Chapter V

Analog Computation

 [1]

A Definition

Although analog computation was eclipsed by digital computation in the second half of the twentieth century, it is returning as an important alternative computing technology. Indeed, as explained in this chapter, theoretical results imply that analog computation can escape from the limitations of digital computation. Furthermore, analog computation has emerged as an important theoretical framework for discussing computation in the brain and other natural systems.

Analog computation gets its name from an *analogy*, or systematic relationship, between the physical processes in the computer and those in the system it is intended to model or simulate (the *primary system*). For example, the electrical quantities voltage, current, and conductance might be used as analogs of the fluid pressure, flow rate, and pipe diameter of a hydrolic system. More specifically, in traditional analog computation, physical quantities in the computation obey the same mathematical laws as physical quantities in the primary system. Thus the computational quantities are proportional

[1]This chapter is based on an unedited draft for an article that appeared in the *Encyclopedia of Complexity and System Science* (Springer, 2008).

to the modeled quantities. This is in contrast to *digital computation*, in which quantities are represented by strings of symbols (e.g., binary digits) that have no direct physical relationship to the modeled quantities. According to the *Oxford English Dictionary* (2nd ed., s.vv. analogue, digital), these usages emerged in the 1940s.

However, in a fundamental sense all computing is based on an analogy, that is, on a systematic relationship between the states and processes in the computer and those in the primary system. In a digital computer, the relationship is more abstract and complex than simple proportionality, but even so simple an analog computer as a slide rule goes beyond strict proportion (i.e., distance on the rule is proportional to the logarithm of the number). In both analog and digital computation—indeed in all computation—the relevant abstract mathematical structure of the problem is realized in the physical states and processes of the computer, but the realization may be more or less direct (MacLennan, 1994a,c, 2004).

Therefore, despite the etymologies of the terms "analog" and "digital," in modern usage the principal distinction between digital and analog computation is that the former operates on discrete representations in discrete steps, while the later operated on continuous representations by means of continuous processes (e.g., MacLennan 2004, Siegelmann 1999, p. 147, Small 2001, p. 30, Weyrick 1969, p. 3). That is, the primary distinction resides in the *topologies* of the states and processes, and it would be more accurate to refer to *discrete* and *continuous computation* (Goldstine, 1972, p. 39). (Consider so-called analog and digital clocks. The principal difference resides in the continuity or discreteness of the representation of time; the motion of the two (or three) hands of an "analog" clock do not mimic the motion of the rotating earth or the position of the sun relative to it.)

B Introduction

B.1 History

B.1.a Pre-electronic analog computation

Just like digital calculation, analog computation was originally performed by hand. Thus we find several analog computational procedures in the "constructions" of Euclidean geometry (Euclid, fl. 300 BCE), which derive from techniques used in ancient surveying and architecture. For example, Problem

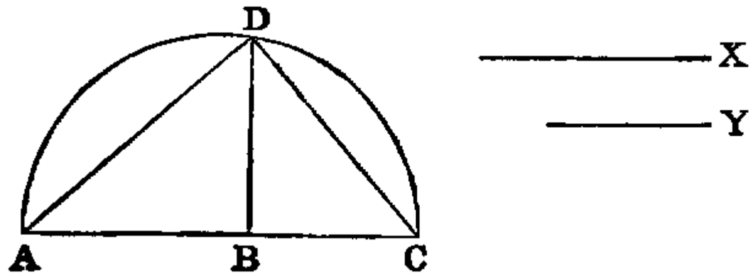


Figure V.1: Euclid Problem VI.13: To find a mean proportional between two given straight lines. Solution: Take on any line AC parts AB, BC respectively equal to X, Y. On AC describe a semicircle ADC. Erect BD at right angles to AC, meeting the semicircle in D. BD will be the mean proportional required.

II.51 is "to divide a given straight line into two parts, so that the rectangle contained by the whole and one of the parts shall be equal to the square of the other part." Also, Problem VI.13 is "to find a mean proportional between two given straight lines" (Fig. V.1), and VI.30 is "to cut a given straight line in extreme and mean ratio." These procedures do not make use of measurements in terms of any fixed unit or of digital calculation; the lengths and other continuous quantities are manipulated directly (via compass and straightedge). On the other hand, the techniques involve discrete, precise operational steps, and so they can be considered *algorithms*, but over continuous magnitudes rather than discrete numbers.

It is interesting to note that the ancient Greeks distinguished continuous *magnitudes* (Grk., *megethoi*), which have physical dimensions (e.g., length, area, rate), from discrete *numbers* (Grk., *arithmoi*), which do not (Maziarz & Greenwood, 1968). Euclid axiomatizes them separately (magnitudes in Book V, numbers in Book VII), and a mathematical system comprising both discrete and continuous quantities was not achieved until the nineteenth century in the work of Weierstrass and Dedekind.

The earliest known mechanical analog computer is the "Antikythera mechanism," which was found in 1900 in a shipwreck under the sea near the Greek island of Antikythera (between Kythera and Crete) (Figs. V.3, V.4). It dates to the second century BCE and appears to be intended for astronomical calculations. The device is sophisticated (at least 70 gears) and well engineered, suggesting that it was not the first of its type, and therefore that other analog

