A panel session—Education of computer professional

ELLIOTT I. ORGANICK, Chairman of Session

University of Houston; currently at
Massachusetts Institute of Technology
Cambridge, Massachusetts

Inter-relating hardware and software in computer science education

by JACK B. DENNIS

Massachusetts Institute of Technology
Cambridge, Massachusetts

The major portion of graduates from curricula in computer science will be professionally involved in the design, specification, implementation or theoretical foundations of computer-based information systems. They will participate in the selection of computer hardware, or will be called on to judge the merits of proposals from suppliers. To be competent in exercising these responsibilities, it is essential that students of computer science thoroughly understand the relationship between computer organization and the implementation of programming languages and information systems.

Given this objective, there is a serious anachronism in the teaching of programming and computer organization in contemporary university curricula: Computer organization is often taught as the final example in a course on logical design by instructors who do not profess knowledge of compiling techniques and software issues. Conversely, programming courses are based on conventional assumptions of computer organization (Von Neuman) as if they were axioms of nature. Moreover, the communication paths that could lead to reorganization and accommodation of the intellectual substance of both areas are frequently blocked by circumstances: Either the areas are the “property” of separate academic departments, or the faculty is divided by disparate interests.

There is critical need for cooperation between faculty in programming and in computer organization to jointly develop curricula that interrelate hardware and software principles for realizing the functional requirements of computer systems. The ACM Curriculum Proposal does not represent sufficient progress toward this objective.

In the undergraduate Computer Science program of the M.I.T. Department of Electrical Engineering, we have developed a three subject sequence in computer systems and programming intended to interrelate software and hardware principles:

1. Programming Linguistics
2. Computation Structures
3. Information Systems

Students enrolling in the sequence are presumed to have had the experience of expressing programs in an algebraic language and seeing them run (with success and failure) at a computer installation.

The first subject, “Programming Linguistics,” treats the important concepts in describing and interpreting algorithmic procedures on the basis of a formal semantic theory. Features of practical programming languages are related to the theory. Discussion of hardware is deliberately omitted so an unencumbered appreciation of linguistic principles can be achieved.

In “Computation Structures” the student learns the properties of memory and logic components that interact strongly with the process of planning a computer organization. A graph model of parallel computations is used both to describe modular hardware systems, and, as a starting point for developing combined