c = A^c \cap B^c$ and $\neg(A \vee B) = \neg A \wedge \neg B$.

² Boole was awarded the Royal Medal from the Royal Society of London in 1844 in recognition of his publications. The Irish Mathematician, Sir Rowan Hamilton (who invented Quaternions), was another famous recipient of this prize.

³ Queens College Cork is now called University College Cork (UCC) and has about 18,000 students. It is located in Cork city in the south of Ireland.

- | | | |
|-----|----------------------|---------------------------|
| 4. | $x + (1 - x) = 1$ | (Not operator) |
| 5. | $x 1 = x$ | (Multiplicative identity) |
| 6. | $x 0 = 0$ | |
| 7. | $x + 1 = 1$ | |
| 8. | $xy = yx$ | (Commutativity) |
| 9. | $x(yz) = (xy)z$ | (Associativity) |
| 10. | $x(y + z) = xy + xz$ | (Distributive) |
| 11. | $x(y - z) = xy - xz$ | (Distributive) |
| 12. | $x^2 = x$ | (Idempotent) |
| 13. | $x^n = x$ | |
-

These operations are similar to the modern laws of set theory with the set union operation represented by '+', and the set intersection operation is represented by multiplication. The universal set is represented by '1' and the empty by '0'. The associative and distributive laws hold. Finally, the set complement operation is given by $(1 - x)$.

Boole applied the symbols to encode propositions of Aristotle's Syllogistic Logic, and he showed how the syllogisms could be reduced to equations. This allowed conclusions to be derived from premises by eliminating the middle term in the syllogism.

Boole refined his ideas on logic further in his book *An Investigation of the Laws of Thought* [Boo:58] published in 1854. This book aimed to identify the fundamental laws underlying reasoning in the human mind and to give expression to these laws in the symbolic language of a calculus. He considered the equation $x^2 = x$ to be a fundamental laws of thought. It allows the principle of contradiction to be expressed (i.e. for an entity to possess an attribute and at the same time not to possess it):

$$\begin{aligned}
 &x^2 = x \\
 \Rightarrow &x - x^2 = 0 \\
 \Rightarrow &x(1 - x) = 0
 \end{aligned}$$

For example, if x represents the class of horses then $(1 - x)$ represents the class of 'not horses'. The product of two classes represents a class whose members are common to both classes. Hence, $x(1 - x)$ represents the class whose members are at once both horses and 'not horses', and the equation $x(1 - x) = 0$ expresses that fact that there is no such class whose members are both horses and 'not horses'. Another words, it is the empty set.

Boole made contributions to other areas in mathematics including differential equations and on finite differences.⁴ He also contributed to the development of probability theory.

⁴Finite differences are a numerical method used in solving differential equations.

He married Mary Everest in 1855, and they lived in Lichfield Cottage in Ballintemple, Cork. She was a niece of the surveyor of India, Sir George Everest, after whom the world's highest mountain is named. They had five daughters, and his daughter Ethel Lilian was the author of the novel *The Gadfly*.⁵ Boole died from fever at the early age of 49 in 1864.

Queens College Cork honoured his memory by installing a stained glass window in the *Aula Maxima* of the college. This shows Boole writing at a table with Aristotle and Plato in the background. An annual Boole Prize is awarded by the Mathematics Department at University College Cork. Des McHale has written an interesting biography of Boole [McH:85].

Boole's work on logic appeared to have no practical use. However, *Claude Shannon* became familiar with Boole's work in the 1930s, and his 1937 Masters thesis showed how Boolean algebra could optimise the design of systems of electromechanical relays which were then used in telephone routing switches. He also proved that circuits with relays could solve Boolean algebra problems.

The use of the properties of electrical switches to process logic is the basic concept that underlies all modern electronic digital computers. All digital computers today use the binary digits 0 and 1, and Boolean logical operations may be implemented by electronic AND, OR and NOT gates. More complex circuits (e.g. arithmetic) may be designed from these fundamental building blocks.

