BIBCON—A General Purpose Software System for MARC-Based Book Catalog Production

Liz GIBSON: Assistant Systems Analyst, California State Library, Sacramento.

The BIBCON file management system, designed for use on IBM 360 system equipment, performs two basic functions: (1) it creates MARC structured, bibliographic records from untagged input data; (2) from these records it produces page image output for book catalogs. The system accepts data from several different input devices and can produce a variety of output formats by line printer, photocomposition, or computer output microform (COM).

INTRODUCTION

BIBCON is a general purpose data management system for BIBliographic records CONtrol (i.e., for creating, manipulating, formatting and outputting of MARC structured bibliographic records from catalog card input data). The system, shown in Figure 1, consists of seven basic programs which functionally divide into two parts: (a) four programs for creation and correction of MARC-like records; and (b) three programs and an IBM utility sort for formation of book catalog entries from these records. Obviously, a detailed description of such a large and complicated system is impossible in one journal article. A detailed description of the system specifications and user instructions has been prepared and published by the California State Library.1

The BIBCON system was cooperatively developed by the Institute of Library Research, Berkeley; the Library Systems Development Project, Santa Barbara; and the Library Systems Offices at the Santa Cruz and Berkeley campuses of the University of California. The system was developed in response to the needs of the University of California (UC) and of the California State Library (CSL) for efficient production of author, title, and added entry listings of their monographic holdings for distribution to their respective clientele groups.

The general system requirements for both libraries were the same:

(a) With a minimum of expensive manual keying, bibliographic data
must be prepared for book catalog production, with any of the standard catalog entries as keys.

(b) Provision must be made for the widest feasible variety of columnar output formats.

(c) The format for any machine-readable records must be compatible with the MARC standard.

The system has been installed with revisions and modifications on an IBM 360 Model 50 computer used by the California State Library. All pro-
grams in this version are written in the IBM Basic Assembler Language (BAL) instead of the original combination of BAL and COBOL.

In its first version, BIBCON processed monographic records exclusively. Various programs have now been modified so that the system will also process serial records in a simplified MARC serials format. This article, however, will describe only the system for processing monographic records.

The system has been used to produce catalogs of monographs for UC Santa Cruz, UC San Diego, and the one million record supplement to the UC catalog of books. Portions of the system were used to produce the initial copies of the University of California Union List of Serials.

The California State Library Automation Project is using this basic file management system to process both monographic and serial records for the production of several book catalogs. These will include, principally, the California Union List of Periodicals, reflecting the periodical holdings of libraries throughout California, the California State Library List of Periodicals, and the Catalog of Books in the California State Library.

AUTOMATIC FIELD RECOGNITION (AFR)

At the heart of the system is the program which creates MARC-like records from unedited input data. This program, called Automatic Field Recognition or AFR, identifies control and variable fields and creates a leader and record directory for each record submitted to it. In order to accomplish this, when a record is submitted to the program, it first sets aside areas into which data for each of the four parts can be placed. The field
### Control Numbers
- **010** LC Card Number
- **011** Linking LC Card Number
- **015** National Bibliography Number
- **016** Linking NBN
- **020** Standard Book Number
- **021** Linking SBN
- **025** Overseas Acquisitions Number (PL 480, LACAP, etc.)
- **026** Linking OAN Number
- **035** Local System Number
- **036** Linking Local Number
- **040** Cataloging Source
- **041** Languages
- **042** Search Code

### Knowledge Numbers
- **050** LC Call Number
- **051** Copy Statement
- **060** NLM Call Number
- **070** NAL Call Number
- **071** NAL Subject Category Number
- **080** UDC Number
- **081** BNB Classification Number
- **082** Dewey Decimal Classification Number
- **086** Supt. of Documents Classification
- **090** Local Call Number

### Main Entry
- **100** Personal Name
- **110** Corporate Name
- **111** Conference or Meeting
- **130** Uniform Title Heading

### Supplied Titles
- **240** Uniform Title
- **241** Romanized Title
- **242** Translated Title

### Title Paragraph
- **245** Title

### Edition Statement
- **250** Edition Statement

### Imprint
- **260** Imprint

### Collation
- **300** Collation

### Bibliographic Price
- **350** Bibliographic Price

### Converted Price
- **360** Converted Price

### Series Notes
- **400** Personal Name—Title (Traced Same)
- **410** Corporate Name—Title (Traced Same)
- **411** Conference—Title (Traced Same)
- **440** Title (Traced Same)
- **490** Series Untraced or Traced Differently

