This digest features 74 papers from 15 conference sessions on topics such as microprocessor testing, memory testing, self-testing, CODEC testing, board testing, measurement and calibration, economics and architecture. Also included are eight papers on system testing, an emphasis requested by attendees of the 1980 conference. 568 pp.

Order #375
Digest of Papers—1981 INTERNATIONAL TEST CONFERENCE
October 27-29, 1981
Members—$27.00
Nonmembers—$36.00

Use order form on p. 95.

This tutorial addresses a broad definition of the software development environment. Thirty articles originally presented at Compac 81, are sectioned to cover goals, language-based environments, Unix, methodologies, configuration, human/computer interaction, organization and ergonomics, as well as personal systems. 476 pp.

Order #385
Tutorial—SOFTWARE DEVELOPMENT ENVIRONMENTS
Edited by Anthony I. Wasserman
1981
Members—$18.75
Nonmembers—$25.00

Cont'd from page 23

to "reverse compile" an object code to a unique high-level source code.

"Under the 1976 act...physical objects that do not communicate messages to human perceivers are not 'copies.' Source programs are directly perceived by and communicated to those who can understand the 'dialect' of English in which they are written." (Page 17)

Object (machine) code was originally the only means of communicating programs to human beings, though it is seldom used for that purpose today.

"The other [example] is 'to add features to the program,' again a procedure that calls for a human being to revise a program intelligible to him, which necessarily means doing so by dealing with an intelligible copy (i.e., a source program, not object code.)" (Page 17)

This is certainly done at the assembly-language level, but we also patch programs directly at the object-code level.

"A derivative work is a new work of authorship based on, and containing perceptible and recognizable elements of, an earlier work. Examples are a translation of an earlier work...." (Page 17)

Here the point that assembly language is source code seems important, since object code is, in fact, a direct translation of assembly language statements. The question of human intelligibility arises again.

"...copyright law distinguishes between useful or utilitarian articles, on the one side, and articles that communicate information between persons on the other. Chairs, saltcellars, and light fixtures are examples of the former; books, musical scores, maps, and posters are examples of the latter." (Page 18)

We do not copyright the physical form of a book, but its contents. The bit pattern conveys the information in the ROM in coded form. Is an encrypted version of a book subject to the original book copyright? Is an encrypted version of a program subject to the original program copyright?

"We believe that intended use—utilitarian or communicative—is the touchstone for determining