12.2.1 Modern Boolean Algebra

Boolean algebra consists of propositions that are either true or false. The proposition ' $2 + 2 = 4$ ' is true, whereas the proposition ' $2 \times 5 = 11$ ' is false. Variables (A , B , etc.) are used to stand for propositions, and propositions may be combined using logical connectives to form new propositions. The standard logical connectives are 'and', 'or' and 'not', and these are represented by the symbols ' \wedge ', ' \vee ' and ' \neg ', respectively. There are other logical connectives that may be used such as implication (\Rightarrow) and equivalence (\Leftrightarrow).

There are several well-known properties of Boolean algebra as described in Table 12.1.

The Boolean constant 'True' is the identity operation for conjunction. In other words, the conjunction of any operand A with the Boolean value 'True' yields the proposition A . Similarly, the Boolean constant 'False' is the identity operation for disjunction.

Truth tables define the truth values of a compound proposition from its constituent propositions. The conjunction of A and B ($A \wedge B$) is true if and only if

⁵ This is a novel about the struggle of an international revolutionary. Shostakovich wrote the score for the film of the same name that appeared in 1955.

Table 12.1 Properties of Boolean algebra

Property	Example
Commutative	$A \wedge B \equiv B \wedge A$ $A \vee B \equiv B \vee A$
Associative	$A \wedge (B \wedge C) \equiv (A \wedge B) \wedge C$ $A \vee (B \vee C) \equiv (A \vee B) \vee C$
Identity	$A \wedge \text{True} \equiv A$ $A \vee \text{False} \equiv A$
Distributive	$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$ $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$
De Morgan	$\neg (A \wedge B) \equiv \neg A \vee \neg B$ $\neg (A \vee B) \equiv \neg A \wedge \neg B$
Idempotent	$A \wedge A \equiv A$ $A \vee A \equiv A$

Table 12.2 Truth tables for conjunction and disjunction

A	B	$A \wedge B$	A	B	$A \vee B$
T	T	T	T	T	T
T	F	F	T	F	T
F	T	F	F	T	T
F	F	F	F	F	F

Table 12.3 Truth table for not operation

A	$\neg A$
T	F
F	T

both A and B are true. The disjunction of A and B ($A \vee B$) is true if either A or B is true. These are defined in Table 12.2.

There are other logical connectives that may be used [ORg:06] such as implication and equivalence. The ‘not’ operator (\neg) is a unary operator such that $\neg A$ is true if A is false and is false if A is true (Table 12.3).

Complex Boolean expressions may be formed from simple Boolean expressions using the logical connectives.

12.2.2 Switching Circuits and Boolean Algebra

Claude Shannon showed in his influential masters thesis that Boolean algebra was applicable to telephone routing switches. It could optimise the design of systems of electromechanical relays, and circuits with relays could solve Boolean algebra problems.

Modern electronic computers use millions of transistors that act as switches and can change state rapidly. The use of switches to represent binary values is the foundation of modern computing. A high voltage represents the binary value 1 with low voltage representing the binary value 0. A silicon chip may contain thousands

Fig. 12.2 Binary AND operation

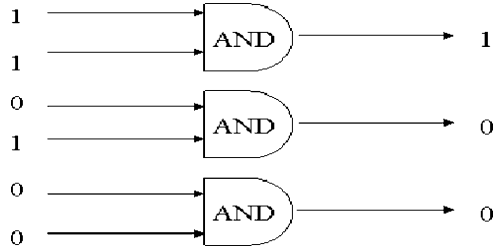


Fig. 12.3 Binary OR operation

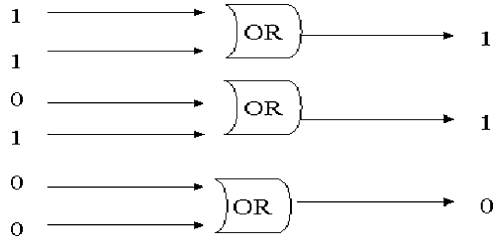


Fig. 12.4 NOT operation

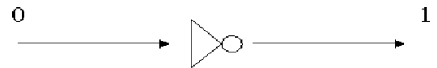
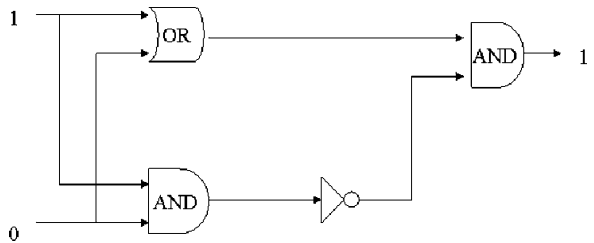


Fig. 12.5 Half adder



of tiny electronic switches arranged into logical gates. The basic logic gates are AND, OR and NOT. These gates may be combined in various ways to allow the computer to perform more complex tasks such as binary arithmetic. Each gate has binary value inputs and outputs.