### Bibliographic Notes
- **500** General Notes
- **501** “Bound With” Note
- **502** Dissertation Note
- **503** Bibliographic History Note
- **504** Bibliography Note
- **505** Contents Note (Formatted)
- **506** “Limited Use” Note
- **520** Abstract or Annotation

### Subject Added Entries
- **600** Personal Name
- **610** Corporate Name (excluding political jurisdiction alone)
- **611** Conference or Meeting
- **630** Uniform Title Heading
- **650** Topical
- **651** Geographic Names
- **652** Political Jurisdictions alone or with Subject Subdivisions
- **660** NLM Subject Headings (MESH)
- **670** NAL Subject Headings

---

*Fig. 3. Variable Field Tags—LC-MARC II*
identification progresses on the basis of two signal symbols which are inserted between fields during input and on the basis of the order and content of the fields. When a control or variable field is identified, a standard MARC record directory entry is created, containing the AFR-MARC II field tag, the length of the field, and the starting character position of the field (Figure 2). Necessary indicators and subfield delimiters are also created and placed in their proper positions in the field’s data stream, and the field, along with its field terminator, is placed into the area set aside for data fields.

AFR-MARC II Records

It is important to emphasize that the system produces MARC-like records rather than full MARC records. While the basic record structure is exactly like that of standard Library of Congress MARC, distinctions such as personal versus corporate main entry are not shown by the field tagging and the degree of subfield delimiting is extremely restricted.\(^5\) Compare the list of variable field tags for AFR-MARC II (Automatic Field Recognition MARC II) records to that for LC-MARC II (Library of Congress MARC II) records (Figures 2 and 3). At present, AFR-MARC II provides detailed subfield tagging for only two fields, call number (090) and title (245). This lack of detailed discrimination causes no problem, however, for output of book catalog entries. It can affect filing sequence, since ALA filing rules depend on such distinctions as personal versus corporate author to determine proper sorting.

The decision to omit detailed subfield discrimination is a concession to cost. The two principal developers (UC and CSL) decided that, for book catalog production, detailed subfield delimiting would be of little value and that the benefits of such detail (i.e., ability to sort according to LC filing rules) would not justify the added costs in editing, input, programming, and processing which would be required to provide this detail.

A sample of an AFR-MARC II record created by the Automatic Field Recognition program is shown in Figures 4 and 6. It can be contrasted with the LC-MARC II record for the same title (Figures 5 and 7). Both a machine-based representation (Figures 4 and 5) and a formatted output example (Figures 6 and 7) of the record are shown.


49 p. 28 cm. (Its educational studies and documents, 7)

**Fig. 4. Sample Library of Congress Card in AFR-MARC II Format**
AFR creates MARC structured records from unedited input data. To what does “unedited” input refer? Without a program such as AFR, each MARC field tag, subfield code, indicator, etc., for every MARC or MARC-like record must be manually supplied by a human editor. With AFR the input keyer simply indicates that some field is beginning; it is then up to the AFR program to identify the field.

AFR will accept input created by a variety of methods. The decision on input method is based principally on cost. Since input costs can vary widely as a result of various local conditions, provision has been made in the BIBCON system to accept data in card or tape format. Keypunch and Optical Character Recognition (OCR) input are the two methods used thus far. A sample OCR input record appears in Figure 8. While input instructions will vary according to the input method used, the four basic keying requirements remain the same:

1. Begin an input record with an identification number.
2. Place a field separator symbol before each field (i.e., each indentation on the catalog card).
3. Place a different symbol (called a “location” symbol) after call number and after the library location data.
Fig. 6. AFR-MARC II PRINTSUS Output Format

4. End each input record with an end-of-record symbol.

Variations on these four basic rules may be required because of restrictions of the input device used, because of variations in content or form of the input data, or because output specifications require nonstandard treatment by the programs. The task of manipulating the varying input into a form which is acceptable to AFR is performed by a program called PREAFR.

