Book Reviews

Computing: Key Developments

From ENIAC to UNIVAC. An Appraisal of the Eckert-Mauchly Computers. NANCY STERN. Digital Press, Bedford, Mass., 1981. x, 286 pp., illus. $21. Digital Press History of Computing Series.

The electronic digital computer was born of the need for numerical computations brought on by the Second World War. But the modern computer age was not an inevitable consequence of just the invention of the machine. In *From ENIAC to UNIVAC*, Nancy Stern argues that, with the war as a catalyst, it was only through the vision and perseverance of two men, J. Presper Eckert and John W. Mauchly, that the computer moved beyond the university and military laboratories to its present position in our society. In short, though the story of the invention of the computer has been told before, Stern stresses Eckert's and Mauchly's crucial importance as innovators—men who made the computer a commercial product. She reminds the reader that in the late 1940's other computer pioneers were envisioning a future in which at most four of five electronic computers would be serving all the computing needs of the country. Eckert and Mauchly, as two young and relatively obscure inventors, had to overcome a lot of barriers—technical, financial, and psychological—before their own vision could prevail.

From ENIAC to UNIVAC is essentially an account of the four computing machines built by the two Philadelphia inventors between 1943 and 1951. These were the ENIAC, the first general-purpose electronic computer; the EDVAC, the first computer designed with the stored-program principle as a central feature; the BINAC, a stored-program computer designed for the Northrop Aircraft Company; and the UNIVAC, the first commercially available computer in the United States, the introduction of which in 1951 marks the true beginning of the computer age. The development of each machine is examined in detail, and the author takes care to sort out and assess the controversies that have arisen over them. And of controversies there is certainly no lack. Were Eckert and Mauchly really the inventors of the ENIAC, or did they get their ideas from others, especially from J. V. Atanasoff of Iowa State College? Alternatively, were Eckert and Mauchly the sole inventors of the ENIAC, or should the credit be shared with the team of engineers and scientists they gathered around them? Who really discovered the principle of storing the computer's program along with the data in its memory, a principle that has since been recognized as fundamental to a computer's power? Have historians until now given the mathematician John von Neumann too much credit for that discovery? On these and other issues, Stern generally supports Eckert and Mauchly, but she is fair to all sides and always carefully documents her arguments. One wishes at times that questions like "Who really invented the computer?" were never asked, since they obviously admit of no simple answers, but since they have been brought up elsewhere Stern has no choice but to deal with them, and she does a good job. In particular, she has examined the suit filed by Honeywell, Inc., against Sperry Rand (holders of the original patent on the ENIAC), in which the question whether Eckert and Mauchly really were the inventors of the first electronic computer was put before a court of law. While acknowledging that a court is an inappropriate place to decide such historical questions, Stern nevertheless has made good use of the thousands of pages of transcripts from the trial—personal recollections and other testimony that has shed much light on how the ENIAC came into being.

In sorting out and evaluating those controversies, Stern focuses on the entrepreneurship of Eckert and Mauchly—how they sought funding for their projects, how they dealt with the demands of their various government and private customers, and so on. Although circuit diagrams and technical specifications are given for each machine, they are presented without comment or explanation and thus are of only limited value to the narrative as a whole. For the reader who is unfamiliar with some aspects of computer design, the focus on external factors can be confusing. For example, the reader is never told why the stored-program principle is so important. On the other hand, Stern shows how questions of marketing and funding were just as important as more technical matters in pointing the way toward the modern computer. Consider the UNIVAC as an example; it turns out that a critical breakthrough was one of salesmanship—Eckert and Mauchly had to convince potential business customers that they could entrust their records to invisible specks of oxide on magnetic tape, instead of to more tangible (and always hand-readable) holes in punched cards. Needless to say, customers were reluctant to take such a leap into the unknown, but without the use of high-speed input-output devices the electronic speeds of the computer's processor could hardly be exploited.

From ENIAC to UNIVAC is well illustrated and carefully documented, with references to the court proceedings mentioned above, to personal interviews with most of the surviving participants, and to an extensive list of primary and secondary documents. In addition, an appendix reprints in full von Neumann's "First Draft of a Report on the EDVAC," one of the first and still one of the most complete discussions of the logical design of a digital computer. (Von Neumann's use of a neural notation to describe a computer's circuits seems dated by today's standards, but it is remarkable how much of what is thought of as new in the computer world is contained in that draft of 1945, before any stored program machines were in existence.)

If Stern's study of the emergence of the computer can be faulted, it is because she has given the reader too narrow a perspective in judging Eckert and Mauchly. She is overly concerned, I think, with establishing their claims to the invention against those of von Neumann, as the latter's are presented by another member of the ENIAC team, Herman H. Goldstine (Stern quotes Goldstine's book *The Computer from Pascal to von Neumann* extensively in nearly every chapter). In treating that controversy she neglects other rival claims that might be just as valid. What about the claim by the British that their COLOSSUS, and not the ENIAC, was the first electronic computer? Or IBM's claim that its SSEC, completed in 1948, was the first computer to store its program internally? Once again, there are limits to the usefulness of trying to establish "firsts," but it should be emphasized that there was a lot of activity going on in electronic computation outside of Philadelphia.

From ENIAC to UNIVAC is the third in the History of Computing Series from Digital Press and is, in my opinion, the

best of the lot. It deserves a wide readership, both among historians of science and technology and among computer professionals, for whom the breathtaking pace of innovation and impact of computing have overshadowed the remarkable personal story of its beginnings.