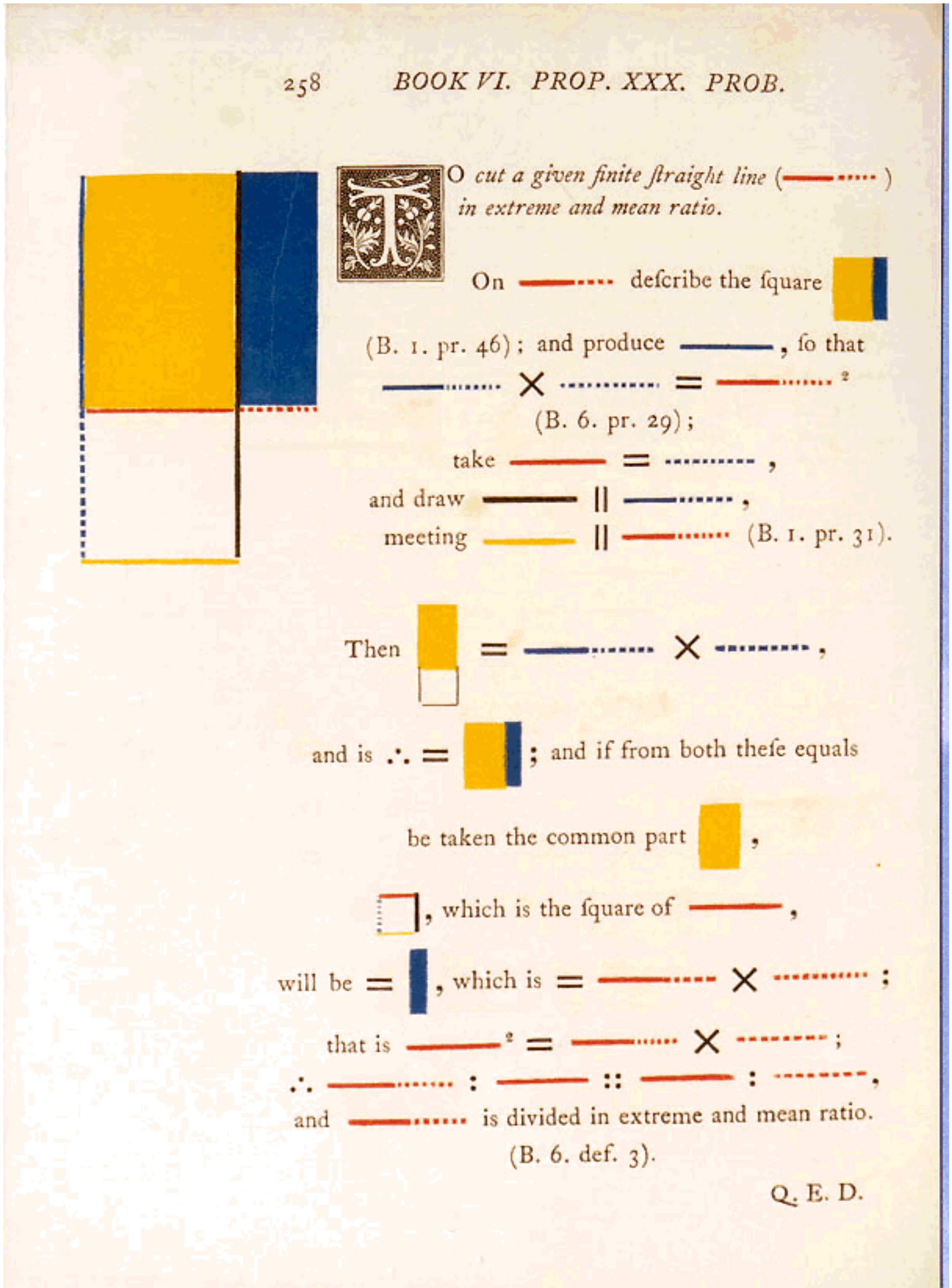
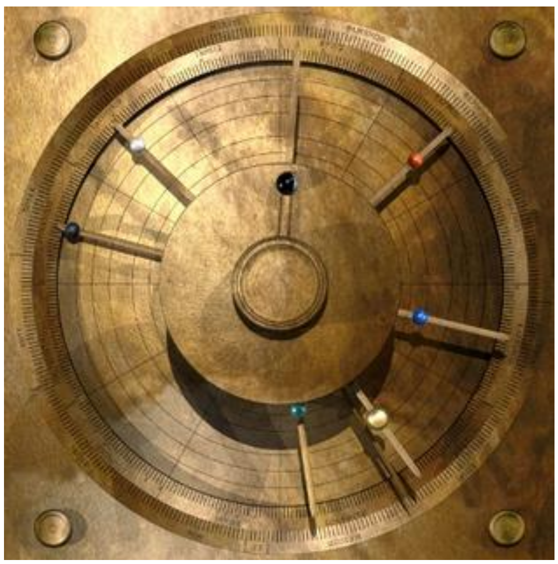
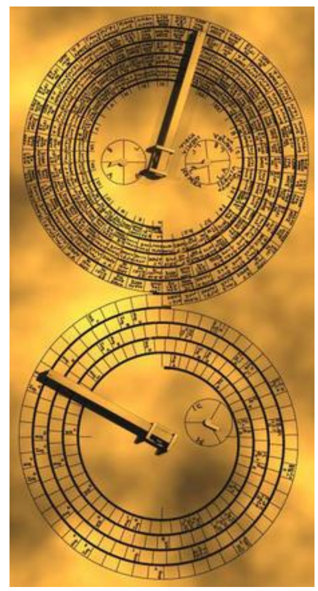


Figure V.2: Euclid Problem VI.30 from Byrne's *First Six Books of Euclid's Elements* (1847). [source: http://www.math.ubc.ca/~cass/Euclid/byrne.html]

Figure V.3: The Antikythera Mechanism. An ancient analog computer (2nd century BCE) for astronomical calculations including eclipses. [source: wikipedia]



(a) front



(b) back

Figure V.4: Computer reconstruction of Antikythera Mechanism. [source: wikipedia]