PRE AUTOMATIC FIELD RECOGNITION (PREAFR)

This program provides the interface between any one of the different input methods and the AFR program. Basically, PREAFR accepts data from keypunched cards, and OCR PREAFR accepts it from tape records. Both forms of the preprocessing program combine input data segments until an end-of-record symbol is reached, indicating that all the data for one bibliographic record have been assembled. A character by character search is made, and special characters and diacriticals which were input as special codes are translated into the values necessary for output processing.
In addition the program can perform several editing and checking functions. These functions are optional and are dependent upon the input equipment and upon the wishes of the user. Options such as deletion of data on the basis of special input symbols, checking to determine that the record control number is valid, and production of a file of control numbers for records in which data could not be interpreted by the input device are standard.

Because this program provides the interface between different, nonstandard input methods and one standard record formatting program, it is very user-dependent. The basic logic will remain the same, but individual options will have to be added or subtracted by each separate user.
FIG. 8. SAMPLE OCR INPUT

PREAFR produces a file of variable length, machine-readable records (Figure 9) which are passed to AFR for formatting into a MARC structure with limited MARC II tagging as described in the section on AFR.

RECORD PROOFING AND CORRECTING

PRINTSUS

PRINTSUS is an output program which provides formatted AFR-MARC II records, showing field tag, subfield delimiters, indicators, etc. This printout is designed for proofing of the MARC records created by AFR. Samples of this type output appear in Figures 6 and 7.
**FIX**

By processing data according to "FIX commands" this program corrects records in MARC format, operating as a context editor. Corrections can be made to content or structure. Entire records can be deleted and new records can be created using FIX "correction" statements. When any change is made, FIX automatically updates the record’s leader and directory to reflect the record as changed.

There are two input files: bibliographic records, in MARC format, and the FIX correction data. The input records file must be in MARC format and must be in the same order (by record I.D. numbers) as the FIX correction data file in order to successfully update the records.

The FIX program method of making corrections is based on the FIX expression, which can be considered as a "language," with rules of grammar governing the structure of expressions (sentences), the order of elements within the expressions and the possible contents of each element (see Figure 10).

**Output Processor**

The output processor consists of three programs and an IBM utility sort program. These general-purpose programs, which are designed to create book catalog page output, allow a variety of options for sorting as well as formatting.

**SORT KEY EDIT (SKED)**

This program performs two major functions (Figure 11) as follows: (a) from a single MARC record it creates a record for each point of access to that record as specified by the program user; and (b) it establishes a 256 character sort key at the head of each record extracted. The file is then passed to an IBM sort package for sequencing.

**Record Extraction**

SKED does not actually extract data from the original MARC record. Instead, it replicates the full record for each access point specified. It is left to the BIBLIST program to extract the required data from these records. Thus, if a particular bibliographic record should have five access points (one for main entry, one for title, two for subjects, and one for some other added entry), SKED would output five full MARC records. Essentially the only differences in the output SKED records would be in the data found in the sort keys prefixed to each record. The record for main entry access would contain main entry data as its first element; the title entry access record would contain title data first, etc.

**Sort Key Creation**

Data for the sort key are selected on the basis of user-specified tables.
### Fig. 10. Sample FIX Data, Illustrating FIX Operations

<table>
<thead>
<tr>
<th>FIX Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td><code>3 1 s '650 11'</code></td>
</tr>
<tr>
<td><strong>INPUT</strong></td>
<td>300 a 1. Loans, Personal - San Francisco.</td>
</tr>
<tr>
<td><strong>OUTPUT</strong></td>
<td>650 1 a 1. Loans, Personal - San Francisco.</td>
</tr>
<tr>
<td>02</td>
<td><code>2 1'c d</code></td>
</tr>
<tr>
<td><strong>INPUT</strong></td>
<td>245 0 a San Francisco. v. 24cm.</td>
</tr>
<tr>
<td><strong>OUTPUT</strong></td>
<td>245 0 a San Francisco. v. 24cm.</td>
</tr>
<tr>
<td>03</td>
<td><code>2 1'v.24cm.' i '300 1</code></td>
</tr>
<tr>
<td><strong>INPUT</strong></td>
<td>245 a San Francisco. v.24cm.</td>
</tr>
<tr>
<td><strong>OUTPUT</strong></td>
<td>300 a San Francisco. v.24cm.</td>
</tr>
<tr>
<td>04</td>
<td><code>2 1' a' cd 11 'Report.'</code></td>
</tr>
<tr>
<td>05</td>
<td><code>2 1'Port.' c</code></td>
</tr>
<tr>
<td><strong>INPUT</strong></td>
<td>245 0 a Report. San Francisco.</td>
</tr>
<tr>
<td><strong>OUTPUT</strong></td>
<td>245 0 a Report. San Francisco.</td>
</tr>
</tbody>
</table>