The example in Fig. 12.2 below is that of an ‘AND’ gate, and this gate produces the binary value 1 as output only if both inputs are 1. Otherwise, the result will be the binary value 0.

The example in Fig. 12.3 is of an ‘OR’ gate, and it produces the binary value 1 as output if any of its inputs is 1. Otherwise, it will produce the binary value 0 as output.

Finally, a NOT gate accepts only a single input which it reverses. That is, if the input is ‘1’, the value ‘0’ is produced and vice versa. The NOT gate may be combined with the AND to yield the NAND gate (NOT AND) (Fig. 12.4).

The logic gates may be combined to form more complex circuits. The example in Fig. 12.5 is that of a half adder of 1 + 0.

The inputs to the top OR gate are 1 and 0, and this yields the result of 1. The inputs to the bottom AND gate are 1 and 0 which yields the result 0, which is then inverted through the NOT gate to yield binary 1. Finally, the last AND gate receives two 1s as input, and the binary value 1 is the result of the addition. The half adder computes the addition of two arbitrary binary digits, but it does not calculate the carry for the operation. The half adder may be extended to a full adder that provides a carry for addition.

12.3 Charles Babbage

Charles Babbage is considered (along with George Boole) to be one of the grandfathers of computing. He was born in Devonshire, England, in 1791 and was the son of a banker. He studied mathematics at Cambridge University in England and was appointed to the Lucasian Chair in Mathematics at Cambridge in 1828. He made contributions to several areas including mathematics, statistics, astronomy, philosophy, railways and lighthouses. He founded the British Statistical Society and the Royal Astronomical Society (Fig. 12.6).

Babbage was interested in accurate mathematical tables as these are essential for navigation and scientific work. However, there was a high error rate in the existing tables due to human error introduced during calculation. His approach to solving this problem was to investigate a mechanical method to perform the calculations as this would eliminate errors introduced by human calculation. Pascal and Leibniz did early work on calculating machines.

He designed the difference engine (no. 1) in 1821 for the production of mathematical tables. A difference engine is essentially a mechanical calculator

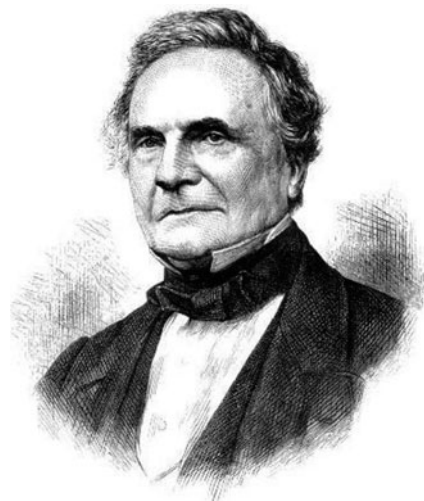


Fig. 12.6 Charles Babbage

(analogous to modern electronic calculators), and it was designed to compute polynomial functions. It could also compute logarithmic and trigonometric functions such as sine or cosine as these may be approximated by polynomials.⁶

The accurate approximation of trigonometric, exponential and logarithmic functions by polynomials depends on the degree of the polynomials, the number of decimal digits that it is being approximated to and the error function. A higher degree polynomial is able to approximate the function more accurately.

Babbage produced prototypes for parts of the difference engine but he never actually completed it. Babbage's Engine was intended to operate on sixth-order polynomials of 20 digits. A polynomial of degree 6 is of the form $p(x) = ax^6 + bx^5 + cx^4 + dx^3 + ex^2 + fx + g$.