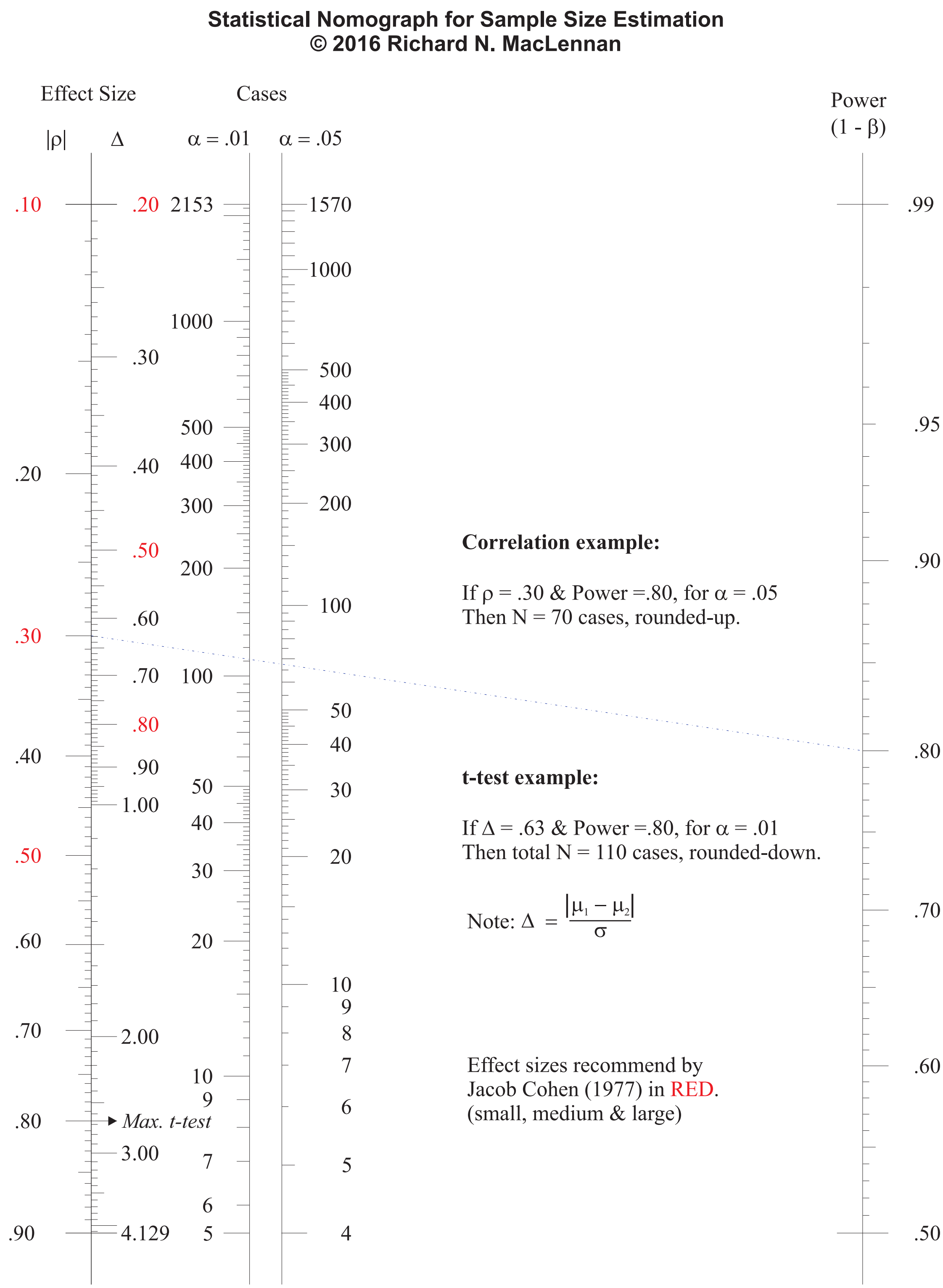


Figure V.5: Example nomograph.

computing devices may have been used in the ancient Mediterranean world (Freeth et al., 2006). Indeed, according to Cicero (*Rep.* 22) and other authors, Archimedes (c. 287–c. 212 BCE) and other ancient scientists also built analog computers, such as armillary spheres, for astronomical simulation and computation. Other antique mechanical analog computers include the astrolabe, which is used for the determination of longitude and a variety of other astronomical purposes, and the *torquetum*, which converts astronomical measurements between equatorial, ecliptic, and horizontal coordinates.

A class of special-purpose analog computer, which is simple in conception but may be used for a wide range of purposes, is the *nomograph* (also, *nomogram*, *alignment chart*) (Fig. V.5). In its most common form, it permits the solution of quite arbitrary equations in three real variables, $f(u, v, w) = 0$. The nomograph is a chart or graph with scales for each of the variables; typically these scales are curved and have non-uniform numerical markings. Given values for any two of the variables, a straightedge is laid across their positions on their scales, and the value of the third variable is read off where the straightedge crosses the third scale. Nomographs were used to solve many problems in engineering and applied mathematics. They improve intuitive understanding by allowing the relationships among the variables to be visualized, and facilitate exploring their variation by moving the straightedge. Lipka (1918) is an example of a course in graphical and mechanical methods of analog computation, including nomographs and slide rules.

Until the introduction of portable electronic calculators in the early 1970s, the *slide rule* was the most familiar analog computing device. Slide rules use logarithms for multiplication and division, and they were invented in the early seventeenth century shortly after John Napier's description of logarithms.

The mid-nineteenth century saw the development of the *field analogy method* by G. Kirchhoff (1824–87) and others (Kirchhoff, 1845). In this approach an electrical field in an electrolytic tank or conductive paper was used to solve two-dimensional boundary problems for temperature distributions and magnetic fields (Small, 2001, p. 34). It is an early example of *analog field computation*, which operates on continuous spatial distributions of quantity (i.e., fields).

In the nineteenth century a number of mechanical analog computers were developed for integration and differentiation (e.g., Lipka 1918, pp. 246–56, Clymer 1993). For example, the *planimeter* measures the area under a curve or within a closed boundary (Fig. V.6). While the operator moves a pointer along the curve, a rotating wheel accumulates the area. Similarly, the *inte-*

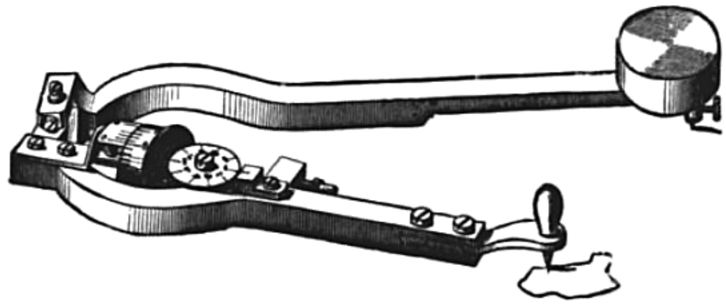


Figure V.6: Planimeter for measuring the area inside an arbitrary curve (1908). [source: wikipedia]

graph is able to draw the integral of a given function as its shape is traced (Fig. V.7). Other mechanical devices can draw the derivative of a curve or compute a tangent line at a given point.

