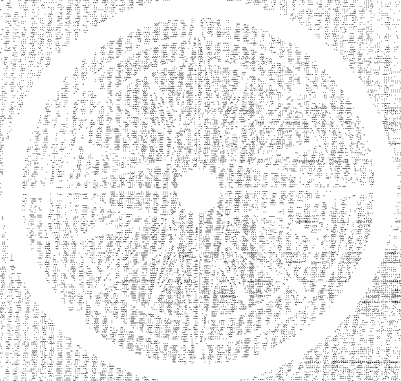


~~CONFIDENTIAL~~ ~~SECRET~~

7



PHASE II FINAL REPORT

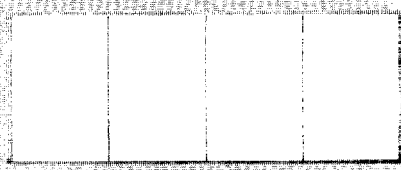
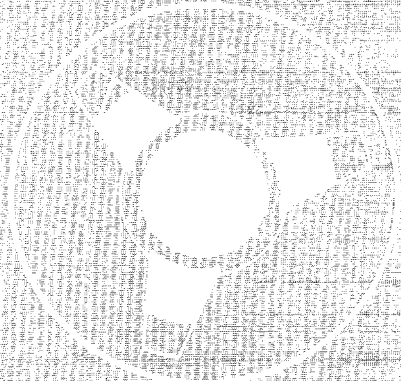
PROJECT CHIVE

Volume VII

EDP SYSTEM

JOB NO. 78-03952A
 BOX NO. 1
 FOLDER NO. 7
 TOTAL DOCS HEREIN 1

CHIVE/R-3-65
1 March 1965



DIRECTORATE OF SCIENCE AND TECHNOLOGY
OFFICE OF COMPUTER SERVICES

~~SECRET~~

GROUP 1
Excluded from automatic
downgrading and
declassification

~~CONFIDENTIAL~~

W A R N I N G

This material contains information affecting the National Defense of the United States within the meaning of the espionage laws, Title 18, USC, Secs. 793 and 794, the transmission or revelation of which in any manner to an unauthorized person is prohibited by law.

~~SECRET~~
CONFIDENTIAL

4
151.2
4799
vol. 7

Project CHIVE:
Phase II Final Report

Volume VII

EDP SYSTEM

CHIVE/R-3-65

1 March 1965

010305

*U.S. Central Intelligence Agency
Office of Computer Services.*

DOC	<u>1</u>	REV DATE	<u>07/12/81</u>	BY	<u>018995</u>
ORIG COMP	<u>—</u>	OPI	<u>63</u>	TYPE	<u>11</u>
ORIG CLASS	<u>5</u>	PAGES	<u>408</u>	REV CLASS	<u>c</u>
JUST	<u>22</u>	NEXT REV	<u>2011</u>	AUTH:	HR 10-2

CONFIDENTIAL

~~SECRET~~
CONFIDENTIAL

TABLE OF CONTENTS

	<u>Page</u>
7.1. CHIVE EDP System Introduction	1
7.2. Summary of Hardware Requirements	7
7.2.1. First Year Hardware Requirements	7
7.2.2. Storage Requirements	8
7.2.3. Optical Page Reader	9
7.3. Files	11
7.3.1. General	11
7.3.2. File Manipulation	12
7.3.3. File Storage	12
7.4. Transaction Types	15
7.4.1. Structure Definition Transactions	15
7.4.2. File Maintenance Transactions	16
7.4.3. Information Retrieval Transactions	19

~~SECRET~~

~~SECRET~~ CONFIDENTIAL

	<u>Page</u>
7.5 Executive Control	23
7.5.1. System Requirements on the Executive Control System	25
7.5.2. Design Philosophy	28
7.5.3. System Input/Output	32
7.5.4. Task Management	35
7.5.5. Core Storage Management	37
7.5.6. Data Management	40
7.5.7. Additional Executive Control Functions	45
7.6. Summary of Internal Data Structures	47
7.6.1. File Format Tables	48
7.6.2. Content of File Format Tables	49
7.6.3. Additional File Format Table Functions	52
7.6.4. File Format Table Modification	53
7.6.5. System Data File Organizations	53
7.6.6. CHIVE System Data File Record Structure	57
7.6.7. Searching for a Value Field	62
7.6.8. Directory File Structure	64

~~SECRET~~ CONFIDENTIAL

~~SECRET~~ CONFIDENTIAL

	<u>Page</u>
7.7. File Maintenance	67
7.7.1. File Maintenance Processing	67
7.7.2. Input Transformations	68
7.7.3. Record Maintenance	68
7.7.4. Directory File Maintenance	69
7.8. Retrieval Hit Processing	71
7.8.1. Hit Processing Functions	71
7.8.2. Structuring Commands	74
7.8.3. Editing Commands	83
7.8.4. The Arithmetic Commands	95
7.8.5. Writing Commands	107
7.9. Input Processing	117
7.9.1. Types of Input	117
7.9.2. Input Language Processor	117
7.9.3. Command Macros and Standing Job Definitions	119
7.9.4. Input Job Batching	123
7.9.5. Input Functions	126

CONFIDENTIAL
~~SECRET~~

~~SECRET~~ CONFIDENTIAL

	<u>Page</u>
7.10. Output Processing	127
7.10.1. Output Routine Definition	127
7.10.2. Report-Related Processing Functions	130
7.10.3. General Report Generating Functions	133
7.11. Security Procedures	153
7.11.1. Problem Statement	153
7.11.2. Mixed Data-Classification Concept (All Source)	154
7.11.3. Mixed EDP Job Classification Concept (All Demand)	156
7.11.4. Problem Solution	157
7.11.5. Decentralized All-Source/All-Service Concept	157
7.11.6. The Role of "Need-to-Know" in CHIVE Security Design	158
7.11.7. Security Isolation of Programs and Data in Storage	159
7.11.8. Security Isolation of Data Transfers	161
7.11.9. Security Isolation of Jobs and Tasks	163
7.11.10. Privileged Access to Logical Files	165

CONFIDENTIAL

~~CONFIDENTIAL~~

	<u>Page</u>
7.11.11. Access to Auxiliary Storage Units	166
7.11.12. File Interaction and Security Isolation	166
7.11.13. Privileged Access to Output Devices	169
7.11.14. EDP Software Security Isolation	170
7.11.15. EDP Hardware Security	176
7.11.16. Procedural Security	178
7.11.17. Computer Facility Security; Premises Security	183
7.11.18. Information Handling Functions and Operations	183
7.12. Backup Procedures	187
7.12.1. Backup Concepts	187
7.12.2. Backup Plan	188
<hr/>	
7.A. User Language Manual	192
7.A.1. Purpose of Manual	192
7.A.2. CHIVE Data Elements and Their Logical Structure	193
7.A.3. General Organization of CHIVE Language	201

~~CONFIDENTIAL~~
~~SECRET~~

~~CONFIDENTIAL~~

	<u>Page</u>
7.A.4. Language Specifications	213
7.A.5. Inter-Record and Inferential Logic	262
7.A.6. File Maintenance	279
7.B. Input Transcription Study	285
7.B.1. Fundamental Assumptions	286
7.B.2. Evaluations of the Candidate Input Systems	292
7.B.3. Summary	321
7.C. EDP Output Formats	335
7.C.1. Document Index File Reports	335
7.C.2. Vocabulary Control File Reports	343
7.C.3. Summary Data File Reports	346
7.C.4. Unsynthesized File Report	346
7.D. System Macros Used in Operational Programs	353
7.D.1. Departure Macros	354
7.D.2. Data Set Control Macros	357
7.D.3. Data Set Accessing Macros	358
7.D.4. Storage Allocation Macros	360

~~SECRET~~

~~CONFIDENTIAL~~

	<u>Page</u>
7.E. Hardware Unit Characteristics	361
7.E.1. Central Processing Unit	361
7.E.2. 2361 Core Storage	371
7.E.3. I/O Channels	372
7.E.4. Bulk Storage Devices and Associated Control Units	375
7.E.5. System Communication Devices	383

FIGURES

	<u>Page</u>
7-1 Overall View of Chive EDP System	3
7-2 Input/Output Interfaces in OS/360	33
7-3 CHIVE System Data File Record Structure	51
7-4 Matrix Representation of Terms in a Phrase Group	59
7-5 Directory File Record Structures	65
7-6 Hit Processing Function	72
7-7 Extract	76
7-8 Merge P	78
7-9 Merge R	84
7-10 Append	88
7-11 Delete	91
7-12 Replace	93
7-13 Tally	97
7-14 Accumulate	100
7-15 Compute	103
7-16 Write	108
7-17 Copy	111
7-18 Explode	114
7-19 Print	116

SECRET

	<u>Page</u>	
7-20	Input Command Statements	121
7-21	CHIVE Input Processing	125
7-22	Page Arrangement - CHIVE Reports	129
7-23	Samples of Line Types	135
7-24	Phrase Print Format 1	142
7-25	Phrase Print Format 2	143
7-26	Phrase Print Format 3	144
7-27	Phrase Print Format 4	145
7-28	Phrase Print Format 5	146
7-29	Phrase Print Format 6	147
7-30	Phrase Print Format 7	148
7-31	Phrase Print Format 8	149
<hr/>		
7.A-1	Logical Structure of CHIVE Records	196
7.A-2	Sample CHIVE Record	197
7.A-3	Schematic and Tree Representation of Document Index Record	199
7.A-4	Schematic and Tree Representation of Information Record	200
7.A-5	Test and Command Lines on Input Coding Sheet	204

SECRET

	<u>Page</u>	
7.A-7	The Numeric OP Entry	217
7.A-8	The Relational OP Entry	218
7.A-9	Value Entries with Phrase Coordination	220
7.A-10	Value Entries with DON'T CARE Characters	221
7.A-11	Value Entries with Tag Occurrence Symbol	223
7.A-13	Value Entry with \$ FILE (name)	224
7.A-14	Test Line (C) Made up of Test Lines (A,B)	226
7.A-15	Phrase Coordination in a COMPLEX Test Line	227
7.A-16	The Uses of '*' for Phrase Coordination	228
7.A-17	Test Lines with M of N Logic	229
7.A-18	Example of Use of X Operator	230
7.A-19	Summary of Test Line Rules	230
7.A-20	COPY and PRINT Command Formats	231
7.A-21	COPY and PRINT Command Lines	233
7.A-22	Test Line	235
7.A-23	COMMAND Lines for COPY and PRINT	235
7.A-24	CRITERIA Entries for Various Data Elements	236
7.A-25	Use of CRITERIA Entry	238
7.A-26	Alternative Use of CRITERIA Entry	239
7.A-26A	DEFINE Command Line	240

	<u>Page</u>	
7.A-27	Example of EXTRACT Command	242
7.A-28	Examples of WRITER Command	244
7.A-29	Examples of Append Command	245
7.A-30	Phrase Appended to a Record	247
7.A-31	Term Appended to Phrase	247
7.A-32	Examples of DELETE Command Lines	248
7.A-33	REPLACE Command Lines	249
7.A-34	EXPLODE Command Lines	251
7.A-35	Exploded Hit Records	252
7.A-36	EXPLODE Terms Command	253
7.A-38	TALLY Command Lines	254
7.A-39	TALLY and APPEND Result for Output	256
7.A-40	Command Lines for Accumulating Across a File	259
7.A-41	Command Lines for Accumulating TALLY Result	258
7.A-42	COMPUTE Command Lines	259
7.A-43	COMPUTE n Referenced in an APPEND Command Line	261
7.A-44	Enriched Querying	264
7.A-45	MERGER with Single and with Multiple Merge Keys	265
7.A-46	Input Records Before MERGER Command	266
7.A-47	MERGED RECORD with Common CNC 20	268
7.A-48	Input Records with Common Values for Multiple Merge Key PNO and CNC	269

SECRET

	<u>Page</u>
7.A-49 Merged Record with Common Multiple Merge Key	270
7.A-50 Test and Command Lines for Inferential MERGER Logic	272
7.A-51 MERGEP with Single and Multiple Merge Keys	274
7.A-52 Phrases with Common Values for Merge Keys	275
7.A-53 Matrix of VALUES for Phrases 1-5	276
7.A-54 New Phrase Merged from Phrases 1,2, and 3	277
7.A-55 New Record after MERGEP Execution	278
7.A-56 MERGE Phrases for Multiple Merge Keys	280
7.A-57 Command Lines for File Maintenance APPEND	281
7.A-58 Command Lines for File Maintenance DELETE	282
7.A-59 Command Lines for File Maintenance REPLACE	283
<hr/>	
7.B-1 Operating Costs Punched Card Systems	294
7.B-2 Operating Costs Paper Tape Systems	298
7.B-3 Operating Costs Paper Tape Systems	299
7.B-4 Operating Costs CRT/KB Systems	308
7.B-5 Operating Costs CRT/KB Systems	309
7.B-6 Operating Costs Farrington Page Reading System	316

SECRET

		<u>Page</u>
7.B-7	Operating Costs Ravinow (CDC) Page Reading System	317
7.B-8	Operating Costs Recognition Equipment Page Reading System	318
7.B-9	Comparative Operating Costs of Candidate Input Systems (off-line)	323
<hr style="width: 25%; margin-left: 0;"/>		
7.C-1	Document ID-Term Report-Cover Page	336
7.C-1	Document ID-Term Report-First Page	337
7.C-1	Document ID-Term Report-Body Page	338
7.C-2	Abbreviated Header-Contents Report	340
7.C-3	Full Header-Contents Report	341
7.C-4	KWIC Title Report	342
7.C-5	Map Catalog Card	344
7.C-6	Organization/Facility Control Report	345
7.C-7	Leader Appearance-Source Citation Report-Cover Page	347
7.C-7	Leader Appearance-Source Citation Report-Body Page	348
7.C-8	Condensed Leader Appearance Report	349
7.C-9	Individual Travel to Foreign Countries Report-Cover Page	350
7.C-9	Individual Travel to Foreign Countries Report-Body Page	351

~~SECRET~~

TABLES

	<u>Page</u>
7-1 File Interaction Security Transition Table	168
7.A-1 Summary of Rules for Command Lines	211
7.B-1 Cost Summary of Candidate Input Systems	324
7.B-2 Summary of Requirements for Typists/ Key punch Operators and Proofing Clerks for Candidate Input Systems	326
7.B-3 Comparison of Alternative Input Trans- cription Methods	328
7.B-4 Subsequent Tasks for OCR and OCS After Choosing the Page Reader Approach	330
7.B-5 Tentative Page Reader Specifications - Project CHIVE	331

~~SECRET~~

Chapter 7.1.

CHIVE EDP SYSTEM INTRODUCTION

The EDP portion of CHIVE will perform the following functions:

- Build and maintain files
- Search files
- Create sub-files from existing files
- Display data

The CHIVE system must process many different data files. The files range from the large document index files to relatively simple and small decode files. Further, it is necessary to provide the capability to add new files to the system and also to revise the structure of existing files. All of the above considerations have dictated a general approach towards the CHIVE EDP design; that is, a single, flexible set of programs will perform the functions outlined above on any of a large class of files. The programs will be implemented in a fashion that will permit new files to be introduced to the system and old files to be restructured with no change to the programs themselves.

SYSTEM INTRODUCTION
7.1.

The feature that gives flexibility to the system is extensive use of parameters. All of the CHIVE programs will rely on external descriptions of the data and transactions to be processed. The data descriptions taken collectively are called the File Format Tables. Each table describes a file and its constituent items. Included in the description are record structure, types of item, storage, addressing rules, and legality parameters. By changing the item descriptions, the user of the system can make the file(s) reflect current needs. It is possible to create a file and then through experience gained from usage, revise the file structure for increased usefulness. Input transactions to the CHIVE system are also broken down into individual functional elements or parameters. The net result is that the maintenance and retrieval operations can be easily controlled by non-EDP users.

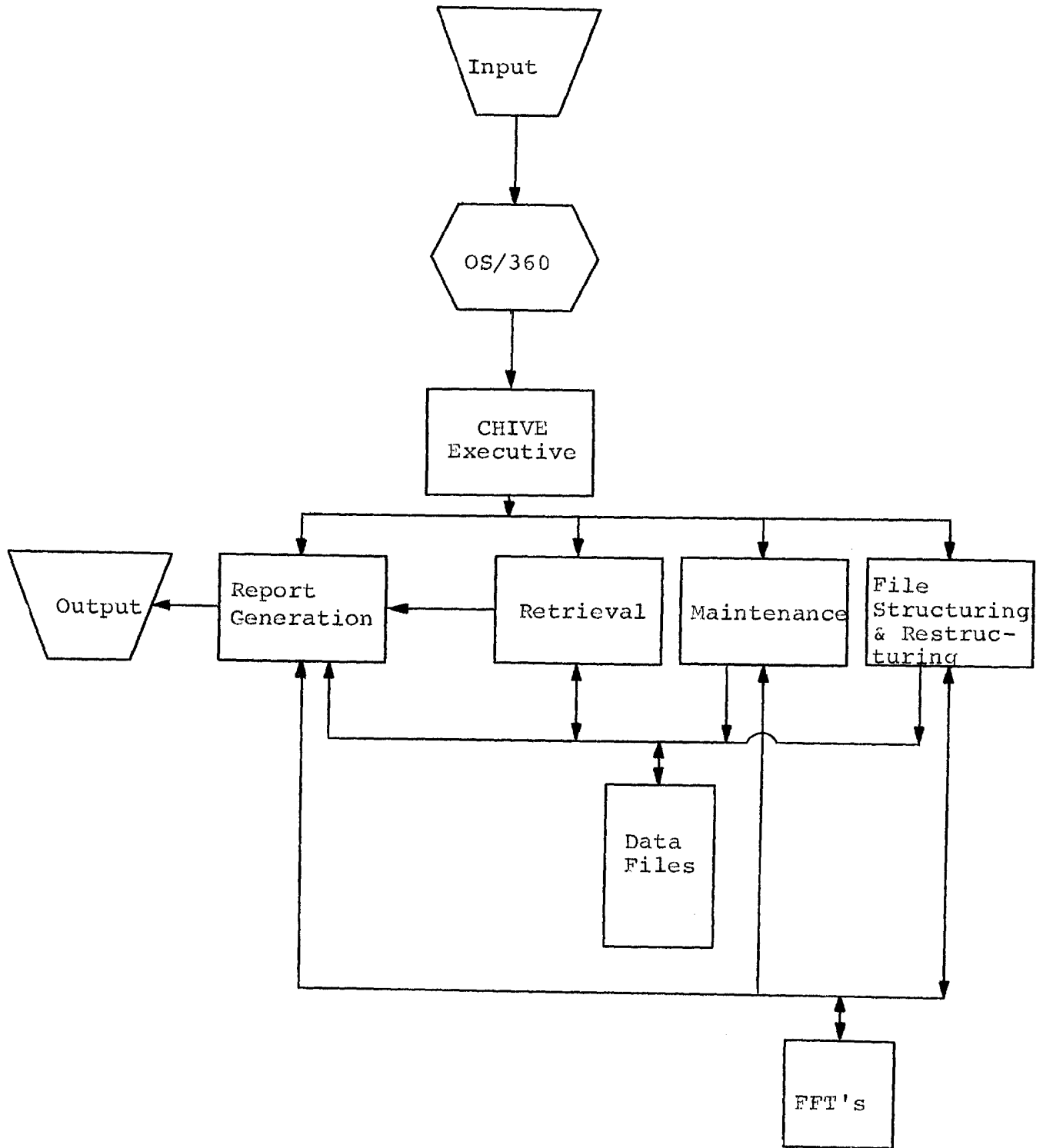
An overview of the CHIVE EDP system is shown in Figure 7-1. The system is divided into five major modules:

- CHIVE Executive
- File Maintenance
- Retrieval
- Report Generation
- Format Table Maintenance and File Restructuring

SYSTEM INTRODUCTION
7.1.

Figure 7-1

OVERALL VIEW OF CHIVE EDP SYSTEM



The CHIVE executive module performs these functions: it interfaces with Operating System/360, it analyzes all CHIVE inputs and switches control to the proper operating module, and bookkeeps the status of all machine files under CHIVE control.

The File Maintenance module creates and updates the CHIVE files. It consists of generalized routines for processing files and special purpose routines for handling raw input. Depending on the type of file to be processed, it will operate in either a demand or batch mode.

The Retrieval module accepts all CHIVE search requests and translates them into search routines. Control is then passed to these generated programs which then search the indicated file. The records selected can be subjected to additional manipulation which can result in the creation of a sub-file or a completely restructured file. This special file manipulation, called Hit Processing is fully described in Chapter 7.8.

The Report module accepts any file in the CHIVE system and prints it according to specified parameters. The system allows the user considerable latitude in formatting reports. Direct control is permitted as to the sequence

SYSTEM INTRODUCTION
7.1.

SECRET

of presentation, the amount of printing, which items should be printed, and their placement and spacing.

The Format Table module generates file format tables and updates them. It also restructures existing files, when their associated format tables have been revised, to insure compatibility with the new tables.

Inspection of Figure 7-1 should make it clear that any processing of a CHIVE data file is always performed in conjunction with its associated file description. This feature is the primary device used in attaining system flexibility. One other important aspect of the system, which is difficult to show pictorially, is the fact that the CHIVE system operates on a demand basis in a multi-programming environment. It will actually be under control of the IBM-provided operating system. Thus, the CHIVE programs will operate concurrently with other computer jobs completely unrelated to CHIVE.

The functional capabilities of the EDP system are controlled by the user via a command language. A description of this language is presented separately in Appendix 7.A. as a user's manual. The information in 7.A. provides the non-EDP reader with the essence of the design goals of the CHIVE EDP system. Also this

SYSTEM INTRODUCTION
7.1.

SECRET

SECRET

section allows all readers to determine the extent of
generality built into the system.

SYSTEM INTRODUCTION
7.1.

SECRET

Chapter 7.2.

SUMMARY OF HARDWARE REQUIREMENTS

The CHIVE Initial System will be implemented on the OCS IBM System/360, Model 60. CHIVE will be operated in a multiprogrammed mode with other OCS jobs. The IBM Operating System/360 will provide the executive control and multi-programming facilities for the CHIVE program modules.

7.2.1. FIRST YEAR HARDWARE REQUIREMENTS

At the end of one year operational test period, CHIVE will require the utilization of the following equipments:

<u>Qty.</u>	<u>Units</u>	<u>(%) Use Estimated</u>
1	2060 - Central Processor	15%
1	1403 - Printer	13%
1	1402 - Reader-Punch	1%
8	2400 - Magnetic Tapes	5%
1	2311 - Disk	13%
1	Console	*
1	Optical Page Reader	*

*The optical page reader and console utilization is not shown since specific selections have not yet been made.

HARDWARE REQUIREMENTS
First Year
7.2.1.

The percent utilizations given for the System/360 are based on 100% utilization being equal to 176 hours per month. Further, these estimates were based on the estimated inputs, processing required, and outputs of the reference and information files projected for the CHIVE System. The type of activities that were used to make these estimates were searching, sorting, maintenance, record building, input processing and output processing. These utilization figures do not contain any time allocated to file backup generation or processing. File backup was left out due to the lack of definition in this area.

The times for the central processor include estimates for all the transactions mentioned previously plus overhead time for the Operating System/360.

7.2.2. STORAGE REQUIREMENTS

A gross volume breakdown of the CHIVE storage needs is shown in the table below. Units are expressed in millions of characters.

	<u>1st Year</u>	<u>2nd Year</u>	<u>3rd Year</u>
Permanent Disk	45	29	70
Temporary Disk	10	10	10
Data Cell (2321)	0	112	212

HARDWARE REQUIREMENTS
Storage Requirements
7.2.2.

It is anticipated that all of the storage requirements for the first year will be assigned to disks. A 2321 Data Cell Drive (400 million characters rated capacity) will be required for CHIVE storage at the beginning of the second year. During the second and third years, CHIVE storage will be divided between the 2321 Data Cell Drive, disks and the 2400 Magnetic tapes.

It should also be pointed out that all of the above estimates do not contain time for conversion, maintenance or operation of any inherited files such as SR or IRS.

7.2.3. OPTICAL PAGE READER

In reaching a decision about the type of input transcription system to be used to input data and queries to the CHIVE EDP system, four alternatives were examined in great detail: a punched card system, a punched paper tape system, a cathode ray tube-keyboard system, and a page reader system. The study considered the cost, usefulness, flexibility, adaptability, speed, dependability, etc., of these four approaches. On the basis of cost-effectiveness considerations and simplicity of operation, the page reader system was selected as most responsive to immediate and long range CHIVE requirements. A detailed discussion of the findings of the study and the methodology employed is in Appendix 7.B.

HARDWARE REQUIREMENTS
Optical Page Reader
7.2.3.

Chapter 7.3.

FILES

7.3.1. GENERAL

The CHIVE EDP system is designed to service a large number of files. There are two characteristics that are common to all files. First, they are all externally described in file format tables. This feature allows a single set of programs to process any files in the system. Second, the structure of the records is such that complex relationships found in documents can be retained. Simpler structures such as those found in card files are also encompassed by the system.

The basic characteristics of the record structure are: sets or groups of values can be linked together. These sets can contain varying combinations of descriptors and in addition the constituent descriptors can occur within a set many times if desired. A set of descriptors and their values when linked together are called a phrase. Several individual phrases can be stored in a record. The descriptors in a file can be subdivided if desired, into mutually exclusive phrase types. It is permitted to

FILES
General
7.3.1.

SECRET

store several different types in a record. When it is desired to control a file with unique record identification, a record control phrase is defined. This phrase is different than other phrase types since it can occur only once in a record. Also, the particular descriptor(s) in the phrase that are chosen to provide the unique designation can occur but once.

7.3.2. FILE MANIPULATION

All CHIVE files are described in external tables. These tables are used to drive maintenance, retrieval and report programs during the processing of a file. Files can be initiated in either of two methods. The first is to manually provide a file description to the system. From this point, it is now possible to build a file from raw input data. The second method is to create files from those already in existence. In this case, the original file format table is revised. The revision is sometimes performed automatically; and, on other occurrences, the revisions must stem from manual inputs.

7.3.3. FILE STORAGE

Provision is made for serial and random storage. Files stored serially will be allocated to magnetic tapes. Random

FILES
File Storage
7.3.3.

SECRET

SECRET

files will be stored on disks and data cells. Those files stored in random fashion will have directory files to facilitate accessing. It should be noted that the IBM provided software allows the files to be I/O device independent. Thus, no special programming effort is required if the storage media of a file is changed.

Storage requirements for CHIVE files have been grossly estimated for three one-year periods. Period one is associated with the first year of operational test concerned 25X6 [REDACTED]. Period two includes the remainder of the Far East while Period three includes the Far East and USSR operations.

The most important files or types of files in the initial system are briefly described below. Their projected volumes and storage allocations are also indicated.

Document Index File (DIF) - contains descriptions of documents residing in CHIVE hard copy files.

Term File (TF) - contains the values of terms in the DIF and the associated storage addresses.

Decode Files (DF) - contain machine stored configurations and desired output equivalents.

FILES
File Storage
7.3.3.

SECRET

Information Files (IF) - contain data on named objects or events associated with named objects, the data itself is organized, evaluated, and non-redundant.

Access Files (AF) - contain information necessary to access randomly stored files. Logically the AF are similar to the TF.

CHIVE Files-Volumes (in millions of characters) and Associated Storage Devices

NAME	I	II	III
DIF	<u>36.0</u> Disk	<u>90.0</u> Data Cell	<u>170.0</u> Data Cell
TF	--	<u>21.0</u> Disk	<u>54.0</u> Disk
DF	<u>.65</u> Disk	<u>1.0</u> Disk	<u>1.0</u> Disk
IF	<u>17.0</u> Tape	<u>18.0</u> Tape	<u>18.5</u> Tape
AF	<u>.2</u> Disk	<u>.5</u> Disk	<u>1.5</u> Disk

FILES
File Storage
7.3.3.

Chapter 7.4.

TRANSACTION TYPES

This Chapter summarizes the following types of transactions in the CHIVE EDP system:

- Structure definition transactions
- File maintenance transactions
- Retrieval transactions

7.4.1. STRUCTURE DEFINITION TRANSACTIONS

There are three major types of data structures which must be defined to the CHIVE EDP system:

- Input formats
- File Format Tables
- Output report formats

In the initial CHIVE system all of these data structures will be defined at the machine language level by assembler or compiler language definitions. The File Format Tables generated by the assembler/compiler process will be treated as data sets which would be catalogued via the Operating System/360 in appropriate files. The structure definitions of inputs and report outputs may contain instructions as well as parameter

TRANSACTION TYPES
Structure Definition
7.4.1.

tables and hence would be catalogued as programs in the system program library.

7.4.2. FILE MAINTENANCE TRANSACTIONS

File maintenance transactions can be categorized by three factors for internal EDP processing. The first factor will be the type of file organization and storage: (a) sequentially accessed tape files, (b) sequentially accessed files on direct access storage devices, and (c) directory accessed files. The second factor will be the type of record specification: (a) specific and (b) conditional. The specific maintenance transactions identify a single record by providing the unique record control terms. Conditional record specifications involve test conditions based on record content which define one or more records for a single maintenance transaction to act upon.

The third factor differentiates between externally generated transactions and the special transactions generated internally to maintain the directories for directory access files. These directory files are maintained by transactions which are automatically generated by the system during the processing of maintenance transactions for a directory accessed file.

TRANSACTION TYPES
File Maintenance
7.4.2.

SECRET

These three factors define seven categories of maintenance transactions on the basis of internal EDP processing requirements. These seven transaction types, their characteristics, and the requirements for handling and internal processing are listed below:

7.4.2.1. File Maintenance Transactions with Conditional Record Specification for Sequential Tape Master Files

- Normally run in limited size batches
- Sequential record search of the file
- File copied with changes onto a new tape
- Each file record compared with the record criteria of all transactions in the batch.

7.4.2.2. File Maintenance Transactions with Specific Record Control Terms Specified for a Sequential Tape Master File

- Normally run in batches
- Transactions must be ordered by record control within the batch
- Not necessary to limit batch size
- Sequential record search of the file
- File copied with changes onto a new tape
- Transaction batch passed in parallel with master file.
- Only one record control from the transaction batch compared with each file record

TRANSACTION TYPES
File Maintenance
7.4.2.2.

SECRET

SECRET

7.4.2.3. File Maintenance Transactions with Conditional Record Specification for Sequentially Accessed Files on Direct Access Storage Device

- Normally run in limited size batches
- Sequential record search of the file
- Only the records changed need to be rewritten onto the storage device
- If space is not available, the record will be relocated and chained into the sequence
- Each file record compared with the record specification of all transactions in the batch

7.4.2.4. File Maintenance Transactions with Specific Record Control Terms Specific for a Sequentially Accessed File on Direct Access Storage Device

- Can be run one at a time or in batches
- File search using OS/360 indexed sequential access method
- Only a small number of file records need to be accessed and compared with record criteria
- Only the records changed need to be rewritten onto the storage device
- If space is not available, the record will be relocated and chained into the sequence

7.4.2.5. File Maintenance Transactions with Conditional Record Specification for Directory Accessed File

- Can be run one at a time or in batches
- Directory access file search using the directory file

TRANSACTION TYPES
File Maintenance
7.4.2.5.

SECRET

SECRET

- Only a small number of master file and directory file records need to be accessed and compared with record criteria
- Only the master records changed need to be rewritten on the storage device
- The required changes to the directory file are generated as separate maintenance transactions on the directory file

7.4.2.6. File Maintenance Transactions with Specific Record Controls for Directory Accessed Files

- Can be run one at a time or in batches
- Directory access file search using the record control directory file
- Only a small number of master file and record control directory file records are accessed and compared with record search criteria
- Only the master records changed need to be rewritten on the storage device
- The required changes to the record control directory file are generated as transactions for these directory files

7.4.2.7. File Maintenance Transactions for Directory Files

- Generated during the processing of master record maintenance on directory accessed files
- Executed in the same manner as specific maintenance transactions to sequentially accessed files on direct access storage devices (Section 7.4.2.4. above)

7.4.3. INFORMATION RETRIEVAL TRANSACTIONS

Information retrieval transactions involve three data handling operations: record retrieval, hit record processing and output generation. There are three classes

TRANSACTION TYPES
Information Retrieval
7.4.3.

SECRET

SECRET

of retrieval commands which indicate the operational character of the transaction.

7.4.3.1. Retrieval Type Commands

The COPY and EXPLODE commands both retrieve records or portions of records and copy them directly onto output files. These files may be used subsequently as further input files within the transaction or as final output of the transaction.

The EXTRACT command retrieves a single record or portion of a record into a staging area for subsequent processing.

7.4.3.2. Processing Type Commands

APPEND, DELETE, REPLACE, and MERGER commands process individual records which are EXTRACTed into a work area. The MERGER command will combine records in a file on the basis of common terms.

TALLY, ACCUM, and COMPUTE are internal processing functions which result in output eventually, either in a specified report or automatically if not otherwise specified.

TRANSACTION TYPES
Information Retrieval
7.4.3.2.

SECRET

7.4.3.3. Output Type Commands

The WRITER and WRITEP commands form a record from data in an EXTRACT staging area and place it in an output file.

The PRINT command causes a designated file to be printed according to the parameters specified in the command.

TRANSACTION TYPES
Information Retrieval
7.4.3.3.

Chapter 7.5

EXECUTIVE CONTROL

The CHIVE programming system will be composed of a number of interacting programs that are continually vying for the resources of the computer. The CHIVE Executive Control System (ECS) brings general overall program management and control to this environment. This section of the report attempts to summarize all pertinent facts relating to the present status of executive control philosophy and design in CHIVE.

Traditionally, monitors or executive control sections of serial programming systems have been looked upon as being second class citizens--their main functions were to pass parameters from one program to the next and to perform some common I/O activities. The advent of sophisticated hardware interrupt structures and multiprogramming have antiquated this viewpoint. Executive control development in CHIVE, rooted in the later philosophy, has undertaken three shifts in the point of view as to how such a multiprogrammed system should be implemented. The present implementation approach and

EXECUTIVE CONTROL
7.5.

the one used in the discussions in this report is that of a modified IBM System/360 operating system. All terms in this section of the report are consistent with Operating System/360 terminology.

To put the discussions on the numerous facets of executive control in perspective, formal and informal requirements and underlying philosophies are covered first. General features and historical design considerations are also presented in order to provide background for the body of the report--Executive Control Functions.

