By the time April, 1975 rolled around, most of us who are any way connected with microcomputers were up to our ears in papers, articles, and specifications on microcomputers, and inundated with announcements of seminars, conferences, courses, and tutorials on the same subject. Why, then, hold a workshop on "The Advanced Architecture and Applications of Microcomputers?"

We — Ted Laliotis (then at Fairchild, now at Atmospheric Sciences), Fred Clegg (then at Santa Clara University, now at Hewlett-Packard), Don Senzig (then at Hewlett-Packard, still at Hewlett-Packard), and I, among others — thought that there was a lot of valuable information which, by its very nature, would never appear in any of the standard communication channels. This information existed in the personal philosophies, experiences, gripes, hopes, frustrations, and insights about microcomputers of the experts in their design, manufacture, programming, and applications.

These thoughts had not appeared in formal presentations or publications because they were controversial, or unconfirmed, or incomplete, or con-
ceptually not quite clear enough to express. And yet, it seemed to us that these very valuable concepts could benefit by open, informal discussion, among peers, which might settle (or heighten) a controversy, confirm (or disprove) a theory, complete (or dissolve) a concept, or clarify (and expand) an idea.

We concluded that an IEEE Computer Society workshop, perhaps modeled after the very popular Lake Arrowhead Workshops, would provide exactly the forum that was needed. So, we isolated five topics of interest, recruited five top people in their respective fields to chair sessions on those topics, and we were on our way.

Session Summaries

The first session, chaired by Ken Rothmuller (Hewlett-Packard), dealt with the “Strengths and Limitations of Current Microprocessors from the Systems Viewpoint.” Several case histories were presented, complete with advances and pitfalls, and really put the current systems state of the art into focus. Ken has incorporated some of the lessons from that session into a paper which examines multi-tasking techniques and suggests future directions for microprocessor design from the systems viewpoint.

The second session was included by popular demand. It seems that practically everyone was involved with, or at least interested in, “Multi-Microprocessor Systems.” John Wakerly (Stanford University) chaired that session, which ran the gamut from practical to theoretical considerations of microcomputer networks. Barry Borgerson (Sperry Research) has written a paper which presents some basic ideas pertaining to the architecture of multi-microcomputer systems.

The third session “Technology Constraints, Present and Future,” was chaired by Peter Verhofstadt (Fairchild Semiconductor). It gave unique insights into what had happened, was happening, and was probably going to happen in terms of microprocessor device technology. Peter has ably summarized his session in a paper for this issue.

Terry Opdendyk (Intel) chaired the fourth session entitled, “Software Considerations of Microprocessors,” which covered a wide range of software topics. Terry has included an overview of this session in which he emphasizes the ever-expanding role of software and its attendant technological implications.

The last formal session, “Microcomputers of the Future: Unique Applications,” was chaired by Bob Winder (RCA). It provided a fitting conclusion to the workshop by graphically demonstrating that some of the “futuristic” applications of microprocessors were already here, or at least a lot closer than many of us may have thought. Bob’s summary of that session is included.

One evening was set aside for an ad hoc session just in case some additional topics might have arisen from the earlier sessions. The response was enthusiastic and the session ran late into the night. Some of the topics covered included interfacing problems, performance measurement, bit serial processors, software transportability, and very large structured arrays. Several lively discussions took place even after the formal session had ended.

Questions and Answers

One of the privileges of being an editor is that one gets to editorialize. So, here are some of the issues raised that I think are particularly important.

First, a question that was raised in my mind by an early presentation: Is there a design philosophy, a set of design guidelines, that is unique to microprocessor-based systems? If so, how do we describe it, learn it, and teach it?

Another question: Why aren’t microprocessors being used more in “smart” peripheral controllers? After all, that was one of their early great expectations. What can and should be done to make a microprocessor more suitable to control, for example, a floppy disk with a minimum of external hardware? A clear example is that floppy disk controllers for microprocessor-based minicomputers use random logic for their controllers!

A new dimension of controversy about microcomputer software was pointed out at the workshop. One view expressed was that programming a microcomputer was something that a designer could pick up without any formal training in software, and that most software tools and techniques are not really needed in most microcomputer applications. Exactly the opposite view was presented in a later session: “Anyone who writes a program is a programmer and should be aware of and adhere to good programming practice!”

How do we reconcile these views? And what impact do they have on the programs that have been and will be written for microcomputers? Also, what effects should they have on software support, and on the education process?

Several points were made which may have been heard before, but bear repeating here:

1. The need for efficient communications between hardware and software personnel is especially critical during microprocessor-based system development.

2. Be careful when trying to improve the price/performance ratio. Remember that it is usually much easier to raise the performance than it is to lower the cost.

3. Beware of creeping elegance, especially in microprocessor systems.

One question raised stands out over the others: what happens when a glass of beer gets spilled all over a microprocessor? This is after all, one of the potential problems, along with physical abuse and static shock hazards, we will encounter in the design
This book develops a sound introduction to linear systems theory for both discrete and continuous systems. The text is suitable for three distinct groups of students: advanced engineering undergraduates; beginning graduate students; and practitioners of automatic controls. It introduces the basic systems concepts and, in addition, prepares the student for more advanced systems topics such as optimal control, estimation theory and system identification. Many worked examples are presented and the text is written on a level that students can understand.

Price: $16.95

INTRODUCTION TO LINEAR SYSTEMS ANALYSIS
George M. Swisher
This book develops a sound introduction to linear systems theory for both discrete and continuous systems. The text is suitable for three distinct groups of students: advanced engineering undergraduates; beginning graduate students; and practitioners of automatic controls. It introduces the basic systems concepts and, in addition, prepares the student for more advanced systems topics such as optimal control, estimation theory and system identification. Many worked examples are presented and the text is written on a level that students can understand.

Price: $16.95

Fred F. Coury is an independent consultant in digital systems, specializing in the hardware aspects of microprocessor-based systems design and applications.

Previously employed by the Hewlett-Packard Company from 1969 to 1973, he was the project manager of the HP2100A Minicomputer Development Project, and later manager of the Minicomputer Engineering Section.

From 1963 to 1969 he was an associate research engineer and lecturer in the Department of Electrical Engineering at the University of Michigan, where he developed and taught courses, and developed laboratory facilities in the digital systems area. During that time he was involved in the design and implementation of the data concentrator at the university’s computing center. He was also the principal investigator on a smart terminal development project in the university’s industrial sciences group.

Earlier, as a research assistant at the university’s Brain Research Laboratory, he developed analog and digital systems for on-line analysis and control of behavioral research experiments.

Coury received the BSE degree in science engineering and the MS in systems engineering from the University of Michigan at Ann Arbor, in 1963 and 1967, respectively. He has written and lectured extensively in the areas of digital system design and applications and has published one book—A Practical Guide to Minicomputer Applications (IEEE Press, 1972).