Figure V.8: Lord Kelvin's analog tide computer. [source: wikipedia]

In the late nineteenth century William Thomson, Lord Kelvin, constructed several analog computers, including a "tide predictor" and a "harmonic analyzer," which computed the Fourier coefficients of a tidal curve (Thomson, 1878, 1938) (Fig. V.8). In 1876 he described how the mechanical integrators invented by his brother could be connected together in a feedback loop in order to solve second and higher order differential equations (Small 2001, pp. 34–5, 42, Thomson 1876). He was unable to construct this *differential analyzer*, which had to await the invention of the torque amplifier in 1927.

The torque amplifier and other technical advancements permitted Vannevar Bush at MIT to construct the first practical differential analyzer in 1930 (Small, 2001, pp. 42–5) (Fig. V.9). It had six integrators and could also do addition, subtraction, multiplication, and division. Input

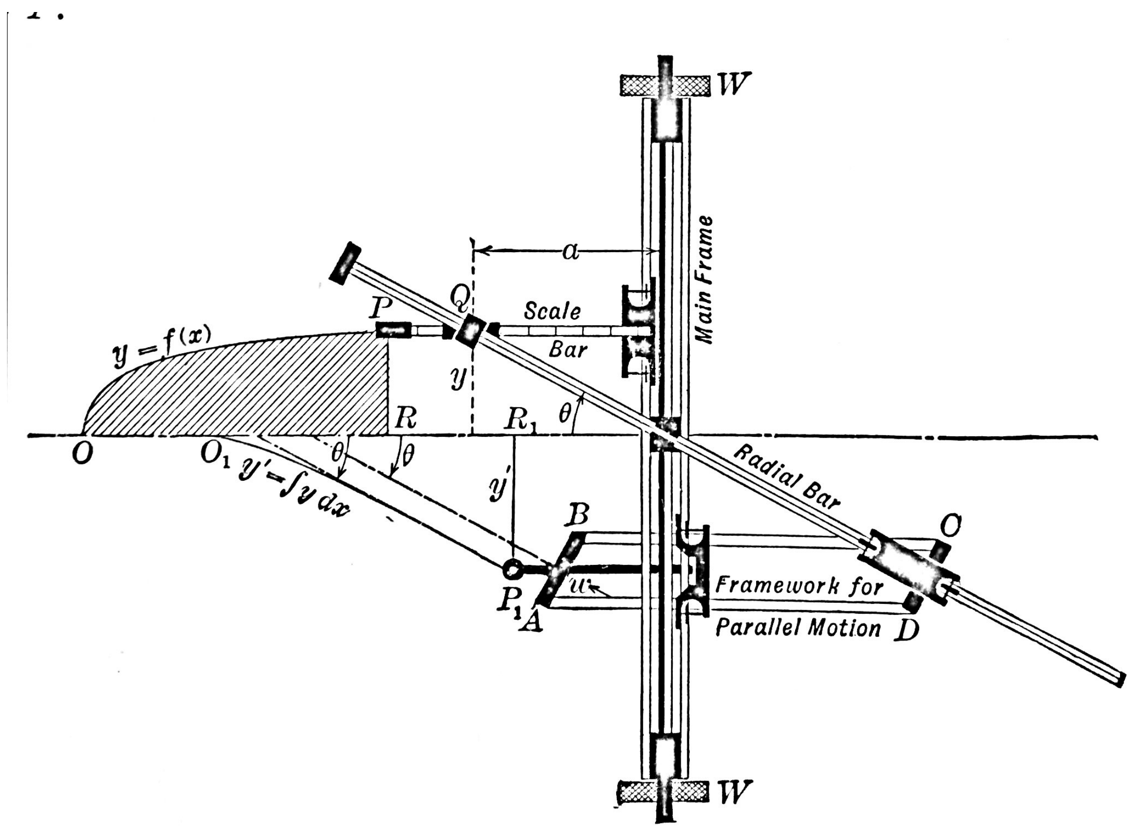


Figure V.7: Integraph for drawing the integral of an arbitrary curve (Lipka, 1918).

