

CONTENTS

Acknowledgments *ix*

BECOMING UNIVERSAL: INTRODUCING A NEW HISTORY OF COMPUTING *1*

- 1 INVENTING THE COMPUTER** *9*
- 2 THE COMPUTER BECOMES A SCIENTIFIC SUPERTOOL** *29*
- 3 THE COMPUTER BECOMES A DATA PROCESSING DEVICE** *55*
- 4 THE COMPUTER BECOMES A REAL-TIME CONTROL SYSTEM** *83*
- 5 THE COMPUTER BECOMES AN INTERACTIVE TOOL** *109*
- 6 THE COMPUTER BECOMES A COMMUNICATIONS PLATFORM** *139*
- 7 THE COMPUTER BECOMES A PERSONAL PLAYTHING** *167*
- 8 THE COMPUTER BECOMES OFFICE EQUIPMENT** *207*
- 9 THE COMPUTER BECOMES A GRAPHICAL TOOL** *243*
- 10 THE PC BECOMES A MINICOMPUTER** *263*
- 11 THE COMPUTER BECOMES A UNIVERSAL MEDIA DEVICE** *293*
- 12 THE COMPUTER BECOMES A PUBLISHING PLATFORM** *329*
- 13 THE COMPUTER BECOMES A NETWORK** *359*
- 14 THE COMPUTER IS EVERYWHERE AND NOWHERE** *385*
- 15 EPILOGUE: A TESLA IN THE VALLEY** *409*

Notes *425*

Bibliography *467*

Index *495*

ACKNOWLEDGMENTS

We are particularly grateful to David Hemmendinger, Gerardo Con Diaz, Marc Weber, Michael J. Halvorson, Alan Staiti, Eugene Miya, and the MIT Press reviewers who all read through the entire manuscript offering innumerable useful suggestions. Paul McJones, Clem Cole, David Brock, Troy Astarte, Tom Lean, Bradley Fidler, Tom Van Vleck, Henry Lowood, Jerome Coonen, Forrest Park, Gordon Bell, and Donald B. Wagner provided important input on specific topics and sections. Series editor Tom Misa had an eagle's eye for typos. Much of the material in this book is adapted from Ceruzzi's earlier *A History of Modern Computing*, and so we also remain in debt to those previously acknowledged there. In particular we appreciate William Aspray's work as longtime series editor, which befitted this book and many of the other books we drew on to create it.

The book's creation was generously supported by Siegen University's Media of Cooperation CRC. This included three periods spent working together in Siegen, Germany, to review the potential for a new overview history, outline the new structure of the book, review existing material in the second edition, and assemble two draft chapters as a test of our new approach. Additional support from Siegen allowed Haigh to focus on writing. We happily acknowledge Erhard Schüttpelz, Sebastian Giessmann, and Tristan Thielman for arranging this support and for their collegial contributions to our project. We talked through the new structure with participants at three Early Digital workshops and benefited from the suggestions of Laine Nooney, David Brock, Stephanie Dick, William Aspray, Martin Campbell-Kelly, Doron Swade, Matthew Kirschenbaum, Len Shustek, Rebecca Slayton, and many others. Parts of the book were later discussed in a Siegen workshop on the history of database management systems with the participation of Moritz Feichtinger and Francis Hunger.

For help with image permissions and scans we are grateful to Debbie Douglas, Janice Hussain, Angela Schad, Katherine Taylor, Brian Daly, Erik Rau, Stephanie

Hueter, Andrea Wescott, Ingrid Crete, Lindsy Wyner, Bryan Roppolo, Amanda Wick, and various contributors to Wikimedia Commons.

Haigh happily acknowledges the support of his family members Maria, Peter, and Paul during the five years of sometimes intense work it took to produce this book. Their love and sacrifice made the book possible. Both authors appreciate the efforts of the MIT Press editorial and production teams. Acquisitions editor Katie Helkie advocated early for the need to update Ceruzzi's book, while Laura Keeler gave prompt answers to all our queries on manuscript preparation. We are particularly grateful for the work of Helen Wheeler and Virginia A. Schaefer at Westchester Publication Services. Stefan Swanson worked heroically on proofreading and reference checking, catching dozens of errors.