Some of the Executive Control reports used in the preparation of this paper are:

CHIVE/N-2-64	Types of Multiprogramming
CHIVE/N-3-64	Preliminary Executive Control Flow Chart Explanation
CHIVE/N-5-64	Preliminary Macro Definitions
CHIVE/N-8-64	Executive Control - Transaction Management
CHIVE/N-9-64	Core Storage Management
CHIVE/N-16-64	General Functional Characteristics of the Executive Control Subsystem

EXECUTIVE CONTROL
7.5.

SECRET

CHIVE/N-1-65 I/O Management

CHIVE/W-28-64 Preliminary Functional Design of
the Executive Control Subsystem

7.5.1. SYSTEM REQUIREMENTS ON THE EXECUTIVE CONTROL SYSTEM

This section contains a compiled set of formal and informal requirements levied on the Executive Control System by other parts of the total CHIVE system. This list of requirements were extracted primarily from CHIVE/W-17-64, however, comments from other reports and meetings have also been used when appropriate.

7.5.1.1. Turn-Around Time Constraints

The most stringent requirement for turn-around time is given for priority information queries. In summary, this requirement reads: One priority information query can be expected every thirty minutes in the initial system, and it should be processed in two minutes (W-17-64). This seemingly innocuous requirement interjects a strong flavor of real-time into the system. This differs from other requirements which point to a more conventional system.

7.5.1.2. Priorities

The system must be able to handle jobs having various processing time constraints on them. These time limita-

EXECUTIVE CONTROL
System Requirements
7.5.1.2.

SECRET

tions are inherently contained in the "priority" of the job. At present, three levels of priority have emerged in the system--priority, normal, and background. Undoubtedly more levels will emerge as design continues. A priority specification will be assigned externally (explicitly) to every job that enters the system. Internal priorities based on these external priorities will also be used within the executive control system to schedule jobs for execution.

7.5.1.3. Input Processing

The primary requirement for the ECS in this area is to accept input jobs into the system on demand. This concept of "demand processing" calls for jobs to be read in and stacked on the input staging unit. They will be picked up in priority sequence from the staging unit when they can be executed.

7.5.1.4. System Logs

In Chapter 5.5. of this report a number of logs are discussed. The ECS will be called on to handle and monitor many of these logs.

7.5.1.5. Operator Communications

In a multiprogrammed system, the operator must be made aware of the status of the processing. This does

EXECUTIVE CONTROL
System Requirements
7.5.1.5.

SECRET

not imply that continuous on-line printouts are required, but the operator should be kept informed on significant internal happenings. For example, advance notice of job requirements would be given, such as the mounting of required storage devices.

7.5.1.6. Non-CHIVE Jobs

It is expected that CHIVE and non-CHIVE jobs will make concurrent use of the same computer. Within the structure of a multiprogrammed system this dual type of operation is certainly possible. Since this is a recent requirement, its full import on CHIVE requirements has not been ascertained.

7.5.1.7. Checkpoint/Restart

Some minimal capability is required for this programming function. Two types of procedures will be used. The ECS will execute macros from operational programs to provide a means of checkpointing and restarting. Checkpoints will include the core storage contents associated with the transaction requesting the checkpoint, as well as designated files. Long operational programs on the other hand can be written in such a way that checkpoints of their data files can be performed by the program itself.

EXECUTIVE CONTROL
System Requirements
7.5.1.7.

SECRET

SECRET

7.5.1.8. Backup Processing

There should be a minimum capability for using CHIVE programs and files on a backup computer. This implies the existence of a minimal executive control system. All basic functions should be provided through the ECS for a backup system, but this capability should not require extensive additional design and reprogramming activities.

7.5.1.9. Interpreting/Parsing

Requests to the CHIVE system will be written in a specially designed user language. This language will have to be interpreted, parsed and compiled into computer instructions. Some means of integrating this activity into the structure of the executive control system is required.

7.5.1.10. Security

Security considerations are discussed in Chapter 7.11. Security in the EDP system will have to be handled for the most part through the ECS.

7.5.2. DESIGN PHILOSOPHY

An overall philosophy of CHIVE Executive Control has evolved since work began in May of last year. This philosophy is based on CHIVE requirements, on a general

EXECUTIVE CONTROL
Design Philosophy
7.5.2.

SECRET

understanding of current computer hardware and software capabilities, and on professional experience and intuition. In a sense, this philosophy serves as a bridge between the CHIVE system requirements and reasonable executive control functions. The main points of this philosophy are listed and discussed below.

7.5.2.1. Multiprogramming

Multiprogramming in CHIVE is a necessity to effectively handle the computer's resources as well as to adequately meet CHIVE processing requirements. An executive control system designed along multiprogramming lines attempts to share all of the resources of the computer (CPU control, storage, I/O communication paths) between jobs in varying stages of execution. This implies that some jobs will have to be queued while they are awaiting the use of a temporarily busy or an overloaded resource. System job control is, then, a major function of the Executive System.

7.5.2.2. I/O Bound

Because of the large volume of external storage required in the CHIVE system, communication paths to external storage units will necessarily be heavily used.

EXECUTIVE CONTROL
Design Philosophy
7.5.2.2.

This condition implies that the CPU will at times be idle while data is being transferred to or from the external units. Since this condition is recognized as a strong possibility, steps should be taken in all phases of design work to use CPU and I/O resources in an optimum manner.

7.5.2.3. Demand Processing

Even though some of the present turn-around time requirements are very tight, the CHIVE system does not have the overall characteristics of a real-time system. However, since input data will be read in immediately upon receipt, it can be termed a "Demand Processing" system.

7.5.2.4. Modular Design

In the basic design of a comprehensive operating system, simplicity and flexibility must be paramount objectives. These characteristics can best be realized by using the concept of modularity in the design of the programming components of executive control. Programming considerations are reduced and growth potential is increased with such a design goal.

EXECUTIVE CONTROL
Design Philosophy
7.5.2.4.

7.5.2.5. Macro Oriented

Since CHIVE's multiprogramming system separates programming responsibilities into executive control and operational, effective communication between all parts of the system must be maintained. This is accomplished through an extensive set of system macros. By using these macros all requests from operational programs are channeled to the appropriate executive control programs which perform the required activity. See Appendix 7.D. for a further discussion of system macros.

7.5.2.6. Programming Services

A number of additional programming services can be provided in the system because of the centralization of control that will exist in the Executive Control System. Such features as debugging aids, checkpoint capability and diagnostic interfaces are possible and desirable through the comprehensive structure of the Executive Control System

7.5.2.7. Bulk Core Storage

Bulk core storage is useful in the CHIVE computer. Bulk core is similar to regular core except that it has a slower access speed and comes in blocks of considerable

EXECUTIVE CONTROL
Design Philosophy
7.5.2.7.

SECRET

size. The main reasons for using bulk core in CHIVE are to:

- Reduce or eliminate reading of programs from external storage units.
- Reduce or eliminate roll-out and roll-in. (See Section 7.5.6.2.)
- Enable temporary results to remain in core rather than in data sets on direct access devices. (See Section 7.5.7.)

7.5.3. SYSTEM INPUT/OUTPUT

The input/output functional area in OS/360 is described diagrammatically in Figure 7-2. The modified OS/360 version will be harmonious with this structure. Possible changes may be necessitated by the CHIVE priority and input language decomposition requirements.

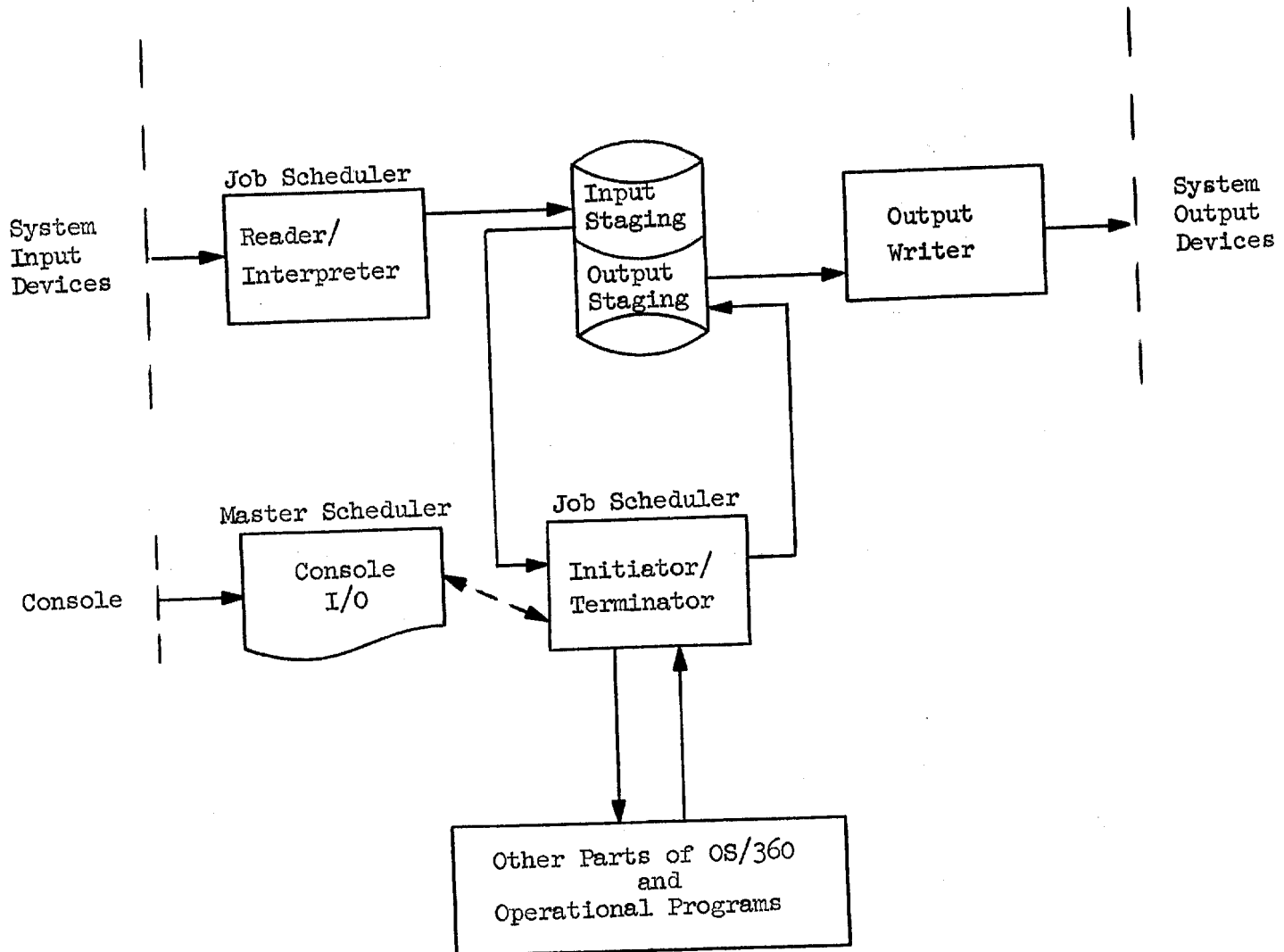
EXECUTIVE CONTROL
System Input/Output
7.5.3.

SECRET

SECRET

Figure 7-2

INPUT/OUTPUT INTERFACES IN OS/360



SECRET

7.5.3.1. Input Processing

The major functions performed at this level are:

- Reading input control information onto the input staging device.
- Error checking and tabulation of job control information.
- Allocation of resources to jobs.
- Scheduling of job steps to process each job through the system.
- Handling all correspondence from the console (operator).

OS/360 will provide appropriate and adequate input programs to perform all phases of input processing. The Job Scheduler is composed of two parts which handle input processing. The Reader/Interpreter will read "on demand" input from all system input devices and queue them into priority sequence on the random access input staging disk. Then it will create a Job Control Table and allocate resources before the job is scheduled for execution. The second section of the Job Scheduler will pick up the jobs from the input staging device and sets up a task control block (TCB) for each task. The Master Scheduler specifically handles all correspondence to and from the console.

7.5.3.2. Output Processing

When individual tasks are being executed within a job, generated output will be written on the random access

EXECUTIVE CONTROL
System Input/Output
7.5.3.2.

SECRET

output staging device. The output will be accumulated in this way until an "end of output" signal is given to the Terminator (output end of Job Scheduler). The Output Writer is relayed the necessary information, and the data on the output staging device will be serially written on the appropriate system output devices.

The revisions to OS/360 in the area of Output Processing will be primarily in the incorporation of remote output terminals to the system. This modification in the initial version of OS/360 will vary in complexity depending on whether common carrier facilities are used.

7.5.4. TASK MANAGEMENT

After a job is entered into the system and the necessary preliminary activities are performed by the Job Scheduler, subsequent processing of that job is handled by the Task Dispatcher and the Task Supervisor. Task Management will dynamically supervise and monitor all tasks that are in the system. This includes such functions as:

- Selection of tasks for execution.
- Maintaining the status of the program related queues.
- Servicing of all types of interrupts (except I/O).
- Maintaining logs of important system events.
- Providing communication services between programs.

EXECUTIVE CONTROL
Task Management
7.5.4.

SECRET

OS/360 provides adequate support in the area of task management.

7.5.4.1. Task Dispatcher

The dispatching procedure centers around the selection and preparation of tasks for execution. The more important steps in this procedure are:

- Selection of the task at the top of the Ready Queue for execution.
- Requesting of core storage and reading in of programs from the system library (if the programs are not already in core).
- Performing dynamic program relocation and assigning storage protection keys (if not already done).

A requirement may emerge to alter the rule by which the Ready Queue is ordered. Also some modifications in the assigning of storage protection keys may be required to meet security specifications.

7.5.4.2. Task Supervisor

The main function of the Task Supervisor is to monitor the execution of tasks. This supervision is characterized by the following:

- Handling all types of communication between programs.
- Routing all interrupts to their respective service programs.

EXECUTIVE CONTROL
Task Management
7.5.4.2.

- Maintaining system queues, lists, and blocks.
- Performing standard and ad hoc logging of system events and activities.

Except for the possible exception of some specialized types of logs, all the above functions will be performed by OS/360.

7.5.5. CORE STORAGE MANAGEMENT

The purpose of core storage management is to:

- Allocate and release core storage blocks on request.
- Maintain control over all temporary core storage areas.
- Resolve conflicting needs for core storage.

The handling of I/O buffers by OS/360 might be changed to give a better utilization of core by assigning I/O buffers just prior to I/O command execution time rather than at data set open time. Core storage usage could be reduced by requiring all programs to be reentrant (a reentrant program is so written that more than one task can use it concurrently) so that system will operate more efficiently when core is overburdened.

In OS/360 tasks and some supervisor functions are the entities that are assigned core storage. The main storage supervisor in its general control over core storage maintains

EXECUTIVE CONTROL
Core Storage Management
7.5.5.

SECRET

interrelated references to a task's storage areas. These references take the form of special lists and keep track of storage blocks assigned to a task as well as free areas within these storage blocks. (OS/360 assigns storage in multiples of 2K byte blocks)

7.5.5.1. Storage Macros

GET MAIN is the macro used to request core storage from OS/360. Requests can be of three types. The amount of space allocated by type of request in OS/360 is as follows:

Types of Requests	One Block Request	Multiple Block Request
Non-Conditional-- Must have specified amount.	Block of size specified	Each block of a specified size
Conditional--Special return if request cannot be met.	Block of size specified	Each block of a specified size
Variable--Special return if minimum length is not available.	Largest block available within specified limits	

If the request is for problem program storage (task storage), the request is filled if possible with available storage from the task's previously assigned blocks.

Otherwise, groups of 2K blocks are allocated to the task

EXECUTIVE CONTROL
Core Storage Management
7.5.5.1.

SECRET

SECRET

to help meet the request. Other types of storage are handled similarly, except for read only program areas. These are handled differently because of future uses that can be made of their storage areas.

FREE MAIN is the macro instruction to dynamically release main storage, so that it becomes available for future use. The return procedure is checked for recognizable errors. If the return frees one or more 2K byte blocks, the blocks are returned from the task storage to overall free storage. A check is made to see if the 2K block could fulfill a pending roll-out or is needed for a roll-in.

7.5.5.2. Roll-out and Roll-in

When storage is requested and no space is available, Core Storage Management tries to free space occupied by a program (read only) that is not being used. If this fails, Core Storage Management will acquire the space occupied by one or more tasks by writing (rolling out) the contents on a direct access device. The decision to roll-out a task is made on the basis of task priority.

Roll-in (reading these sections back into core) takes place automatically as soon as the original space is available again. If conflicts over core use occur often,

EXECUTIVE CONTROL
Core Storage Management
7.5.5.2.

SECRET

it may be desirable in OS/360 to have data sections organized in a way they could be rolled into areas other than the original area.

7.5.6. DATA MANAGEMENT

A multiprogramming system such as that envisaged for CHIVE implies that the I/O Supervisor will take on more functions than the traditional IOCS. Brief definitions of those terms used in this section are as follows:

- Data Sets: a named collection of data (previously, this was called a logical file)
- Volume: a physical piece of hardware used for storing data sets (previously, this was called a physical file)
- Catalog: a hierarchical structure of indexes of data sets names for locating data sets.

The additional functions (beyond those of conventional IOCS systems) included in OS/360 are:

- Allocation of space on direct access devices.
- Expanded library and catalog organization including automatic location of data sets by name alone.
- More device independence given in part by the capability to defer specifications such as buffer size, blocking factor, device identification, and in some cases device type until the job is submitted for actual execution.
- Wide variety of supported data organizations.

EXECUTIVE CONTROL
Data Management
7.5.6.

SECRET

- Protection of data sets by:
 - Not executing I/O orders for a data set outside of its extent, i.e., its logical and physical boundaries.
 - Protection of a classified file by use of an external "password" or key.
 - Protection against concurrent updating of same record by different tasks.

The protection of classified files by use of a password requires operator action in OS/360. If it is desired to have an internal security tag on a data set rather than on record or lower level, an internal data set password, handled completely internal to the programming system, would be more desirable and would be incorporated in modified OS/360.

7.5.6.1. Data Set Location Control

The functions of controlling space on direct access devices and of finding and keeping track of data sets are covered in this section.

Every direct access volume has a volume table of contents used to keep track of its data sets and free space. When a user desires to create a data set on the volume, Data Management assigns the data set space from the volume's free space, and enters a data set label for this data set in the volume table of contents. If the

EXECUTIVE CONTROL
Data Management
7.5.6.1.

SECRET

SECRET

data set is to be cataloged, additional work as described in the following section is required.

Cataloging of a data set permits its storage and retrieval by name alone, otherwise the volume number would be needed. Generated data sets are easy for the user to reference, because the system automatically recatalogs them when a new member is added. Also a set of control statements and macro instructions are provided for maintenance of the library and catalog.

7.5.6.2. Data Set Organization and Use

OS/360 will probably support the access methods and data set organizations presented below. There are three record formats:

- Fixed-length logical records
- Variable-length logical records
- Record of unspecified characteristics (each physical record is treated as one block without regard to internal structure)

7.5.6.3. Data Set Organization

Four data set organizations will be available. In the sequential organization each record in a data set is related to others by its position.

The indexed sequential method applies to any direct access device and provides sequential and random accessing.

EXECUTIVE CONTROL
Data Management
7.5.6.3.

SECRET

A data set is created in sequence of both physical location and collating sequence. If subsequent insertions cannot be put into proper sequential location, they are placed in a separate area and chained in logical sequence. Cylinder and track indexes provide random accessing of information.

The direct organization for direct access devices lets the user specify where each record is located. A relative track address is used as a starting point of search for the referenced record.

The partitioned data set organization lets the programmer collect groups of sequential records on a direct access device for rapid retrieval through a directory within the data set.

7.5.6.4. Data Set Accessing Languages

The two supported language levels are Basic and Queued. The Basic level is for physical record accessing; the Queued is for logical records. Queued describes the automatic queuing of input and output records on buffers. The supported access methods are identified as follows:

<u>Data Organization Types</u>	<u>Queued</u>	<u>Basic</u>
Sequential	QSAM	BSAM
Partitioned	XXX	BPAM
Indexed Sequential	QISAM	BISAM
Direct	XXX	BDAM

EXECUTIVE CONTROL
Data Management
7.5.6.4.

SECRET

OS/360 may provide for user provided machine language access methods and record formats.

7.5.6.5. I/O Buffers

Buffers are located in user core and are allocated either at assembly time or dynamically at execution time.

Techniques for their use are:

- Simple Buffering: Buffer belongs only to data set using it.
- Cyclic Buffering: Buffers for physical records are shared; and ownership may pass from one data set to another, depending on use.
- Exchange Buffering: Buffers for logical records are shared between input and output data sets.

Buffers belonging to data sets of a task as in OS/360 simplify the I/O Supervisor. This does not give as good core utilization as dynamic assignment of buffers just prior to I/O command execution.

7.5.6.6. OPEN and CLOSE Macros

The OPEN macro operates on a Data Control Block (DCB) that refers to the desired data set. The DCB provides data set descriptive information such as name, organization, language, buffering techniques, and location. This information is provided by the program symbolic deck, Data Definition Statement (from the Job Control Language),

EXECUTIVE CONTROL
Data Management
7.5.6.6.

SECRET

SECRET

and the Data Set Control Block. The CLOSE macro restores the DCB to its original condition prior to open.

7.5.7. ADDITIONAL EXECUTIVE CONTROL FUNCTIONS

In addition to the primary functions discussed above, several secondary functions can be easily implemented due to the amount of control inherent in OS/360. A few of the more important extras incorporated into OS/360 are discussed.

7.5.7.1. Debug Supervisor

Since there are many activities being performed concurrently in a multiprogrammed system, the necessity for a central control over program errors is paramount. Increased debug services can be offered through the operating system to the operational programmers. These services provide a set of debug macros unparalleled in conventional operating systems.

7.5.7.2. Checkpoint/Restart Supervisor

Most programs that require a long execution time make periodic checkpoints of part or all of their intermediate results. These checkpoint records would be initialized or restarted if an error occurred in later stages of processing. In addition to this standard practice, OS/360 will provide

EXECUTIVE CONTROL
Additional Functions
7.5.7.2.

SECRET

the user, upon request, dynamic checkpoints of the areas in core that are related to his programs and data.

7.5.7.3. Diagnostic Supervisor

Even on today's highly reliable equipment, malfunctions still occur. Many of the hardware errors can be analyzed and isolated by special programming packages called diagnostic programs. OS/360 has several types of diagnostic programs built into its structure. Whenever a hardware failure has been recognized, OS/360 will call in automatically the appropriate diagnostic program. If possible after the reason for failure has been found, normal processing will be continued with the malfunctioning hardware blocked off until it can be repaired.

EXECUTIVE CONTROL
Additional Functions
7.5.7.3.

SECRET

CHAPTER 7.6

SUMMARY OF INTERNAL DATA STRUCTURES

The CHIVE system data files will contain data organized according to the following well defined structure:

- Term: A term is the basic logical data unit. This is made up of two parts, a tag and a value. The tag defines the "term type" or use in the system and the value is the actual data of the term.
- Phrase: A phrase is made up of two or more terms linked together to form a meaningful unit of data. (In some cases, phrases with only one term can occur.)
- Phrase type: Two types of phrases are defined in the CHIVE system: header phrases and content phrases.
- Header Phrase: A CHIVE record contains exactly one header phrase.
- Content Phrase: Any number of content phrases can occur in each record of a CHIVE file.
- Phrase Group: A set of content phrases in a CHIVE record terms of the same types, i.e., with the same tags.
- Record Control Terms: Each record a CHIVE file will have one or more terms which taken together contain unique data value(s) in the file. The record control terms therefore uniquely identify each record. There can be only one occurrence of these record control values in a record.

SUMMARY OF DATA STRUCTURES
7.6.

SECRET

7.6.1. FILE FORMAT TABLES

The format of each CHIVE system data file is defined by a File Format Table (FFT). The File Format Table describes the format and content of each system data file and provides an interface between the CHIVE files and problem programs. The declaration of file content and structure via an FFT frees CHIVE programs from consideration of a particular file. Program development and implementation need be oriented only towards the general classification of structures which can be represented in an FFT and the processing techniques which can be applied to these structures.

The FFT also provides a mechanism by which a program can define an intermediate file to subsequent programs requiring access to the intermediate file.

7.6.1.1. FFT Jurisdiction

The employment of File Format Tables in CHIVE will be restricted to the definition of system data files. System processing files (e.g., program and subroutine libraries), directory files, and OS/360 input and output staging files are excluded from FFT definition.

The FFT will provide the problem programs, which are not oriented towards a particular CHIVE file, sufficient information to process any system data file.

SUMMARY OF DATA STRUCTURES
File Format Tables
7.6.1.1.

SECRET

7.6.2. CONTENT OF FILE FORMAT TABLES

File Format Tables will contain various structural definitions and process controls in a tabular format. The structural definitions outlined below will be carried primarily at the tag level, supplying an entry point to a record for each tag declared eligible in the file.

7.6.2.1. Phrase Level of Tag

Indicates the phrase group to which the tag belongs.

7.6.2.2. Occurrences of Tag Allowed in a Phrase

The number of terms which can be represented by the tag. The eligible occurrence modifiers are one (1) and variable (V).

7.6.2.3. Occurrences of Tag Required in a Phrase

The requirement parameter specifies if the appearance of a term with the tag is mandatory or optional.

7.6.2.4. Term Value Length

The number of characters which can be employed to represent the value of the term in the record. This parameter consists of the maximum and minimum number of characters allowed.

SUMMARY OF DATA STRUCTURES
Content of Format Tables
7.6.2.4.

SECRET

7.6.2.5. Value Location Within EDP Record

The addresses required to calculate the location of the values for the tag in each record. The values of fixed-length terms which appear will be referenced by a single displacement vector pointing to the high order position of the value, relative to the starting position of each record.

The values of all other terms not meeting the above criteria will be located by the displacement vectors listed below. Figure 7-3 and Section 7.6.6. describe their use in accessing all the terms within a specified record.

- An address vector in the FFT specifying the relative location of a Term Locator Control Block. This TLCB is located in the same position in each record of the file.
- The Term Locator Control Block of each record contains a second displacement vector locating a Value Locator Block in the record servicing the tag of interest. The position of the VLB will vary from record to record.
- A second displacement vector in the FFT locates the position in the Value Locator Block which contains the actual displacement vector to the values of all terms with the particular tag in the record. The second FFT vector is required because the Value Locator Block will specify the actual displacement vectors for multiple tags.

SUMMARY OF DATA STRUCTURES
Content of Format Tables
7.6.2.5.

SECRET

Figure 7-3

CHIVE System Data File Record Structure

Dedicated Fields

RECORD LENGTH	RECORD CONTROL		TAG 3	TAG 4	TAG i
Count	TAG 1	TAG 2	Value	Value	Value
	Value	Value			

PHRASE GROUP MASK LENGTHS			TERM LOCATOR CONTROL BLOCKS			
P.G. 1	P.G. 2	P.G. j	TLCB 1	TLCB 2	TLCB k	TLCB h.p.
Length	Length	Length	Vector to VLB 1	Vector to VLB 2	Vector to VLB k	
TAG 5 - HEADER PHRASE TAG						
# of Values	Length 1	Value 1				
	Length 2	Value 2				
	Length n	Value n				

TAG 6 - PHRASE GROUP 1 TAG			
# of Values	Length 1	Mask 1	Value 1
	Length 2	Mask 2	Value 2
	Length n	Mask n	Value n

TAG 7 - PHRASE GROUP 2 TAG			
# of Values	Length 1	Mask 1	Value 1
	Length 2	Mask 2	Value 2
	Length n	Mask n	Value n

TAG M - PHRASE GROUP j			
# of Values	Length 1	Mask 1	Value 1
	Length n	Mask n	Value n

VALUE LOCATOR FIELDS			
VLB 1		VLB 2	VLB k
Vector to TAG 5	Vector to TAG 6	Vector to TAG 7	Vector to TAG m

SECRET

7.6.2.6. Mask Length Location

The address of the field in the record which contains the phrase mask length for the phrase group of which the tag is a member.

7.6.3. ADDITIONAL FILE FORMAT TABLE FUNCTIONS

Special FFT services required by problem programs cannot be completely identified until program design has progressed to a more advanced state. A preliminary outline of required process controls include the following:

- Identification of tags employed as record and phrase controls.
- Tags dedicated to contain record security classification and dissemination.
- Identification of the directory files servicing the system data files.
- An indication if the values for the tag are eligible for conditional maintenance.
- Justification procedures for each tag, indicating the method of aligning variable length values for comparison. The modifiers include:
 - Right justification
 - Left justification
 - Floating justification: a special character in the value is to be employed as a reference point.
- An indicator for each tag specifying if the tag is represented in a directory file.

SUMMARY OF DATA STRUCTURES
Additional Functions
7.6.3.

SECRET

- Subroutine or table declarations for each tag whose values require special processing for conversion from external to internal format. The declaration will be employed by retrieval, unconditional and condition maintenance programs.
- General value characteristics, which at a minimum specify if a value is in binary coded decimal, packed decimal, or binary representation.

7.6.4. FILE FORMAT TABLE MODIFICATION

The CHIVE retrieval programs will have some capability to modify the data structure and content of an existing system data file. The modified files produced by retrieval are candidates for inclusion as system data files. In addition to the modified file, retrieval produces a File Format Table specifying the structure of the revised file, which must be included in the system library if the revised file is to become a permanent member of the CHIVE EDP system.

File revision requirements not within the purview of retrieval must be processed as new file creations, with a program assigned to accomplish the required transformation. The supporting File Format Table must be generated and introduced into the CHIVE library as a new file format definition.

7.6.5. SYSTEM DATA FILE ORGANIZATIONS

The organization of the system data files for storage and retrieval is dependent on the storage devices

SUMMARY OF DATA STRUCTURES
File Organizations
7.6.5.

SECRET

SECRET

and the expected activity of the files. The files will be organized either as ordered sequential files on tapes or direct access storage devices, or as directory accessed files stored on direct access storage devices. CHIVE/N-1-65 describes the various standard file organizations and accessing methods provided by OS/360. The sequential files will be physically ordered by their record control terms. Sequential files stored on direct access storage devices will be processed using the OS/360 indexed sequential access method. This technique provides direct access to records through index tables which relate the record control terms to the physical location of the records. The processing of these files will use routines and macros which are a standard part of the Operating System/360.

7.6.5.1. Sequential Files on Tape

Files stored on magnetic tape will be ordered on the terms designated for record control. File maintenance transactions, and brand new records will be processed in batches defined as jobs to the Operating System/360. Unconditional maintenance transactions in these batches will be sorted on the record control terms into the same order as the file records prior

SUMMARY OF DATA STRUCTURES
File Organizations
7.6.5.1.

to the actual maintenance processing. The original file and the maintenance batch will then be processed sequentially against each other using the OS/360 sequential access method. A new updated file will be produced on another tape. Logical maintenance transactions can be processed in the same batch as the specific unconditional maintenance transactions by appending a blank sort key to each conditional transaction before ordering the batch. They can be grouped together and matched against all the file records.

7.6.5.2. Sequential Files on Direct Access Storage Devices

Files may also be stored on direct access storage devices for sequential access. The records in these files will be ordered on the record control terms and stored sequentially using the OS/360 indexed sequential technique. This will permit direct access to specific records for maintenance and output. Specific file maintenance transactions can be processed individually even if inputted as batches, or in batches ordered by record control. Logical maintenance transactions or retrieval transactions require a complete search of the file and should be processed in batches for efficiency to avoid multiple passes against the file.

SUMMARY OF DATA STRUCTURES
File Organizations
7.6.5.2.

7.6.5.3. Directory Accessed Files

Directory accessed system data files will consist of three distinct segments or subfiles. The main segment will contain the master data records in the order they are received by the system. The other two segments will contain directory files. One directory file will provide access to the records through the record control terms and the other through access terms designated in the File Format Table. These access terms can be selected from the header and content phrases of the master data records.

The directory files will be stored on direct access storage devices using the OS/360 indexed sequential access method. The records in the directory file will consist of a term and a set of postings, one posting for each phrase which contained the term. Each posting will contain a storage address for the master record and a phrase number to show which phrase in the record contained the term. The records in the directory file will be ordered by the tag and the specific value. Changes and additions to the directory files will be automatically generated from maintenance and record addition transactions processed against the system data

SUMMARY OF DATA STRUCTURES
File Organizations
7.6.5.3.

SECRET

file. No changes can be specifically directed against the directory files.

7.6.6. CHIVE SYSTEM DATA FILE RECORD STRUCTURE

CHIVE system data file records will have the same general record structure for all types of physical storage devices. The data terms and required control data will be contained in fixed length and variable length fields. The tags can be partitioned into arbitrary tag groups by the user, according to their frequency of co-occurrence. These tag groups do not have to be related in any way to the phrase group(s) to which the tags belong. The phrase linkages of terms are represented by phrase mask fields which are associated with each individual term value. This mask in each record will have a single binary bit position dedicated to each content phrase of the phrase group to which the term belongs. A mask bit position set to one indicates that the term value was contained in the phrase represented by that position. A bit position set to zero indicates that the term was not contained in that phrase. The phrase masks from all the terms in a phrase group would form a coincidence matrix of binary bit positions with each row representing a term and each column a

SUMMARY OF DATA STRUCTURES
Record Structure
7.6.6.

SECRET

phrase. This coincidence matrix concept is illustrated in Figure 7-4.

The records in a system data file will have two major sections as shown in Figure 7-3--a dedicated section and an optional section. The dedicated section of a record will be made up of fixed length data and control fields which will have the same position and space allocated to each such field in all records of the file. The position and length of the dedicated data fields and internal control fields will be stored in the File Format Table. The optional section of the records will be made up of fixed and variable length data fields and control fields. If they do not occur in a particular record of the file no storage space is allocated to them. The position and length of the optional data fields is stored in Value Locator Blocks in the optional section of the record. The position of these Value Locator Block fields is, however, stored in Term Locator Control Block fields located in the dedicated portion of the record.

7.6.6.1. Record Length

This control field contains a binary count of the total record length in bytes. This controls the

SUMMARY OF DATA STRUCTURES
Record Structure
7.6.6.1.