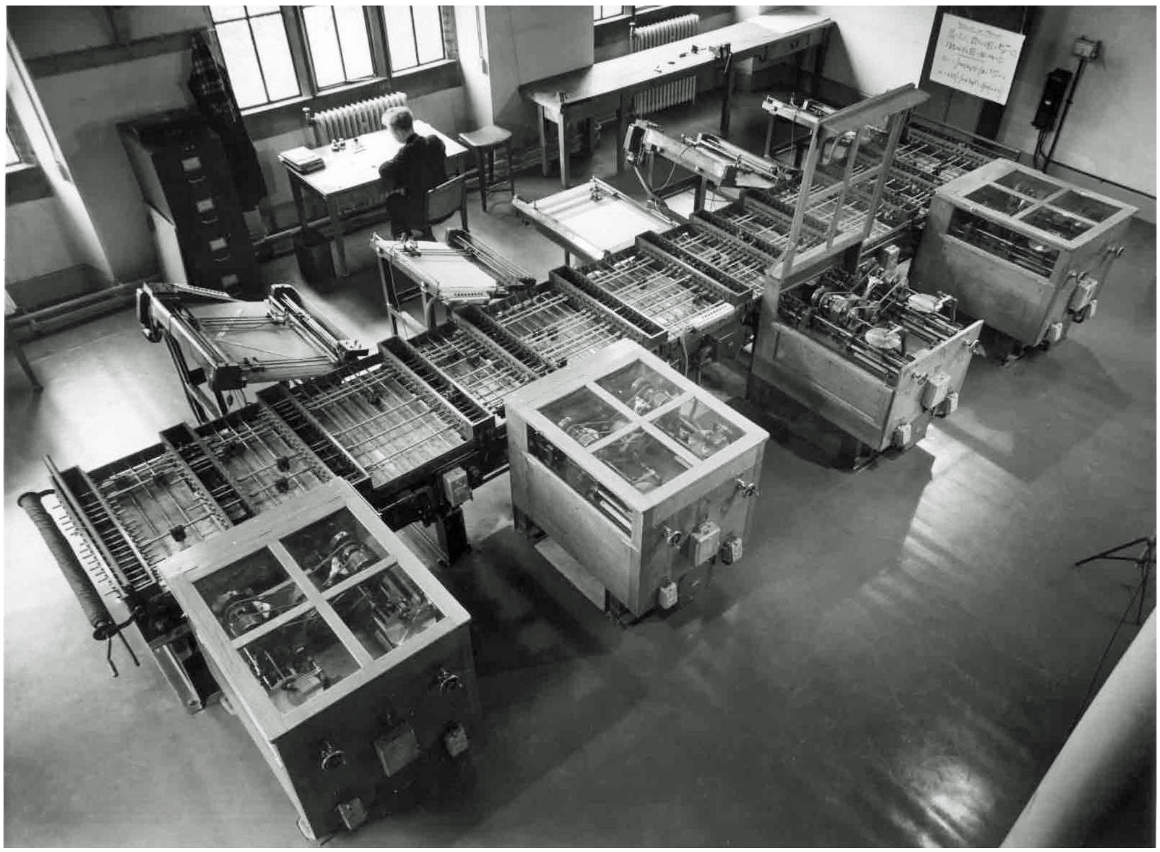


Figure V.9: MECCANO differential analyzer at Cambridge University, 1938. The computer was constructed by Douglas Hartree based on Vannevar Bush's design. [source: wikipedia] data were entered in the form of continuous curves, and the machine automatically plotted the output curves continuously as the equations were integrated. Similar differential analyzers were constructed at other laboratories in the US and the UK.

Setting up a problem on the MIT differential analyzer took a long time; gears and rods had to be arranged to define the required dependencies among the variables. Bush later designed a much more sophisticated machine, the Rockefeller Differential Analyzer, which became operational in 1947. With 18 integrators (out of a planned 30), it provided programmatic control of machine setup, and permitted several jobs to be run simultaneously. Mechanical differential analyzers were rapidly supplanted by electronic analog computers in the mid-1950s, and most were disassembled in the 1960s (Bowles 1996, Owens 1986, Small 2001, pp. 50–5).

During World War II, and even later wars, an important application of optical and mechanical analog computation was in "gun directors" and "bomb sights," which performed ballistic computations to accurately target artillery and dropped ordnance.

B.1.b Electronic analog computation in the 20th century

It is commonly supposed that electronic analog computers were superior to mechanical analog computers, and they were in many respects, including speed, cost, ease of construction, size, and portability (Small, 2001, pp. 54–6). On the other hand, mechanical integrators produced higher precision results (0.1%, vs. 1% for early electronic devices) and had greater mathematical flexibility (they were able to integrate with respect to any variable, not just time). However, many important applications did not require high precision and focused on dynamic systems for which time integration was sufficient; for these, electronic analog computers were superior.

Analog computers (non-electronic as well as electronic) can be divided into *active-element* and *passive-element* computers; the former involve some kind of amplification, the latter do not (Truitt & Rogers, 1960, pp. 2-1–4). Passive-element computers included the *network analyzers* that were developed in the 1920s to analyze electric power distribution networks, and which continued in use through the 1950s (Small, 2001, pp. 35–40). They were also applied to problems in thermodynamics, aircraft design, and mechanical engineering. In these systems networks or grids of resistive elements or reactive elements (i.e., involving capacitance and inductance as well as resistance) were used to model the spatial distribution of physical quantities such as voltage, current, and power (in electric distribution networks), electrical potential in space, stress in solid materials, temperature (in heat diffusion problems), pressure, fluid flow rate, and wave amplitude (Truitt & Rogers, 1960, p. 2-2). That is, network analyzers dealt with partial differential equations (PDEs), whereas active-element computers, such as the differential analyzer and its electronic successors, were restricted to ordinary differential equations (ODEs) in which time was the independent variable. Large network analyzers are early examples of *analog field computers.*

Electronic analog computers became feasible after the invention of the *DC operational amplifier* ("op amp") c. 1940 (Small, 2001, pp. 64, 67–72). Already in the 1930s scientists at Bell Telephone Laboratories (BTL) had developed the DC-coupled feedback-stabilized amplifier, which is the basis of the op amp. In 1940, as the USA prepared to enter World War II, D. L. Parkinson at BTL had a dream in which he saw DC amplifiers being used to control an anti-aircraft gun. As a consequence, with his colleagues C. A. Lovell and B. T. Weber, he wrote a series of papers on "electrical mathematics," which described electrical circuits to "operationalize" addition, subtraction, integration, differentiation, etc. The project to produce an electronic gun-director led to the development and refinement of DC op amps suitable for analog computation.

The war-time work at BTL was focused primarily on control applications of analog devices, such as the gun-director. Other researchers, such as E. Lakatos at BTL, were more interested in applying them to general-purpose analog computation for science and engineering, which resulted in the design of the General Purpose Analog Computer (GPAC), also called "Gypsy," completed in 1949 (Small, 2001, pp. 69–71). Building on the BTL op amp design, fundamental work on electronic analog computation was conducted at Columbia University in the 1940s. In particular, this research showed how analog computation could be applied to the simulation of dynamic systems and to the solution of nonlinear equations.

Commercial general-purpose analog computers (GPACs) emerged in the late 1940s and early 1950s (Small, 2001, pp. 72–3) (Fig. V.10). Typically they provided several dozen integrators, but several GPACs could be connected together to solve larger problems. Later, large-scale GPACs might have up to 500 amplifiers and compute with 0.01%–0.1% precision (Truitt & Rogers, 1960, pp. 2–33).