Fragments of chapters 6 and 12 were adapted from Haigh's contribution to *The Internet and American Business* (edited by William Aspray and Paul Ceruzzi) and Ceruzzi's contribution to *Social Media Archaeology and Poetics* (edited by Judy Malloy), both published by MIT Press. Portions of chapter 3 are modeled after Haigh's "The Chromium-Plated Tabulator: Institutionalizing an Electronic Revolution, 1954–1958." *IEEE Annals of the History of Computing* 23, no. 4 (October–December 2001): 75–104 and "How Data Got Its Base: Information Storage Software in the 1950s and 60s." *IEEE Annals of the History of Computing* 31, no. 4 (October–December 2009): 6–25.

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—Project-ID 262513311-SFB 1187 Media of Cooperation.

BECOMING UNIVERSAL: INTRODUCING A NEW HISTORY OF COMPUTING

This book is a comprehensive reimagining of *A History of Modern Computing*, first published in 1998 and expanded with a new chapter in 2003. A lot has changed since 1998 when the Web was a novelty, iPhones didn't exist, and the founders of Google and Facebook were in graduate school and high school, respectively. Doing justice to those changes required more than just adding a few more chapters at the end of the book. For example, as the first edition was being written and conceived, the Internet was still quite an obscure system. Today we view the development of computer communications as a central thread in the history of computing, not just in the 1990s but also in the 1960s and 1970s. The wholesale shift of video and music reproduction to digital technologies likewise challenges us to integrate media history into the long history of computing. Since the original book was written, the computer had become something new, which meant that the book also had to become something new.

The unmistakable importance of the Internet, digital media devices, and video games to modern life has driven public interest in their stories. Yet this discussion is rarely grounded in the longer and deeper history of computer technology. For example, as we finalized our revisions to this book, one of us chanced upon *How the Internet Happened: From Netscape to the iPhone* by Brian McCullough, a tech industry insider.¹ It is readable, admirably tight, and solidly researched—based on two hundred interview podcasts McCullough recorded with company founders. We recommend it to you. Yet we were also struck by how little engagement such approaches to history have with the larger story of computing. As his title suggests, McCullough starts the story of the Internet in 1994 with the first commercial Web browser, giving only occasional flashbacks to the first twenty-five years of the Internet (and its precursor, the ARPANET). He says little about where the core technologies, protocols, or algorithms of the Web came from, or about the evolving technologies personal computing, such as new processors and operating systems, that made the rapid spread of Web browsers possible. He

says nothing about Web server technology, or the programming languages and practices that evolved alongside Web browsers. Similar observations can, and have, been made about popular histories of video games and personal computing. Our aim here is to integrate Internet and Web history into the core narrative of the history of computing, along with the history of iPods, video game consoles, home computers, digital cameras, and smartphone apps.

To write the history of a technology only as a series of models, inventors, and refinements is to miss the point. Thomas J. Misa once suggested that the great challenge facing historians of computing is to explain “How Computing Has Changed the World.”² Doing that might not seem so hard. The computer has a relatively short history, which for our purposes begins in the 1940s. Set against technologies such as agriculture, Arabic numerals, or alphabets its span looks like the blink of an eye. Despite its ever-growing importance, its influence on our lives has so far been less fundamental than that of industrial age technologies such as electric light or power, automobiles, or antibiotics.