SECRET

Figure 7-4

MATRIX REPRESENTATION OF TERMS IN A PHRASE GROUP

Logical Phrase Group				
Phrase Number	Terms			
1	A1	B1	C1	D1
2	A2	B2	C1	D2
3	A3	B2	C2	D3
4	A2	B1	C2	D1
5	A3	B3	C2	D1
6	A1	B2	C1	D2

A_i, B_i, C_i, D_i, = Term with tag A, B, C, or D and value i

		Phrase Numbers							
Terms		1	2	3	4	5	6	7	8
A1		1	0	0	0	0	1	0	0
A2		0	1	0	1	0	0	0	0
A3		0	0	1	0	1	0	0	0
B1		1	0	0	1	0	0	0	0
B2		0	1	1	0	0	1	0	0
B3		0	0	0	0	1	0	0	0
C1		1	1	0	0	0	1	0	0
C2		0	0	1	1	1	0	0	0
D1		1	0	0	1	1	0	0	0
D2		0	1	0	0	0	1	0	0
D3		0	0	1	0	0	0	0	0

SECRET

storage of the record on the storage devices and defines its length for processing routines.

7.6.6.2. Record Control Fields

These fields contain the value of the record control terms. They are located in their ranking order immediately following the record length fields. They are used to order sequential files and to uniquely identify a single record for file maintenance operations.

7.6.6.3. Dedicated Data Fields

These fields contain the values of fixed length data terms which are required to appear in every record. Their location is given by a displacement vector specified directly in the File Format Table.

7.6.6.4. Phrase Group Mask Length Fields

These control fields contain a binary count of the actual number of phrases in each phrase group of a particular record. The actual length of the phrase mask field will be the minimum number of bytes required to store this number of bits.

7.6.6.5. Term Locator Control Blocks

The term types or tags will be partitioned into tag groups according to the expected co-occurrence of the

SUMMARY OF DATA STRUCTURES
Record Structure
7.6.6.5.

data terms in records. A block will be designated in the record for each tag group which has been defined in the File Format Table. Each block will contain a 16-bit binary vector address showing the relative location of the Value Locator Control Block in the record if any terms with a tag belonging to the particular tag group are present in the record. It will contain all zero bits if no tag of its tag group is present in the record. A special TLCB will also be reserved for tags not defined in the FFT which are appended to the record during retrieval "Hit Processing." This occurs when retrieval processing is not under FFT Control.

7.6.6.6. Value Locator Blocks

These blocks will normally occur as the last fields of a record. Their presence in the record is indicated by the presence of a non-zero vector address in the associated Term Locator Control Block. The length of each Value Locator Block will be indicated in the File Format Table. A Value Locator Block will be made up of a set of 16-bit Value Locator Fields, one for each tag in the group of tags controlled by the block. Each Value Locator Field in the Value Locator Block

SUMMARY OF DATA STRUCTURES
Record Structure
7.6.6.6.

will contain either all zeros if no terms with the particular tag are present in the record, or a location vector showing the relative location of the Value Count Control Field for the set of values for terms with that tag in the record.

7.6.6.7. Optional Value Fields

The value fields will be partitioned by term type tag. The individual values for a particular tag will be ordered in /360 collating sequence within the partition for that tag. Each partition will be headed by a Value Count Control Field showing the number of values with this tag in the record. Each value field will be preceded by a Value Length Control Field showing the number of bytes of data in the value field. If the value field is for a term in a phrase group that can occur more than once, it will also be preceded by a Phrase Mask Field. Terms appearing in a phrase group that can occur only once per record will have an entry made up of a Value Length Field, and a Value Field.

7.6.7. SEARCHING FOR A VALUE FIELD

The dedicated value fields will be searched directly by obtaining their relative location vector and length

SUMMARY OF DATA STRUCTURES
Searching for a Field
7.6.7.

SECRET

from the File Format Table. The optional value fields will be searched indirectly. This access will use the File Format Table entries and control fields in the record as described below.

Step 1: Using the File Format Table extract the relative location in the record of the Term Locator Control Block for the particular tag. Also extract the location in the Value Locator block of the Value Locator Field for the particular tag (for use in Step 3).

Step 2: Extract the relative location of the Value Locator Block within the record from the Term Locator Control Block.

Step 3: Extract the relative location of the Value Count Field from the Value Locator Field in the Value Locator Block.

Step 4: Search through the individual value fields using the Value Count Field, the Value Length Fields and the Phrase Mask Length to control the search stepping. A search for a particular value would involve match comparisons against all the value fields. A search for a term value in a particular phrase would also match against the particular bit position representing that phrase in the phrase mask.

SUMMARY OF DATA STRUCTURES
Searching for a Field
7.6.7.

SECRET

7.6.8. DIRECTORY FILE STRUCTURE

The directory files will have a very simple structure. The Record Control Directory File and the Term Director File are the same in overall file organization. They will be stored and accessed using the Operating System/360 indexed sequential technique. They are defined and processed as distinct file types, however, because they have different record structures (see Section 7.7.5.).

7.6.8.1. Term Directory File Record Structure

The Term Directory File records will be variable length. The record structure is shown in Figure 7-5. The first field in the record will be a record length field containing a binary number defining the record length in characters or bytes. This will be followed by the term entry consisting of a one byte code for the tag, a value length field and a variable length value field. The record postings will consist of a 24 bit record address and an 8 bit phrase number. Each posting will be 32 bits or one /360 computer word. For processing ease the first posting will be located in the first complete computer word following the term entry. From one to three blank bytes will be inserted in the record following the value field to make the length of the record count and the term entry length an even multiple of four bytes.

SUMMARY OF DATA STRUCTURES
Directory File

-64- 7.6.8.1.

SECRET

Figure 7-5

DIRECTORY FILE RECORD STRUCTURES

Term Directory File Record

Record Length	Tag	Val. Lgth	Value
---------------	-----	-----------	-------

Posting 1		Posting 2		}	}	Posting (n)	
Record Location	Ph. No.	Record Location	Ph. No.			Record Location	Ph. No.

Record Control Directory File Record

Record Control Fields	Record Location
-----------------------	-----------------

SECRET

7.6.8.2. Record Control Directory File Record

The record control directory file records will be fixed in length. The record control terms which are located in the dedicated section of the system data file record are fixed length. There will be only one record address for each record of this file. This structure is also shown in Figure 7-5.

SUMMARY OF DATA STRUCTURES
Directory File
7.6.8.2.

SECRET

Chapter 7.7.

FILE MAINTENANCE

7.7.1. FILE MAINTENANCE PROCESSING

File maintenance processing covers two major activities, adding new data and changing data. The three types of file organization each present a separate problem in maintenance. Tape files must be copied to change records. Therefore file maintenance transactions should be batched for economy. The indexed sequential files on direct access storage have a direct access capability using the record control terms. These files can be changed without copying all the records onto another storage unit. New records can be chained into the sequence by reserving extra space throughout the file for this purpose. The file can then be copied onto a tape and reloaded when this chaining becomes too cumbersome. Directory accessed files must also have concurrent directory file maintenance transactions generated whenever a system data file change affects a directory file entry. These transactions cannot be completed until the system data file record is stored in its storage device.

FILE MAINTENANCE
Processing
7.7.1.

SECRET

Record changes, i.e., adding, deleting and replacing terms or phrases in records, can be classed in two ways according to the method of specifying the record to be changed. Specific maintenance transactions use the record control terms to specify an exact record. Adding complete records is always classed as specific maintenance. Conditional maintenance transactions use some combination of test conditions to specify one or more records to be changed. Conditional maintenance transaction which indicate the specific record to be changed but conditionally specify the phrases or terms to be updated will be handled as specific changes.

7.7.2. INPUT TRANSFORMATIONS

Transcript sheets or input records for adding records to files will be processed at input time into CHIVE format records. Special preprocessor routines will be developed to process the various types of inputs for each CHIVE file. The processing of these records will be controlled by the job specification card (see Appendix 7.A.3.) which will be transferred to the maintenance processing routines with the records.

7.7.3. RECORD MAINTENANCE

Record maintenance, which involves the modification of a single record after this record is found by the

FILE MAINTENANCE
Record Maintenance
7.7.3.

appropriate search mechanism will be similar for all files. The location of the data to be added, deleted or changed will be determined with record search routines. These routines use the File Format Table, to find a particular data field. After the field is deleted inserted or changed the necessary changes will be made in the Record Length Control Field, Tag Locator Blocks and Value Locator Blocks to complete the operation. After all the required changes have been made to the record it will be stored back in the file.

7.7.4. DIRECTORY FILE MAINTENANCE

Directory file maintenance transactions are generated whenever a change is made to a term in a directory accessed file which has been flagged in the File Format Table as an access term. A term posting deletion is generated if a term is deleted, a term posting addition if a term is added and both an addition and a deletion if a term is changed. These transactions will be processed as soon as the record has been returned to the file. . If a complete record is added to the file, a posting addition transaction for each access term in the record is generated before the record is stored. These transactions will be processed as soon as the system data record has

FILE MAINTENANCE
Directory File
7.7.4.

been stored in its file. If a record is deleted, one or more posting deletion transactions are generated and processed. The record is flagged and written back into the file and its location saved to be used for storing new records.

If a posting addition is processed for a term which has not previously been entered in the directory file, the term will be added to the directory file and an entry made on an output history tape to be used by the file analyst to check new terms for possible errors. If a delete posting transaction is processed which deletes the last record posting of an access term, the term will be deleted. An entry is made on the history tape and the disposition notice showing that the term is no longer used in the file.

FILE MAINTENANCE
Directory File
7.7.4.

Chapter 7.8.

RETRIEVAL HIT PROCESSING

The hit processing in the CHIVE EDP system involves those record modification actions taken after specified search criteria have been met, but before any output processing occurs. The relation of hit processing to the other major elements of a retrieval transaction are illustrated in Figure 7-6.

7.8.1. Hit Processing Functions

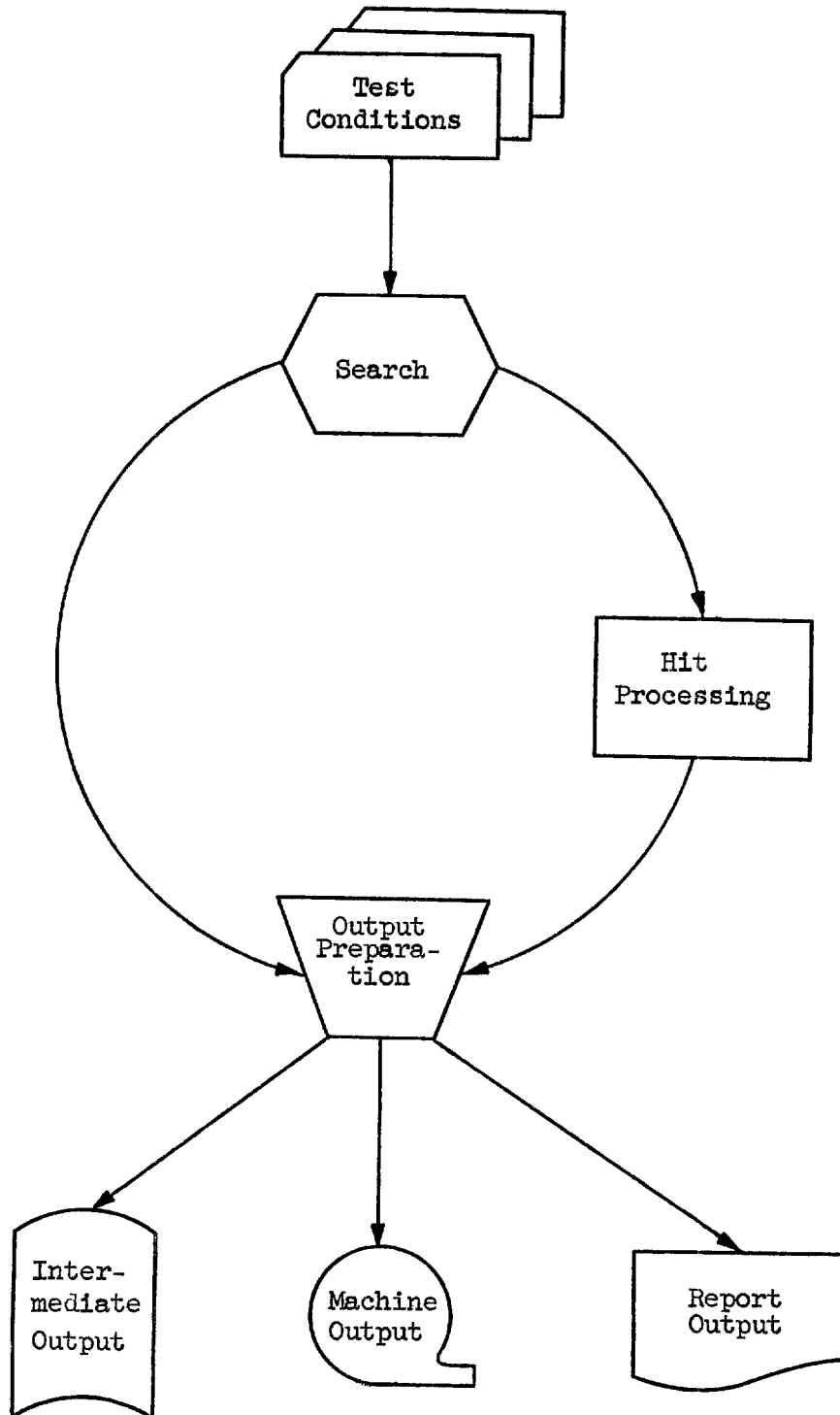
Hit processing functions are classified into four categories:

- Structuring Commands
 - Extract
 - Merge Records
 - Merge Phrases
- Editing Commands
 - Append
 - Delete
 - Replace
 - Define

HIT PROCESSING
Functions
7.8.1.

Figure 7-6

HIT PROCESSING FUNCTION



~~SECRET~~

- Arithmetic Commands
 - Tally
 - Compute
 - Accumulate
- Writing Commands
 - Copy
 - Explode
 - Write
 - Print

In general a command will include one or more operands:

Data - such as terms, sort keys, print fields, computed results, etc.

Files - either input or output files.

Search Criteria - such as the term, phrase or record specification for a "hit."

Limit values - a numeric value to indicate the maximum number of times that a command is to be performed.

Appendix 7.A. gives a complete description of the command syntax and symbol conventions.

HIT PROCESSING
Functions
7.8.1.

~~SECRET~~

SECRET

7.8.2. STRUCTURING COMMANDS

7.8.2.1. EXTRACT

COMMAND	LIMIT	SEARCH CRITERIA			FILE		DATA
		Record	Phrase	Term	I	O	
EXTRACT	(Number)	(Ident)	--	--			
		(Ident)	(Ident)	--	(Name)		
		(Ident)	(Ident)	(Ident)			

Command Description: EXTRACT causes the operand(s) specified in the Search Criteria to be written into a system generated work area for subsequent manipulation. No explicit reference to the work area is necessary. Its presence is assumed by the system.

The data elements that may be specified are terms, phrases, or whole records. No restrictions are placed upon the type of element that may be extracted.

The EXTRACT command operates upon records only and is the entry point for the inner loop of the Hit Processor. It exists principally to preserve records or record fragments for additional processing. Specific reference must be made to the input file upon which the command is to operate.

HIT PROCESSING
Structuring Commands
7.8.2.1.

SECRET

SECRET

The Limit feature will cause the EXTRACT operation to cease when the number of elements extracted equals the Limit number.

- Processing Logic: See Figure 7-7 EXTRACT Logic

7.8.2.2. MERGE Phrases

COMMAND	LIMIT	SEARCH CRITERIA			FILE		DATA
		Record	Phrase	Term	I	O	
MERGE P	-	-	-	-	-	-	(Key)

Command Description: MERGE P causes all phrases, in the same record, containing a common value (or values) for the tag (or tags) specified as the merge key, to be combined into a single phrase. The merge key is entered in the Data field. This command operates within a given record; never across records, and then only upon the record elements present in the EXTRACT work area.

- Processing Logic: See Figure 7-8 MERGE P Logic

7.8.2.3. MERGE Records

COMMAND	LIMIT	SEARCH CRITERIA			FILE		DATA
		Record	Phrase	Term	I	O	
MERGE R	-	-	-	-	(Name)	(Name)	(Key + Result C & L Tag)

HIT PROCESSING
Structuring Commands
7.8.2.3.

SECRET

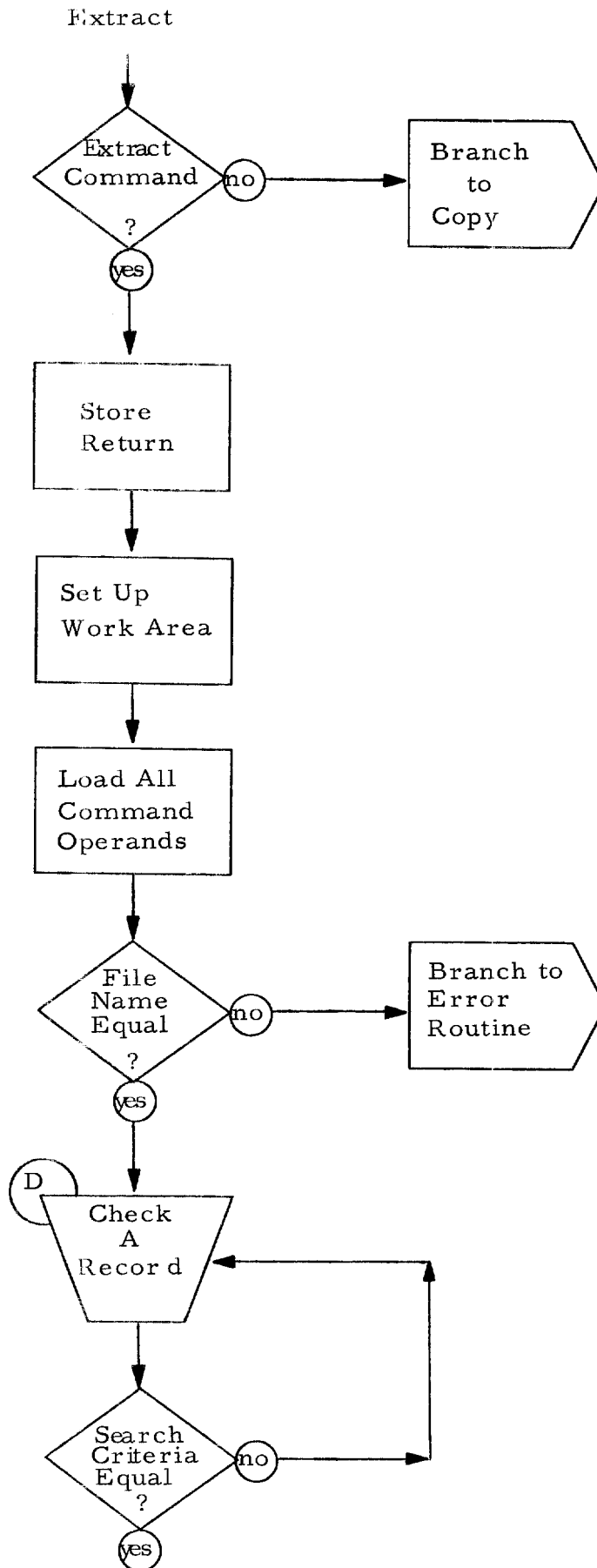


Figure 7-7

Extract

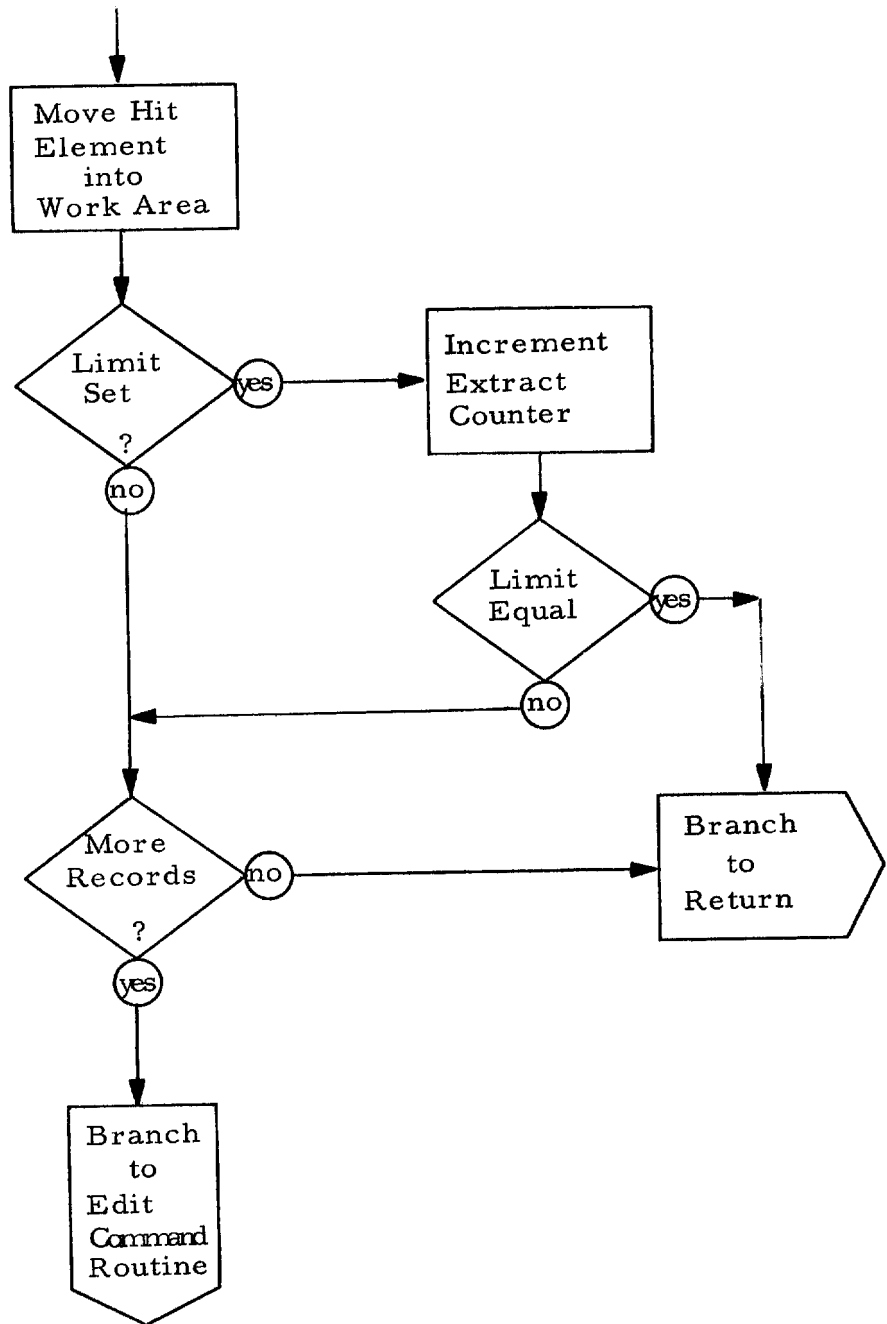
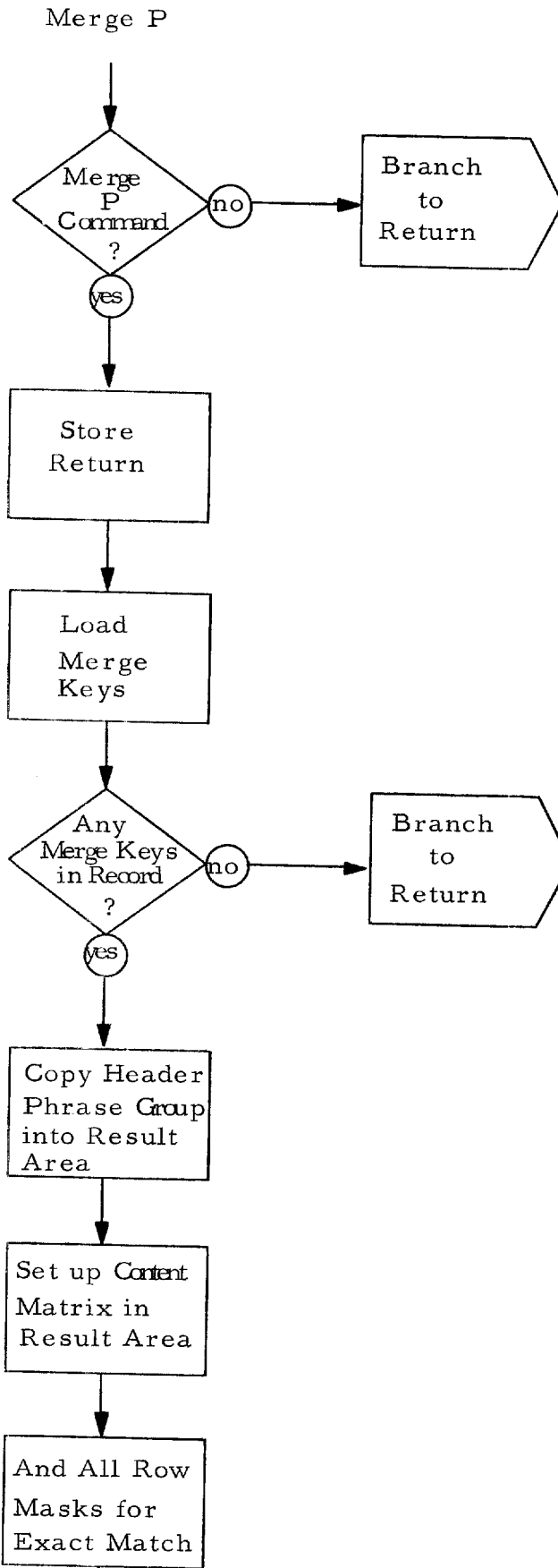


