C. Gordon Bell was born in Kirksville, Missouri in 1934. He received the BS degree in Electrical Engineering in 1956 and his MS in 1957, both from Massachusetts Institute of Technology, Cambridge.

In 1959 he was with the Speech Communications Laboratory at the MIT Division of Sponsored Research, then worked as a Research Engineer with the Electronic Systems Laboratory at MIT. From 1960 to 1966 he served as manager in charge of computer design at the Digital Equipment Corporation, Maynard, Massachusetts. He is at present a Professor of Electrical Engineering and Computer Science at Carnegie-Mellon University, Pittsburgh, Pennsylvania. He is a coauthor of the book, *Computer Structures: Readings and Examples*, McGraw-Hill, 1971. His research interests include general systems design and design management, design of multiple processor computer systems for either parallel or multiprocessing and design automation.

Mr. Bell is a member of the Association for Computing Machinery and Eta Kappa Nu, and a Senior Member of IEEE.

JACK McCREDIE was born in Pittsburgh, Pennsylvania in 1940. After receiving a BS and MS in Engineering and Applied Science at Yale he worked with the Department of Defense in Washington. He joined the Computer Science Department at Carnegie-Mellon University in 1968 after more graduate work at Carnegie's Graduate School of Industrial Administration. He teaches courses in simulation and operations research techniques. Present research interests include probability and simulation models of computers, scheduling theory, and measurement and evaluation of computer systems. Jack has done consulting work with the RAND Corporation, IBM, and a number of small businesses. He is a member of the Association for Computing Machinery and The Institute of Management Science.

The impact of minicomputers on simulation—an overview

by C. G. BELL and
J. W. McCREDIE
Department of Computer Science
Carnegie-Mellon University
Pittsburgh, Pa.

INTRODUCTION

The minicomputer population explosion impacts all facets of computing: manufacturers, users, and theorists. A 1969 review of capabilities and performance characteristics of digital computers in the up-to-$20,000 price range (published in *Modern Data*[1]) listed 35 different systems manufactured by 26 different firms. Only one year later a similar survey in *Computer Decisions*[2] covered 77 systems manufactured by 48 firms. Presently there are approximately 20,000 operational minicomputers. The proliferation of low-cost computers having increased capabilities allows users to reformulate computing habits. Problems inherent to the design of a large complex centralized computer have caused users to question the trend to one general facility serving all needs of a diverse user community. A benefit of large size and great generality is the ability to respond to a wide and unspecified spectrum of applications, but the price is often decreased cost effectiveness for specific tasks.

Some computing requirements can be met only on very large computers, while others may be met in a number of diverse ways. Recently minicomputers have solved problems that were previously thought to be exclusively in the domain of larger machines. For example, a program written for a minicomputer came in second in the 1970 ACM computer chess tournament. We normally think of artificial intelligence applications as requiring large primary memories. A fast minicomputer with effective management of secondary-primary memory does very well.

Before discussing the minicomputer's impact on simulation, we shall make its definition more precise.

MINICOMPUTER CHARACTERISTICS

Minicomputers are a state of mind. Current logic technology and selected characteristics of larger machines are combined in a low-cost package. Manufacturers emphasize different capabilities in their resolution of the conflicts between lower cost and higher performance. Minicomputers may be classified at least two ways:

1. It is the minimum computer (or very near it) that can be built with state-of-the-art technology.
2. It is that computer which may be purchased for a given, relatively minimal, fixed cost (*e.g.*, $10K in 1970).

The first definition would probably preclude hardwired floating-point arithmetic from being part of a minicomputer, while the second definition would convey the attitude of the larger general-purpose computer.

Several dimensions of a computer space (Figure 1) characterize today's minicomputers. The dimension values are: a mainframe cost of four to ten 1971 kilo-dollars, a 12- to 16-bit wordlength (as measured between the processor and primary memory), a minimum 4-Kword memory with addressing capability to 32 or 65 Kwords, and a processor state of about four words (program counter, accumulator, accumulator extension, and possibly an index register).