#### FIX Commands in This Sample

<table>
<thead>
<tr>
<th>Expression</th>
<th>Code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>s</td>
<td>Set field tag</td>
</tr>
<tr>
<td>02</td>
<td>c</td>
<td>Copy data</td>
</tr>
<tr>
<td>03</td>
<td>i</td>
<td>Identify field</td>
</tr>
<tr>
<td>04</td>
<td>cd</td>
<td>Copy &amp; Delete data</td>
</tr>
<tr>
<td>05</td>
<td>c</td>
<td>Copy data</td>
</tr>
</tbody>
</table>

---

**EXECUTIVE SUMMARY**

The Executive Summary provides an overview of the key findings and conclusions from the study. It highlights the main results, implications, and recommendations derived from the analysis and research conducted. This section is designed to be concise, focusing on the most impactful aspects of the study, ensuring that stakeholders and decision-makers can quickly grasp the essence of the research without delving into the detailed methodology and data analysis presented in the main body of the report. The summary serves as a bridge to the conclusions and recommendations, setting the stage for readers to understand the broader implications and practical applications of the research findings.
GENERAL DESCRIPTION OF SKED SUBSYSTEM

1. EXPLOSION OF RECORD
   - Field table is used to determine the number of "Sked Records" to make.

2. EXTRACTION OF DATA; SORT KEY BUILD
   - For each Sked record to be made, the program cascades through the three tables as required, in order to properly extract the data from the MARC record to build sort keys.
   - 256 byte sort key is created.

3. CREATION OF SKED RECORDS
   - Original MARC record is replicated, and unique sort key is affixed to the front end.

EXEMPLARY CONDITIONS REQUIRE 3 SKED RECORDS

- Conditions
- Initial Articles
- MISCELLANEOUS EDITING: All upper case; remove blanks; drop punctuation; etc.

Fig. 11. General Description of SKED Subsystem
The user is allowed almost unlimited freedom in determining what fields are placed in this sort key and the order of their placement.

Field and subfield selection and sequencing are performed on the basis of information contained in the three basic tables set up by the user. The program contains both table-driven and automatic editing routines. The three principal tables are called: (a) FIELD TABLE (FLDTABLE); (b) FIELD CONTROL TABLE (FLDCNTL); and (c) SUBFIELD CONTROL TABLE. A portion of one SKED table appears in Figure 12.

When the data are placed into the sort key, various editing functions are performed; some are required and others are performed only at the user’s request. This editing includes translation of all alphabets to upper case so that they will all have the same sort value, deletion of initial articles in several languages, and insertion of blanks in certain locations in order to provide for a proper sort.

Another option allows the user to specially prepare the data to be placed in the sort key. Thus, if the title of a book, for example, contained numeric data or abbreviations, this option would allow the user to prepare data in a specified field by translating the numerics or abbreviations to their alphabetic equivalents so that the title would sort according to standard library filing rules. This specially prepared field would then be placed in the sort key for title added entry, instead of the actual title as found in the title field.

BIBLIOGRAPHIC RECORDS LISTING (BIBLIST)

BIBLIST is the program which formats individual entries for output. BIBLIST, like SKED, is a table driven program, and on the basis of user-specified options it extracts the fields needed for each type of book catalog entry. The program adds necessary spaces, numerals, words, phrases, and symbols as requested. The column width is specified by the user, and BIBLIST formats the entries according to this specification. The data stream is broken only at a blank, so no words are split between lines. A list and a description of some of the standard BIBLIST options are included in Figure 13.

BIBLIST creates a file of records, which are also, incidentally, MARC structured and which contain all of the instructions necessary for the final program (PAGEFORM) to array full pages of the book catalog.

PAGE FORMATTING (PAGEFORM)

PAGEFORM relies on user specified options in table format to establish: the number of columns per catalog page; the length of these columns; the width of the left, right, top, and bottom margins; and the width of the gutters between columns. PAGEFORM numbers the pages at center bottom, establishes entry headings, and combines two or more entries under identical headings.