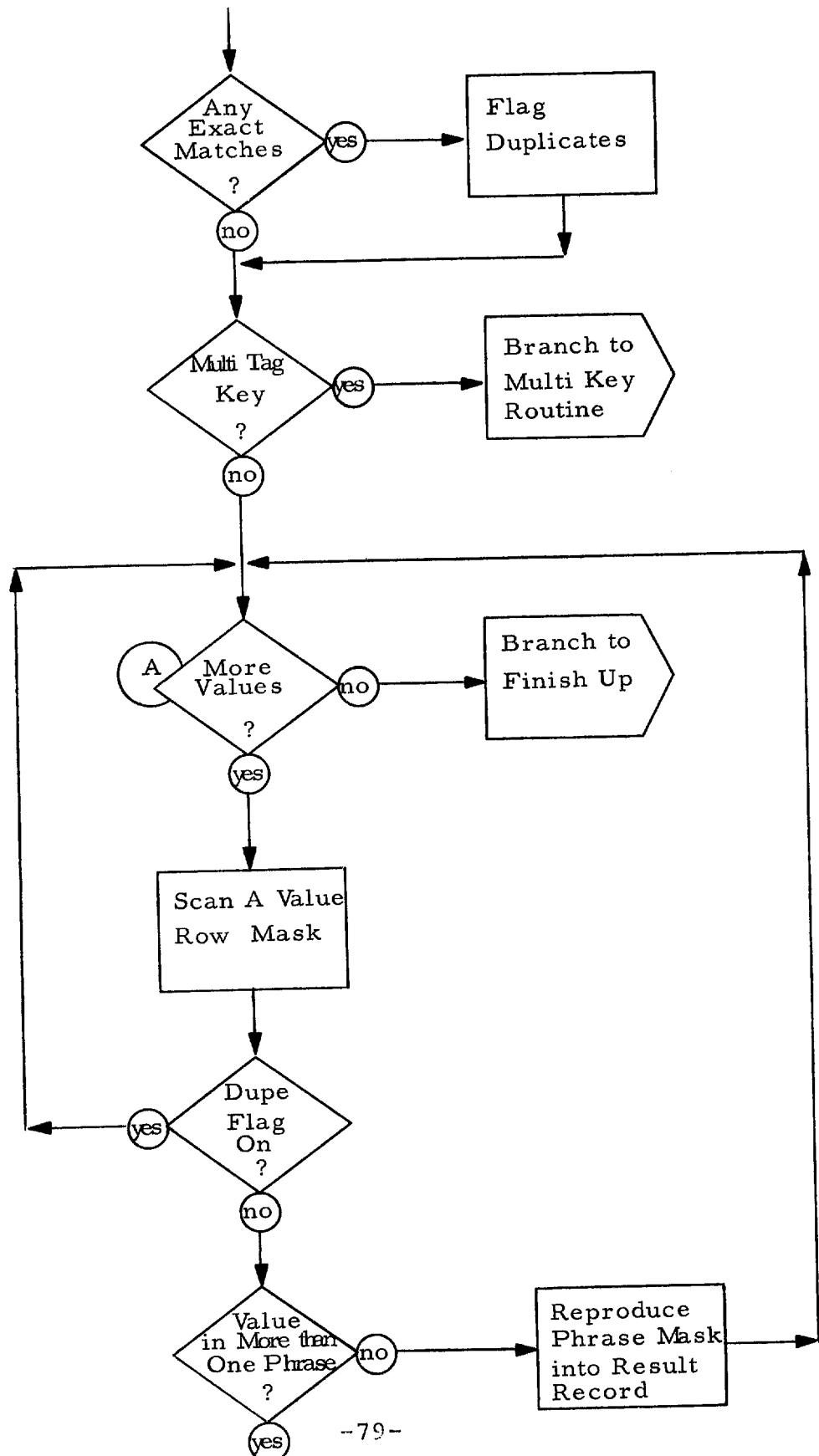
Figure 7-8



SECRET

Figure 7-8

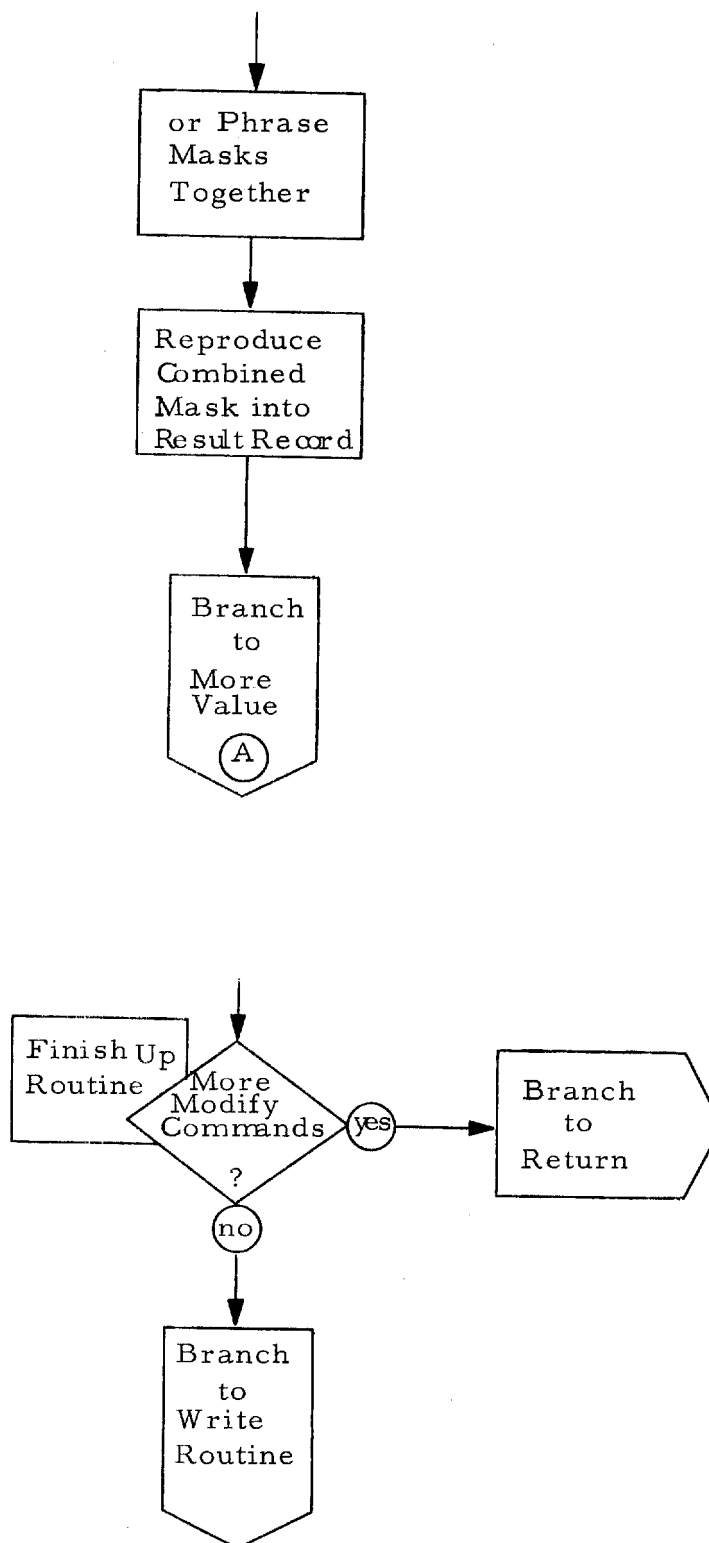
Merge P



SECRET

Figure 7-8

Merge P



Merge P

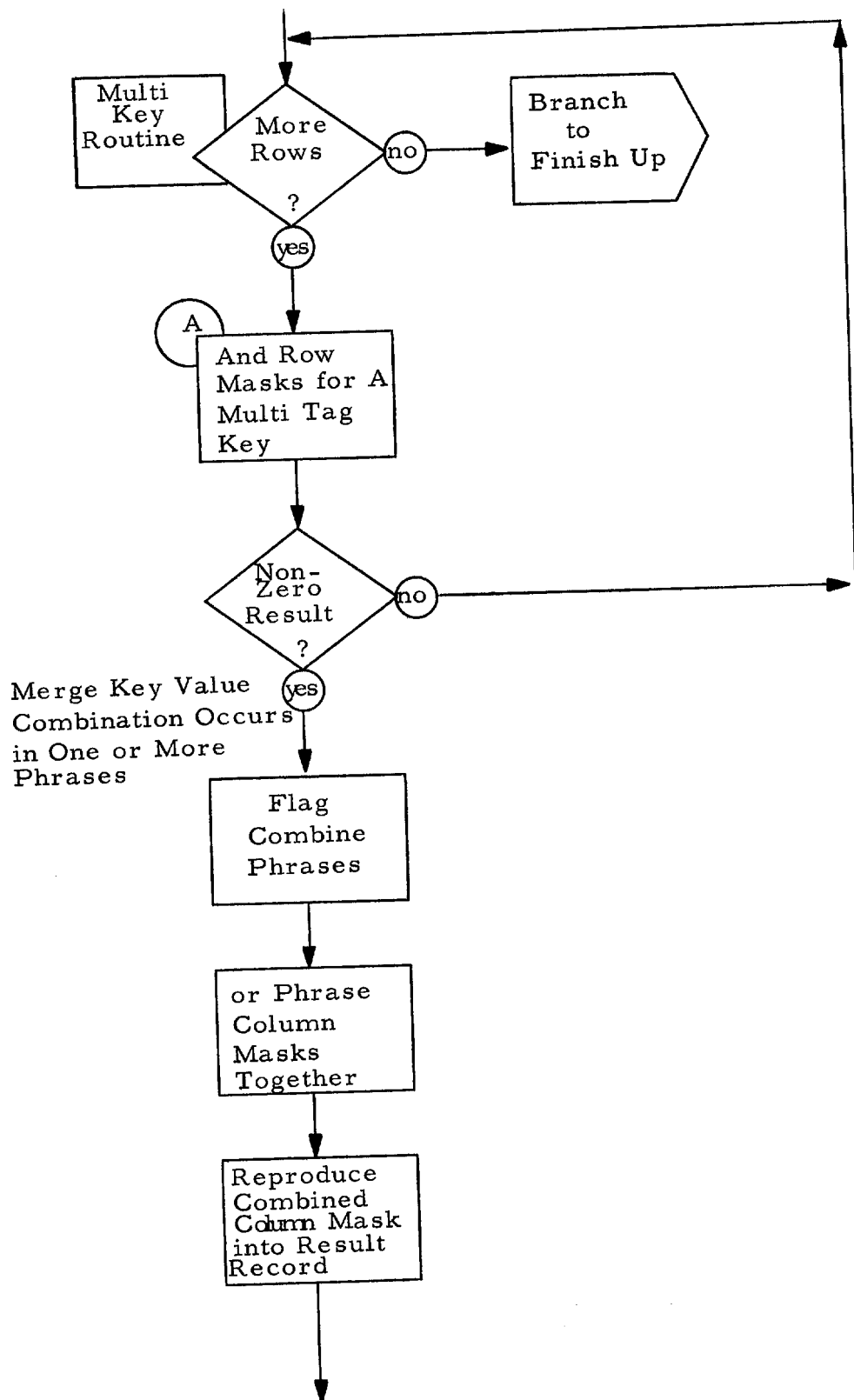
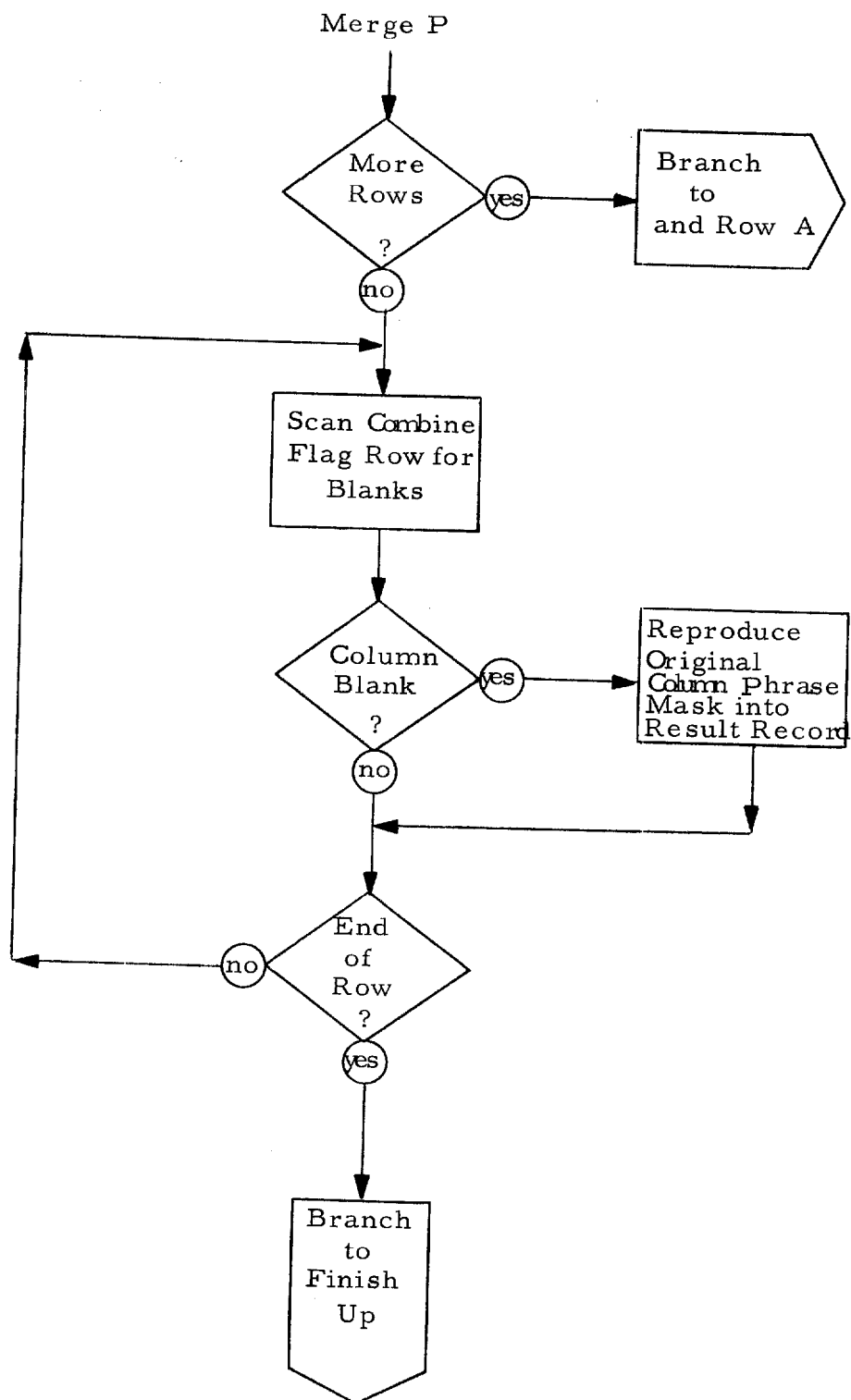


Figure 7-8



The MERGER command causes all records, in a named input file, which contain a common term (or terms) to be physically co-located in a named output file. One or more tags may be specified as the merge key in the Data field.

The MERGER command operates across records, but within a given file. Its action is similar to MERGEP but it operates on a broader boundary.

As in the MERGEP, the result of the MERGER is a record (or several records) which combine common terms within a searchable entity that were previously not coordinated.

- Processing Logic: See Figure 7-9 MERGER Logic

7.8.3. EDITING COMMANDS

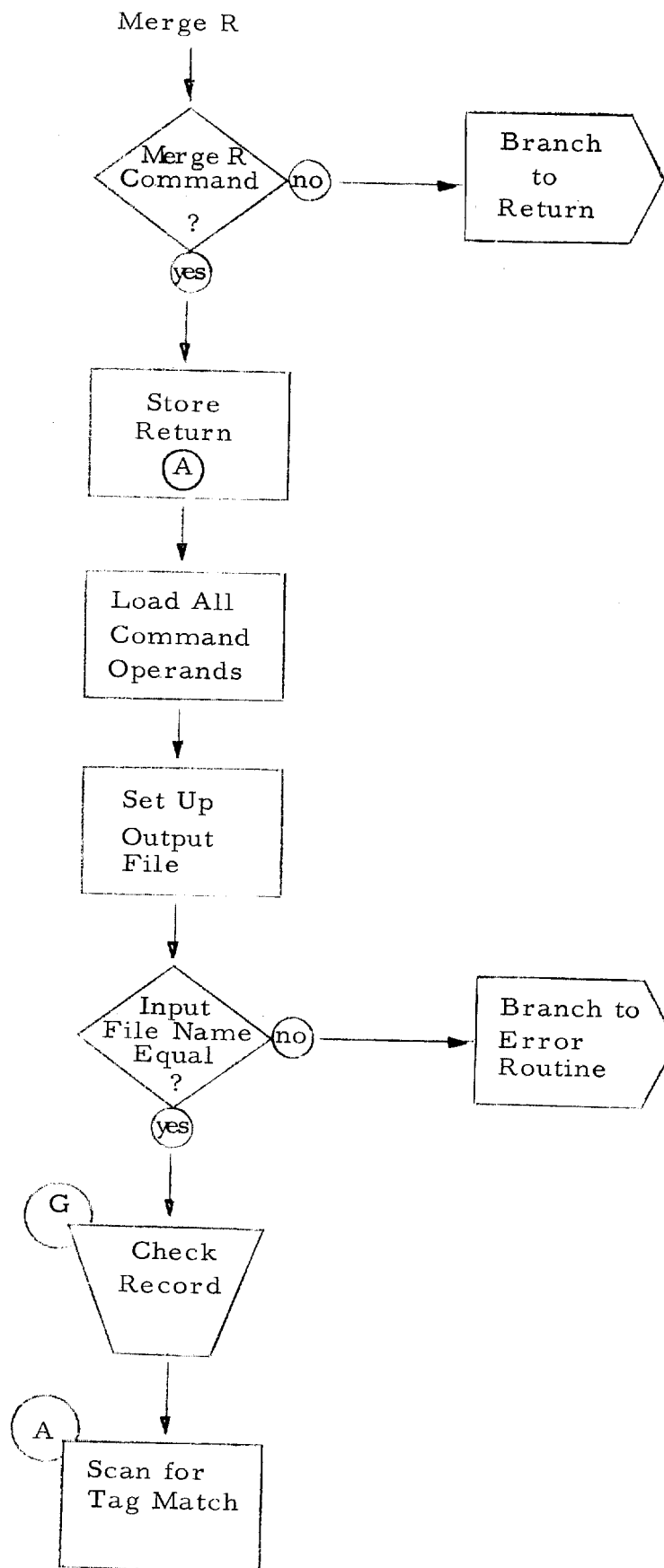
7.8.3.1. APPEND

COMMAND	LIMIT	SEARCH CRITERIA			FILE		DATA
		R	P	T	Input	Output	
APPEND	(Number)	(Ident) (Ident)	- (Ident)	- -	-	-	(Data)

Command Description: APPEND causes the data elements specified in the DATA field to be added to the phrase or record specified by the search criteria. The resultant

HIT PROCESSING
Editing Commands
7.8.3.1.

Figure 7-9



SECRET
Figure 7/9

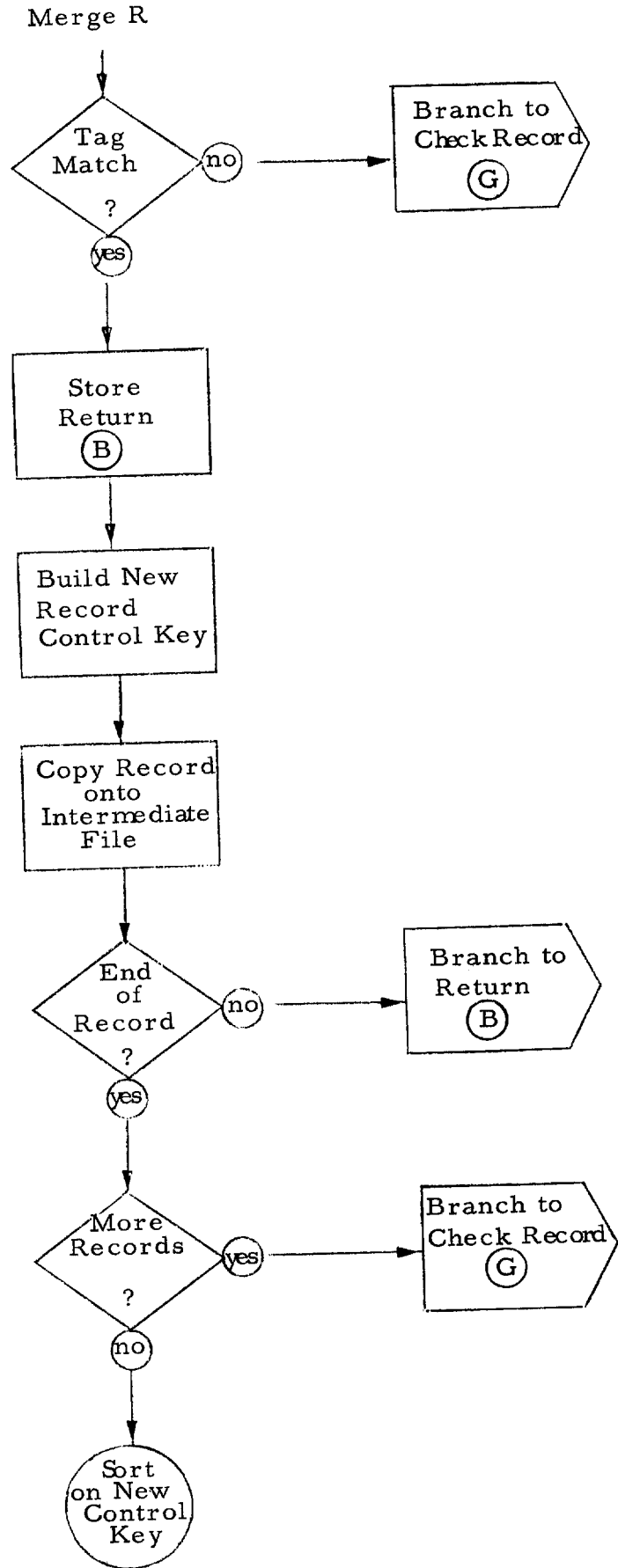
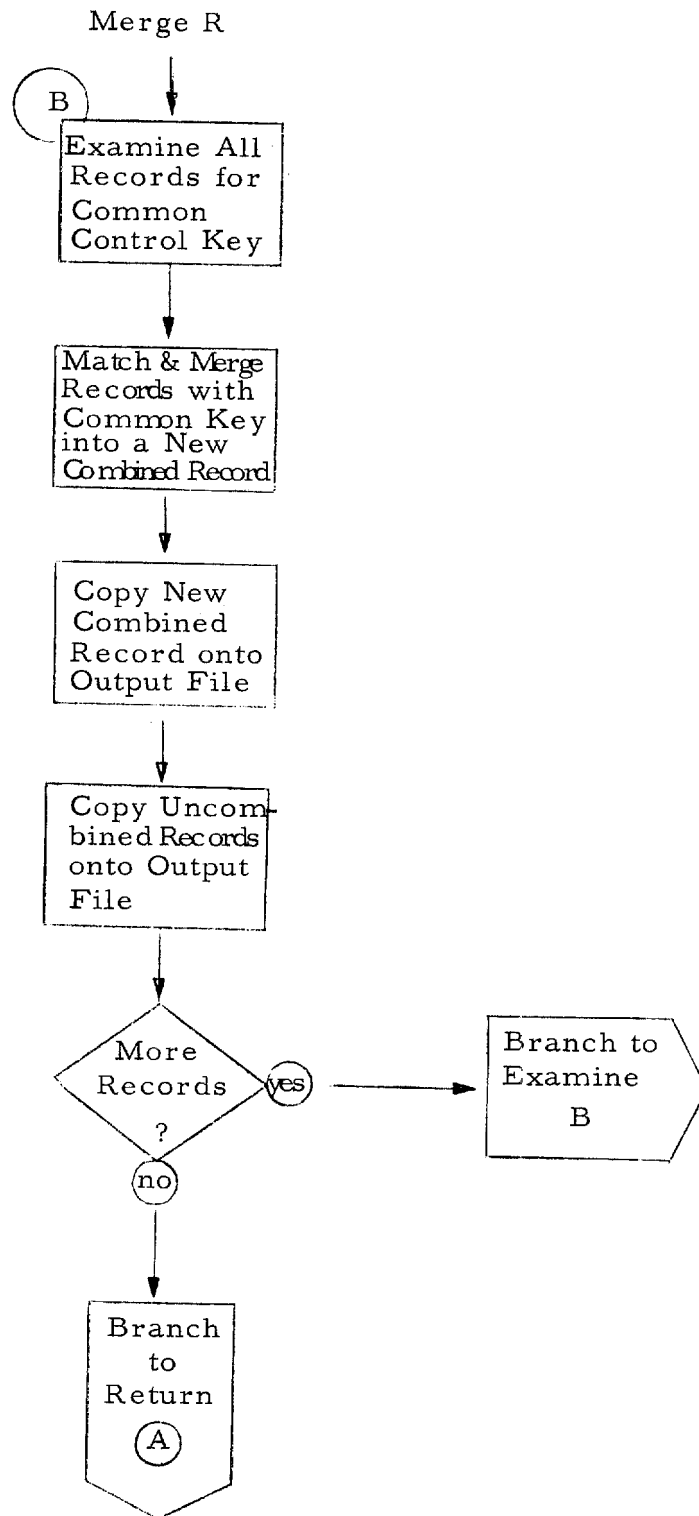


Figure 7-9



SECRET

record is located in a system generated work area that was created for this purpose by the EXTRACT command. No explicit reference need be made to the work area by the user. Its presence is assumed.

The data elements that may be specified include terms, tally results, accumulation results, computational results and defined terms. The command operates only upon record elements that have been previously extracted.

If hit processing is under FFT control (see Appendix 7.A.3. for a complete description of FFT Control), the data being supplied and the action of appending the data must obey the rules and conventions established by the File Format Table of the specified input file. Failure to do so will result in error conditions. The fundamental concern of FFT control is to insure that files produced by hit processing commands are still defined by a File Format Table. Thus enabling these files to be subsequently processed by the CHIVE system programs.

- Processing Logic: See Figure 7-10 APPEND Logic

7.8.3.2. DELETE

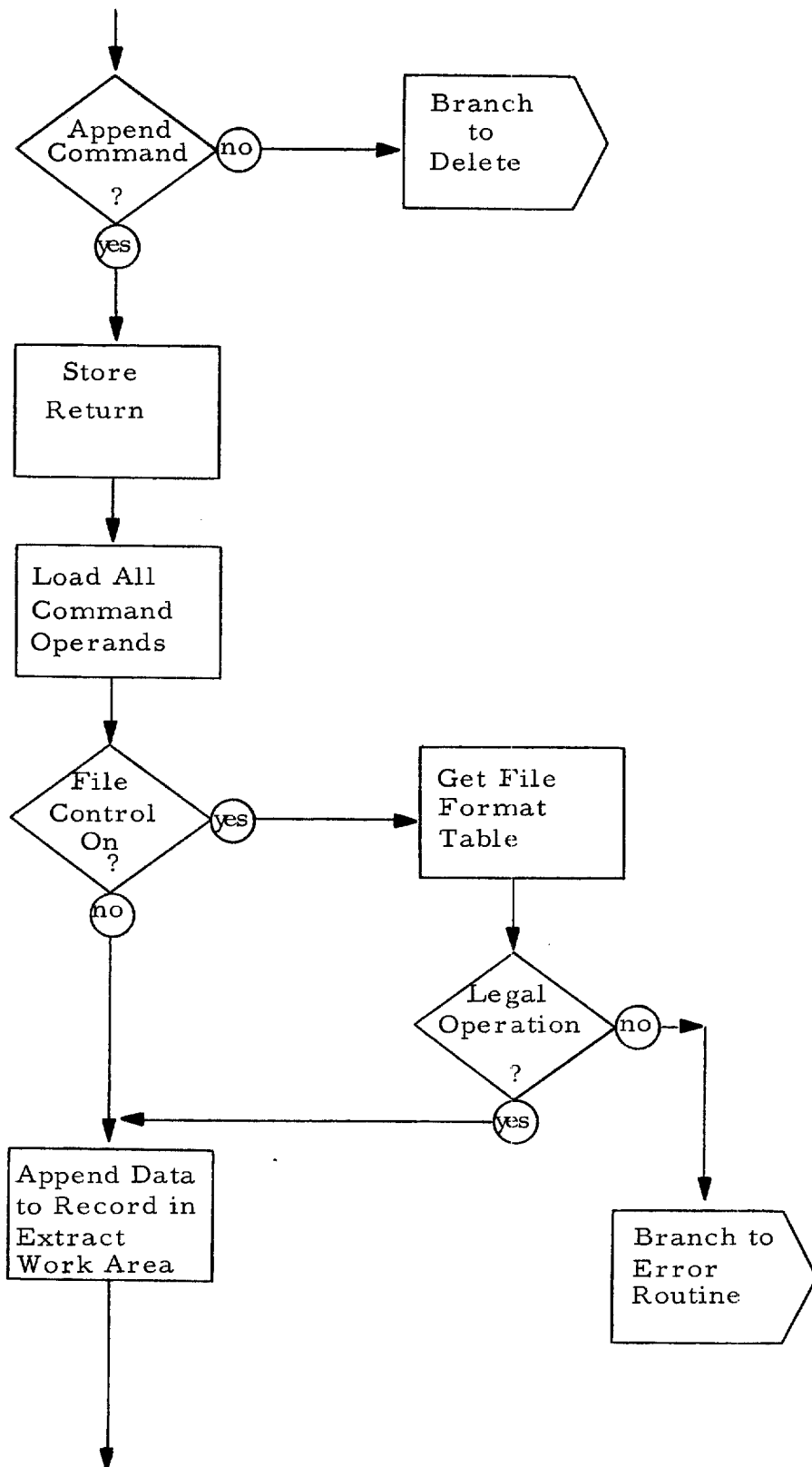
COMMAND	LIMIT	SEARCH CRITERIA			FILE		DATA
		Record	Phrase	Term	I	O	
DELETE	(Number)	(Ident)	-	-	-	-	-
		(Ident)	(Ident)	(Ident)			
		(Ident)	(Ident)	(Ident)			

HIT PROCESSING
Editing Commands

SECRET

Figure 7-10

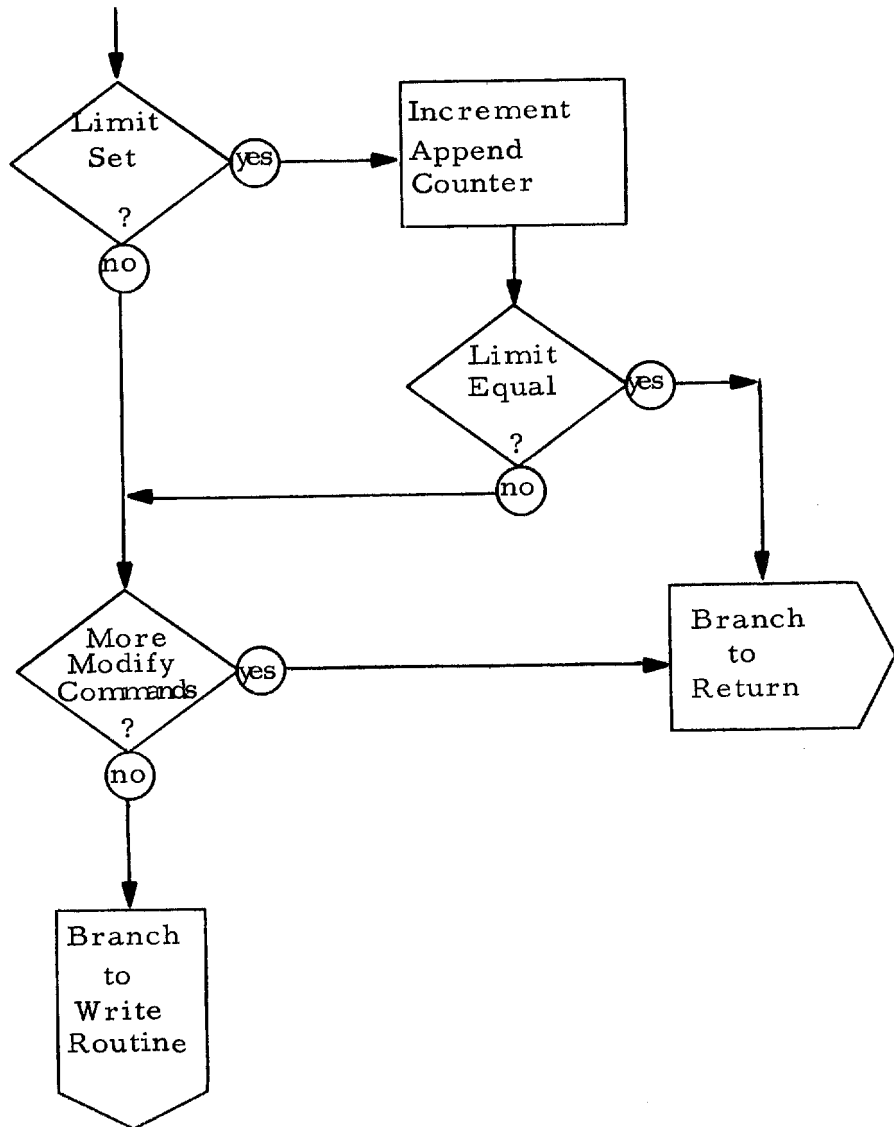
Append



SECRET

Figure 7-10

Append



Command Description: DELETE causes the terms, phrases, or records specified in the search criteria to be deleted from the record located in the work area. As in the APPEND command, the record elements being operated upon were introduced into the work area by the EXTRACT command.

If FFT control is specified in the job specification, the delete operation must obey all rules established by the File Format Table of the input file. Failure to do so will result in an error condition.

- Processing Logic: See Figure 7-11 DELETE Logic

7.8.3.3. REPLACE

COMMAND	LIMIT	SEARCH CRITERIA			FILE		DATA
		RECORD	PHRASE	TERM	I	O	
REPLACE	(Number)	(Ident) (Ident)	(Ident) (Ident)	(Ident)	-	-	(Data)

Command Description: REPLACE causes the data elements specified in the DATA field to be substituted for the terms or phrases specified in the search criteria, and the result located in a system generated work area. As in APPEND and DELETE, the record elements being operated upon were placed in the work area by the EXTRACT command.

- Processing Logic: See Figure 7-12 REPLACE Logic

HIT PROCESSING
Editing Commands
7.8.3.3.

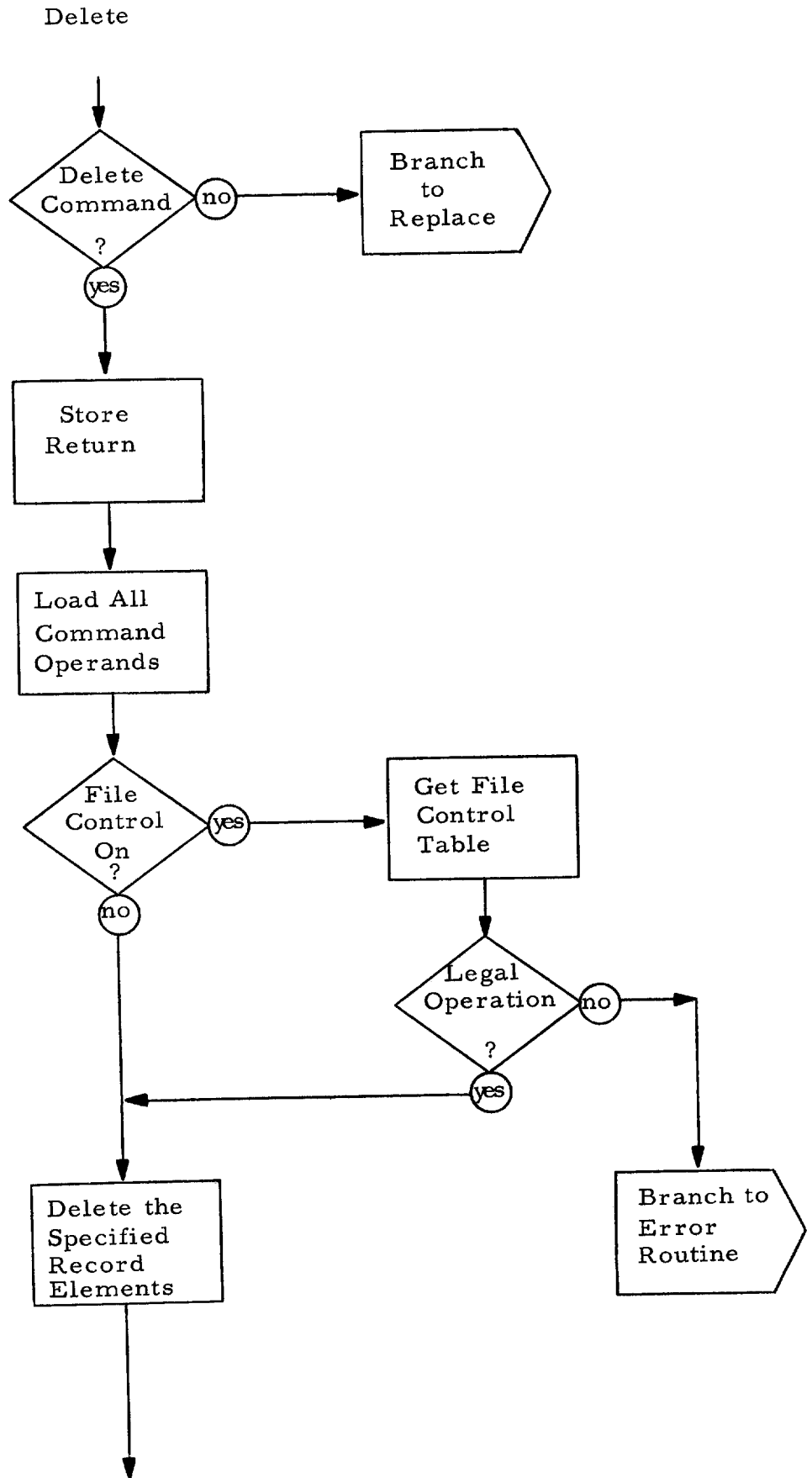
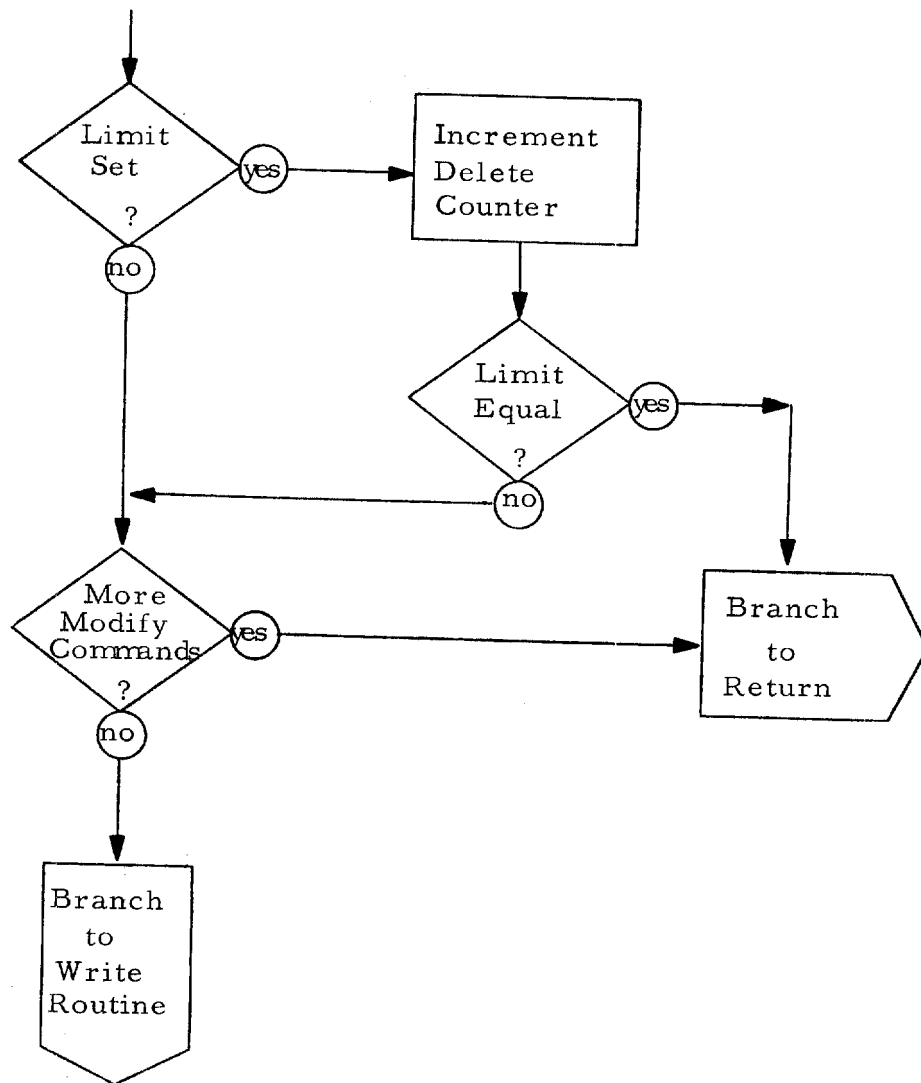