Important technologies have complex histories. The automobile, for example, was made possible by the development of earlier technologies such as coaches and bicycles. Its mass adoption with the Ford Model T took place decades after its invention, and was made possible by the development of big business and the invention of mass production as a way to build complex machines cheaply in huge quantities. The Model T’s users discovered new uses for it, building new body work or adapting it as a portable power source for agricultural machinery. The automobile facilitated, but did not dictate, a mass exodus from America’s cities into sprawling suburbs and exurbs. Because most Americans came to rely on cars to shop, get to work, and socialize, their national culture grew up around the technology. The resulting need for massive quantities of oil reshaped American foreign policy and transformed the fortunes of nations from Norway to Nigeria, usually for the worse.³

Doing justice to that story would challenge even the most ambitious historian, but the automotive historian has a crucial advantage over the historian of computing: over the century from 1920 to 2020, the typical car had a roughly stable physical form: a large self-propelled metal box able to move between two and eight people over asphalt at a maximum speed that has roughly doubled, from forty miles an hour to a (legally mandated) seventy or eighty. Cars are still built on assembly lines by large, capital-intensive companies. Ford, General Motors, and Chrysler were the “big three” US automakers of the 1920s and retain that status today. Cars are still distributed by franchised dealers. A basic but functional car costs a skilled worker a few months of salary.

The story of computing offers us no comparable continuities. Few, if any, other technologies have changed their scale, dominant applications, and users so often and so fundamentally. The computer started out as esoteric and specialized as the cyclotron

and has finished up only slightly less ubiquitous than clothing or food. In the 1940s, computers were used by a few hundred people worldwide to carry out complex numerical calculations. They were built as one-of-a-kind pieces of custom lab equipment, each costing the equivalent of several million present-day dollars.

Computer scientists have adopted a term from Alan Turing, the *universal machine*, to describe the remarkable flexibility of programmable computers. To prove a mathematical point he described a class of imaginary machines (now called Turing machines) that processed symbols on an unbounded tape according to rules held in a table. By encoding the rules themselves on the tape, Turing's universal machine was able to compute any number computable by a more specialized machine of the same ilk. Computer scientists came to find this useful as a model of ability of all programmable computers to carry out arbitrary sequences of operations, and hence (if unlimited time and storage were available) to mimic each other by using code to replicate missing hardware.⁴

In practice, however, the first modern computers faced severe practical limits on their capabilities. As those restraints were gradually lifted, the scope of what could feasibly or economically be computerized grew dramatically as the computer evolved toward what economists call a "general purpose technology" with highly varied applications. Today about half the world's inhabitants use hand-held computers daily to facilitate almost every imaginable human task. They carry out their work millions of times faster than those early models, fit easily in a pocket, and are cheap enough to be thrown away when a cracked piece of glass needs repair.

Computers will never do everything, be used by everyone, or replace every other technology, but they are more nearly universal than any other technology. In that broader sense the computer began as a highly specialized technology and has moved toward universality and ubiquity. We think of this as a progression toward *practical universality*, in contrast to the theoretical universality often claimed for computers as embodiments of Turing machines.

To the extent that it has become a universal machine, the computer might also be called a *universal solvent*, achieving something of that old dream of alchemy by making an astounding variety of other technologies vanish into itself. Maps, filing cabinets, video tape players, typewriters, paper memos, and slide rules are rarely used now, as their functions have been replaced by software running on personal computers, smartphones, and networks. We conceptualize this convergence of tasks on a single platform as a *dissolving* of those technologies and, in many cases, their business models by a device that comes ever closer to the status of universal technological solvent.

In many cases the computer has dissolved the insides of other technologies while leaving their outward forms intact. Although computer technology is universal, most actual computers are configured and deployed to carry out extremely specialized tasks.

They hide inside cars and consumer appliances to replace the guts of many of the technologies of everyday life, such as telephones, photocopiers, televisions, pianos, and even light bulbs. These computers outnumber humans many times over and cost as little as three cents apiece in bulk. They still have processors and memory and run software, but only computer scientists habitually think of them as computers.

This shape shifting makes the construction of a satisfactory overall history of computing exceptionally difficult. How to tell a story when the scale of the stage and the cast of characters changes so fundamentally? The easiest way to write a book like this would be to devote one or two chapters to each decade. But we want to tell a story with a plot, not just arrange a succession of facts and anecdotes in roughly chronological order. Our answer was to focus on constructing each individual chapter to tell the story of a transformation, in which particular communities of users and producers remade the computer into something new. Each chapter tells a coherent story with a stable cast of characters, even though the companies, applications, and communities relevant to that chapter may not appear in others.