Finally, if the user wishes, PAGEFORM provides that individual entries
**Output Options Currently Provided by BIBLIST**

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>OPTIONS AVAILABLE WITH ONE SET OF BIBTABLES</th>
<th>PROVIDED BY</th>
<th>DOCUMENTED IN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Records Processed</td>
<td>Any number of records, in a sequence, from any portion of the input file may be specified for processing.</td>
<td>File Control Card</td>
<td>BIBLIST APPENDIX II</td>
</tr>
<tr>
<td>2. Column Width</td>
<td>From 1 to 132 characters.</td>
<td>Program Parameter Card, Columns 13-15</td>
<td>BIBLIST APPENDIX III</td>
</tr>
<tr>
<td>3. Output Character Set</td>
<td>Provision is made for specification of 100 different character sets. Each set must be specified by the user. The IBM TN print train set and the standard all upper case set are currently specified by CSL. (The TN train contains lower case and more special characters than the standard set.)</td>
<td>Program Parameter Card, Columns 17-18</td>
<td>BIBLIST APPENDIX III</td>
</tr>
<tr>
<td>4. Print Diacriticals</td>
<td>Diacriticals may be printed or suppressed. (Not implemented as of 4/1/72.)</td>
<td>Program Parameter Card, Column 20</td>
<td>BIBLIST APPENDIX III</td>
</tr>
<tr>
<td>5. Print BIBLIST Records</td>
<td>The file of entries created by BIBLIST may be printed in list form before the file is passed to PAGEFORM.</td>
<td>Program Parameter Card, Column 22</td>
<td>BIBLIST APPENDIX III</td>
</tr>
<tr>
<td>6. Catalog Entries Selected</td>
<td>Any MARC field may be selected as the first (or heading) field in a catalog entry.</td>
<td>Field Format Table, Bytes 0-9</td>
<td>BIBLIST APPENDIX IV</td>
</tr>
<tr>
<td>7. Selection of Succeeding Fields in an Entry</td>
<td>Any MARC field may be selected to follow the first field in a catalog entry. These succeeding fields may appear in any sequence, regardless of their order in the AFR-MARC record.</td>
<td>Field Format Table, Bytes 0-9</td>
<td>BIBLIST APPENDIX IV</td>
</tr>
<tr>
<td>8. Bold Face Type</td>
<td>Bold face type may be specified for heading field print lines.</td>
<td>Field Format Table, Byte 10</td>
<td>BIBLIST APPENDIX IV</td>
</tr>
<tr>
<td>9. Missing Field</td>
<td>When a field which has been specified by the user is missing from a record, the record may be rejected and processing continued with the next record; or from 1 to 255 Field Format Table entries may be skipped, with processing of the record continuing from the next entry.</td>
<td>Field Format Table, Byte 11</td>
<td>BIBLIST APPENDIX IV</td>
</tr>
</tbody>
</table>

*Fig. 13. Sample List of BIBLIST Options*
are not split between columns or pages and for the repetition of entry headings on succeeding columns or pages.

The various users of the BIBCON software have produced both print files for line printer and driver tapes for photocomposition of book catalogs.

**Book Catalog Samples**

Figures 14 and 15 show samples of catalogs produced with the BIBCON system. For these catalogs the page masters were formatted by BIBCON and printed on a line printer. These page masters were then photo-reduced and the resultant paper masters were duplicated by usual offset methods.

**Processing Costs**

**OCR Keying:**

- 400 character record @ 3,000 strokes/hr
- 8-10 min/record @ $.06/min = $ .48

**OCR Scanning:**

- 15 records/min @ $.45/min = $ .03

**Program Processing:**

- All programs: Est. 5 sec @ $.10/sec = $ .50
- **TOTAL** $1.01/output record

More detailed costs will be prepared by the California State Library and will be reported as they are available.

**Evaluation**

The BIBCON system accepts unedited data, formats it into a MARC-like record, and produces book catalog output with a variety of options. The system is particularly useful for "listing" projects that require a range of output products and formats.

The advantages and disadvantages are summarized as follows:

**Advantages:**

1. **Tagged Input Unnecessary:**
   
   Because of the formatting and tagging abilities of the Automatic Field Recognition program, BIBCON can produce MARC records from input which has not been manually supplied with any of the MARC field tags.

2. **Versatility:**
   
   The output processing programs provide for a wide variety of output formats. With the addition of programs to produce files for photocomposition, the output options will be even more varied.