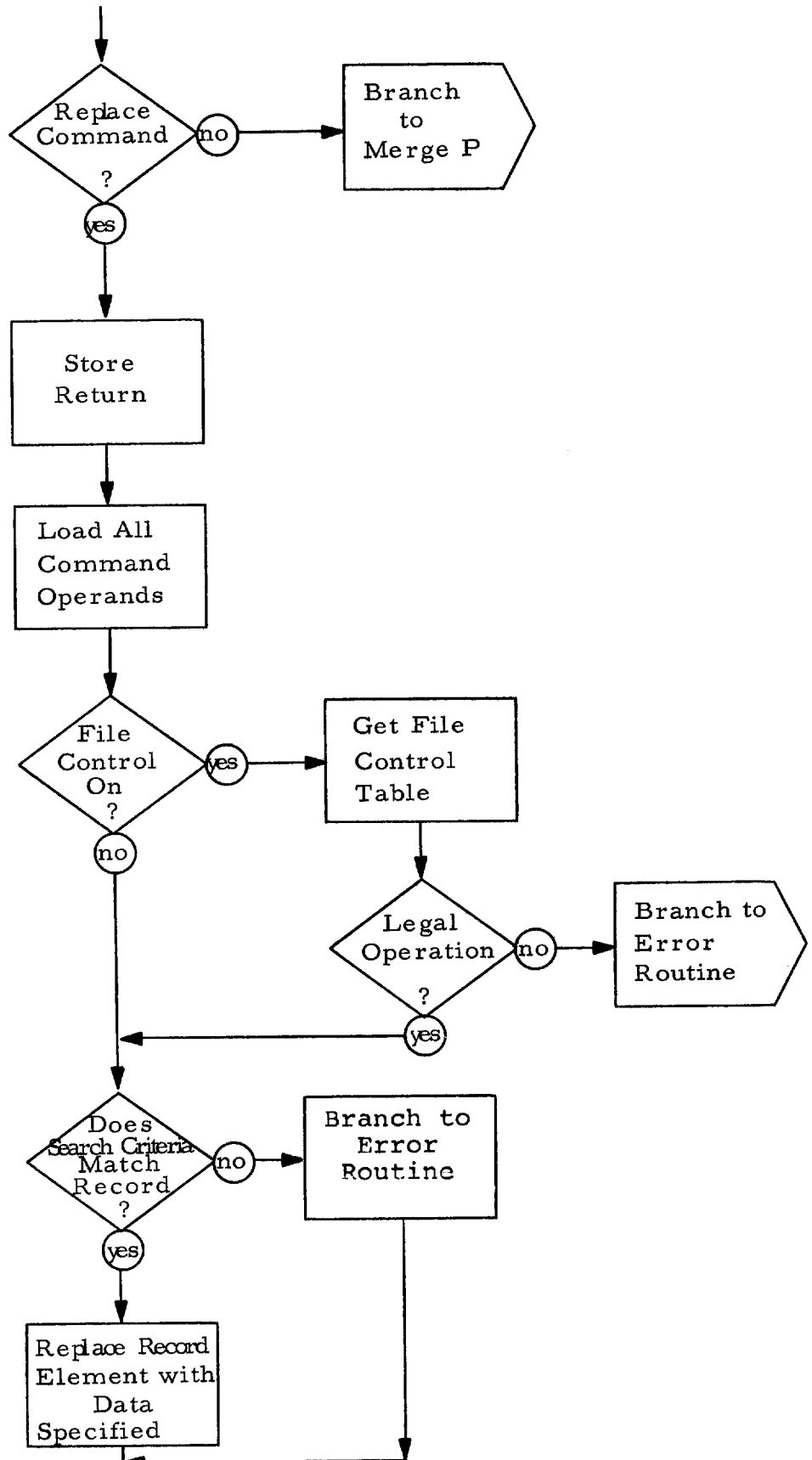
Figure 7-11

Delete



SECRET

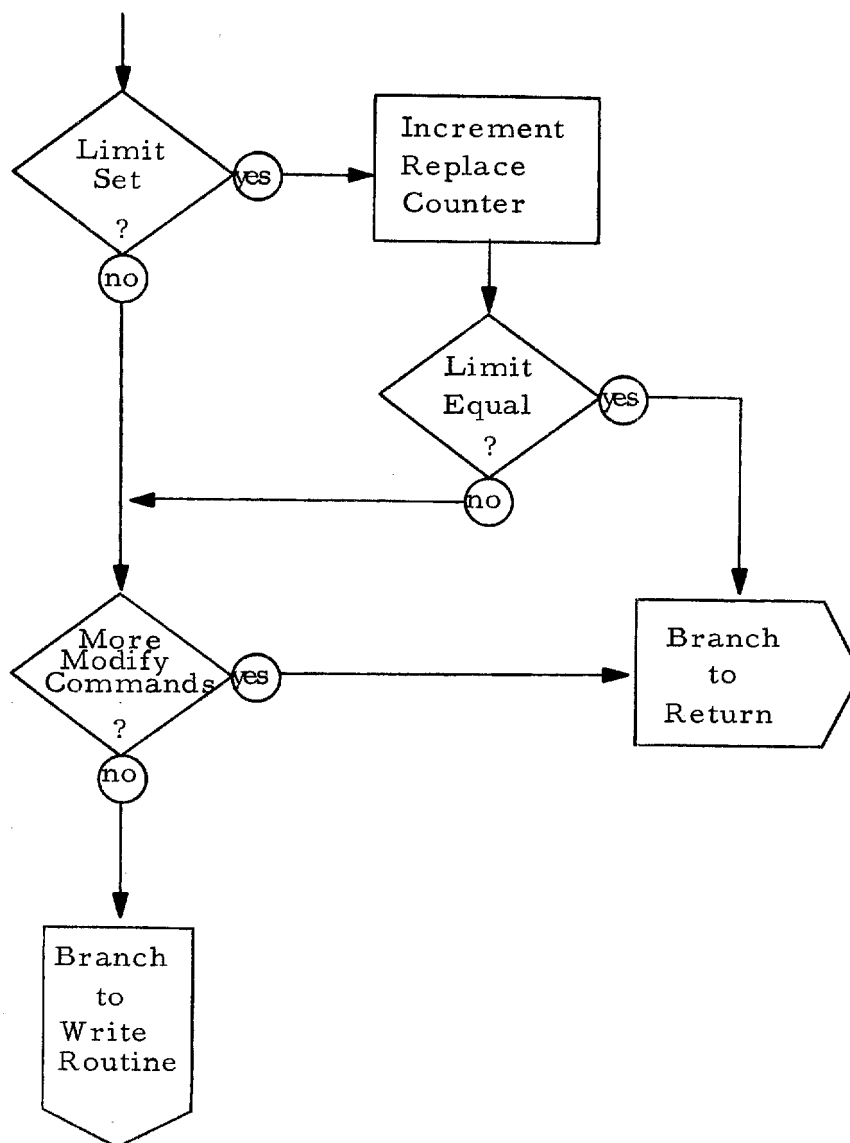
Replace



SECRET

Figure 7-12

Replace



SECRET

7.8.3.4. DEFINE n

COMMAND	LIMIT	SEARCH CRITERIA			FILE		DATA
		Record	Phrase	Term	I	O	
DEFINE n	-	(Ident) (Ident)	(Ident) (Ident)	- (Ident)	(Name)	-	-

DEFINE n uniquely identifies a term or phrase in a record according to the specified search criteria. The DEFINE command is used in conjunction with other hit processing commands. For example, when used in conjunction with the arithmetic commands it allows the user to uniquely "define" the terms that enter into a computation. In addition, it insures control via test conditions that the required value will not occur more than once in the argument.

7.8.4. THE ARITHMETIC COMMANDS

7.8.4.1. TALLY n

COMMAND	LIMIT	SEARCH CRITERIA			FILE		DATA
		RECORD	PHRASE	TERM	I	O	
TALLY n	number	(Ident) (Ident) (Ident)	- (Ident) (Ident)	- (Ident) (Ident)	(Name)	-	-

HIT PROCESSING
Arithmetic Commands
7.8.4.1.

SECRET

SECRET

Command Description: TALLY n causes a numbered tally counter (N) to be incremented by one each time that the specified search criteria are satisfied, for either a Record, Phrase, or Term specification. It operates upon the named Input File and stores the result in a symbolic location named TALLY n. This result may be referenced by name and its value used as an operand by any command group including the other arithmetic commands.

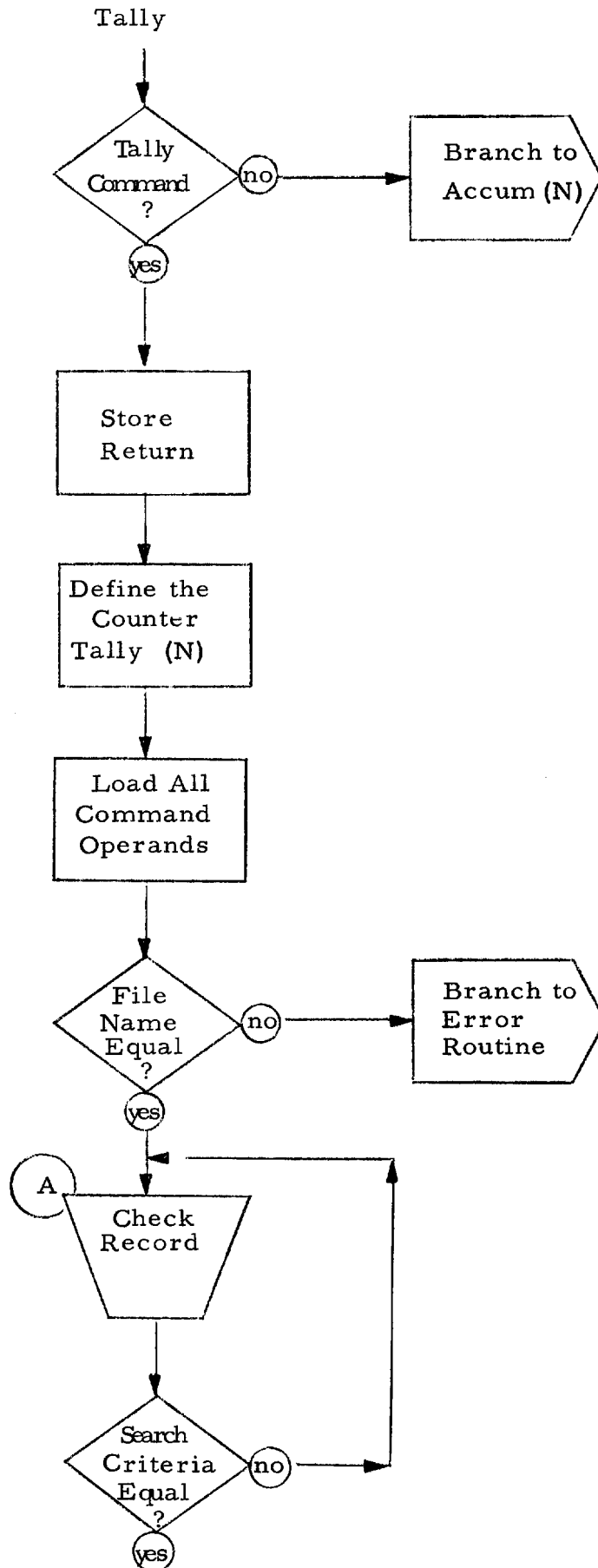
The tally counter is reset to zero for each new record brought into the job when Phrase or Term search criteria are specified. However, specification of Tally Records provides a running count of all "hit" records and the tally counter is not reset until the end of the retrieval transaction.

Optional inclusion of a value in the Limit field will cause the tally operation to cease when the count equals the specified value.

- Processing Logic: See Figure 7-13 TALLY n Logic

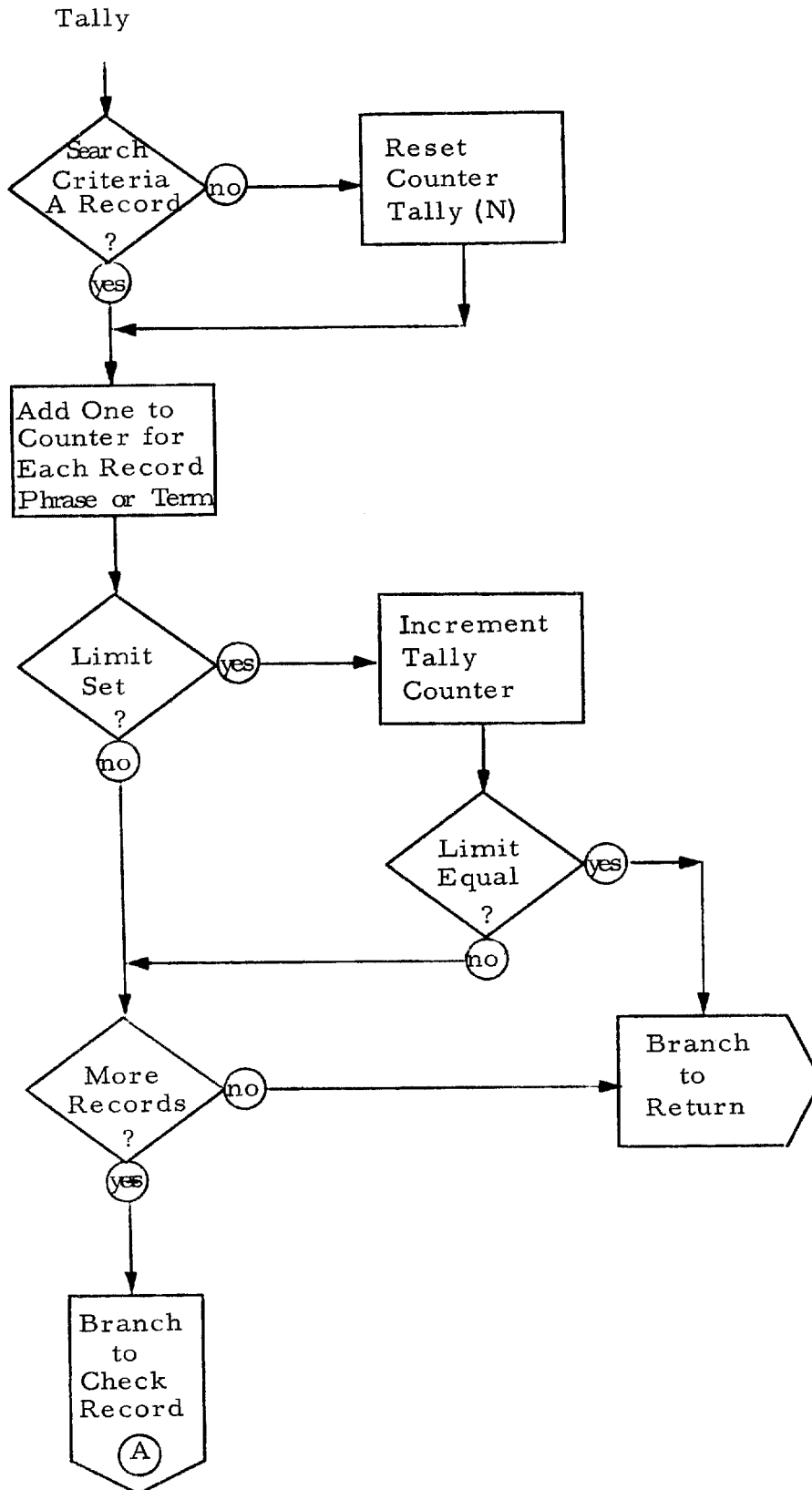
HIT PROCESSING
Arithmetic Commands
7.8.4.1.

SECRET



SECRET

Figure 7-13



7.8.4.2. ACCUMULATE

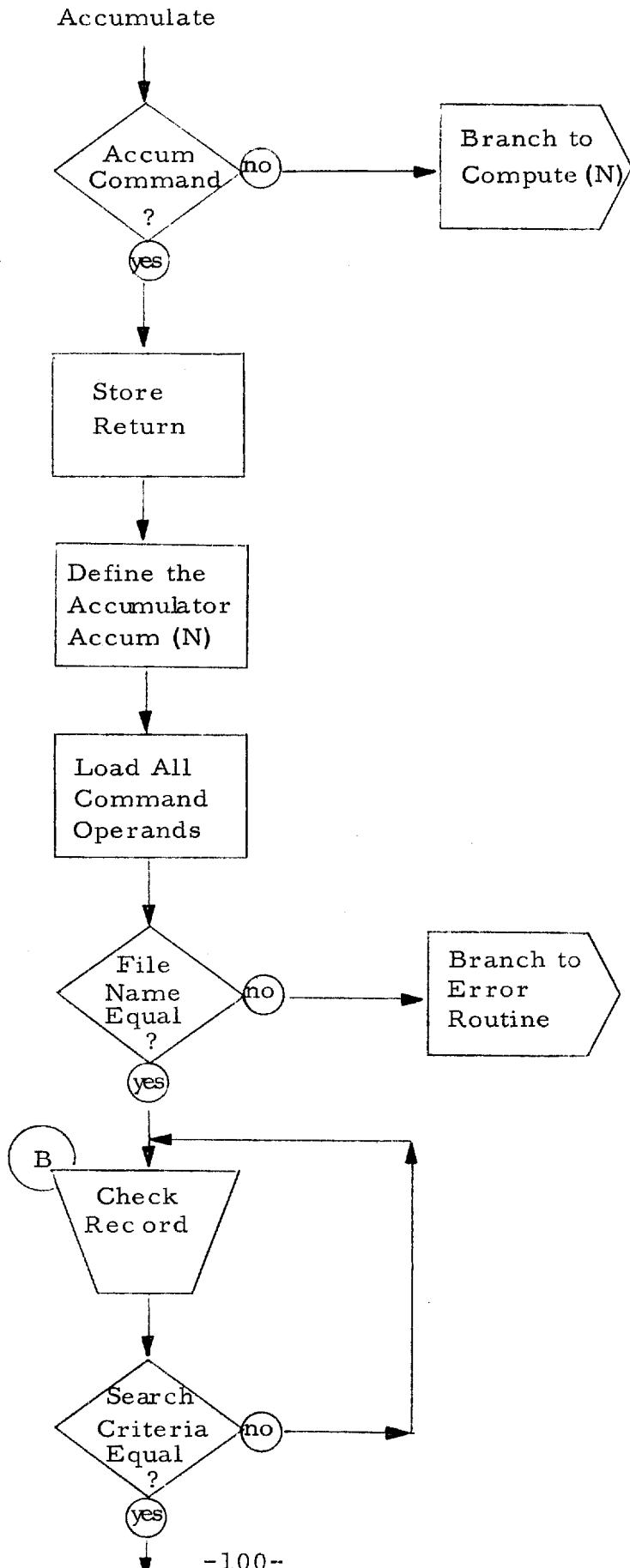
COMMAND	LIMIT	SEARCH CRITERIA			FILE	DATA
		Record	Phrase	Term I		Ω
		-	-	-		
ACCUM n	(Number)	(Ident)	(Ident)	(Ident)	(Name)	- -

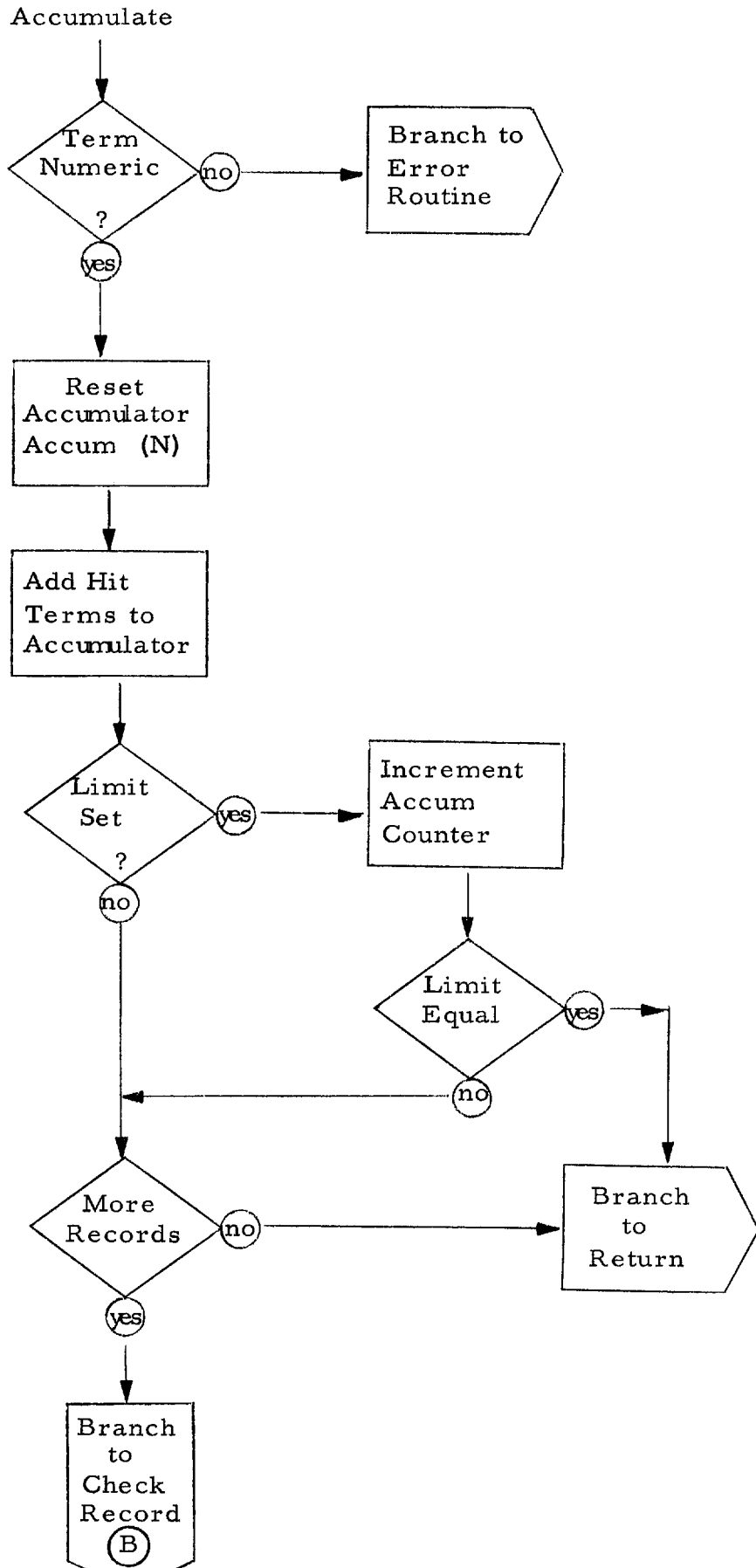
Command Description: ACCUM n causes the term operand(s) specified in the Search Criteria to be arithmetically summed, and the results are stored in a symbolic location named ACCUM n. The command operates upon the named input file. If a value is inserted in the Limit field, the accumulation operation will cease when the number of operations performed reaches the specified Limit value.

All search criteria term operands must be numeric. Detection of a non-numeric character in any term operand will create an error condition.

- Processing Logic: See Figure 7-14 ACCUM n Logic

HIT PROCESSING
Arithmetic Commands
7.8.4.2.





7.8.4.3. COMPUTE

COMMAND	LIMIT	SEARCH CRITERIA			FILE	DATA
		Record	Phrase	Term	I	O
COMPUTE n	(Number)	-	-	-	(Name)	(Expression)

Command Description: COMPUTE n causes the computational expression inserted in the Data field to be executed and the result stored in the symbolic location named COMPUTE n. This expression is stated in the following form:

Data Field

Operand	OP	Operand . . .	OP	Operand
---------	----	---------------	----	---------

Where "OP" can be any of the following:

- * Multiply
- / Divide
- + Add
- Subtract

Operands may be either numeric valued tags, TALLY or ACCUM results, or numeric constants. The symbolic location Compute n is reset to zero for every record encountered in the job.

- Processing Logic: See Figure 7-15 COMPUTE n Logic

HIT PROCESSING
 Arithmetic Commands
 7.8.4.3.

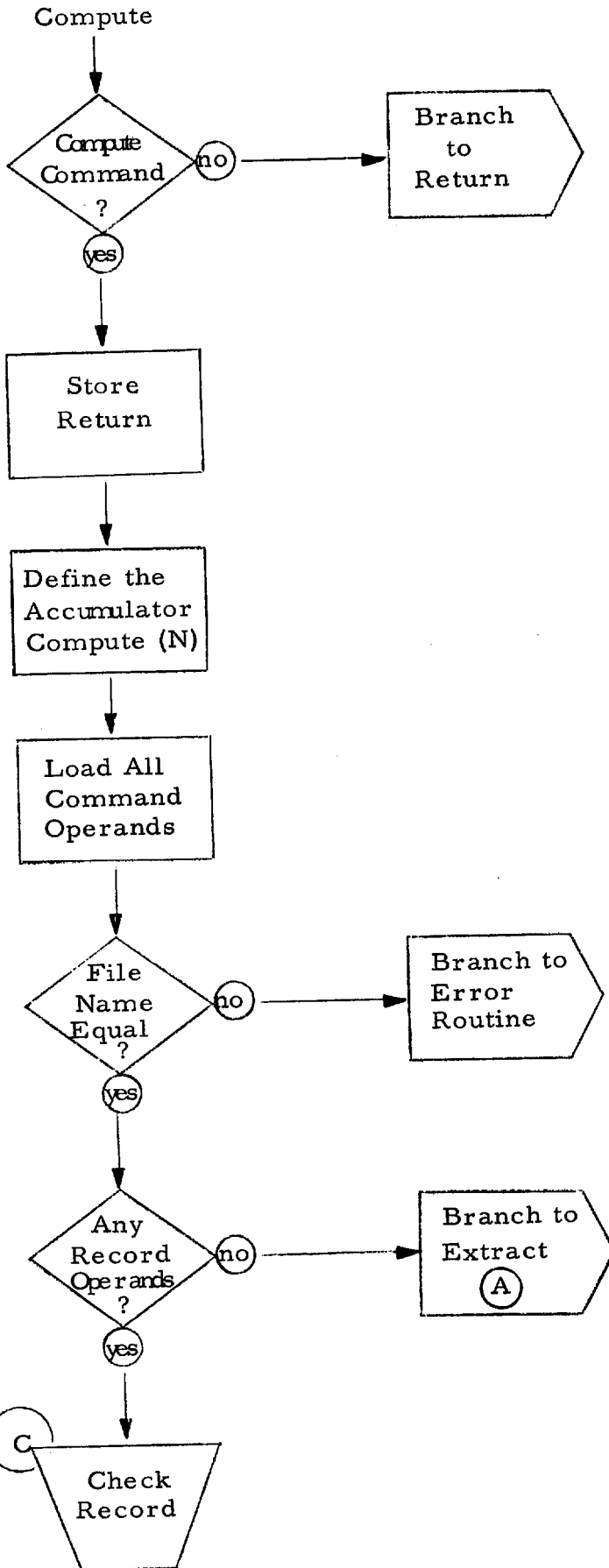


Figure 7-15

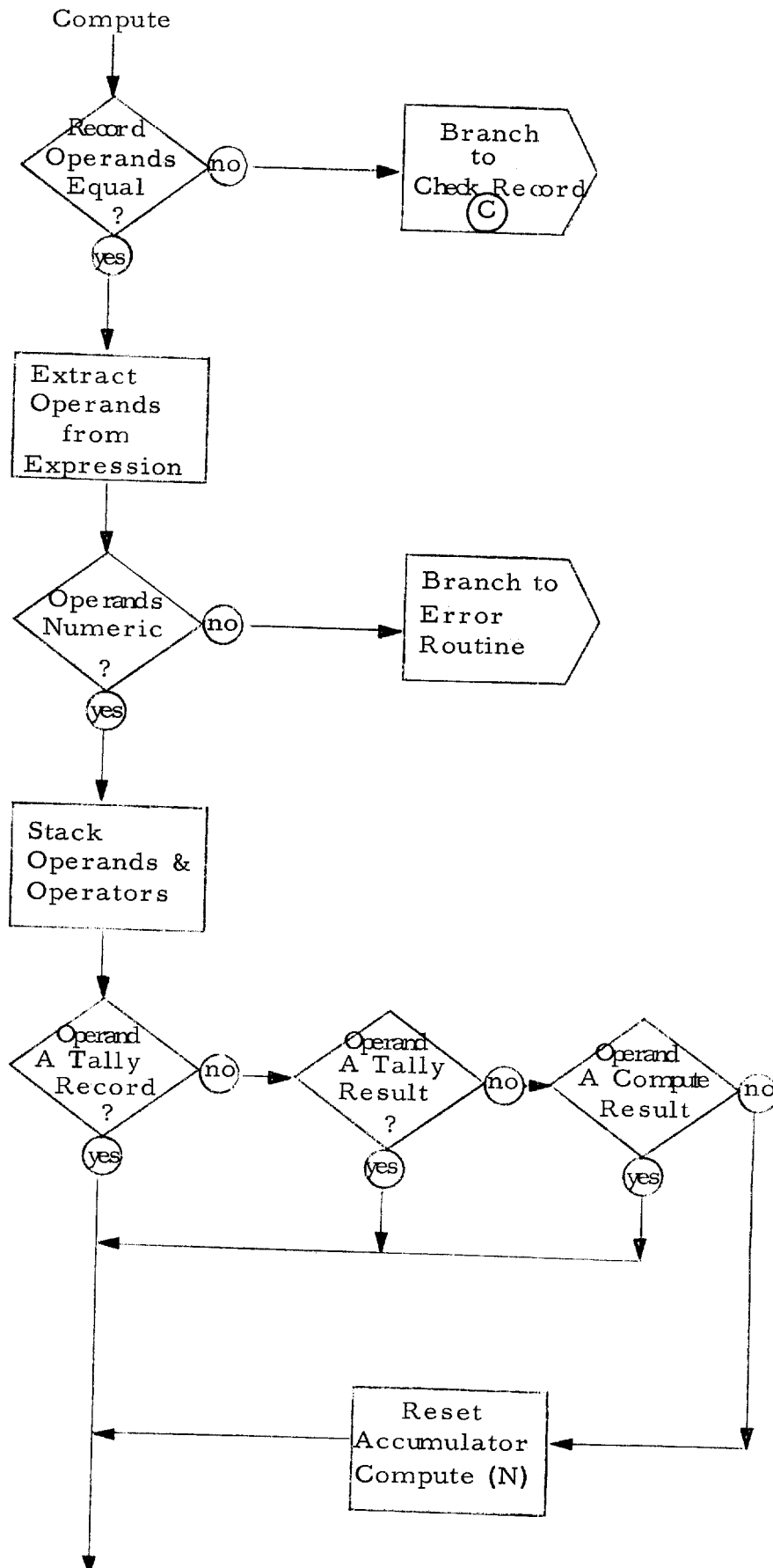
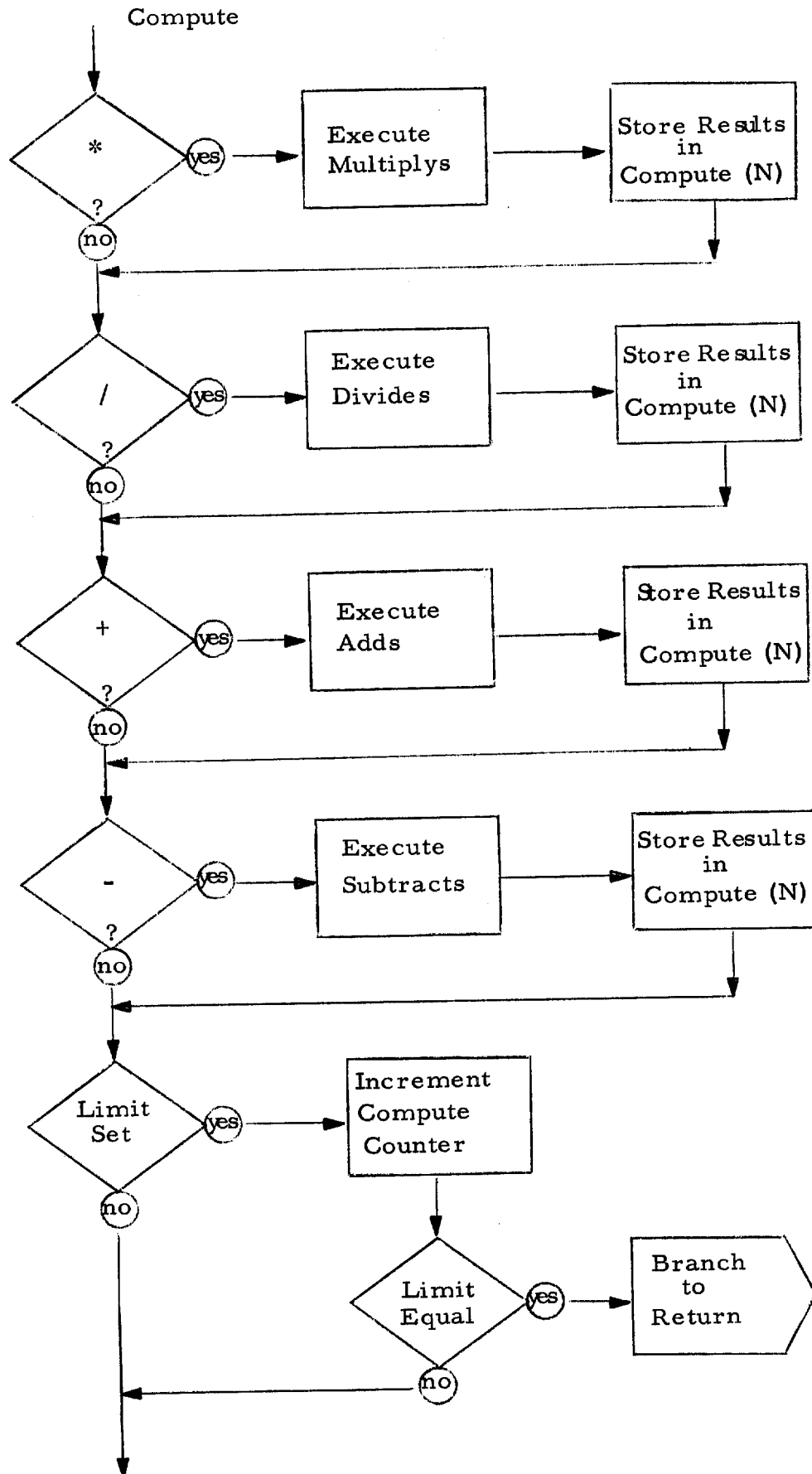
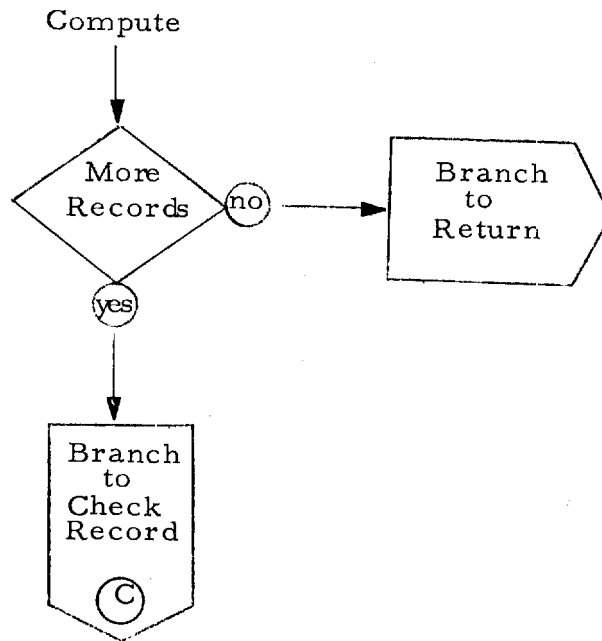


Figure 7-15



SECRET

Figure 7-15



SECRET

7.8.5. WRITING COMMANDS

7.8.5.1. WRITE

COMMAND	LIMIT	SEARCH Record	CRITERIA Phrase	FILE Term	DATA Input	Output
WRITEP WRITER	(Number)	-	-	-	-	(Name) -

Command Description: WRITE occurs in two modes, WRITEP and WRITER. However, the form of the commands and the structure of the operands is identical. Basically, WRITE causes the record elements contained in the EXTRACT work area to be written onto an output file whose name is specified by the Output File field. When WRITER is specified, the entire contents of the work area are written out as a single record. When WRITEP is specified, each phrase in the work area is written out as an individual record.