Besides integrators, typical GPACs provided adders, subtracters, multi-

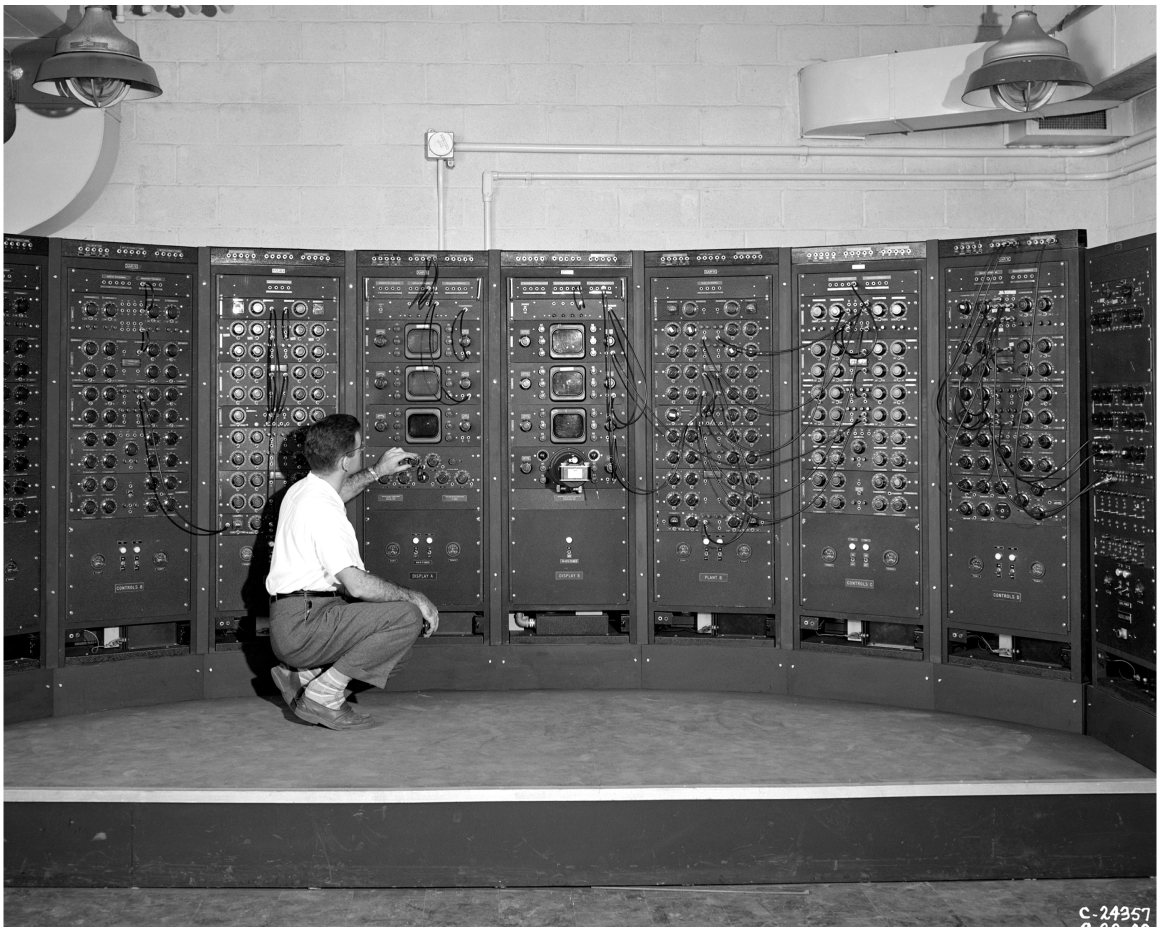


Figure V.10: Analog computer at Lewis Flight Propulsion Laboratory circa 1949.

pliers, fixed function generators (e.g., logarithms, exponentials, trigonometric functions), and variable function generators (for user-defined functions) (Truitt & Rogers, 1960, chs. 1.3, 2.4). A GPAC was programmed by connecting these components together, often by means of a patch panel. In addition, parameters could be set by adjusting potentiometers (attenuators), and arbitrary functions could be entered in the form of graphs (Truitt & Rogers, 1960, pp. 1-72–81, 2-154–156). Output devices plotted data continuously or displayed it numerically (Truitt & Rogers, 1960, pp. 3-1–30).

The most basic way of using a GPAC was in *single-shot mode* (Weyrick, 1969, pp. 168–70). First, parameters and initial values were entered into the potentiometers. Next, putting a master switch in "reset" mode controlled relays to apply the initial values to the integrators. Turning the switch to "operate" or "compute" mode allowed the computation to take place (i.e., the integrators to integrate). Finally, placing the switch in "hold" mode stopped the computation and stabilized the values, allowing them to be read from the computer (e.g., on voltmeters). Although single-shot operation was also called "slow operation" (in comparison to "repetitive operation," discussed next), it was in practice quite fast. Because all of the devices computed in parallel and at electronic speeds, analog computers usually solved problems in real-time but often much faster (Truitt & Rogers 1960, pp. 1-30–32, Small 2001, p. 72).

One common application of GPACs was to explore the effect of one or more parameters on the behavior of a system. To facilitate this exploration of the parameter space, some GPACs provided a *repetitive operation mode*, which worked as follows (Weyrick 1969, p. 170, Small 2001, p. 72). An electronic clock switched the computer between reset and compute modes at an adjustable rate (e.g., 10–1000 cycles per second) (Ashley, 1963, p. 280, n. 1). In effect the simulation was rerun at the clock rate, but if any parameters were adjusted, the simulation results would vary along with them. Therefore, within a few seconds, an entire family of related simulations could be run. More importantly, the operator could acquire an intuitive understanding of the system's dependence on its parameters.

B.1.c The eclipse of analog computing

A common view is that electronic analog computers were a primitive predecessor of the digital computer, and that their use was just a historical episode, or even a digression, in the inevitable triumph of digital technology. It is supposed that the current digital hegemony is a simple matter of technological superiority. However, the history is much more complicated, and involves a number of social, economic, historical, pedagogical, and also technical factors, which are outside the scope of this book (see Small 1993 and Small 2001, especially ch. 8, for more information). In any case, beginning after World War II and continuing for twenty-five years, there was lively debate about the relative merits of analog and digital computation.

Speed was an oft-cited advantage of analog computers (Small, 2001, ch. 8). While early digital computers were much faster than mechanical differential analyzers, they were slower (often by several orders of magnitude) than electronic analog computers. Furthermore, although digital computers could perform individual arithmetic operations rapidly, complete problems were solved sequentially, one operation at a time, whereas analog computers operated in parallel. Thus it was argued that increasingly large problems required more time to solve on a digital computer, whereas on an analog computer they might require more hardware but not more time. Even as digital computing speed was improved, analog computing retained its advantage for several decades, but this advantage eroded steadily.