3. **National Standard:**
   
   BIBCON produces and processes records in the Library of Congress MARC format which has been established as the international standard for computer-based bibliographic records.
Fig. 14. Sample BIBCON Output; CSL Science and Technology Catalog

4. System Is Operational:
The BIBCON system has been installed and is operational at the University of California and in Sacramento for the California State Library. It has already been used to produce catalogs of all sorts, from small, topical catalogs to large union lists of monographs and of se-

DIFFERENTIAL EQUATIONS, PARTIAL.
Gilbert, Robert P., 1932-
QA 377 .G6 1969 Gen
Horváth, László
QA 377 .H66 1969 Gen
Petrowski, Ivan Georgievich
QA 377 .P43 1969 Gen

DIFFERENTIAL EQUATIONS, PARTIAL-DIFFERENTIAL SOLUTIONS.
Hartman, Philip, 1908-
QA 913.3 .H35 1969 Gen

DIFFERENTIAL OPERATORS.
Horváth, László
QA 377 .H66 1969 Gen

DIFFERENTIATION GRADINGS.
Davis, Sumner P.
QA 415 .B35 1969 Gen

DIFFUSION.
Frank-Kaart, David Albertovitch
QA 543 .F6613 1969 Gen

DIGITAL COMPUTER SIMULATION.
Daub, Donald E.
TS 157 .D42 1969 Gen

TP 155 cr2 1971 Gen

DIGITAL COMPUTER SIMULATION.
Chu, Yuhao, 1920-
QA 76.5 .C49 1963 Gen
Gordon, Geoffrey
QA 76.5 .G63 1969 Gen
Harbaugh, John Marealle, 1926-
QA 76.5 .H37 1970 Gen
Mire, Joe H.
QA 76.5 .M56 1971 Gen
Pritsker, A. Alan R., 1933-
QA 76.5 .P53 1969 Gen
Schiidt, Joseph William
T 57-62 53 Gen
Siegel, Arthur I.
TA 367 .S53 1969 Gen
Yusen, Forrest Paul
QA 76.5 .Y94 1969 Gen

DIGITAL COMPUTERS.
Lancaster, Donald E.
TK 7877 .C7 13 Gen
Kalmstadt, Howard V., 1942-
TK 7877 .C7 83 Gen
Horváth, László
TK 7886 .59 86 Gen

DIGITAL ELECTRONICS.
Cook, Alfred
TK 7886 .56 65 Gen
Fig. 15. Sample BIBCON Output: CSL Education Catalog

Disadvantages:

1. Personnel Dependency:
   BAL: The system is written in Basic Assembler Language, thus necessitating the services of an experienced programmer.

MARC: Because the system operates upon MARC structured record format, the average programmer may well have a difficult time in dealing with the added complexities introduced by this aspect.
OPTIONS: The wide range of options provided by the system necessitates highly complex programs which may be difficult for the average programmer to grasp readily.

2. Equipment Dependency:
IBM: Because the programs are written in IBM Basic Assembler Language, the system is presently usable on IBM equipment only.

Conclusion
The BIBCON-360 system is a versatile and inexpensive method for producing book catalogs, when a wide range of format options are required and when the catalogs must contain bibliographic information with more than one entry or access point per bibliographic record. If a simple, main entry catalog is needed, microfilm reproduction of the catalog cards may still be much cheaper.

BIBCON-360 is most useful for producing large scale catalogs (e.g., union catalogs) to be distributed widely to assist in the effort to provide the widest possible dissemination of library information at the least possible cost.

REFERENCES
1. California. State Library, Sacramento. Automation Project, A Users' Manual for BIBCON 360; A File Management System for Bibliographic Records Control (Sacramento: California State Library, 1972), 274p. (This manual, produced in limited quantities, is now available only on interlibrary loan.)
2. University of California, Santa Cruz, Author-Title Catalog of the University Library (Santa Cruz: University of California, 1970), 32 v.
3. University of California, San Diego, Author-Title Catalog; Subject Catalog (San Diego: San Diego Medical Society-University Library, 1969), 350p.
7. California. State Library, Sacramento, Special Education Problems; A Catalog of Materials in the California State Library (Unpublished). (This topical catalog was output only to test refinements to the BIBCON-360 programs. It was not published, but the sample pages produced illustrate further refinements in formatting and sorting routines.)