Newly-appointed technical sessions editor John Hiles (see Update) attended the Asilomar Workshop on Microprocessors held at the end of April; his report is presented below. As with most Computer Society workshops, which attempt to encourage free and candid communication among participants, this one published no proceedings and in fact prohibited tape recorders as well as photographs of slides. In the spirit of observing that requirement, Hiles' commentary is kept at a fairly general level. However, without attributing specific comments or discussing specific presentations, it does summarize the major thrusts of the discussions.

The Second Asilomar Workshop on MICROPROCESSORS

John Hiles
Rockwell International

As microprocessors continue their spread in all directions into technology and applications—in the process reaching areas that were previously untouched by computers—they give rise to whole new orders of problems, approaches, and solutions. These were the topics of discussion among the hundred-odd practitioners in the field who attended the Second Asilomar Workshop on Microprocessors, held April 28-30 at the Asilomar Conference Center near Monterey. Sponsored by the Western Area Committee of the IEEE Computer Society, the workshop focused this year on the theme of "Unique Aspects of Microprocessor-Based Systems."

Parallel to the trend toward delivering systems to new settings is another trend of developing systems in new settings. Absent from some of these development settings are most of the accoutrements that just a few years ago conveyed the unmistakable message, "Computer Industry Here." Just as the minicomputer moved development beyond the raised floor, microcomputers have moved it beyond the Teletype. Large differences can separate development environments: at the low end, with a small budget and staff, development tools may consist of not much more than a coding pad and PROM programmer; in contrast, a variety of specialized, computer-based tools are available for the sophisticated high-volume or software-intensive development effort.

Presentations covered several types of new ICE (in circuit emulation) development systems—the microprocessor programmer's equivalent of an oscilloscope. Though debate on development tools spilled over into the related issue of hand assembly of code versus compilers, the apparent conclusion was that the range of development environments was too broad to be satisfied by any single approach. Adequacy of support tools depends on the size, complexity, and budget of each project.

In addition to new development tools and environments, new developers are producing microprocessor-based systems. Software they had written in a simple development environment was referred to in the workshop as having come from the engineer's bench. Similarly, software produced with the aid of editors, debuggers, operating systems, and perhaps compilers was labeled as a product of the software laboratory. Not for the first time in the mixed company of programmers and engineers, the relative merits of software from the bench or from the laboratory was a vexing issue. But as one participant stated, the essential differences do not separate engineers from programmers; rather, they distinguish good engineering from bad engineering, good programming from bad programming: hardware and software should be documented, testable, and correct.

The diversity of programming methods at work in the world of microprocessors was brought home in one session, when a presentation on machine-oriented higher-level languages immediately followed the description of a procedure for hand assembling programs into absolute hexadecimal code. One useful technique that has been used on several projects involves the concept of "programming into." The absence of a clear and suitable higher-level language on a project (whether a result of choice or necessity) does not prohibit the use of a high-level notation in understanding programming problems and defining their solutions. Once defined, the notation can be manually translated into whatever form required. Just as applicable to descending from flowchart to symbolic instruction to machine language as from some Fortran superset to Fortran, "programming into" is based on sound principles of considerable utility.

Because of the expense of programming even moderately large (i.e., greater than eight or ten thousand instructions) microprocessor-based systems, many participants commented on the need for processor architectures that would provide improved support for programming. Other attention focused on the benefits of balancing current trends toward increased speed and memory capacity with development of processors whose distinguishing characteristics would be their simplicity and consistency.

Whereas programming expense and processor capacity may determine whether or not an application is built, an equally important aspect of the system—serviceability in the
field—influences the growth and lifetime of the product. As the ratio of system cost to service cost continues to decrease, this influence will strengthen. Self-test capabilities, based on PROM resident routines, were cited by many system builders as an effective means of limiting the expense of field service. A built-in self-test capability allows systems to verify that their components respond properly to test inputs. Emphasis was placed on the importance of planning for self-test capability at the beginning of the design process.

The rapid changes in microprocessor capacity, cost, and configuration are nowhere more evident than in the development of the hobby computer. After tremendous growth in the last year, representatives of that market speculated on the promising future of their products and compared the prospects favorably with the spread of home stereo and amateur radio equipment.

Toward the end of the workshop, one participant considered the rush of events during the microprocessor’s brief history. He recommended that certain intellectual equipment was indispensable to an engineer operating in the midst of this acceleration of change and diversity:

- competence in defining a problem;
- ability to renew technical knowledge;
- healthy skepticism toward answers, manuals, textbooks, and teachers.

The workshop’s program committee is planning the Third Annual Asilomar Workshop, which will convene in April of 1977.