He also designed the analytic engine (the world's first mechanical computer). The design of the analytic engine included a processor, memory and a way to input information and output results.

12.3.1 *Difference Engine*

The first working difference engine was built in 1853 by the Swedish engineers George and Edward Scheutz. They based their plans on Babbage's design and received funding from the Swedish government. The Scheutz machine could compute polynomials of degree 4 on 15-digit numbers. A copy of the third Scheutz Difference Engine is on display in the Science Museum in London.

This machine was the first to compute and print mathematical tables mechanically. The working engine produced by the Scheutz brothers was accurate, and it was used by scientists and engineers in their calculations. It showed the potential of mechanical machines as a tool for scientists and engineers (Fig. 12.7).

The difference engine consists of N columns (numbered 1– N). Each column is able to store one decimal number, and the numbers are represented by wheels. The difference engine (no. 1) has seven columns with each column containing 20 wheels. Each wheel consists of ten teeth, and these represent the decimal digits. Each column could therefore represent a decimal number with up to 20 digits. The seven columns allowed the representation of polynomials of degree six.

The only operation that the difference engine can perform is the addition of the value of column $n + 1$ to column n , and this results in a new value for column n .

⁶The power series expansion of the sine function is given by $\sin(x) = x - x^3/3! + x^5/5! - x^7/7! + \dots$. The power series expansion for the cosine function is given by $\cos(x) = 1 - x^2/2! + x^4/4! - x^6/6! + \dots$. Functions may be approximated by interpolation and the approximation of a function by a polynomial of degree n requires $n + 1$ points on the curve for the interpolation. That is, the curve formed by the polynomial of degree n that passes through the $n + 1$ points on the function to be approximated is an approximation to the function.

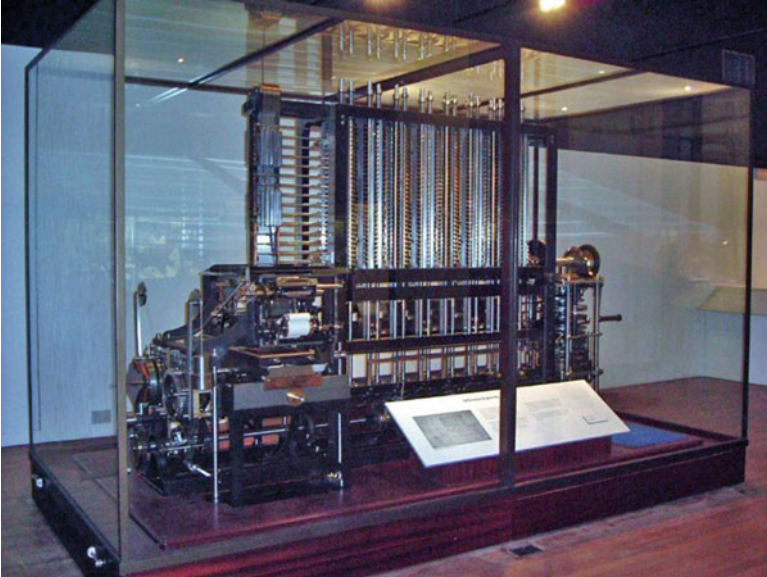


Fig. 12.7 Difference engine no. 2 (Photo public domain)

Column N can only store a constant and column 1 displays the value of the calculation for the current iteration. The machine is programmed prior to execution by setting initial values to each of the columns. Column 1 is set to the value of the polynomial at the start of computation; column 2 is set to a value derived from the first and higher derivatives of the polynomial for the same value of x . Each column n from 3 to N is set to a value derived from the $(n - 1)$ first and higher order derivatives of the polynomial.

The Scheutz's difference engine was comprised of shafts, shelves and wheels. The scientist could set numbers on the wheels⁷ and turn a crank to start the computation. Then, by reading down each shaft, he could find the result of the calculation. The difference engine was able to print out the answers to the computation. The decimal numbering system was employed, and there was also a carry mechanism.

The machine is unable to perform multiplication or division directly. Once the initial value of the polynomial and its derivatives are calculated for some value of x , the difference engine can calculate any number of nearby values using the numerical method of finite differences. This method replaces computational intensive tasks involving multiplication or division by an equivalent computation which just involves addition or subtraction.