Another important issue was the comparative *precision* of digital and analog computation (Small, 2001, ch. 8). Analog computers typically computed with three or four digits of precision, and it was very expensive to do much better, due to the difficulty of manufacturing the parts and other factors. In contrast, digital computers could perform arithmetic operations with many digits of precision, and the hardware cost was approximately proportional to the number of digits. Against this, analog computing advocates argued that many problems did not require such high precision, because the measurements were known to only a few significant figures and the mathematical models were approximations. Further, they distinguished between precision and *accuracy*, which refers to the conformity of the computation to physical reality, and they argued that digital computation was often less accurate than analog, due to numerical limitations (e.g., truncation, cumulative error in numerical integration). Nevertheless, some important applications, such as the calculation of missile trajectories, required greater precision, and for these, digital computation had the advantage. Indeed, to some extent precision was viewed as inherently desirable, even in applications where it was unimportant, and it was easily mistaken for accuracy. (See Sec. C.4.a for more on precision and accuracy.)

There was even a social factor involved, in that the written programs, precision, and exactness of digital computation were associated with mathematics and science, but the hands-on operation, parameter variation, and approximate solutions of analog computation were associated with engineering, and so analog computing inherited "the lower status of engineering *vis-à-vis* science" (Small, 2001, p. 251). Thus the status of digital computing was further enhanced as engineering became more mathematical and scientific after World War II (Small, 2001, pp. 247–51).

Already by the mid-1950s the competition between analog and digital had evolved into the idea that they were complementary technologies. This resulted in the development of a variety of *hybrid* analog/digital computing systems (Small, 2001, pp. 251–3, 263–6). In some cases this involved using a digital computer to control an analog computer by using digital logic to connect the analog computing elements, to set parameters, and to gather data. This improved the accessibility and usability of analog computers, but had the disadvantage of distancing the user from the physical analog system. The intercontinental ballistic missile program in the USA stimulated the further development of hybrid computers in the late 1950s and 1960s (Small, 1993). These applications required the speed of analog computation to simulate the closed-loop control systems and the precision of digital computation for accurate computation of trajectories. However, by the early 1970s hybrids were being displaced by all-digital systems. Certainly part of the reason was the steady improvement in digital technology, driven by a vibrant digital computer industry, but contemporaries also pointed to an inaccurate perception that analog computing was obsolete and to a lack of education about the advantages and techniques of analog computing.

Another argument made in favor of digital computers was that they were general-purpose, since they could be used in business data processing and other application domains, whereas analog computers were essentially special-purpose, since they were limited to scientific computation (Small, 2001, pp. 248–50). Against this it was argued that *all* computing is essentially computing by analogy, and therefore analog computation was general-purpose because the class of analog computers included digital computers! (See also Sec. A on computing by analogy.) Be that as it may, analog computation, as normally understood, is restricted to continuous variables, and so it was not immediately applicable to discrete data, such as that manipulated in business computing and other nonscientific applications. Therefore business (and eventually consumer) applications motivated the computer industry's investment in digital computer technology at the expense of analog technology.

Although it is commonly believed that analog computers quickly disappeared after digital computers became available, this is inaccurate, for both general-purpose and special-purpose analog computers have continued to be used in specialized applications to the present time. For example, a general-purpose *electrical* (vs. electronic) analog computer, the Anacom, was still in use in 1991. This is not technological atavism, for "there is no doubt considerable truth in the fact that Anacom continued to be used because it effectively met a need in a historically neglected but nevertheless important computer application area" (Aspray, 1993). As mentioned, the reasons for the eclipse of analog computing were not simply the technological superiority of digital computation; the conditions were much more complex. Therefore a change in conditions has necessitated a reevaluation of analog technology.

B.1.d Analog VLSI

In the mid-1980s, Carver Mead, who already had made important contributions to digital VLSI technology, began to advocate for the development of analog VLSI (Mead, 1987, 1989). His motivation was that "the nervous system of even a very simple animal contains computing paradigms that are orders of magnitude more effective than are those found in systems made by humans" and that they "can be realized in our most commonly available technology—silicon integrated circuits" (Mead, 1989, p. xi). However, he argued, since these natural computation systems are analog and highly nonlinear, progress would require understanding neural information processing in animals and applying it in a new analog VLSI technology.

Because analog computation is closer to the physical laws by which all computation is realized (which are continuous), analog circuits often use fewer devices than corresponding digital circuits. For example, a four-quadrant adder (capable of adding two signed numbers) can be fabricated from four transistors (Mead, 1989, pp. 87–8), and a four-quadrant multiplier from nine to seventeen, depending on the required range of operation (Mead, 1989, pp. 90–6). Intuitions derived from digital logic about what is simple or complex to compute are often misleading when applied to analog computation. For example, two transistors are sufficient to compute the logarithm or exponential, five for the hyperbolic tangent (which is very useful in neural computation), and three for the square root (Mead, 1989, pp. 70–1, 97–9). Thus analog VLSI is an attractive approach to "post-Moore's Law computing" (see Sec.

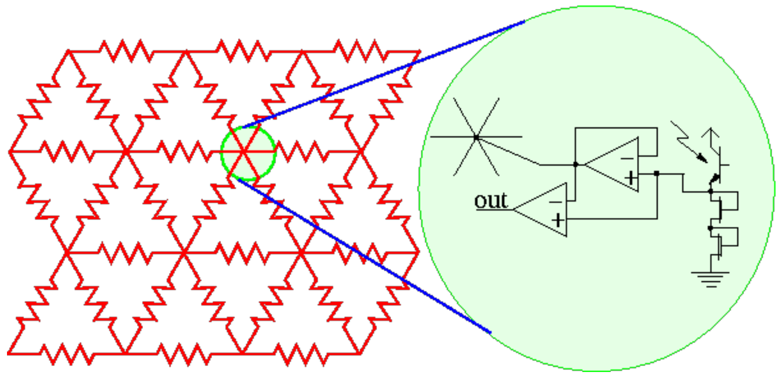


Figure V.11: Mahowald/Mead silicon retina.

H, p. 284 below). Mead and his colleagues demonstrated a number of analog VLSI devices inspired by the nervous system, including a "silicon retina" and an "electronic cochlea" (Fig. V.11) (Mead, 1989, chs. 15–16), research that has lead to a renaissance of interest in electronic analog computing.