The WRITE command operates upon the contents of the work area only and acts as the exit point of a group of hit processing commands. It is always paired with at least one EXTRACT command.

The Limit operand causes the WRITE operation to terminate when the number of records written equals the Limit value.

- Processing Logic: See Figure 7-16 WRITE Logic

HIT PROCESSING
Writing Commands
7.8.5.1.

Figure 7-16

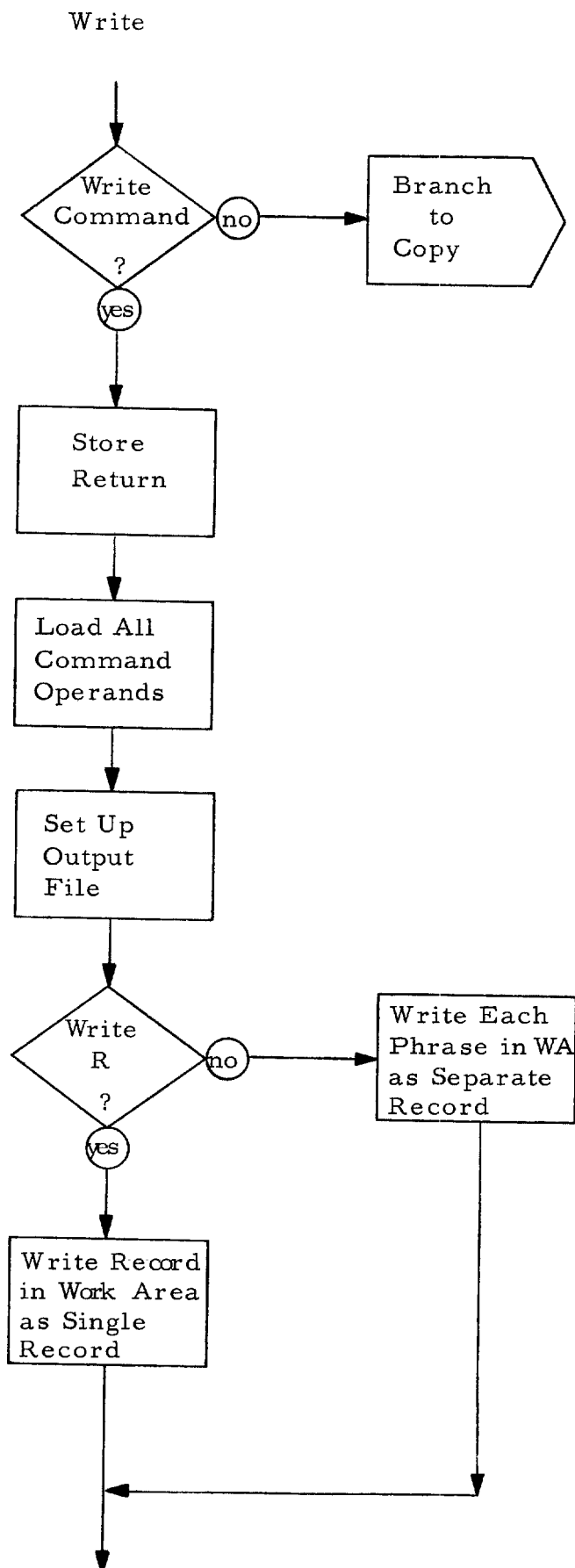
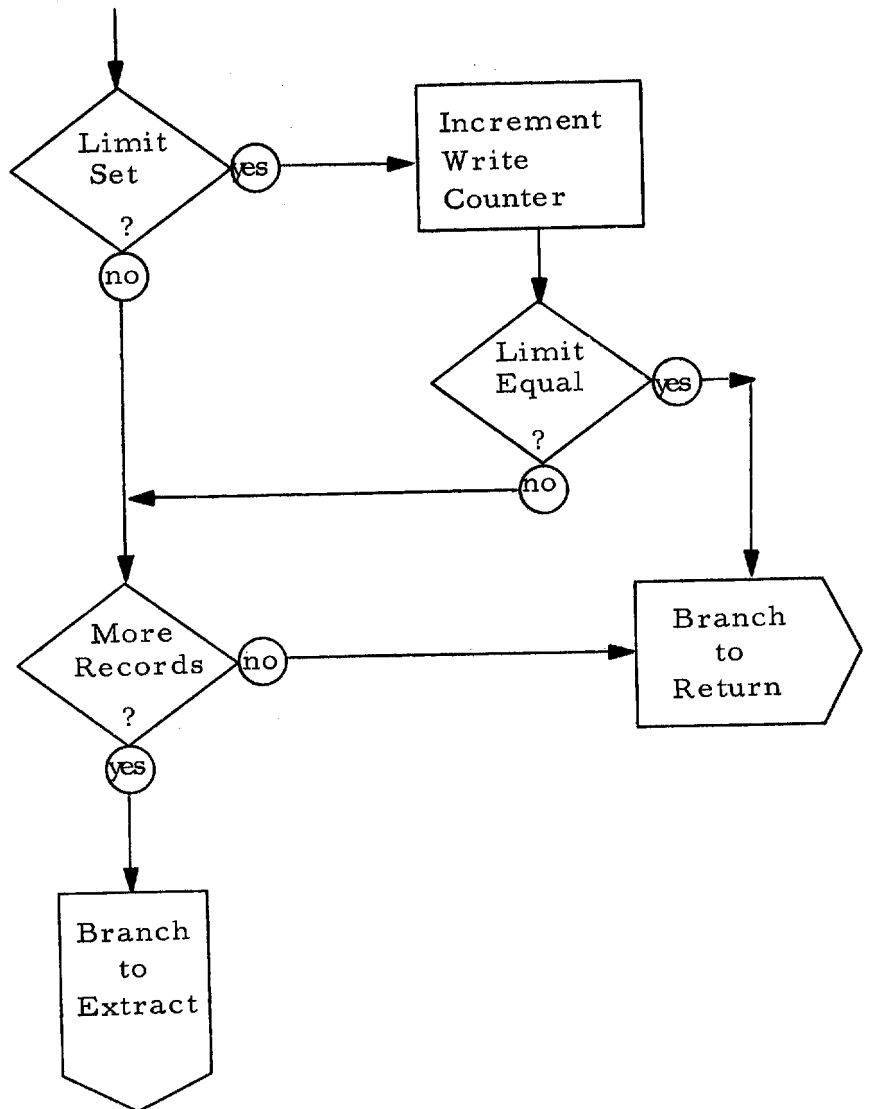


Figure 7-16

Write



SECRET

7.8.5.2. COPY

COMMAND	LIMIT	SEARCH CRITERIA			FILE	DATA
		Record	Phrase	Term I	O	
COPY	(Number)	(Ident)	-	-	(Name)	(Name) -
		(Ident)	(Ident)	-	(Name)	(Name) -
		(Ident)	(Ident)	(Ident)		

Command Description: COPY causes the record elements specified in the Search Criteria field to be written onto an output file whose name is specified by the Output File operand.

The command operates upon records, terms, or phrases, contained in the named input file. If a value is inserted in the Limit field, the COPY operation will terminate when the number of records copied equals the specified Limit.

When the COPY command is used, there can be no additional manipulation of the record elements specified.

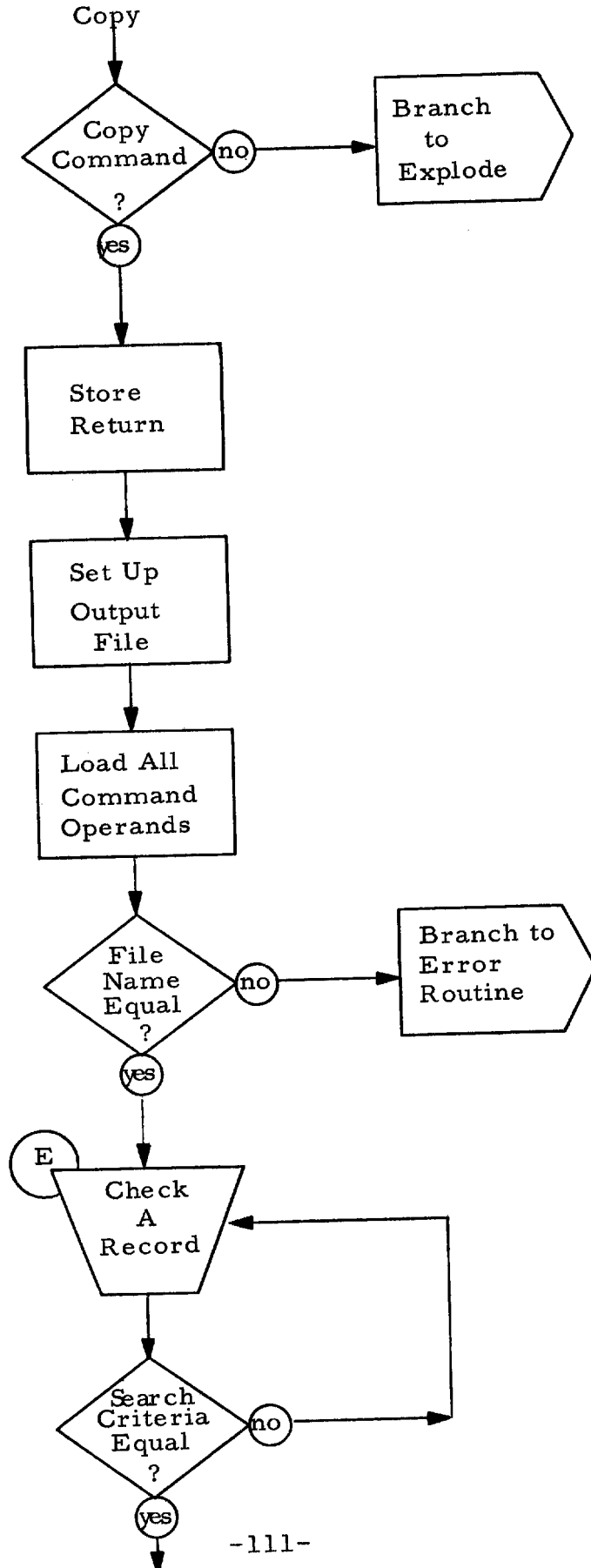
- Processing Logic: See Figure 7-17 COPY Logic

7.8.5.3. EXPLODE

COMMAND	LIMIT	SEARCH CRITERIA			FILE	DATA
		Record	Phrase	Term I	O	
EXPLODE	(Number)	(Ident)	(Ident)	-	(Name)	(Name) -
		(Ident)	(Ident)	(Ident)		

HIT PROCESSING
Writing Commands
7.8.5.3.

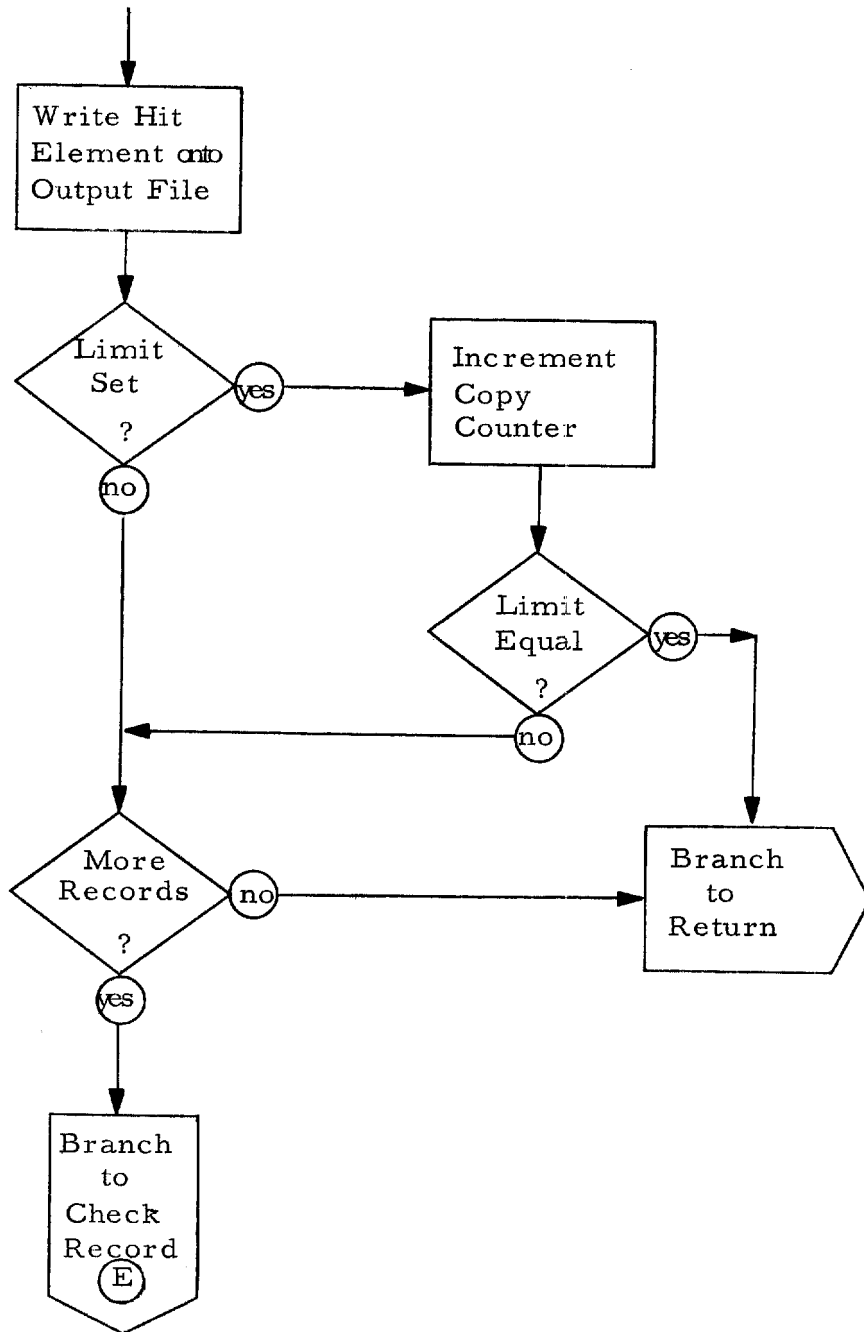
SECRET



SECRET

Figure 7-17

Copy



SECRET

Command Description: EXPLODE causes the generation of a separate, distinct record for each term or phrase specified in the Search Criteria. It operates upon the input file so designated and writes the resultant records onto an output file whose name is specified by the Output File Operand.

EXPLODE is a variant of COPY. There is one important difference, it allows terms and phrases to become separate records. As a result these new files can be regrouped for meaningful displays. It should be noted that the use of EXPLODE causes the record relationships to be lost.

- Processing Logic: See Figure 7-18 EXPLODE Logic

7.8.5.4. PRINT

COMMAND	LIMIT	SEARCH CRITERIA			FILE		DATA
		Record	Phrase	Term	I	O	
PRINT	(Number)	-	-	-	(Name)	(Name)	(Parameters)

The PRINT command causes the Input File records to be routed to and processed by the report programs. Sort sequence, format, and other control parameters are indicated in the Data field.

-Processing Logic: See Figure 7-19 PRINT Logic

HIT PROCESSING
Writing Commands
7.8.5.4.

Figure 7-18

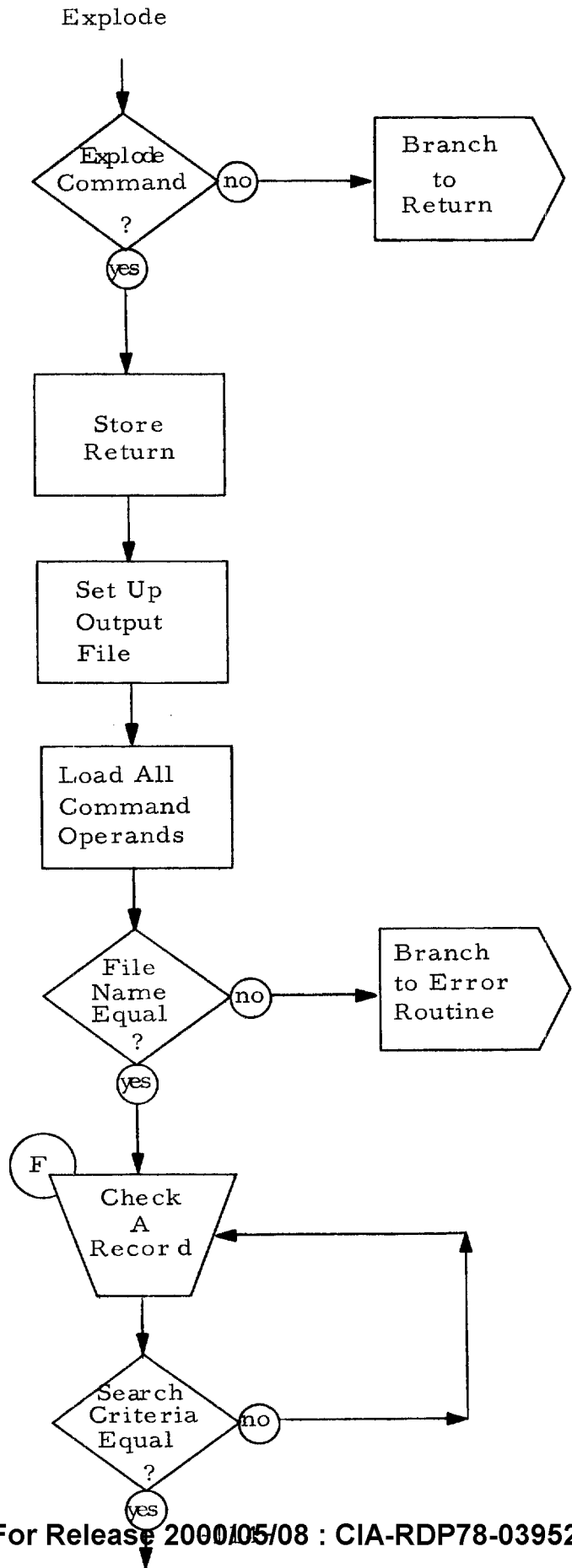
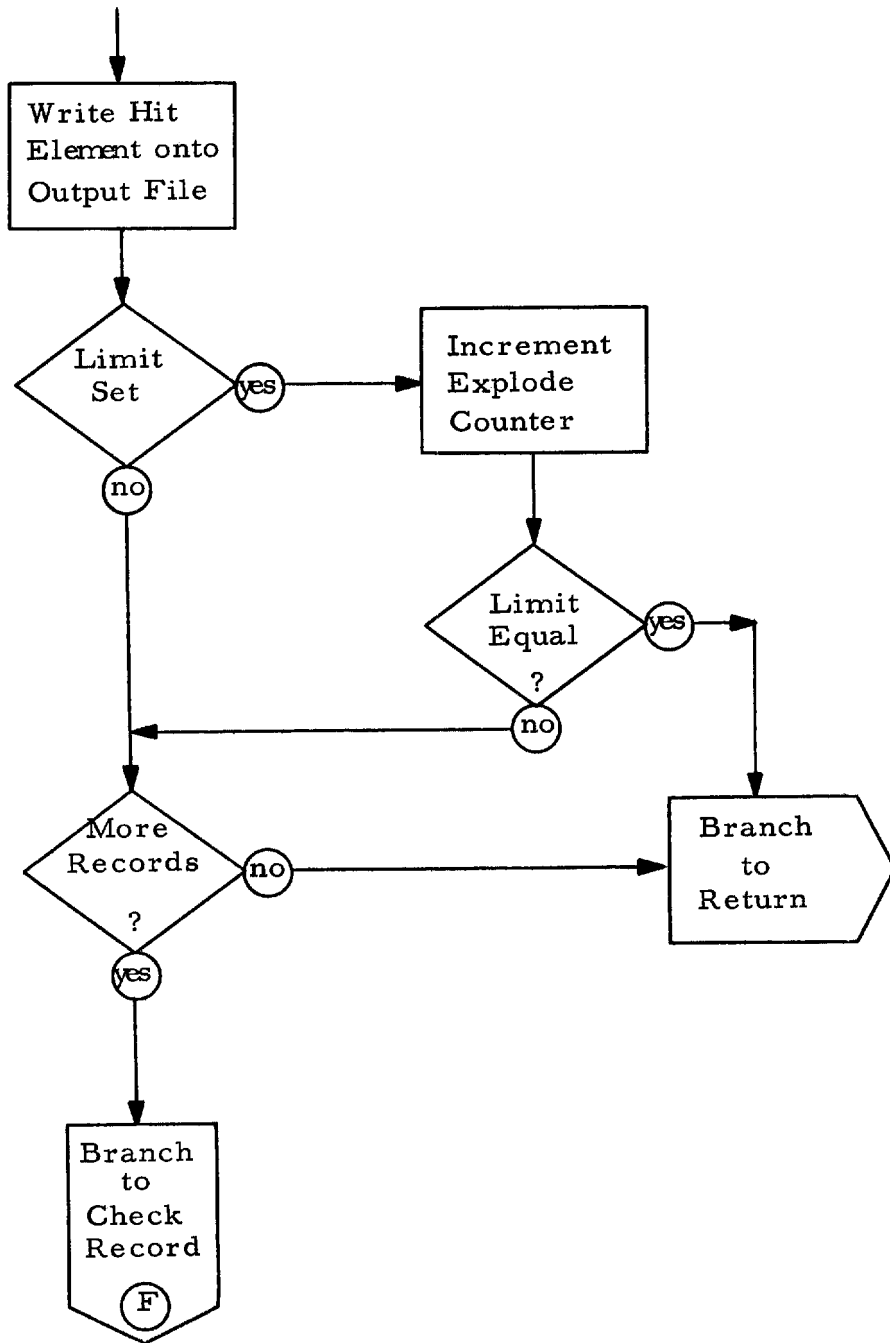


Figure 7-18

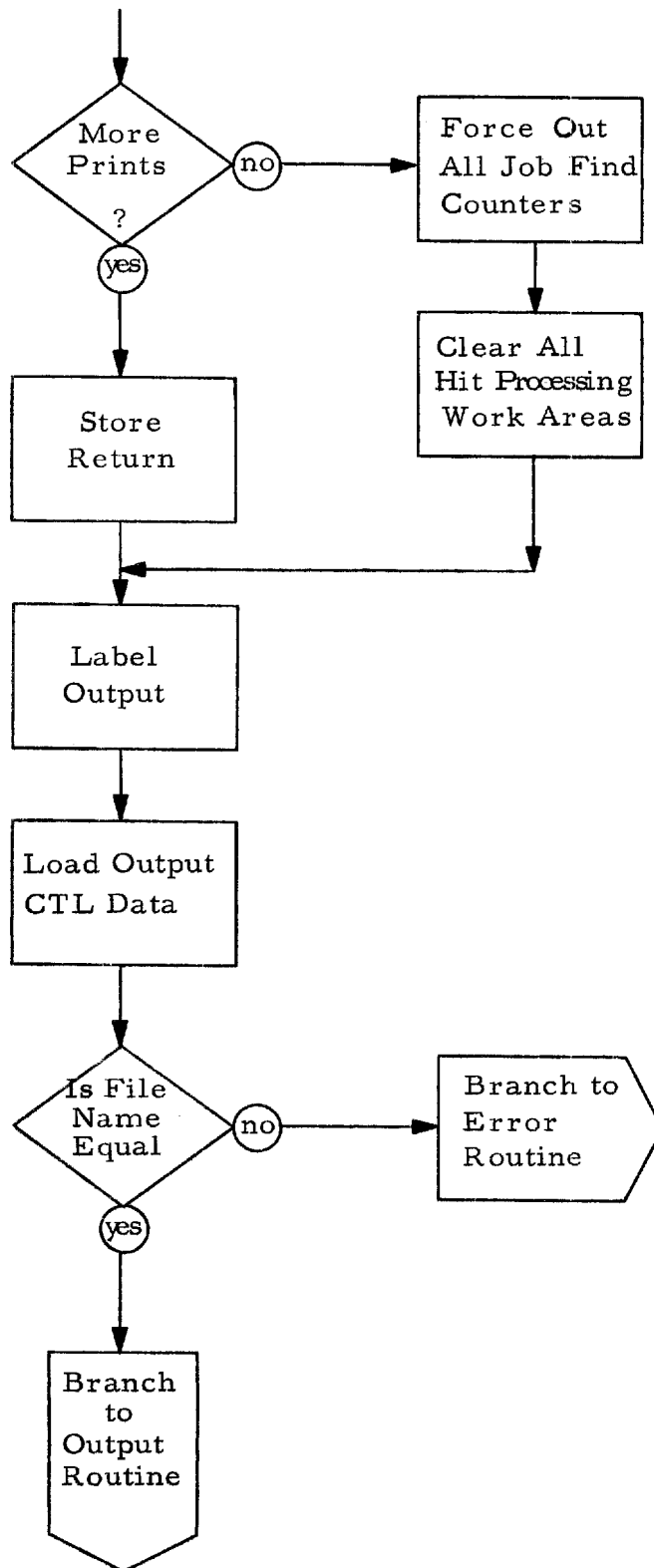
Explode



SECRET

Figure 7-19

Print



SECRET

Chapter 7.9.

INPUT PROCESSING

7.9.1. TYPES OF INPUT

CHIVE EDP inputs will consist primarily of data and commands (maintenance and retrieval) entered on coding sheets. These will be processed by the optical page reader. In the initial system structuring information will be entered via assembled or compiled tables. Examples of structuring inputs are File Format Tables and report format definitions.

Control statements will also be directed at both Operating System/360 and the operational programs. Control information (such as job start, job end, and job priority) are directed at the operating system. More detailed information such as input data file identifications are used by operational programs.

7.9.2. INPUT LANGUAGE PROCESSOR

The input processor or generator interprets and analyzes input command statements and generates, as output, a list of references (program calls) to problem programs.

INPUT PROCESSING
Language Processor
7.9.2.

7.9.2.1. Placement

Because of the expected use of Operating System/360 in the CHIVE system, a logical placement may be within the OS/360 input processing framework. Thus, two alternatives present themselves--either place the generator within OS/360, or outside of OS/360 as the first problem program to handle the input stream. Tentative indications are that the generator may be best used as the first operational program.

A Transaction Control Block (TCB) will be set up for each job being compiled. Core storage blocks will be allocated to the TCB, and the generator output will be stored in these blocks. The final action of the input generator can be a transfer to the first generated instruction. The instructions will consist primarily of LINK macros which will cause transfers to the program(s) required to execute each command.

7.9.2.2. Design

The various points concerning the design of the generator can be narrowed to two subjects--complexity of design and the method of generation.

The processing complexity of a generator is directly proportional to how similar the structure of the user

INPUT PROCESSING
Language Processor
7.9.2.2.

SECRET

language is to the problem program system. A close similarity exists when one statement (command) in the language can be processed by one problem program. In this case, the generator would only need to perform a single table look-up operation to translate each statement into a program "call." In contrast to this simple approach, the generator may be required to perform a complicated analysis on each statement in a sophisticated user language. The CHIVE language is simple enough to support the first approach.

The generator can perform its generation functions in two ways--(a) complete preprocessing, or (b) interpretive. Complete preprocessing is defined as a thorough analysis of the entire set of commands before any of them is executed. When the preprocessing is completed, the commands are in a form usable by the programs; and no further pass is needed through the generator. On the other hand an interpretive generator decodes one statement at a time, executes the statement, then steps to the next statement. The process is continued until the job is completed.

7.9.3. COMMAND MACROS AND STANDING JOB DEFINITIONS

In addition to the regular jobs containing explicit command, transactions can contain references

INPUT PROCESSING
Job Definitions
7.9.3.

SECRET

SECRET

to commands that were previously defined to the generator in the following two ways:

- Command Macro Definition Statements
- Standing Job Definitions Statements

These definitions can be used in a load-and-go job or simply as a stand alone "definition job."

Figure 7-20 shows the input command processing performed on jobs containing the various types of command statements.

There are major differences between the two types of definition statements. A command macro is used to build a command character string, which may contain parameters. It is designed to simplify coding of repetitive command strings and would normally be used within a job. On the other hand, a standing job definition "defines" a complete job which may include control statements, command statements, and command macros.

7.9.3.1. Command Macros

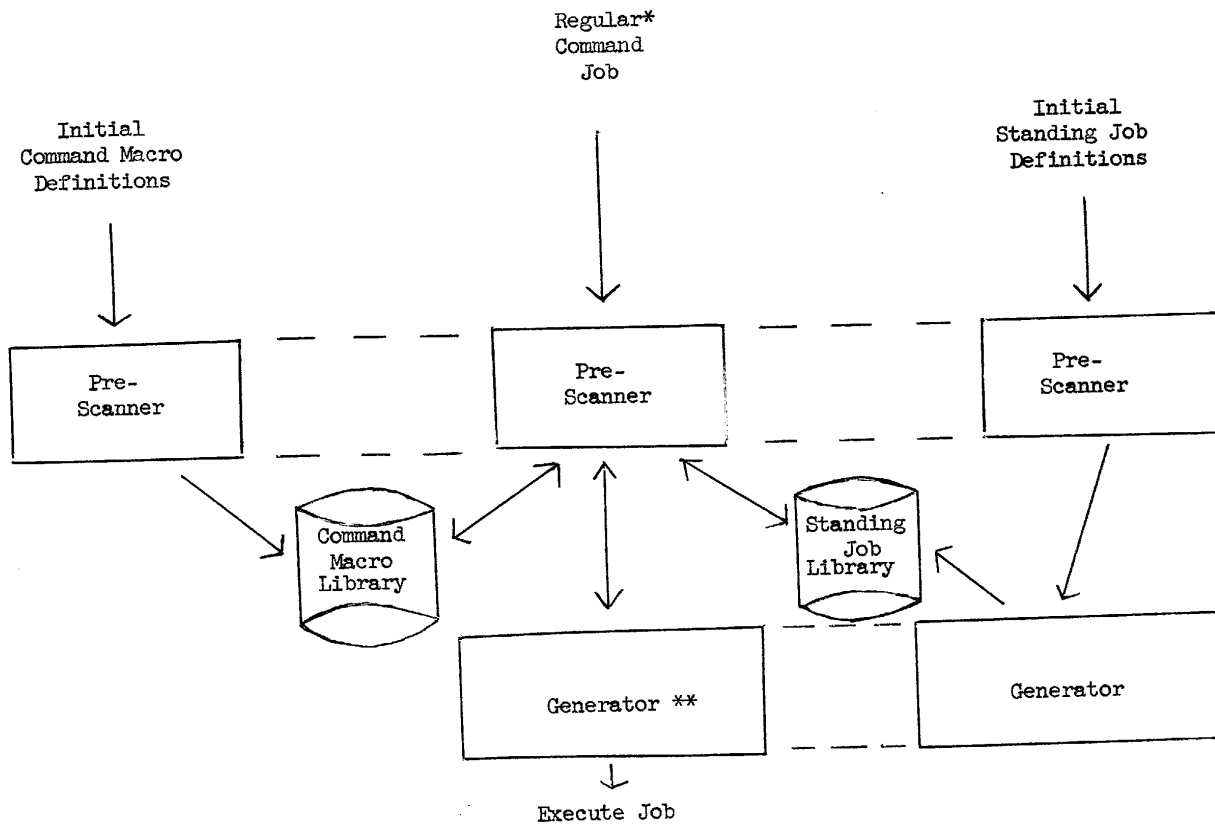
A command macro definition specifies a set of contiguous standard CHIVE commands and a set of dummy parameters which will be "filled in" when this macro is called by a macro statement. Once defined and entered into the system, any subsequent job can use this macro

INPUT PROCESSING
Job Definitions
7.9.3.1.

SECRET

Figure 7-20

INPUT COMMAND STATEMENTS



*May include command macro statements or be a standing job request.

**Standing jobs bypass this stage because they have been processed previously by

SECRET

SECRET

by referring to it by name. It will be "compiled" each time that it is used in a job. When a command macro is detected by the pre-scanner section of the generator, the following events occur;

- Locate the appropriate command macro skeleton in the macro library.
- Fill in the dummy parameters with the real ones given as part of the macro.
- Replace the command macro with the included standard CHIVE commands.

7.9.3.2. Standing Jobs

The creation of a standing job involves a complete specification of a entire job (some parameters may be specified at execution time). The job is processed through the generator as a normal job, but instead of being executed, it is given an identification tag and stored in the standing job library. Henceforth, this job need only be requested by identification number (and, optionally, some parameters supplied to it), and it will be pulled from the library and executed. It is "compiled" once, but may be used as often as necessary.

A job, if it is not a standing job and contains no references to command macros, is processed directly by the generator. A job that refers to one or more command macros undergoes some additional pre-scanner steps

INPUT PROCESSING
Job Definitions
7.9.3.2.

SECRET

which merge the requested commands into the job. A request for a standing job is interpreted by the pre-scanner which starts the standing job through the system.

7.9.4. INPUT JOB BATCHING

Transactions that enter the CHIVE System can be introduced singly or grouped together into a batch job. A job composed of a single transaction poses no requirement for special processing, for it is immediately handled by the generator and executed. Batch jobs, however, require several additional processing steps.

7.9.4.1. Definition of a Batch

A batch job is defined as that set of CHIVE transactions that are present between Operating System/360 "job header" and "end" control statements. The job control capabilities of the OS/360 are consequently used to control batches of CHIVE transactions. Batching in the CHIVE system, however, requires two levels of control statements. Between the two OS/360 control cards are CHIVE control statements that are necessary to define completely the individual transactions within the batch. These CHIVE control statements will be interpreted first by the CHIVE generator and subsequently by the CHIVE problem programs. Transactions using different files

INPUT PROCESSING
Job Batching
7.9.4.1.

SECRET

and having different priorities may be included in the same batch.

7.9.4.2. Batch Processing

The initial processing of batch jobs involves a series of preprocessing steps that are performed by Phase I and II of the CHIVE generator. These steps, some of which may be performed asynchronously, are discussed below. Figure 7-21 shows the major interactions between the various steps.

