

Electronic Composition in Printing

Proceedings of a Symposium



U.S. Department of Commerce
National Bureau of Standards
Special Publication 295

UNITED STATES DEPARTMENT OF COMMERCE
Alexander B. Trowbridge, *Secretary*
NATIONAL BUREAU OF STANDARDS • A. V. Astin, *Director*

Electronic Composition in Printing

Proceedings of a Symposium
National Bureau of Standards
June 15-16, 1967

Edited by

Richard W. Lee and Roy W. Worrall

Center for Computer Sciences and Technology
National Bureau of Standards
Washington, D.C. 20234



National Bureau of Standards Special Publication 295

Issued February 1968

System 70

John W. Seybold, President

Rocappi, Inc.

I wish to compliment all of those who have played a part in the planning and carrying forward of this most successful conference on electronic composition. The large turnout is eloquent evidence of the escalating interest in this subject.

A preliminary comment I would like to make is that this meeting points up the fact that insufficient attention has been paid to the interrelationships between the computer processing of information generally, and the typesetting application. What conventions can be established to make it generally possible to take data from computers, wherever situated, and create a typeset product, by means of all-purpose typesetting software? What conventions can and should be established to represent format instructions and to access the wide variety of typographic characters now available? This is a matter in which the National Bureau of Standards should concern itself, if it has not already done so. The adoption of specific code structures is more or less irrelevant, but the establishment of related "families" of type characters is a minimum prerequisite to effective interchange of information when there is a possibility that the end product will be photocomposed.

Not only must type-family conventions be agreed upon, but ways of signalling for these must be specified. Where information is described in fixed-field format it will not be difficult to precede this field with whatever relevant format and typesetting instructions are required, but it will also be necessary to provide for midfield flags. This is really not very difficult, but it will be necessary to modify standard programming applications so that such flags, and uppercase identifiers, will be ignored in normal data processing runs. The edit-insert programs which the Government agency users of the Linotron are expected to provide are of this character. Formatting for typesetting is not difficult where the information to be typeset is very limited in its typographic requirements and character set. But we should not permit ourselves to be constricted by unimaginative adaptations.

Let me turn, now, to my topic: an exposition of Rocappi's System 70. System 70 is so named because we conceive of it as a valid solution to the

problem of computerized photocomposition until the early or mid-1970's. It represents the third major rewriting of our basic programs since we came into existence in 1963. System 70 is the product of many minds, including those of us now with Rocappi and many who have gone before. Former key staff members are now with Rocappi, Ltd., with IBM, with the RCA Graphic Systems Division, and with Alphanumeric. We derive some satisfaction from the fact that some of our ideas and visions are becoming a reality not only in our own company but also generally throughout the industry.

System 70 is first of all a complete operating system with its own diagnostic routines, table-generating capabilities, its own assembly language, and its own sort procedures. It consists of more than 20 separate but related programs, many of the major ones, all of which are read into memory from the same program library tape, which usually remains mounted on its tape station for the entire day of research and production activity.

System 70 differs from our earlier System 65 in the following major respects:

- (1) Its table-generating capabilities.
- (2) Its ability to handle complicated tabular matter (up to ten columns across the page).
- (3) Its ability to move into and out of justification and hyphenation—that is, to substitute ragged setting for justified composition at any point in text processing.
- (4) Its recovery capabilities.
- (5) The fact that the system uses its own unique language, which is different from common machine language. But at least four different code structures are required for the processing of each job.
- (6) Its command over five different levels of hyphenation accuracy.
- (7) Its HYCOR or hyphenation verification procedures.
- (8) Its ability to nest indents within indents (up to 5 combinations) and to handle indents of a definite duration. Left, right, center, hanging, and variable indents are possible with a predetermined duration or until further notice.

- (9) Its ability to incorporate nontypesetting information at any point in the text, bypassing this information for purposes of typesetting.
- (10) Its castoff and pagination procedures (Phase III and Phase V), including the ability to correct pagination parameters following inspection of the proposed pagination solutions.
- (11) Its treatment of justified lines of composition as if they were slugs (Phase IV), thus permitting merging of lines for multicolumn setting, and rearranging line sequences, or line substitutions, at the last minute.
- (12) Its magnetic-tape-to-paper-tape procedures for producing paper tape output for a variety of different typesetting devices, including ways of keeping track of output on a galley-by-galley basis, starting and stopping the punching-out process at any designated galley, and controlling the length of each "take."
- (13) The corollary programs which
 - a. provide keyboard shorthand facility and the use of generalized macros which may be interpreted in a variety of different ways;
 - b. make possible extractions for further data manipulation: abstractions, explosions, permutations, etc;
 - c. permit indexing, with page references.

It is evident that System 70 consists of a comprehensive set of modular, interrelated programs which make up a total system. These programs are input-oriented, as indeed the entire system is; that is, they seek to simplify the problems of keyboarding and try to take into account the capabilities and limitations of keyboard operators. The System is operational without complicated markup and instructional procedures.

Let me digress to observe that I believe keyboard operators are not unskilled people, but re-

quire a high degree of skill. They should be well paid. For the most part, at least for the kind of business we do, they are creating a permanent file or data-bank, and it is worth any amount of pains to get this file correct. It will be used for reiterative typesetting, explosions, and updates for many years to come.

Our System is also based upon a keyboard philosophy which seeks to separate input from output so that there is in no sense a one-for-one correspondence of codes; we do not believe in "position" keyboarding.

All photo-units activated by our System are used as slaves. We do not permit any photo-unit to do its own justification. Complete flexibility is thus afforded, which usually goes beyond the normal capabilities of these typesetting devices.

It is possible to use the same input for any typesetting device capable of coping with the requirements of the job, whether or not the same font or grid configurations obtain. If ligatures are called for they will be generated. If accents are desired they will be accessed, either as floating accents or as distinct characters. If no accents exist the unaccented equivalent characters will be produced.

We are still working on more elaborate pagination solutions, but we are also planning to move as quickly as possible into the development of an entirely new series of programs for third generation computers. Some of the things we do now we will keep because we believe our solutions are generally valid. Some of the features will be changed. At present all programs operate within 20 K (6 bit) core memory with a minimum of overlays. We have four tape stations but no disk storage.

We do not plan to write our third generation programs so tightly. Nevertheless, we think they must be written for a computer configuration which can justify its installation solely for a typesetting and related data processing application.