B.1.e Field-programmable analog arrays

Field Programmable Analog Arrays (FPAAs) permit the programming of analog VLSI systems comparable to Field Programmable Gate Arrays (FPGAs) for digital systems (Fig. V.12). An FPAA comprises a number of identical Computational Analog Blocks (CABs), each of which contains a small number of analog computing elements. Programmable switching matrices control the interconnections among the elements of a CAB and the interconnections between the CABs. Contemporary FPAAs make use of floating-gate transistors, in which the gate has no DC connection to other circuit elements and thus is able to hold a charge indefinitely **cite**. Therefore the floating gate can be used to store a continuous value that governs the impedance of the transistor by several orders of magnitude. The gate charge can be changed by processes such as electron tunneling, which increases the charge, and hot-electron injection, which decreases it. Digital decoders allow individual floating-gate transistors in the switching matrices to be addressed and programmed. At the extremes of zero and infinite impedance the transistors operate as perfect switches, connecting or disconnecting circuit elements. Programming the connections to these extreme values is time consuming,

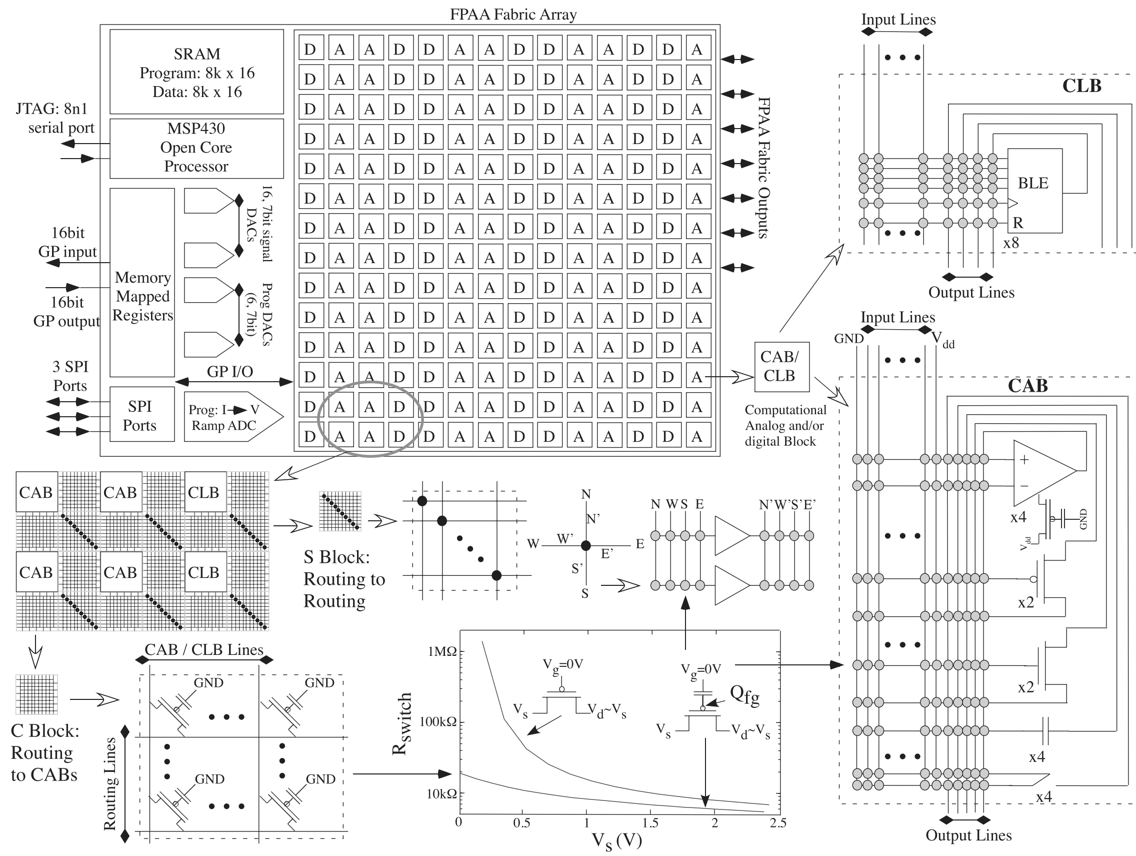


Figure V.12: RASP 3.0 FPAA. "RASP 3.0 integrates divergent concepts from multiple previous FPAA designs ... along with low-power digital computation, including a 16-bit microprocessor (μP), interface circuitry, and DACs + ADCs. The FPAA SoC die photo measures 12 mm 7 mm, fabricated in a 350-nm standard CMOS process. The die photo identifies μP, SRAM memory, DACs, and programming (DACs + ADC) infrastructure; the mixed array of the FPAA fabric is composed of interdigitated analog (A) and digital (D) configurable blocks on a single routing grid. DACs and programming infrastructure are accessed through memory-mapped registers." (George et al., 2016) however, and so in practice some tradeoff is made between programming time and switch impedance. Each CAB contains several Operational Transconductance Amplifiers (OTAs), which are op-amps whose gain is controlled by a bias current. They are the principal analog computing elements, since they can be used for operations such as integration, differentiation, and gain amplification. Other computing elements may include tunable band-pass filters, which can be used for Fourier signal processing, and small matrix-vector multipliers, which can be used to implement linear operators. Current FPAAs can compute with a resolution of 10 bits (precision of 10^{-3}).

B.1.f Non-electronic analog computation

As will be explained later in this chapter, analog computation suggests many opportunities for future computing technologies. Many physical phenomena are potential media for analog computation provided they have useful mathematical structure (i.e., the mathematical laws describing them are mathematical functions useful for general- or special-purpose computation), and they are sufficiently controllable for practical use.

B.2 Chapter roadmap

The remainder of this chapter will begin by summarizing the fundamentals of analog computing, starting with the continuous state space and the various processes by which analog computation can be organized in time. Next it will discuss analog computation in nature, which provides models and inspiration for many contemporary uses of analog computation, such as neural networks. Then we consider general-purpose analog computing, both from a theoretical perspective and in terms of practical general-purpose analog computers. This leads to a discussion of the theoretical power of analog computation and in particular to the issue of whether analog computing is in some sense more powerful than digital computing. We briefly consider the cognitive aspects of analog computing, and whether it leads to a different approach to computation than does digital computing. Finally, we conclude with some observations on the role of analog computation in "post-Moore's Law computing."