PAUL E. CERUZZI
Department of History,
Clemson University,
Clemson, South Carolina 29631

Software

History of Programming Languages. Papers from a conference, June 1978. RICHARD L. WEXELBLAT, Ed. Academic Press, New York, 1981. xxiv, 760 pp., illus. $45. ACM Monograph Series.

Modern computing systems are as much a product of software as of hardware. One of the most important parts of such software is the set of compilers and interpreters that enable a computer to be programmed in a so-called high-level language. The first such programming languages date from the early 1950's, and their history is as interesting as that of the computers whose usability they so greatly increased. In 1977, when the conference of which the present book is a record was held, comparatively little had been done to record or study this history. The conference was in fact a deliberate attempt to remedy this situation, at least with regard to a selected set of programming languages, namely ALGOL 60, APL, APT, BASIC, COBOL, FORTRAN, GPSS, JOSS, JOVIAL, LISP, PL/1, SIMULA, and SNOBOL. The reasoning behind this choice from among the literally hundreds of programming languages that have been designed and implemented in the last 30 years is quite understandable. Though each language had been introduced at least ten years earlier, all were still in active use and had undoubtedly had a major influence on the field of computing. Indeed, most present-day computer users will be familiar with, and have had their view of how computers should be programmed colored (for better or worse) by, one or another of the chosen languages. Very rarely, however, will they have any clear idea of the circumstances surrounding the development of the language or of the motivations and intentions of its developers. Any such user with even a modicum of interest in the past should find the present book fascinating, consisting, as it does, largely of accounts given by the original developers themselves. For example, FORTRAN programmers will learn that one of the major spurs to the development in 1954 of their language by a team at IBM led by John Backus was the advent of hardware for performing floating-point arithmetic. COBOL programmers will learn that its designers thought they were developing in 1959 just "a short-range composite approach (good for at least the next year or two) to a common business language for programming digital computers." And ALGOL programmers will get, all too clearly, an impression of the intensity of the debates that occurred in the international committee whose deliberations led to Peter Naur's magnificent ALGOL 60 Report.

This book is not, however, a mere collection of personal reminiscences by pioneers. Rather it is the outcome of a carefully organized process that sought to maximize the historical value of the conference. Thus for each language one, or in some cases two, of the leading figures in its original conception and development were invited to prepare a detailed written account of the origins of and rationale behind the design of the language. Each was given guidance as to the information it was hoped his or her paper would contain. This guidance, in the form of a lengthy questionnaire, covered both general matters and specific technical questions relative to the particular language and was complemented by a careful and constructive reviewing process.

Each author was, as a result, motivated to supplement his or her personal recollections by undertaking extensive historical research. The resulting accounts are all excellent and full of fascinating and often surprising information, though they are far from uniform in style and differ greatly in emphasis. For example, the accounts of languages designed by committees, such as COBOL and ALGOL, tend to stress the often painful process by which agreement on the various major features of the languages was reached. In contrast, the papers on languages that were essentially, at least initially, the product of a single individual typically concentrate more on the reasoning behind the various detailed technical decisions that were made; a prime example of such a paper is that by Kenneth Iverson on APL.

The papers were made available in draft form before the conference, at which ample time was provided for discussion and questions. The published proceedings include, in addition to the final versions of the papers, transcripts of the actual conference presentations and of the discussion sessions and technical summaries of each language. These last play an important role in helping readers to appreciate the accounts of programming languages with which they are unfamiliar. The book, therefore, provides, as was hoped, a very valuable and readable source of historical information on the development of some of the most important and influential programming languages. Yet for all its serious intent, the conference was obviously an enjoyable and entertaining affair, and this is well reflected in the present volume. Thus this is a book that should appeal not just to people with a serious interest in the history of computing but to anybody who has experienced the delights and frustrations of computer programming and who has an appropriate curiosity as to the origins of the language or languages that provide the arena for his or her programming exertions.

BRIAN RANDELL
Computing Laboratory, University of Newcastle upon Tyne, Newcastle upon Tyne NE1 7RU, England

Vertebrate Morphology

Basic Structure and Evolution of Vertebrates. ERIK JARVIK. Academic Press, New York, 1980. In two volumes. Vol. 1, xvi, 576 pp., illus. $94.50. Vol. 2, xiv, 338 pp., illus. $56.50.

Vertebrate morphology is not a fashionable subject. The textbook most commonly cited is still E. S. Goodrich's *Studies on the Structure and Development of Vertebrates* (1930), now more than 50 years old. The heyday of the subject was over by about 1920, when it seemed that the comparative anatomy and embryology of vertebrates were well enough understood for the framework of morphology to be permanent, so that attention could be turned to newer fields. Then in 1921 Erik Stensiö published the first of a series of brilliant monographs, analyzing fossil lower vertebrates in an entirely novel way. Before Stensiö fossil fishes had been treated much like fossil invertebrates—as shells, whose external features were sufficient for diagnosis. Stensiö's innovation was to treat fossil anatomy in the same detail as is found in classical morphologists' work on Recent fishes. By new and painstaking methods of preparing fossils and by close comparison with Recent adults and embryos, Stensiö reconstructed not only bone but cartilage, nerves, vessels, and muscles in group after group of Paleozoic and early Mesozoic fishes. Goodrich viewed Sten-