⁷ Each wheel represented a decimal digit (each wheel consisted of ten teeth).

Table 12.4 Finite Differences (Calculation)

x	$p(x)$	Diff. 1	Diff. 2
1	6		
2	15	9	
3	28	13	4
4	45	17	4
5	66	21	4

Table 12.5 Finite Differences (In Use)

x	$f(x)$	Diff. 1	Diff. 2
1	6	9	4
2	15	13	4
3	28	17	4
4	45	21	4
5	66	25	4

12.3.2 Finite Differences

A finite difference is a mathematical expression of the form $f(x + h) - f(x)$. If a finite difference is divided by h , then the resulting expression is similar to a differential quotient, except that it is discrete.

Finite differences may be applied to approximate derivatives, and they are often used to find numerical solution to differential equations. They provide a very useful way to calculate the value of a polynomial used to approximate a given function.

The method of finite differences is used in the production of tables for polynomials using the difference engine. Consider the quadratic polynomial $p(x) = 2x^2 + 3x + 1$ and consider Table 12.4.

The first difference is computed by subtracting two adjacent entries in the column of $p(x)$. For example, $15 - 6 = 9$, $28 - 15 = 13$ and so on. Similarly, the second difference is given by subtracting two adjacent entries in the Difference 1 column; for example, $13 - 9 = 4$, $17 - 13 = 4$ and so on. The entries in the second difference column are the constant 4. In fact, for any n -degree polynomial the entries in the n -difference column are always a constant.

The difference engine performs the computation of the table in a similar manner, although the approach is essentially the reverse of the above. Once the first row of the table has been determined, the rest of the table may be computed using just additions of pairs of cells in the table (Table 12.5).

The first row is given by the cells 6, 9 and 4 which allows the rest of the table to be determined. The numbers in the table below have been derived by simple calculations from the first row. The procedure for calculation of the table is as follows:

1. The Difference 2 column is the constant 4.
2. The calculation of the cell in row i for the Difference 1 column is given by $\text{Diff. 1}(i - 1) + \text{Diff. 2}(i - 1)$.
3. The calculation of the cell in row i for the function column is given by $f(i - 1) + \text{Diff. 1}(i - 1)$.

In other words, to calculate the value of a particular cell, all that is required is to add the value in the cell immediately above it to the value of the cell immediately to its right. Therefore, in the second row, the calculations $6 + 9$ yield 15, and $9 + 4$ yields 13, and, since the last column is always a constant, it is just repeated. Therefore, the second row is 15, 13 and 4, and $f(2)$ is 15. Similarly, the third row yields $15 + 13 = 28$, $13 + 4 = 17$ and so on. This is the underlying procedure of the difference engine.

The initial problem is to compute the first row which allows the other rows to be computed. Its computation is more difficult for complex polynomials. The other problem is to find a suitable polynomial to represent the function, and this may be done by interpolation. However, once these problems are solved, the engine produces pages and columns full of data.

Babbage received £17 K of taxpayer funds to build the difference engine, but for various reasons he only produced prototypes of the intended machine. Therefore, the British government was reluctant to continue funding, and the project was cancelled in 1842.

The prototypes built by Babbage and his engineer Joseph Clement were limited to the computation of quadratic polynomials of six-digit numbers. The difference engine envisaged by Babbage was intended to operate on sixth-order polynomials of 20 digits.

He designed an improved difference engine (no. 2) in 1849. It could operate on seventh-order differences (i.e. polynomials of order 7) and 31-digit numbers. The machine consisted of eight columns with each column consisting of 31 wheels. However, it was over 150 years later before it was built (in 1991) to mark the 200th anniversary of his birth. The science museum also built the printer that Babbage designed, and both the machine and the printer worked correctly according to Babbage's design (after a little debugging).

12.3.3 *Analytic Engine*

The difference engine was designed to produce mathematical tables and required human intervention to perform the calculation. Babbage recognised its limitations and proposed a revolutionary solution. His plan was to construct a new machine that would be capable of executing all tasks that may be expressed in algebraic notation. The analytic engine envisioned by Babbage consisted of two parts (Table 12.6).