For example, the first transformations began in the 1950s, as computers were remade for use as scientific supercomputers capable of feats of number crunching, data processing devices able to automate the work of hundreds of clerks, and real-time control systems used to coordinate air defense. We tell those three parallel stories in three parallel chapters, each reaching into the 1970s. Chapters continue to overlap in time, although as you move through the book you will draw gradually closer to the present. In later chapters the computer becomes a communications medium, a graphical tool, a personal plaything, and so on. The full list is longer, as you can read in the table of contents. We hope that you find the new structure clear and coherent. After sketching out this version, we began by rearranging passages of the existing text within it, filling in the gaps with new material. If you are familiar with Ceruzzi's original book you will be able to find compressed versions of almost all the topics it covered somewhere in our new text.⁵

You will find many examples here of how computerization changed specific parts of the world, but not every part of the world has been changed in the same way. Misa's question, "How did the computer change the world," which was posed to an entire field, admits no single answer. We have instead tried to give a reasonably comprehensive answer to a more tractable question: "How did the world change the computer?" Read together, the smaller stories told in each chapter add up to a larger one. The protagonist of this story is "the computer" itself. To talk about "the computer" might sound a little ridiculous, in a world where some computers are thrown away inside hotel key cards and others cost millions of dollars. Yet at the core of each machine is a package of programming techniques and architectural features reflecting a shared descent. Architectural advances pioneered by Cray supercomputers now help your phone to play Netflix

video more effectively. The original *A History of Modern Computing* engaged more deeply than any other overview history of computing with the evolution of computer architecture. Preserving and deepening that focus on the origins and diffusion of new architectural features contributes to knitting the new book together.

Technologies are shaped by societies or, more specifically, by institutions such as governments and corporations, by inventors responding to incentives, and by users who apply and reshape technologies in ways unimagined by their original creators. Another distinctive feature of the original *A History of Modern Computing* was its interest in the stories of computer users, with deeply researched case studies of NASA, the Internal Revenue Service, and other influential organizational users of computers. We have retained these and have also woven shorter examinations of the experiences of computer users into each chapter. This complements our focus on architecture, because new architectural features and software technologies were originally created to serve the specific needs of specific users. This structure builds on the insights of Michael S. Mahoney in his classic paper “Histories of Computing(s).” Mahoney argued that “the histories and continuing experience of the various communities show that they wanted and expected different things from the computer. They encountered different problems and levels of difficulty in fitting their practice to it. As a result, they created different computers or (if we may make the singular plural) computings.”⁶ Whenever the computer became a new thing it did not stop being everything it had been before. Computers are still used by nuclear weapons labs and banks. These stories intertwined, as new capabilities move from one domain to another.

Our story starts in the 1940s with programmable electronic computers and not, like more traditional overview histories, with mechanical calculators or Charles Babbage. To tell the story of a new technology one would ideally begin by documenting the practices it was applied to and the earlier technologies used and then explore its origin, its spread, and the new practices and institutions that coevolved with it. Decades ago, when the scope of computing was smaller, it made sense to see electronic computing as a continuation of the tradition of scientific computation. The first major history of computing, *The Computer from Pascal to von Neumann* by computing pioneer Herman Goldstine, concluded in the 1940s with the invention of the modern computer. In *A History of Computing Technology*, published in 1985, Michael Williams started with the invention of numbers and reached electronic computers about two thirds of the way through. By the 1990s the importance of computer applications to business administration was being documented by historians, so it was natural for Martin Campbell-Kelly and William Aspray, when writing *Computer: A History of the Information Machine*, to replace discussion of slide rules and astrolabes with mechanical office machines, filing cabinets, and administrative processes.⁷

Giving up their coverage of earlier technologies carries a cost. To understand what changed in the world because of the adoption of a technology, we need to know the “before” as well as the “after.” Yet its influence will confront future historians of every kind, whether they are writing about presidential politics or pop music. The breadth of technologies displaced by the computer and practices remade around it makes it seem arbitrary to begin with chapters that tell the stories of index cards but not of televisions; of slide rules but not of pinball machines; or of typewriters but not of the postal system. But to include those stories, each of our chapters would need to become a long book of its own, written by different experts.