Phase I - Sorting, Compiling, and Separation: All the input transactions within a batch will be sorted and classified into three groups -

- Retrieval and conditional maintenance transactions
- Unconditional maintenance transactions
- Special page reader inputs, e.g., document index header and content phrase sheets which comprise a new record

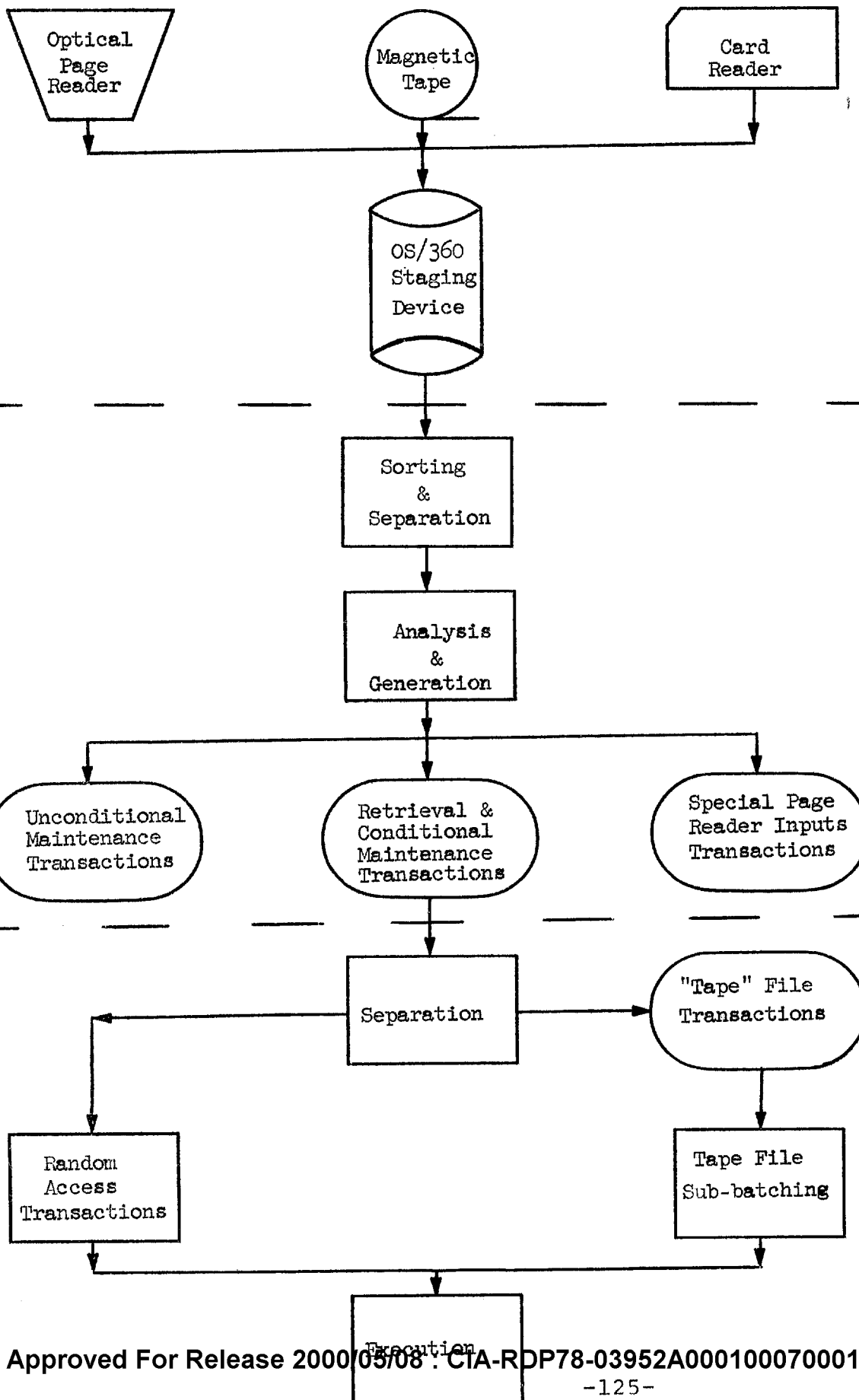
Phase I - Analysis and Generation: The test-lines and commands of each transaction will be compiled into a sequence of program calls.

Phase II - Separation: All transactions that reference direct access files are ready for processing and are introduced immediately into the system. Transactions

INPUT PROCESSING
Job Batching
7.9.4.2.

Figure 7-21

CHIVE INPUT PROCESSING



SECRET

that process a serially access tape file receive additional processing.

Phase II - Tape File Sub-batching: To optimize the accessing of magnetic tape files Phase II will analyze the compiled transactions and rearrange them into meaningful (sub) batches associated with each individual tape file. After this is done, the batches are ready for execution.

7.9.5. INPUT FUNCTIONS

As a summary, a list of the major input functions are given below. All of them are directed at preparing a job for execution in the CHIVE system.

- Read input (data, commands, and control information) from the system input devices.
- Perform checks on input information (such as, formatting and sequencing checks).
- Decompose command statements into program calls (Phase I of the Generator).
- Partition batch jobs and set up optimum processing steps (Phase II of the Generator).
- Provide input services (such as command macro and standing job capability).

INPUT PROCESSING
Input Functions
7.9.5.

SECRET

Chapter 7.10

OUTPUT PROCESSING

7.10.1. OUTPUT ROUTINE DEFINITION

Output reports will provide the vehicle for conveying to the information analyst or a research analyst the data extracted from the CHIVE files. Based upon their intended use, the urgency of the response and the files concerned, the makeup of a printed page may vary from a direct presentation of the file contents, with minimum editing, to a well-organized composition of selected items, arranged hierarchically to accentuate the most significant fields of information, with the more discriminating fields reordered so as to simplify their interpretation. Summary features of complete reports may be further isolated from the body of the report, by placing them on cover pages, to facilitate a preliminary assessment of its contents.

The initial routines for report generation will be prepared to fit the general requirements for the most common reports anticipated. They will be maintained in the system library subject to call by the analyst, with a

OUTPUT PROCESSING
Routine Definition
7.10.1.

~~SECRET~~

PRINT command. Each report can have parameters which must be specified in the PRINT command. An enumeration of the reports to be defined for the various files is included in Table 7.C-1 Same Report Formats and a number of sample layouts for such reports appear in Appendix 7.C.

A report will contain three types of pages as shown in Figure 7-22: cover pages for the complete report, boundary pages for the body of the report, and content pages which represent the bulk of the report.

Variation in the appearance of content pages will be due to differences in the data contained in individual records being represented or due to changes in pages, such as the page number. The format will generally be fixed, subject to any conditions placed upon lines or parts of lines in the report specifications.

Variation in the appearance of boundary pages will be due primarily to the report itself, though these pages may actually contain some information extracted from records. They will contain enough information about the report to assure that if the cover pages are removed the nature and basis of the report can be determined. The first, or initial, pages will contain the identity

OUTPUT PROCESSING
Routine Definition
7.10.1.

SECRET

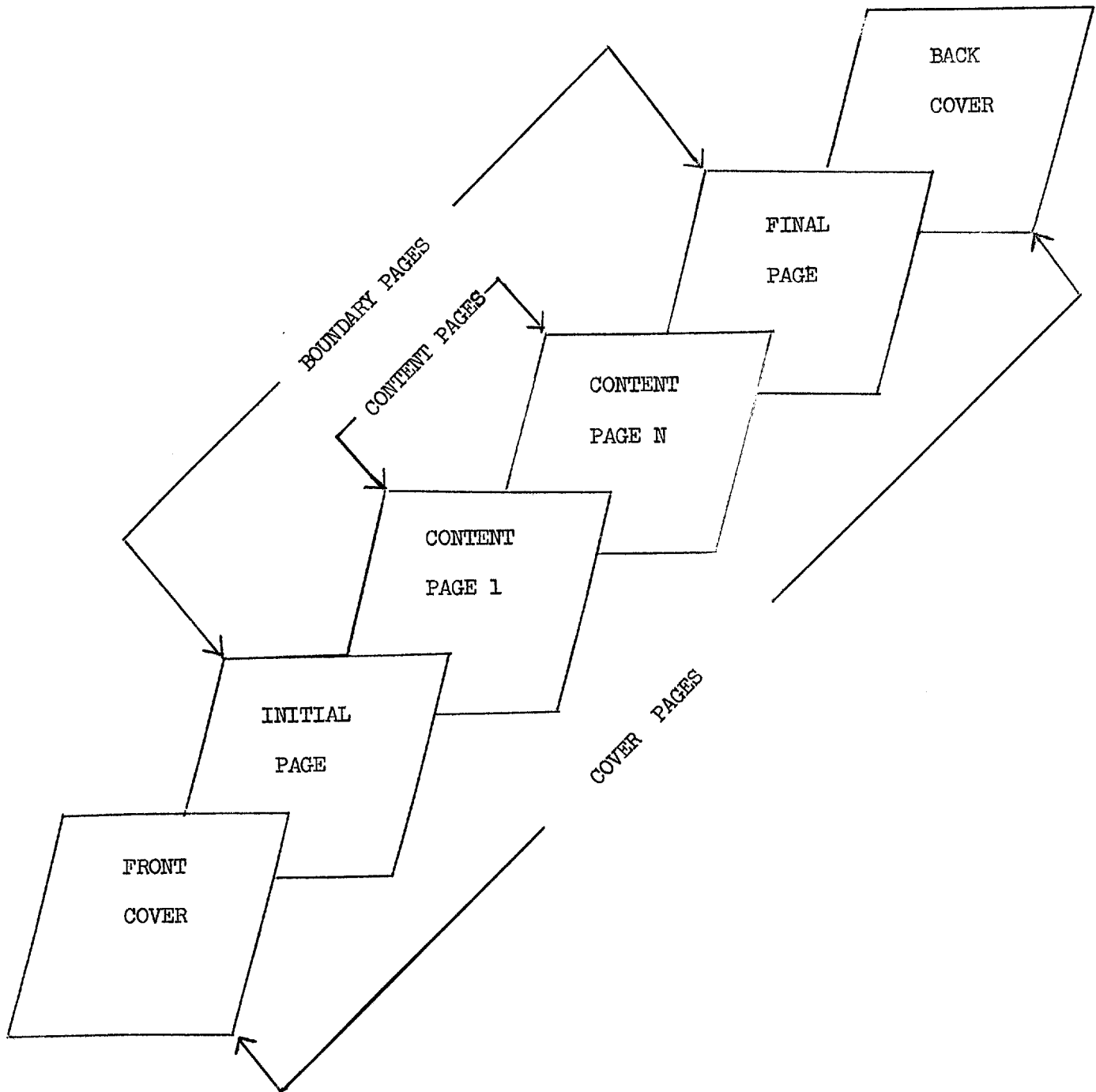


Figure 7-22. PAGE ARRANGEMENT - CHIVE REPORTS

of the report and the request which caused it to be prepared. The last, or final, pages will contain summary information such as totals or counts for the complete output file.

Variations in the appearance of cover pages will be due to the entire retrieval transaction which issues the PRINT command. Front covers will contain clear identification of the request and the requestor with such routing information as is available. It may also present any data descriptive of the file which is known before the body of the report is printed. Back covers will similarly contain identification and routing information and will include entries derived from the entire file reported upon, such as the identity of records (or documents) from which the reported results were extracted. This may be used as a request sheet for subsequent retrieval of selected documents after appropriate annotation by the analyst.

7.10.2. REPORT-RELATED PROCESSING FUNCTIONS

In order that information published in reports be as meaningful as possible to the reader the data may be rearranged and translated from its compact internal form. These processes will be performed automatically as part

OUTPUT PROCESSING
Processing Functions
7.10.2.

SECRET

of the report generation routines on the basis of parameters supplied by the analyst in calling for the report with a PRINT command. The routines which complete these processes may operate at several levels during the report preparation.

At the level of the transaction, the specifications for the transaction may be examined and enough data extracted and stored in cover and boundary page print areas to define the request number, the name and location of the requestor, and, the search criteria. Where no specific PRINT command appears in the transaction specifications, but printing is implied by special commands such as TALLY or ACCUM commands, the entire report generation will be accomplished at the transaction level.

At the level of the output file the process is one of sorting. The PRINT command in the retrieval transaction specifications which initiates the report generation may designate one or more terms which will serve as keys for sorting. This will cause the records in the file to be sorted in the order indicated prior to the actual generation of the report.

At the level of the individual record is the process of cross-referencing source citations. As output

OUTPUT PROCESSING
Processing Functions
7.10.2.

SECRET

records are built up the identity of the document from which the information is derived may be attached to the item placed in the record. Each item and its source document identity may both be linked with a unique sequential flag so that the two parts may be separated and later reassociated (for example, see Figure 7.C-7 in Appendix 7.C.). This will permit the printing of items of information and cross referenced source citations during report generation.

At the level of the term the report-related processing functions are based upon the desire to decode or translate the tags and/or values of the terms and to simplify the representation or interpretation of the meaning of these parts.

The decoding of tags will transform the internal representation of the tag into either a fixed or a variable length equivalent by means of a table lookup. If the ultimate user is familiar with the 3-character mnemonic, this form may be used; otherwise a narrative equivalent will be used. The equivalents must be inserted into the translation table at the time the decoding routines are prepared.

Document titles may be converted into a more easily interpreted form by permuting the contents cyclically,

OUTPUT PROCESSING
Processing Functions
7.10.2.

to align key words in a fixed position and replicating the record for each such key word. The copies may be sorted on the key words in a single group for the record copied or all copies of all records may be assembled as a single file and sorted on the key words which appear at the designated position.

7.10.3. GENERAL REPORT GENERATING FUNCTIONS

A very comprehensive choice of report generation capabilities can be provided with the combination of:

- A varied selection of line types
- Conditional control over line or line element printing
- Line or column overflow for variable length fields
- Vertical spacing through channel carriage control
- Editing and decoding by table lookup or subroutine
- Sorting of a file prior to report generation
- Arithmetic actions taken at report time

7.10.3.1. Line Types

Line types fall into two groups, those which contain only non-record data which is either invariant or varies only with respect to the transaction or the progress of the report itself, and those which contain

OUTPUT PROCESSING
Generating Functions
7.10.3.1.

some record data which may vary between records of the file being reported.

Lines which contain only non-record data may be composed from actual character strings supplied at definition time, character strings previously designated for use in other line definitions or elements of data derived from the transaction definition statements.

Figure 7-23 illustrates the following non-record line types:

- Leader lines which are to be printed at the top and bottom, respectively, of each page printed
- Trailer lines
- Title lines which may appear on the covers or initial pages or on other pages conditionally

Lines which contain record data may be composed from actual character strings supplied at definition time, character strings previously designated for use in other line definitions, elements of data derived from the transaction definition statements, data derived from terms extracted from data records which may have been edited or decoded, or data obtained as the result of arithmetic actions. The record line types are:

- Label lines which may appear on initial, content or final pages,

OUTPUT PROCESSING
Generating Functions
7.10.3.1.

25X1A

Approved For Release 2000/05/08 : CIA-RDP78-03952A000100070001-5

Approved For Release 2000/05/08 : CIA-RDP78-03952A000100070001-5

SECRET

- Term lines which may appear on content pages or the content portion of initial or final pages. In term lines the terms will be assigned to fixed column locations within a line on the basis of their tags. Overflow, if permitted, will remain within the column
- Phrase lines which may appear on content pages or the content portion of initial and final pages. In phrase lines the terms will be positioned on a line in the order in which they appear in the phrase. Possible overflow conventions appear in Section 7.10.3.3., Overflow, below
- Final lines which usually appear conditionally on the terminating page.

The likelihood of occurrence of a line of a particular type in a specified type of page is shown by the following matrix in which

R indicates the line type is required

P indicates the line type is probable

O indicates the line type is optionally possible

Page Type Line Type	Front Cover	Initial Page	Content Page	Final Page	Back Cover
Leader	R	R	R	R	R
Title	P	P	O	O	P
Label		O	O	O	
Phrase		O	P	O	
Term		O	P	O	
Final				P	O
Trailer	R	R	R	R	R

OUTPUT PROCESSING
Generating Functions
7.10.3.1.

SECRET

Briefly the line samples in Figure 7-23 may be described as follows:

- Leader Line: This consists of date, security classification and page number. It will be printed at the top of each page of the report.
- Title Lines: These consist of a title for this type of transaction and information about the request which will be extracted from the transaction specifications.
- Label Line: This is a line of headings which may appear above columns of information describing documents selected during the retrieval process.
- Phrase Lines and Term Lines: These are examples of the presentation of one record of information which has been selected in response to a request concerning men attending a particular conference.

In the case of the phrase lines the fields are printed successively across the page with a fixed 3-character space between fields and the actual phrase content printed in its entirety.

In the term lines the same fields are placed in columns, 30 characters wide, of which 3 characters are used for the tag, 1 character (-) is used to separate the tag and value and 26 characters are used for the value, the remaining characters of the value being lost due to truncation.

In both these line samples it is assumed that the requestor wishes the tags to be printed as 3-character mnemonics, rather than having them decoded into natural English.

- Final Lines: These are examples of information which may be presented to summarize the results of a search through the document index records.
- Trailer Line: This contains information extracted from the transaction specifications as well as the security classification. It will appear at the bottom of each page printed.

OUTPUT PROCESSING
Generating Functions
7.10.3.1.

SECRET

7.10.3.2. Conditional Control of Print Lines

In addition to the conditional retrieval of records, it is possible to condition the printing of reports made from these records. Tests of whether a field is greater than, less than, equal to or not equal to specified values may be accomplished by test conditions similar to those employed in retrieval and conditional maintenance. The serial character of report generation and the ordering of records in files or data in records introduce additional conditions which can be exploited for the enhancement of output. Particular conditions which may be applied to report generation are:

- Change or repetition of the value of a term which recurs in successive print lines or records being reported.
- The completion (exhaustion) of reporting of an identifiable subset of an ordered set of values or data collections, at any level of the hierarchy.
- The satisfaction of test conditions by record data.
- The satisfaction of test conditions by non-record data.

Examples of the use of such conditions might be:

- To cause a label line to be printed whenever a record changes.
- To suppress the printing of part of a line which contains terms which are designated as the sort key whenever that part would not change from line to line.

OUTPUT PROCESSING
Generating Functions
7.10.3.2.

SECRET

- To cause the printing of special intermediate lines when all records in the file containing a particular value of a sort key term have been processed.
- To cause a particular line format to be used for records meeting test conditions.
- To cause printing to be halted after a specified number of records has been processed.

The manner in which these tests may be applied to the various line types is:

- Leader, title, final and trailer lines or line elements may be conditioned by tests on non-record data.
- Label or term lines or line elements and phrase lines may be conditioned upon tests, changes or completion of record data.
- Label and term lines may be associated with one another so that the label line will appear only if its associated term line does. The conditional control of label lines is therefore subordinated to the control of the term line associated with it.

7.10.3.3. Overflow

The printing of term lines may cause overflow in two ways, (a) the variable number of occurrences of a tag (i.e., different values for the same tag) or (b) the variable length of values of terms with a given tag which is to be printed in a fixed-length field. The first cause, variable occurrences of the same tag, may be accommodated by printing the term lines as many times as the greatest number of occurrences of any term

OUTPUT PROCESSING
Generating Functions
7.10.3.3.

SECRET

type which appears in the term line. For those term types having fewer values than this greatest number, blank fields will appear for each term line element for that tag after its values have been exhausted. As a result the appearance of a specified term line type may be:

Tag 1	Tag 2	Tag 3
Value 1	Value 1 with Overflow	Value 1
Value 2	Value 2	Value 2 with Overflow
	Value 3	
	Value 4	

For the second case, variable length values, there are three options which may be selected in cases where the term value is not of constant length:

- Values may be converted to a fixed length by use of a predefined equivalence table.
- Overflowing values may be truncated to the length available with attendant loss of the excessive characters
- Overflowing values may be carried over to succeeding lines in that column with some editing to preserve readability. This is the procedure assumed in the example above.

The printing of phrase lines, because of the greater variability possible in phrase contents, offers more

OUTPUT PROCESSING
Generating Functions
7.10.3.3.

SECRET

alternatives in format and the concomitant handling of overflow:

- The terms of a phrase may be sequenced horizontally across the page or they may be arranged vertically in a column only one term wide.
- A term from the phrase may be arranged in a single line as tag followed by value, or in a double line as a tag over a value.
- Columns for terms may be in fixed positions or may float in response to the maximum width of a value for terms in a phrase.

The nature of the record printout, in phrase-line format, which will be created by each of the eight possible combinations of these options are shown in Figures 7-24 through 7-31.

7.10.3.4. Carriage Control

The format of a report can be affected significantly by the control of spacing between lines or parts of a report. This may be attained in three different ways: spacing a specified number lines, ejecting to the next page or skipping to a specified line number on the page. Skipping may cause the carriage to move forward on the current page or to skip to the line number on the next page depending upon whether the current line number before the skip is or is not less than the specified number.

OUTPUT PROCESSING
Generating Functions
7.10.3.4.

Figure 7-25

Phrase Print Format 2

Horizontal Arrangement; Single Line; Fixed Term Position

HEADER	Tag 1 (xxxxx)	Tag 2 (xxxxxxxxxx)
	Tag 3 (xxxxxxxxxxxxxxxxxx)	Tag 4 (xxxxxxxxxxxxxxxxxx xx)
	Tag 5 (xxxxxxxxxxxxxxxxxx xxxxxx)	Tag 6 (xxxxxxxxxxxxxxxxxx xxxxxxxxxx)
	Tag 7 (xxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxx)	
Phrase 1	Tag 1 (xxxxx)	Tag 2 (xxxxxxxxxx)
	Tag 3 (xxxxxxxxxxxxxxxxxx)	Tag 4 (xxxxxxxxxxxxxxxxxx xx)
	Tag 5 (xxxxxxxxxxxxxxxxxx xxxxxx)	
Phrase 2	Tag 1 (xxxxx)	Tag 2 (xxxxxxxxxx)

Figure 7-26

Phrase Print Format 3

Horizontal Arrangement; Double Line; Floating Term Position

HEADER	Tag 1	Tag 2	Tag 3
	(xxxxx)	(xxxxxx)	(xxxxxxx)
	Tag 4	Tag 5	Tag 6
	(xxxxxxxx)	(xxxxxxxx)	(xxxxxxxx)
	Tag 7		
	(xxxxxxxxxxx)		
Phrase 1	Tag 1	Tag 2	Tag 3
	(xxxxx)	(xxxxxx)	(xxxxxxx)
	Tag 4	Tag 5	
	(xxxxxxxx)	(xxxxxxxx)	
Phrase 2	Tag 1	Tag 2	Tag 3
	(xxxxx)	(xxxxxx)	(xxxxxxx)
	Tag 4	Tag 5	
	(xxxxxxxx)	(xxxxxxxx)	

Figure 7-27

Phrase Print Format 4

Horizontal Arrangement; Double Line; Fixed Term Position

HEADER	Tag 1	Tag 2	Tag 3
	(xxxxxx)	(xxxxxxxx)	(xxxxxxxxxx)
	Tag 4	Tag 5	Tag 6
	(xxxxxxxxxx)	(xxxxxxxxxxxx x)	(xxxxxxxxxx xxx)
	Tag 7		
	(xxxxxxxxxx xxxxx)		
Phrase 1	Tag 1	Tag 2	Tag 3
	(xxxxxx)	(xxxxxxxx)	(xxxxxxxxxx)
	Tag 4	Tag 5	Tag 6
	(xxxxxxxxxx)	(xxxxxxxxxxxx x)	(xxxxxxxxxx xxx)
Phrase 2	Tag 1	Tag 2	
	(xxxxxx)	(xxxxxxxx)	

SECRET

Figure 7-28

Phrase Print Format 5

Vertical Arrangement; Single Line; Floating Term Position

<u>Header</u>	<u>Phrase 1</u>	<u>Phrase 2</u>
Tag 1 (xxxxxxxx)	Tag 1 (xxxxxxxxxxxxxxxxxxxx)	Tag 1 (xxxxxxxxxxxxxxxx)
Tag 2 (xxxxxxxxxxxxxxxxxxxx)	Tag 2 (xxxxx)	Tag 2 (xxxxxxxx)
Tag 3 (xxxxxxxxxxxx)	Tag 3 (xxxxxxxxxxxx)	Tag 3 (xxxxxxxxxxxxxxxxxxxx)
Tag 4 (xxxxx)		Tag 4 (xxxx)
Tag 5 (xxxxxxxxxxxxxxxxxxxx)		Tag 5 (xxxxxxxx)
		Tag 6 (xxxxxxxxxxxxxxxxxxxx)

<u>Phrase 3</u>	<u>Phrase 4</u>
Tag 1 (xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)	Tag 1 (xxxxxxxxxxxx)
Tag 2 (xxxxxxxxxxxx)	Tag 2 (xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)

SECRET

Figure 7-29

Phrase Print Format 6

Vertical Arrangement; Single Line; Fixed Term Position

<u>Header</u>	<u>Phrase 1</u>	<u>Phrase 2</u>
Tag 1 (xxxxxxxxxxxxxxxxx xxxxx)	Tag 1 (xxxxxxxxxxx)	Tag 1 (xxxxxxxxxxxxx)
Tag 2 (xxxxxxxxx)	Tag 2 (xxxxxxxxxxxxxxxxx xxxxxxxxxxxxx)	Tag 2 (xxxxxxxxxxxxxxxxx xxxxx)
Tag 3 (xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx)	Tag 3 (xxxxxxxxxxxxx)	
	Tag 4 (xxxxx)	
 <u>Phrase 3</u>	 <u>Phrase 4</u>	 <u>Phrase 5</u>
Tag 1 (xxxxxxxxxxx)	Tag 1 (xxxxxxxxxxxxxxxxx xxxxx)	Tag 1 (xxxxxxxxxxxxx)
Tag 2 (xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx)	Tag 2 (xxxxxxxxxxxxxxxxx xx)	Tag 2 (xxxxxxx)
	Tag 3 (xxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxx)	

All three methods of controlling spacing may be conditioned by test, change or completion conditions as explained in Section 7.10.3.2. above. They may also be called for unconditionally, before printing of a designated line type or between records which are being reported.

7.10.3.5. Decoding

Decoding may involve the use of subroutines or tables or a combination of these. The most common method will be the use of a table in which the argument will be the internal representation of the item to be converted and the values will be one or more representations of the item in a converted form. Where the form used for reporting is the same as that used for input there will probably be inverted tables used for input conversion or encoding.

The increased readability of reports whose fields are of fixed length tends to encourage the use of conversion tables. Some of the fields which may lend themselves to this type of treatment are:

- Tags
- Subject Codes
- Area Codes

OUTPUT PROCESSING
Generating Functions
7.10.3.5.

- Security Classification
- Evaluation Codes
- Dissemination and Collection Codes
- Codewords

Some types of items lend themselves to direct algorithmic subroutine handling. Dates are more easily compared internally if they are in numeric form in the order of year, month and day, whereas for reading the user is more accustomed to their appearance in the order of day, month and year. The rearrangement is a transposition of characters while the month may be converted with tables. For similar reasons the characters in the Intelligence Information Report Number require transposition as well as any conversion which may be desired.

7.10.3.6. Sorting

Sorting of a file will be accomplished during output processing by a system sort program. When any specifications for a sort key are recognized in the PRINT command, the entire sort key will be determined. It will then be specified to the sort program and the sort will be performed before the file is processed for printing.

OUTPUT PROCESSING
Generating Functions
7.10.3.6.

If one or more iterative tags are named in a sort key there is an implicit exploding of each record on each sort key value in the record. The exploded records are printed out in sort order.

7.10.3.7. Arithmetic Functions

Arithmetic functions which may be included as part of output processing are:

- Tallying of items which satisfy specified test conditions across records. The tally will be stored in a six digit field.
- Accumulating sum of numeric fields across records. The total will be stored in a field twice as long as the field being accumulated.
- Addition of numeric fields occurring in one record. The sum will be placed in a field one digit longer than the longest term in the sum.
- Subtraction of fields occurring in one record. The difference will be placed in a field one digit longer than the longest term in the difference.
- Multiplication of fields occurring in a record. The product will be placed in a field as long as the sum of the lengths of the factors.
- Division of one field by another. The quotient will be stored in a field as long as the dividend.

For arithmetic purposes fields derived from the performance of arithmetic functions upon fields in the same record may be considered fields occurring in the record for further computation.

OUTPUT PROCESSING
Generating Functions
7.10.3.7.

Chapter 7.11.

SECURITY PROCEDURES

7.11.1. PROBLEM STATEMENT

CHIVE EDP security requirements are fundamentally unique when compared with EDP security requirements in existing intelligence processing systems. CHIVE security requirements ultimately stem from the multiprogramming character of the CHIVE system. With multiprogramming, jobs of different sensitivity are run concurrently. Multiprogramming gives rise to the practice of maintaining diverse data files simultaneously on line--files containing data of different classification. This is in contrast with the normal mode of operations in the intelligence community, where jobs are batched by security classification and run one-at-a-time through the system. Storage units at one security level are mounted or dismounted for each batch. Conventionally, security isolation between files of varying sensitivity is achieved by physical human intervention, not automatically.

CHIVE, on the other hand, maintains diverse physical files on-line, ready at all times to serve demands made

SECURITY PROCEDURES
Problem Statement
7.11.1.

SECRET

by the processor. Because the CHIVE data base is all-source, CHIVE maintains files of data at varying security levels; but because CHIVE service is tailor-made to customers of different clearance, output records from homogeneous files of different classification must be intermixed, and heterogeneous files must be screened.

7.11.2. MIXED DATA-CLASSIFICATION CONCEPT (ALL SOURCE)

There are two types of CHIVE data files with respect to security (a) heterogeneous and (b) homogeneous.

Heterogeneous files contain records which are security controlled. Each record may be at a different classification level from the others. Homogeneous files contain records all of which are classed at a single level, that of the file itself.

One may consider a heterogeneous or homogeneous logical or physical file, to distinguish the data contents from the physical storage medium on which they are recorded. One may also regard a single storage device such as a disk to consist of one or more physical files (e.g., each cylinder may be a physical file), or a single storage device may be one segment of a larger physical file. One

SECURITY PROCEDURES
Mixed Concept
7.11.2.

SECRET

logical file may span or comprise one or more or less than one physical file. Therefore, several logical files may be stored on one physical file, or one logical file may be strung over several physical files.

In CHIVE the classification of data must be distinguished from the classification of the storage media upon which they reside. All dismountable or fixed storage units within CHIVE are all source. If a dismountable unit is released to an outside agency, the unit is classified at the level of the most sensitive data it contains or has contained since last erased for declassification of the physical storage medium.* If the unit is erased, it is reclassified.

Logical files may be assigned to any CHIVE-internal physical file. In CHIVE, all allocations of logical to physical files are allowed, irrespective of the security levels of the physical or logical files, except that no logical file may be assigned to a physical file which is not at a security level at least equal to that of the data. Of course, since only dismountable devices releasable to outside agencies are not all-source, any fixed device may receive any logical file.

*Note: See Section 7.11.14.4. on protection against laboratory restoration techniques. A similar technique for dismountable disks will be developed to permit downgrading of the physical medium.

SECURITY PROCEDURES
Mixed Concept
7.11.2.

SECRET

Furthermore, two or more logical files may occupy the same physical storage device without regard to the level of the device or to each other, provided that the level of the device is not exceeded by the level of either file. Each logical file retains its own security level.

In summary, within the stipulations of the mixed data-classification concept, any combination of logical and physical files at any security level may reside on-line simultaneously. Security isolation of data is achieved by software checking procedures.

7.11.3. MIXED EDP JOB CLASSIFICATION CONCEPT (ALL DEMAND)

In CHIVE, without regard to security level, different jobs may be simultaneously in the job queue and the ready queue, while still another security level may be held by the job currently in control of the CPU. In other words, the EDP system will process jobs of different levels at the same time.

Jobs contending for the CPU are isolated whenever possible. Such isolation is achieved by means of safeguards in the Executive Control. The EC employs the IBM/360 storage protect key, which uniquely specifies data and programs associated with a given job-task combination.

SECURITY PROCEDURES
Classification Concept
7.11.3.

SECRET

A job is classified only because it operates upon classified data. Therefore, different job-task classifications exist in the system when and only when information consumers make simultaneous demands on the CHIVE data base for different levels of data.

7.11.4. PROBLEM SOLUTION

The combination of the mixed data-classification concept and the software job/task safeguards in the EC is the means in principle by which the CHIVE EDP system solves the fundamental problems posed by the multi-programming environment. The implementation of this solution is attained at the cost of added overhead in the Executive Control and an increased burden on the CPU. Cost has been held to a minimum consistent with the degree of confidence sought. Especially sensitive ad hoc projects which cannot fit into the security philosophy of CHIVE in general, must be treated separately. Ad hoc projects will be reviewed as they arise to determine whether minor alterations to CHIVE procedures can make them acceptable.

7.11.5. DECOMPARTMENTALIZED ALL-SOURCE/ALL-SERVICE CONCEPT

Compartmentalization is the technique of achieving security isolation by physically segregating operational

SECURITY PROCEDURES
Decompartamentalized Concept
7.11.5.

personnel, processor and files, and by timing jobs in accordance with security level and need-to-know. Compartmentalization internal to the system is in direct contradiction to the CHIVE concept of all-source/all-service.

Decompartmentalization affects all aspects of CHIVE processing, both within the EDP system and without. From the standpoint of computer operations, all operators are qualified to run all jobs. From the standpoint of the EDP processor, by exercising suitable precautions, jobs may be intermixed and files may be physically available on-line irrespective of security. From the standpoint of information handling operations performed by the information analyst, any analyst may service any recipient of information, placing a significant burden for routine security screening upon the EDP system.

7.11.6. THE ROLE OF "NEED-TO-KNOW" IN CHIVE SECURITY DESIGN

The aim of CHIVE security design has been to achieve strict security control over outright security breaches through software as well as hardware safeguards. A security breach is defined as a violation of security level, where level is interpreted broadly to include groups of security classifications such as T/KH/SI/

SECURITY PROCEDURES
Need-to-Know
7.11.6.

Collateral or combinations of these. The security breach also applied to dissemination caveats, when these are significant. For example, "CIA Internal Use Only" is treated as a classification of weight approaching that of any code word.

When security control has been balanced against cost, only need-to-know has been affected. Thus, the accidental writing of one SECRET job output into the printer assigned to another SECRET job output is treated as a tolerable risk when safeguards are costly, while writing a TOP SECRET job output into a SECRET output is treated as intolerable.

7.11.7. SECURITY ISOLATION OF PROGRAMS AND DATA IN STORAGE

7.11.7.1. Storage Protect Feature

The storage protect feature is a hardware feature which significantly affects the handling by the EC of core blocks of data and programs. With this feature, core may not be illegally written.

The EC dynamically assigns matching keys to programs and data and to blocks of core in accordance with the task using them. When a program issues an OPEN macro which calls upon a data file in an I/O device, the EC

SECURITY PROCEDURES
Isolation in Storage
7.11.7.1.

will identify the task which is working. It will then obtain the storage protect key assigned to the task and assign it to the core block into which the data will be written. Data read into core from auxiliary storage by means of a READ macro is buffered into the block(s) assigned for the use of this task. The peripheral equipment checks to insure that the task key matches the key of the affected core block.