Babbage intended that the operation of the analytic engine would be analogous to the operation of the Jacquard loom.⁸ The latter is capable of weaving (i.e. executing

⁸ The Jacquard loom was invented by Joseph Jacquard in 1801. It is a mechanical loom which used the holes in punch cards to control the weaving of patterns in a fabric. The use of punch cards allowed complex designs to be woven from the pattern defined on the punch card. Each punch card corresponds to one row of the design, and the cards were appropriately ordered. It was very easy to change the pattern of the fabric being weaved on the loom, as this simply involved changing cards.

Table 12.6 Analytic engine

Part	Function
Store	This contains the variables to be operated upon as well as all those quantities which have arisen from the result of intermediate operations
Mill	The mill is essentially the processor of the machine into which the quantities about to be operated upon are brought

on the loom) a design pattern that has been prepared by a team of skilled artists. The design pattern is represented by punching holes on a set of cards, and each card represents a row in the design. The cards are then ordered and placed in the Jacquard loom, and the exact pattern is produced by the loom.

The Jacquard loom was the first machine to use punch cards to control a sequence of operations. It did not perform computation, but it was able to change the pattern of what was being weaved by changing cards. This gave Babbage the idea to use punch cards to store programs to perform the analysis and computation in the analytic engine.

The use of the punch cards in the analytic engine allowed the formulae to be manipulated in a manner dictated by the programmer. The cards commanded the analytic engine to perform various operations and to return a result. Babbage distinguished between two types of punch cards:

- Operation cards
- Variable cards

Operation cards are used to define the operations to be performed, whereas the variable cards define the variables or data that the operations are performed upon. This planned use of punch cards to store programs in the analytic engine is similar to the idea of a stored computer program in von Neumann architecture. However, Babbage's idea of using punch cards to represent machine instructions and data was over 100 years before digital computers. Babbage's analytic engine is therefore an important step in the history of computing.

The analytic engine was designed in 1834 as the world's first mechanical computer [Bab:42]. It included a processor, memory and a way to input information and output results. However, the machine was never built as Babbage was unable to receive funding from the British Government.

Babbage intended that the program be stored on read-only memory using punch cards, and that the input and output would be carried out using punch cards. He intended that the machine would be able to store numbers and intermediate results in memory that could then be processed. There would be several punch card readers in the machine for programs and data. He envisioned that the machine would be able to perform conditional jumps as well as parallel processing where several calculations could be performed at once (Fig. 12.8).

Lady Ada was introduced into Babbage's ideas on the analytic engine at a dinner party. She was a mathematician and the daughter of Lord Byron. She was fascinated by the idea of the analytic engine and communicated regularly with Babbage with ideas on its applications. She predicted that such a machine could be used

Fig. 12.8 Lady Ada Lovelace



to compose music, produce graphics as well as solving mathematical and scientific problems. She suggested to Babbage that a plan be written for how the engine would calculate Bernoulli numbers. This plan is considered to be the first computer program, and Lady Ada Lovelace is therefore considered to be the first computer programmer. The Ada programming language is named in her honour.

12.4 Review Questions

1. Describe how Boole's symbolic logic is used in digital circuit design.
2. Explain how finite differences are used to produce tables in the difference engine.
3. Describe the analytic engine and compare/contrast to von Neumann architecture.

12.5 Summary

This chapter considered the contribution of George Boole and Charles Babbage who are considered to be the grandfathers of the computing field.

Boole was a nineteenth-century English mathematician who made contributions to logic, probability theory and calculus. His calculus of logic (Boolean logic) introduced two quantities (i.e. 0 and 1) and is the foundation of all modern digital computers. It was Claude Shannon who recognised the applicability of Boolean logic to circuit design in his influential 1937 masters thesis. He showed that Boole's symbolic logic provided the perfect mathematical model for switching theory and for the design of digital circuits.

Babbage was a nineteenth-century scientist and inventor who did pioneering work on calculating machines. He invented the difference engine (a sophisticated calculator that could be used for the production of mathematical tables), and he designed the analytic engine (the world's first mechanical computer). The design of the analytic engine included a processor, memory and a way to input information and output results.