A History of Modern Computing was the most widely cited scholarly overview history of computing. For many of the people who picked up a copy in a library, or were assigned it for a class, it gave a first introduction to the topic. We hope our new book is a starting point, and not an end point, for your engagement with this rich history. To help guide you, we have systematically added citations to, and quotations from, some of the many outstanding works of scholarly history on different aspects of the history of computing. Our challenge here is to condense stories big enough to fill entire books into a page or a paragraph. Most histories focus on a specific aspect, occasionally something as broad as the software industry, but more often a single company or computer platform. There are more histories of Google, Microsoft, or Apple than there are of the computer itself. We do not include “further reading” lists for each chapter, but when we mention a book in the text you can safely assume that it is an outstanding and highly relevant source of further reading. These books have inspired and informed us, and we would like to share that gift with you.

Broad as this book is, we must warn you that it is a history of computing technology and practice, not of computer science. Computer science is an academic discipline. It began to come together intellectually in the late 1950s and was institutionalized during the 1960s and 1970s via university departments, corporate research labs, funding agencies, conferences, and journals. When specific work done by computer scientists has a major influence on practice we discuss its contribution, but we cannot try to squeeze into this book the stories of research areas, influential departments, intellectual schools, or the development of subdisciplines such as architecture, theory, graphics, databases, networking, and artificial intelligence. Historians of science have paid remarkably little attention to computer science (and, alas, computer scientists to history), so unfortunately there are no major histories of computer science or of any of its subdisciplines for us to point you toward.⁸

Another question we won’t be answering is “What was the first computer?” Arguments about firsts once drove lengthy lawsuits and patent proceedings. They continue to dominate much general discussion of early electronic computing, particularly in

Internet forums. Any answer depends on one's definition of *computer*. In the 1940s the question would not even have made sense, because *computer* usually meant a person employed to carry out complex calculations. The new machines being built at the time were called automatic computers or computing machines. Even those weren't the first calculating machines, which is why we call this a history of *modern* computing.

But we do have to start this book somewhere. We start it in 1945 with the first operation of a machine called ENIAC at the University of Pennsylvania. A truce reached in the 1980s, as professionally trained historians began to engage with this topic, established strings of adjectives to qualifying the "firstness" of the various novel machines constructed during the 1940s. ENIAC is usually called something like the "first electronic, general purpose, programmable computer."⁹

Those qualifying adjectives separate it from two earlier groups of machines. *Electronic* distinguishes it from electromechanical computers whose logic units worked thousands of times more slowly. Often called relay calculators, these computers carried out computations one instruction at a time under the control of paper tapes. They were player pianos that produced numbers rather than music. Among the best known were the Harvard Mark 1, produced by International Business Machines to meet the specification of Harvard's Howard Aiken, and the Z3 designed by German computing pioneer Konrad Zuse. *General purpose* and *programmable* separated ENIAC from special purpose electronic machines whose sequence of operations was built into hardware and so could not be reprogrammed to carry out fundamentally different tasks. The ABC, or Atanasoff-Berry Computer, built at Iowa State, used a fixed program to solve systems of linear equations.¹⁰ The British wartime Colossus machines applied logical tests to inputs from encrypted messages and electronically simulated code wheels. Their basic sequence of operations was likewise fixed.¹¹

The ENIAC project introduced the vocabulary of programs and programming and the automation of higher-level control functions such as branches and looping. It was publicized around the world, stimulating interest in electronic computation. Its two main designers founded the first electronic computer company. And even before ENIAC was finished, design work on a planned successor, EDVAC, had defined the key architectural features of the modern computer.