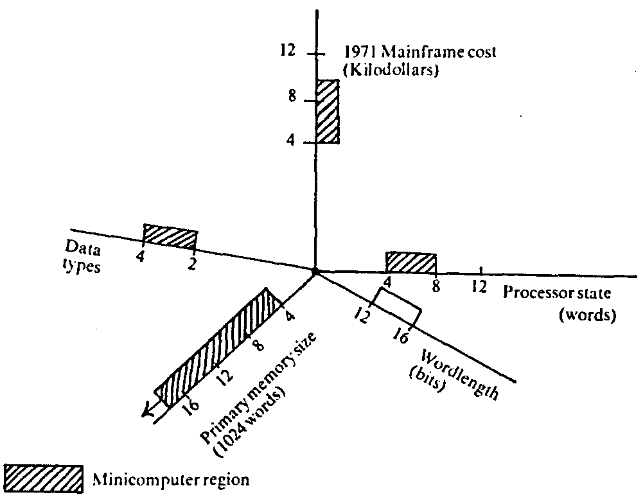

Figure 1 – Some computer-space dimensions

Common characteristics for nearly all 1971 technology minis are: small-to-medium scale integrated circuitry, core primary memory (with MOS memories likely to be delivered during the year, memory cycle times of 0.7 to 2 micro-seconds with correspondingly fast processor times, internal clock rates of 5 to 10 MHZ, and uni-processor structures.

Characteristics less easily defined are the processor's data types. Initially, one-word integers and one-word boolean vectors (usually called logical words) were available. Vectors and character strings are accessible indirectly via index registers. Data types for byte and double-word integers are now common, and floating-point undoubtedly will be added.

Attributes have been acquired from larger computers (see Table 1) because of necessity (*e.g.*, interrupts and some form of base addressing) or utility (*e.g.*, index registers, general registers, and specialized processors). If a feature on a large machine enables it to perform well, some minicomputer will adopt it.

Table 1 – Characteristics derived from larger computers

Internal central processor
Indirect addressing; interrupts; index registers; multiple, general-purpose registers; base and/or page addressing; floating-point data-types*; paging, segmentation, and inter-process communication*

Structural
Specialized processors, multiprocessors*

Implementation
Microprogramming*, lookahead*, cache*

Software
Compilers (e.g., Fortran) time-sharing monitors*

*Not extensively used, but use likely to increase

THE OBSERVED IMPACT

The interdisciplinary nature of simulation causes a wide range of computing demands. Applications require analog, digital, and hybrid equipment. Since human interaction with simulations ranges from all-man to all-machine applications, time-sharing and graphical capabilities have been used since their introduction. Simulators have been quick to adapt any new computer technology to their problems.

This issue of *Simulation* assesses the impact minicomputers are having along different dimensions of the simulation space, and it projects some future effects we may observe. Our goal is to expose readers to a rapidly changing technology and to indicate how analysts have adapted to its capabilities and limitations.

In the initial call for papers we asked for papers from the following areas:

1. The use of minicomputers for modeling and simulation.
   a. Hybrid computation. Without minicomputers, because of cost reasons, the hybrid computer would probably not exist.
   b. Continuous and discrete system simulation. We commonly think of only large computers having this capability. Clearly minicomputers can be used for some simulations, but are they practical considering cost and performance?
2. The need for modeling and simulation of minicomputer architecture. Such simulation may include the instruction set, memory-processor-peripheral organization, registers, logic, memory-hierarchies (caches), etc.

3. Modeling and simulation of systems which use minicomputers. Some systems would not exist without minicomputers. These systems are typically concerned with process control, data communications, traffic control, small-scale time-sharing, instrument control, etc., all of which require modeling and simulation.

We appealed to authors to include any other relevant areas in order to help decide whether there is, has been, or will be an effect of minicomputers on the art and practice of modeling and simulation. The call for papers was not very widely circulated, the deadline was short, and the response seemed to be relatively small. (We do not have comparative data for other similar calls.) We are not prepared to say there has been no impact; but we have not measured it as well as we would like. The range of papers in this issue indicates that there will be many future minicomputer applications to the field of simulation.

We tried other approaches to measure the impact. IBM has two computers, the 1130 and the 1800, which could be classified as minicomputers by their technical characteristics and performance, although they are too expensive to qualify. These computers have been delivered to organizations who use them for simulation (*e.g.*, small technically-oriented companies, and universities). The simulation packages for these computers include Continuous System Modeling Program (CSMP), and General Activity Simulation Program (GASP). Therefore, we conclude that users are probably simulating on these machines. On Digital Equipment Corporation's PDP-8, 11, and 15, only standard languages are available: the two interactive languages BASIC and FOCAL, and FORTRAN. Users have contributed CSMP and GASP. FOCAL is the basis of one of the simulation languages described by Sias and Coleman in this issue. In addition, we expect conventional languages are used without modification for simulation. For example, the Carnegie-Mellon University Electrical Engineering Department has a PDP-8, running only FOCAL, for the sole purpose of modeling electrical circuits in sophomore and junior courses. Here the mini is performing the same role as that of a console on a large general-purpose computer, but at a considerably lower cost. The campus could buy two to three *hundred* PDP-8's for the same price as *one* of our large general-purpose computation-center computers with its 30 on-line users.

Our findings are:

1. Modeling and simulation are done directly via minicomputers (and sub-micro-computers). The most impressive report, considering models per computer dollar, is the work of Spain (Michigan Technological University) using an Olivetti Programma 101 desk calculator (which is actually a stored-program computer with approximately 100 digits of storage).
2. Simulations are implemented using modified interactive languages. Sias and Coleman (University of Mississippi) have modified DEC's interactive FOCAL language to include continuous modeling elements.
3. New languages have been created for continuous system simulation. Benham (Interactive Mini Systems, Inc.) has already developed an interactive language for a minicomputer and is planning to implement it on other machines. The facility does not use floating point and users must do time- and amplitude-scaling of the model. However, for this inconvenience they are able to use a minicomputer for a number of applications at about the same speed as a large-scale machine with floating point hardware.
4. Minicomputers are used with analog computers for hybrid computational facilities. These facilities allow users to solve special classes of problems which require large amounts of computation if done on a digital computer. Paul (Carnegie-Mellon University) discusses the role of a hybrid facility.
5. Larger machines are being used to simulate minicomputers. Mallach (Computer Usage Company) discusses the problem of developing software for minicomputers and presents techniques for simulating minicomputer systems. As minis become more complex and their architectures include such features as cache memories and multiple processors, simulation will become a necessary design tool.
6. Interesting systems using minicomputers require simulation for design and optimization. For example, at the University of Wichita, a two minicomputer time-sharing system is being simulated as it is designed. The simulation runs on a medium-scale computer, using the GPSS language. Process control applications often require simulation during the design or checkout process.
7. The minicomputer has been used as a simulator of physical equipment. Although a paper which discussed this application was delayed and cannot be published, the author used a mini as a universal waveform generator. He also carried out waveform analysis, Fast Fourier Transforms, and convolutions. Minis have also been programmed to be music synthesizers.
8. The minicomputer has been used to simulate (or emulate) general-purpose computers. High instruction execution rates, and relatively constant input-output equipment speed, allow minis to simulate older business computers, such as the IBM 1401, in a production environment. Such systems already

exist. When future very-high-speed minicomputers are constructed, they could be used like microprogrammed computers to emulate other computers. This approach would be a flexible alternative to direct conventional implementation of a large general-purpose machine.

Some effects we had expected to see, but for which we received no response, include the following:

1. Discrete event simulation applications. We believe the reason that this type of simulation is not common on minicomputers is that the models are often large and would exceed the minicomputer's primary-memory capacity. However, continuous applications and languages are available, and minis do not presently have floating-point hardware, a common specification for continuous models. Clearly many discrete simulations of limited size could be carried out as fast on a minicomputer as on a large-scale computer of comparable memory speed. Interactive use of minis is often very cost effective when compared to large-scale timesharing systems. With proper memory management discrete simulation can be done effectively.
2. Discrete simulation languages such as SIMSCRIPT, SIMULA, or GPSS. GASP for the IBM 1130 does qualify for this type of language, but it was not designed for a small machine. Since many of the operations carried out in discrete simulations are of a list-processing nature and do not require floating-point arithmetic, a language designed expressly for a mini would encourage discrete modeling. GPSS/360, one of the most widely used discrete languages, is based upon integer arithmetic. A subset of GPSS for a minicomputer would be both feasible and welcome.
3. Boolean algebra or register transfer simulation languages for logical design. The minicomputer is fast enough, but a simulator for reasonably large problems, together with run-time packages, might be quite large. However, for educational use, the minicomputer would be adequate.

CONCLUSION

The most probable evolution of minicomputer structure will be to include more features of current larger computers (see Table 1). For example, future minicomputers will have larger primary memories and built-in floating-point arithmetic, both of which are prerequisites for larger, more general-purpose simulations. The minicomputer industry is growing at a fast rate. To sustain this expansion, manufacturers will probably look closely at important application areas such as simulation as they design new equipment.

The papers in this issue of *Simulation* represent only a small fraction of the potential applications of minicomputers to the field of simulation and modeling. The present impact, although not revolutionary, is observable; the future impact will be more significant.

REFERENCES

1 COHEN J W
Technology profile — minicomputers
Modern Data August 1969

2 *Mini-menagerie*
Computer Decisions August 1970