When the program issues a move instruction which reads and writes data in core, it will use the storage-protect key given to it by the EC to unlock the data area. If this key is incorrect, or if the key is correct but the wrong core block is accessed, an interrupt will occur. When such an illegal condition exists, the access will be denied and an appropriate error message sent to the operator. The error will be recorded in the system log and the Error Supervisor will be called.

Resident programs such as the Monitor always will be assigned to protected core.

SECURITY PROCEDURES
Isolation in Storage
7.11.7.1.

7.11.7.2. Write-Protect Feature

With this feature core blocks may be read only by programs having the legal key. Software support parallels that for the read-only feature.

7.11.7.3. Branch-Protect Feature

A branch protect feature will not be utilized. Storage-protect and write-protect together obviate the need for a branch protect, since an illegal branch, although honored, can only trigger a security breach when an I/O results, and such a breach is stopped by the other two features in combination.

7.11.8. SECURITY ISOLATION OF DATA TRANSFERS

Core-to-core transfers of data and programs are isolated by means of the storage-protect/write-protect features. The EC finds the key in the Program Status Word (PSW) at the time the problem program calls for the move. The key in the PSW is compared by the circuitry with the key in the core block, which automatically interrupts when an improper condition (mismatch) is sensed. The actual transfer is made with complete parity checking to protect not only the integrity of the transferred information, but the routing assignment as well.

SECURITY PROCEDURES
Isolation of Data Transfers
7.11.8.

Core-to-auxiliary storage transfers and the reverse are made secure under management of the control programs, which utilize the protect keys for this purpose. The data channel uses the storage protect key. The usual parity checks also are made.

Parity checks are deemed sufficient to secure peripheral control over routing addresses. However, routing assignments must be confirmed by the EC in the manner described below.

When the file is being accessed for the first time after opening by a specific task, the File Header Record (FHR)* is interrogated by the EC to insure that the file on the device is the one sought. This check insures that the physical file actually accessed on auxiliary storage is the one called by the problem program and addressed by the EC.

A sought file may differ from the actual file found at an auxiliary storage location for these reasons.

(a) In dismountable files, the actual cell or disk pack mounted on the storage drive may differ from the one

*Note: See Section 5.5.1.5.5 for a discussion of the FHR.

SECURITY PROCEDURES
Isolation of Data Transfers
7.11.8.

SECRET

specified. (b) Although unlikely, a file address may be accidentally altered at issuance time by a machine error in the storage control unit or in the CPU.

After the first access to a file made by a particular routine working on a specific task, remaining accesses are not checked against the FHR. This reduces overhead without materially increasing risk.

The above checks are made as appropriate for both file retrieval and maintenance. Writing into files is checked in order to preclude accidental writing of sensitive information into less-sensitive files. Although these checks are very significant security safeguards, they also are part of the normal checking procedures built in to the operating system because of other EDP concerns. Thus, they are available bonus features whose cost need not be borne by security requirements.

7.11.9. SECURITY ISOLATION OF JOBS AND TASKS

Security isolation of jobs is achieved by assigning them a security classification. Tasks assume the classification of the job to which they belong. The classification of a retrieval job is the highest level of the sought output. That of an input (maintenance) job is the level of the most sensitive data being entered.

An explicit statement of job security level is made by the file analyst in the external EDP environment

SECURITY PROCEDURES
Isolation of Jobs and Tasks
7.11.9.

SECRET

using the request language. Any job type may involve any security classification. A specific job of a given type may have a level differing from that of other jobs of that type. For example, a specific reference retrieval job may be ALL-SOURCE, but another reference retrieval job may be limited to SECRET.

Another form of explicit statement is made in standing requests, which are held in a table of standing requests, with output security level an entry. For standing requests which may take on varying levels, the user will either specify the level as a parameter in the request language statement when he calls upon the standing request, or separate standing requests for a single job type will be made up for the several security levels the job type can take on. In the latter case, each standing request of that type will be referred to by a different name.

For some job types, there may be only one legal security level. When parent tasks generate sub-tasks, the sub-tasks will have a security classification of their own, since they have an independent life in the system. This classification can be equal to, less

SECURITY PROCEDURES
Isolation of Jobs and Tasks
7.11.9.

than or greater than the level of the parent task. The correct level is determined by any of the criteria outlined above.

EDP system processing files are not "seen" by the request writer or referred to explicitly in the request language, but are required to process a job (see Section 7.11.10.). These files are called or opened under control of the Operating System or the problem program. Hence, opening legality is achieved by correct design of programs.

7.11.10. PRIVILEGED ACCESS TO LOGICAL FILES

Logical internal files are those files which are automated within the EDP system. There are two kinds. Internal information files or data files are files of data which usually have counterpart files (that is, Printouts) which they generate for use in the external, non-EDP environment.

A second kind of internal file is the system processing file. This kind of file is used by the EDP system for bookkeeping, logging, system status monitoring and other purely EDP functions. System processing files

SECURITY PROCEDURES
Access to Logical Files
7.11.10.

may not be directly accessed by the request language. Access to these files is privileged; it is limited to CHIVE system programmers.

7.11.11. ACCESS TO AUXILIARY STORAGE UNITS

All programs may have access to any file of data on any storage unit. After the file (or a block of data from the file) has been moved into core, it will reside alone under protection of a storage-protect key.

7.11.12. FILE INTERACTION AND SECURITY ISOLATION

The security level of a data file which is operated on is not the sole criterion for security isolation. Two data sets or files under the same storage key may be vulnerable to security violation even though they are at an identical level. The combination of source data security level, data file character (i.e., homogeneous or heterogeneous), target file level and character, and task (i.e., file interaction) determine the isolation criteria.

For example, consider the case in which two TOP SECRET data files share a storage protect key. They reside in core. Task I is to screen and convert part of

SECURITY PROCEDURES
File Interaction
7.11.12.

data file I which is heterogeneous into data file IA which is homogeneous SECRET. Task II is to move data from file II which is homogeneous into data file IIA which is homogeneous TOP SECRET. If Task I should inadvertently operate on data file II, no screening will take place, and TOP SECRET data could find its way into SECRET data file IA.

The File Interaction Security Transition Table (see Table 7-1) depicts the various combinations of file interaction. These apply equally to I/O and core-to-core transfers.

The job classification (as used in Table 7-1) is the classification for which the information recipient is cleared. It is the screening criterion, when retrieval is involved, or it is the input legality criterion when record maintenance is involved.

Entry type #1 lists the information unit which sets the level assigned to data units moved out of a file. For instance, consider a job at a level greater than a homogeneous source file which moves data out of the file and into a homogeneous target file which is at a level beneath that of the job. The level of the moved data becomes the level of the target file.

SECURITY PROCEDURES
File Interaction
7.11.12.

Table 7-1

FILE INTERACTION SECURITY TRANSITION TABLE

Source File	Target File Job Class	One Homogeneous File			One Heterogeneous File		
		Less Than Target File	Equal To Target File	Greater Than Target File	Less Than Target File	Equal To Target File	Greater Than Target File
One or More Homogeneous File(s)	Less Than Source File(s)	1. Target File 2. Nothing 3. Source File Target File	Illegal	Illegal	1. Source File 2. Extracted Unit 3. Source File Target File	Illegal	Illegal
	Equal To Source File(s)	1. Target File 2. Nothing 3. None	1. Source File/Target File/Task 2. Nothing 3. None	Illegal	1. Source File 2. Extracted Unit 3. None	1. Source File 2. Extracted Unit 3. None	Illegal
	Greater Than Source File	1. Target File 2. Nothing 3. None	1. Target File 2. Nothing 3. None	1. Target File 2. Nothing 3. Source File Target File	1. Source File 2. Extracted Unit 3. None	1. Source File 2. Extracted 3. None	1. Source File 2. Extracted Unit 3. Source File Target File
One or More Heterogeneous File(s)	Less Than Source File(s)	1. Target File 2. Tag Dropped 3. Source Unit Target File	1. Target File 2. Tag Dropped 3. Source Unit Task/Target File	1. Target File 2. Tag Dropped 3. Source Unit Target File	1. Extracted Unit 2. Extracted Unit (Tag Unchanged) 3. Source Unit Target File	1. Extracted Unit 2. Extracted Unit (Tag Unchanged) 3. Source Unit Task/Target File	1. Extracted Unit 2. Extracted Unit (Tag Unchanged) 3. Source Unit Target File
	Equal To Source File(s)	1. Target File 2. Tag Dropped 3. Source Unit Target File	1. Target File 2. Tag Dropped 3. Source Unit Task/Target File	1. Target File 2. Tag Dropped 3. Source Unit Target File	1. Extracted Unit 2. Extracted Unit (Tag Unchanged) 3. None	1. Extracted Unit 2. Extracted Unit (Tag Unchanged) 3. None	1. Extracted Unit 2. Extracted Unit (Tag Unchanged) 3. Source Unit Target File
	Greater Than Source File	1. Target File 2. Tag Dropped 3. None	1. Target File 2. Tag Dropped 3. None	1. Target File 2. Tag Dropped 3. Source Tag Target File	1. Extracted Unit 2. Extracted Unit (Tag Unchanged) 3. Source Unit Target File	1. Task/Extracted Unit 2. Extracted Unit (Tag Unchanged) 3. None	1. Extracted Unit 2. Extracted Unit (Tag Unchanged) 3. Source Unit Target File

Entry type #2 lists the security level tagging activity which takes place when the information unit is moved. It alternatively lists the information unit which is tagged.

Entry type #3 lists any legality condition which must be met.

The legality conditions set forth in this table are followed implicitly by the information analyst in composing his request. These rules apply to all CHIVE files. For system-defined files and ad hoc files (working files which may be destroyed after the job is run), the analyst specifies and controls legality. For files not "seen" by the request writer but used by the EDP system for processing, and for file interactions generated by sub-tasks, legality is handled internally by the CHIVE system programs.

In order to simplify the assignment of storage-protect keys, all data sets called into core by a single task will be isolated under a unique key.

7.11.13. PRIVILEGED ACCESS TO OUTPUT DEVICES

Output devices are all-source. Protection is gained by means of physical isolation of the operations area in

SECURITY PROCEDURES
Access to Output Devices
7.11.13

which the device is located. Access to areas in which output devices are located is limited to qualified (all-source) personnel.

7.11.14. EDP SOFTWARE SECURITY ISOLATION

7.11.14.1. Tamper-proof Operating System

A number of software security features are implemented by means of the Executive Control. To insure the correct function of these features, it is necessary to isolate the EC from the action of problem programs and programmers. This is accomplished by running the computer in the problem state when problem programs are in the system. The EC may be modified by means of privileged instructions which are honored only when the system is in the supervisory state.

7.11.14.2. Core Erasure

When blocks of core are released by problem programs, the EC will erase that core by writing zeros in place of classified data.

7.11.14.3. Auxiliary Storage Erasure

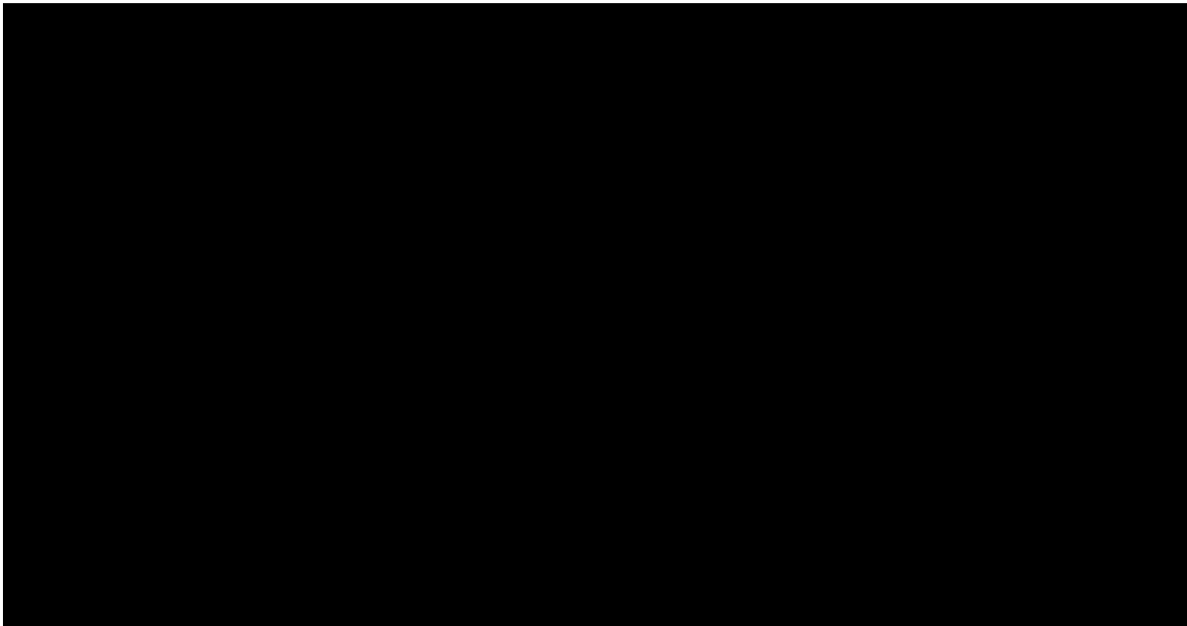
When files of data on fixed auxiliary storage devices are no longer needed by the system, they will be erased by

SECURITY PROCEDURES
EDP Software Isolation
7.11.14.3.

means of a job specially submitted by the computer operator for this purpose. The job will cause the writing of zeros on the recording surfaces. Working files not seen by analysts will be erased under control of the EC. In this case, the problem program will issue a DESTROY macro which will be honored by EC.

This form of erasure relates to classified data stored on on-line storage devices such as disk, diskpacks and data cell strips. It does not apply to degaussing of tape.

25X1A



7.11.14.5. Security Communication with the Computer Operator

In Operating System/360, the EC communicates with the operator by means of console devices. When a problem

SECRET

program requires access to a secure auxiliary file, the EC can request the operator to supply the password, which is needed by the program to open the file. This feature will not be employed in CHIVE unless extremely sensitive situations arise.

7.11.14.6. Dumps

When a core dump is requested as the result of an error condition, the SI/TKH/TS Collateral classification will be routinely provided at the top and bottom of the printout. Only the Error Supervisor or error programmer may initiate a dump; problem programmers or Information Analysts may not. The Information Analyst who initiated the job aborted by the error will receive a message alerting him to the problem, but will not receive the problem job output.

Dumps will be treated as all-source material, since the data they contain may be derived from any source. Appropriate logging of the dump will take place. This logging will contain lists of files dumped and other security relevant information.

7.11.14.7. Debug Macro

There will be periods after the system is operational during which programs are tested and debugged. Such

SECURITY PROCEDURES
EDP Software Isolation
7.11.14.7.

SECRET

programs will run in the multiprogramming environment together with fully tested operational programs.

These undebugged programs may employ a Debug macro to provide the programmer with information needed for evaluation and error correction. A Debug macro may also be utilized from time to time if problems should arise in operational programs.

To isolate programs using Debug macros, and to insure that Debug macros do not result in a breach of security, it is necessary to treat the debug output material as an all-source file, and to log the print event.

The printout will contain markings of the security level of the output (that is, all-source) and a routing assignment showing the individual or group responsible for the debugging.

7.11.14.8. Printout of EDP System Processing Files

Request for printout of an EDP system processing file is privileged. A printout of the system library, system tables, the decode dictionary, etc., is restricted to (and needed only by) a CHIVE system programmer or an EDP File Analyst.

SECURITY PROCEDURES
EDP Software Isolation
7.11.14.8.

7.11.14.9. Information Suppression in the Output Phase

Suppression of retrieved material which research analysts are not authorized to have can take place in the search phase or in the output phase. Search logic provides for a search phase suppression. However, it also is necessary to extend the suppression into the output phase.

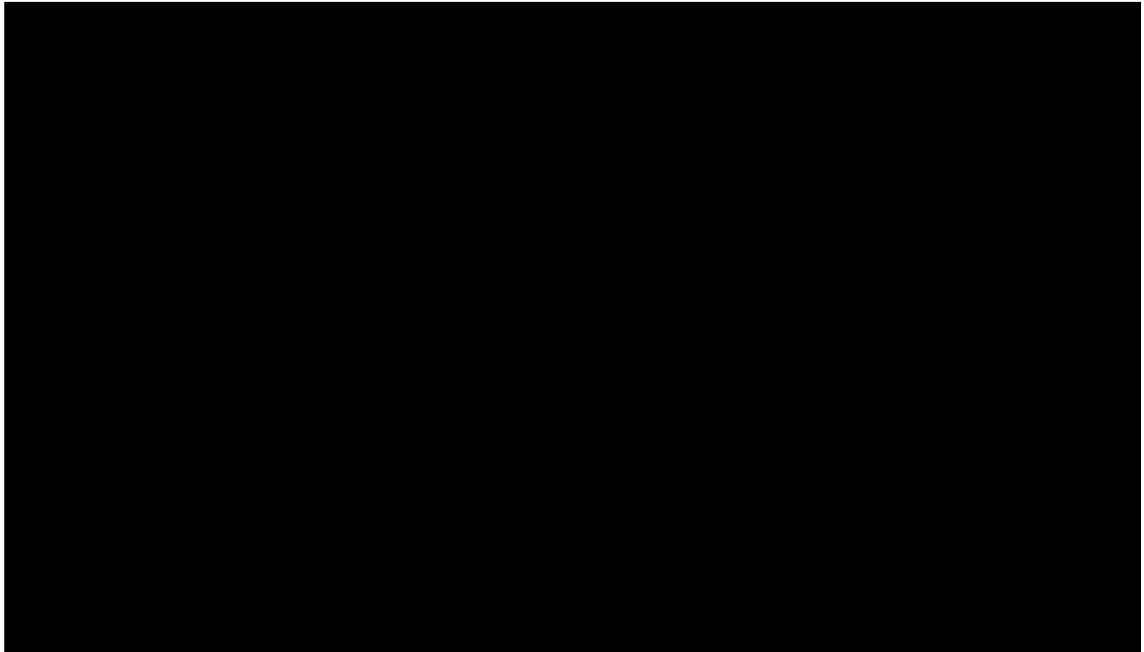
Information may be intentionally suppressed in the output phase for two reasons: (a) a security restriction may be discovered upon decoding which could not be discovered on searching, (b) a heterogeneous file may be screened to produce a report (in report mode) which is at a level beneath that of the file.

An occurrence of either case may result in the suppression of the information record, or part of the record, depending upon the job. The output routines will contain the logic to determine the suppression option, and to implement it.

25X1A

SECURITY PROCEDURES
EDP Software Isolation
7.11.14.10.

25X1A



7.11.14.11. Remote Operations

Transmission of data to a remote I/O station (such as a printer not in the computer operating room) must be accomplished under safeguards of two classes.

Class 1 - Data transmission lines must be secure.

Class 2 - The I/O device must be protected from use by unauthorized personnel, and the device must be made secure against tampering. This security control is achieved by limiting the personnel having access to devices in the computer room and the operational areas. Procedurally, users of consoles will erase displays when finished with them.

SECURITY PROCEDURES
EDP Software Isolation
7.11.14.11.

SECRET

CHIVE EDP system interfaces with other systems must be safeguarded against improper automatic handover of information. For example, if the direct control feature of automatic interruption is employed to mate the page reader with the CHIVE central processor, the EC must inspect the direct control command for legality.

7.11.15 EDP HARDWARE SECURITY

7.11.15.1. Hardware Security Levels

All hardware subsystems will be assigned a "lifetime" security level of ALL SOURCE signifying the all-source jobs or data with which they may be associated.

CHIVE-internal fixed physical storage files used with hardware also will be all-source. Dismountable files sent to external agencies will be classified in accordance with the data they contain.

7.11.15.2. Input/Output Hardware

I/O devices located within the EDP facility are lifetime cleared for all-source data.

A display user will erase (if possible) when he is finished using a device.

SECURITY PROCEDURES
EDP Hardware Security
7.11.15.2.

SECRET

Any I/O devices located outside the computer facility are routed through a patch panel or similar physical or electronic disconnect device, so that they may be physically and/or electronically isolated from the EDP system when not in use. The devices will be manually enabled at the start of a use period, and disenabled afterward.

7.11.15.3. Auxiliary Storage Devices

These are of two kinds: permanent and dismountable.

Permanent storage devices are devices such as disks which are kept on-line most or all of the time. When the recording surface is removed, a modification to the drive is required, or the entire drive itself must be removed. Dismountable devices are designed to facilitate removal of the recording surface, as in the case of a diskpack, tape or cell on a strip file.

The lifetime level of the drive and the recording surface may differ. They may differ when the dismountable surface is routed to an agency other than CHIVE. The lifetime security state of the recording surface of a permanent device is the same as that of the drive.

SECURITY PROCEDURES
EDP Hardware Security
7.11.15.3.

Dismountable devices are color coded in accordance with the security level of the data currently recorded thereon. They also bear a lifetime code, to indicate whether or not they have been released outside of CHIVE.

Such a releasable surface also bears a code to indicate the most sensitive data it has borne since the last complete erasure (in compliance with degaussing and downgrading regulations). It is treated in handling in accordance with the most sensitive of the two following criteria: either the data it now bears or the data it has borne since complete erasure.

They also are coded in a color system compatible with that used by the outside agency. They may be mounted on-line while CHIVE all-source files are on-line. Data from CHIVE may be written directly on these devices.

If data recorded on external files is to be brought into CHIVE, it will be recorded in a CHIVE-compatible file format. That is, File Header Records must be available on the recording surface.

7.11.16. PROCEDURAL SECURITY

7.11.16.1. Definition

Procedural security is defined to encompass security procedures relating to computer control room operations.

SECURITY PROCEDURES
Procedural Security
7.11.16.1.

SECRET

7.11.16.2. Decomartmentalized Operating Concept

All CHIVE computer operating personnel are cleared to run all EDP jobs. The output of every printer or other I/O device in the computer area may be inspected by any operator as part of his assigned duties. All equipment in the control room is open and unscreened. Physical security is considered an adequate barrier to the prevention of unauthorized release of information in the computer facility.

7.11.16.3. Operator Responsibility

The Executive Control is designed to free the computer operators from undue routine burdens, and at the same time to make them an integral part of the system. For example, in Operating System/360 an operator could be asked by EC to supply a password needed to open a secure file.

In addition to their usual responsibilities, operators are expected to monitor the proper operation of the system with respect to security functions. Examining printer output, for example, is a desirable check which operators can make to maintain watch over the system. Insuring that output is routed to the proper recipient is another.

SECURITY PROCEDURES
Procedural Security
7.11.16.3.

SECRET

SECRET

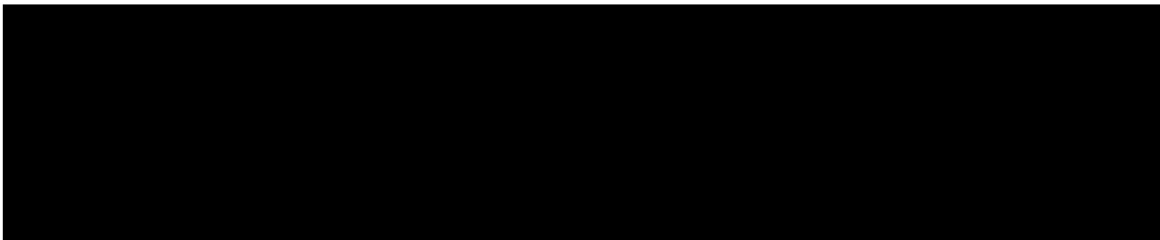
7.11.16.4. Output Routing

All CHIVE output destined for research analysts (that is, all output which leaves CHIVE per se or is exposed to non-CHIVE personnel) must be screened by an information analyst. No request can be written by other than an authorized member of the CHIVE staff.

Some authorized offices within the government will issue standing requirements for information. These will be expressed in standing requests and might provide a routing destination which is printed out together with the answer. When authorized by the Controlling Information Analyst, the operator will use this routing to transmit the copy to the originating office.

Output destined for information analysts will also contain routine information, including the name of the Information Analyst and the name of the research analyst (if any). Responsibility for release rests with the Information Analyst.

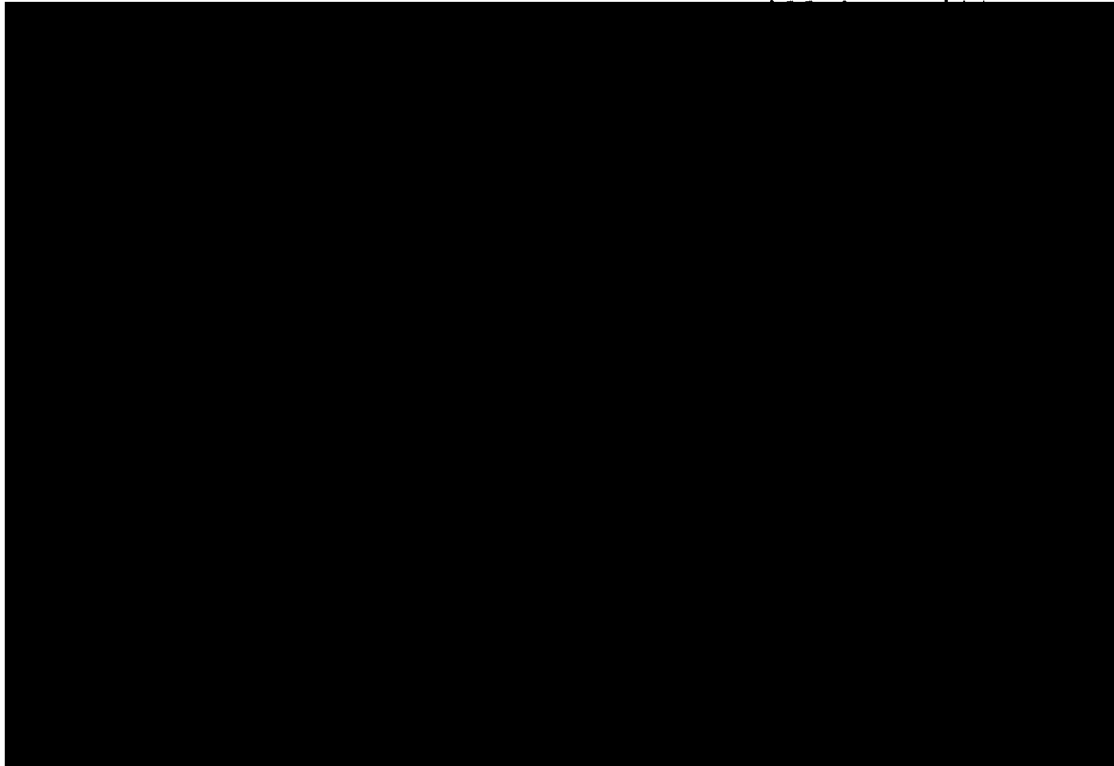
25X1A



SECURITY PROCEDURES
Procedural Security
7.11.16.5.

SECRET

25X1A



7.11.16.6. Environmental Isolation

Environmental isolation refers to security aspects of CHIVE interfaces with other systems, which comprise the environment.

When interfacing systems are steady customers of CHIVE, or when they frequently contribute to the CHIVE data store, a file circulation system will be set up. Storage devices which are sent to customers containing CHIVE data will be returned for reuse.

SECURITY PROCEDURES
Procedural Security
7.11.16.6.

SECRET

7.11.16.7. Printouts

The machine room will retain no copies of printouts. Codeword printouts will not be logged by the machine room. Printouts are logged manually by analysts only if released directly to CHIVE customers and only for TKH and TS Collateral material. Offices originating requests will be responsible for logging the output received from CHIVE. Logging will be performed by the cognizant security office(s).

Information analysts may destroy any unlogged printout without logging the destruction. Printouts are treated as classified waste within CHIVE boundaries.

7.11.16.8. CHIVE Vault

A vault in the computer area is used to safeguard CHIVE machine files. The file library is maintained by a CHIVE librarian, who logs and controls the use of files.

7.11.16.9. CHIVE Courier Service

Couriers are used to transport printouts and documents between the computer room and the service desk(s), which are not behind the same barrier.

SECURITY PROCEDURES
Procedural Security
7.11.16.9.

SECRET

25X1A

7.11.17. COMPUTER FACILITY SECURITY; PREMISES SECURITY

Over-the-counter service will be provided to customers in a secure area separated from the computer area. Desks providing service to one type of customer (that is, a customer cleared for one combination of codeword/collateral) will be isolated from those providing service to another type.

7.11.18. INFORMATION HANDLING FUNCTIONS AND OPERATIONS

7.11.18.1. Reliance on Personnel

Maintaining security control in CHIVE is a joint man/machine undertaking. Screening of output by information analysts is supported by EDP techniques intended to reduce the manual burden.

SECURITY PROCEDURES
Information Handling
7.11.18.1.

7.11.18.2. Information Analyst

Service will be provided to customers who fall into the following classes.

- TKH/SI/Collateral all levels
- SI/Collateral all levels
- Collateral all levels
- Collateral SECRET and below

Dissemination caveats will be observed within each group.

The Information Analyst is expected to determine the customer's classification and need-to-know. He is expected to reflect these items in EDP requests, providing indicators in the request language which the EDP system may use to determine screening requirements. He is expected to exercise options in stipulating the desired output (records, documents, etc.) in accordance with security requirements.

The Information Analyst will also security control the dissemination of microimages, printouts, hard copies of documents, etc. He will often provide information retrieval directly to the customer out of prepared printouts, imposing screening restrictions at the time of service. He is required to keep records of the release of secure information and/or documents.

SECURITY PROCEDURES
Information Handling
7.11.18.2.

7.11.18.3. File Analyst

The File Analyst defines the content and structure of CHIVE EDP files. He also defines content and structure of the security aspects of files, including their homogeneity or heterogeneity, levels or ranges of data, caveats, etc. These are used by the EDP system programs to interpret requests, determine their legality, and to perform checking of various kinds. No file may be defined without a security statement.

The File Analyst verifies the file definitions by examining a machine-produced copy of the File Format Table. This insures that the EDP system has properly accepted the definition.

7.11.18.4. Document Control and Security Officer

A CHIVE officer will be appointed to assist in procedural security. He will exercise control over procedures regarding printouts and documents which are input or output from the EDP system.

In performing CHIVE security functions, the security officer will not represent the cognizant security offices. Instead, he will perform liaison with these offices.

SECURITY PROCEDURES
Information Handling
7.11.18.4.

SECRET

Chapter 7.12.

BACKUP PROCEDURES

7.12.1. BACKUP CONCEPTS

File duplication is the primary means by which CHIVE data or programs are preserved against complete loss. In the EDP system, both data files and system processing files are preserved.

No backup system is foolproof. A backup system seeks a high degree of confidence in protection against complete loss due to handling damage, loss, damage due to program bugs, machine damage, disaster, etc.

Confidence may be achieved by a combination of precautionary methodology and recovery techniques. Manual and automatic means are available to reduce the likelihood of total loss. The tape file protect ring is an instance of manual prevention; EDP file opening legality checking is an instance of automatic prevention.

Recovery consists of maintenance or resumption of normal system operations after information loss due to an extraordinary incident by use of reserve copies of files, by reconstruction of files, or by transformation of data

BACKUP PROCEDURES
Concepts
7.12.1.

SECRET

stored in one form or medium into another form or medium more suitable to the attainment of system goals.

Resources committed for backup are usually limited by the expectation of infrequent system breakdown due to failure. The backup design is a compromise between reduced performance during down periods and the relative frequency of reduced performance. In CHIVE a fall-back concept has been employed. At a time of loss, operations are not crippled although capabilities are reduced. The severe cost burden of striving for full performance capabilities during temporary periods of failure thus is not borne.

In the CHIVE fall-back approach, the chosen temporary cost may consist of some system feature given up entirely, or some degradation in performance, or a combination of these. For example, it may be decided to suspend service to an unimportant customer class, or to provide this service with a longer response time. Instead of using an EDP file to perform a search, a printout may be used.

7.12.2. BACKUP PLAN

There are two versions of a backup capability for CHIVE. Version I is implemented in the initial system. It represents a simplified low-cost solution which will

BACKUP PROCEDURES
Plan
7.12.2.

meet the limited preliminary system needs of the first period. Version II represents the full backup capability which will be implemented gradually. Both versions are an integrated complex of human procedures, EDP software features, and hardware capabilities.

There are two fundamental procedures in the file backup for the Version I plan. First, the entire system is dumped once per month, with the exception of files which have had no activity during the period. Secondly, all transactions for the current month are saved in their original form against a possible loss of the current monthly file.

If lost, the previous monthly file is used in conjunction with the saved transactions to build back another current file. Multiple copies of the dump are available in hard copy form on printouts in case the magnetic form of backup is lost. These copies are dispersed against possible disaster.

Standard office policy is established to determine the handling procedures for duplicate copies. These procedures are developed jointly by OCS and OCR. Some special procedures may be developed for certain classes of file. For example, slowly changing authority files

BACKUP PROCEDURES
Plan
7.12.2.

may be backed up and built back differently than rapidly changing personality summary files. When an information analyst creates a new permanent (not working) file, he is responsible for declaring that the file is a system file of a said class. The procedures for backup are then automatically instituted throughout the CHIVE system for that file.

Hardware backup is achieved in Version I primarily by utilization of the existing OCS computer facility. Storage units (drives) will be shared by the two organizations. Multiple storage units will be available in case of the failure of a single unit. In Period Two (that is, roughly in the second year of operation) CHIVE files will grow to a size which requires a special CHIVE strip file. This file will be backed up by another drive used by OCS. Should a failure occur in either file, the data cells on both can share the operating drive.

One of the most important CHIVE backup techniques is based on the page reader (PR). From a cost-effectiveness point of view, the combination of printout dumps of files and the PR is one of the most attractive forms of backup, due to the low cost of paper as a recording medium and the ease with which multiple hard copies may be made. Hard

BACKUP PROCEDURES
Plan
7.12.2.

copy recording will be used to supplement the usual magnetic tape recording for backup. The first generation copy will be stored on tape; succeeding generations will be stored on printouts.

The Version II plan is a developmental one. As the size of CHIVE files increases, more sophisticated backup techniques will be needed. For example, duplicate copies of incoming index records may be built on-line on different strips of data cell drives.

The phase-over from the initial backup plan to a more sophisticated plan will be gradual. The principal criteria for instituting changes in the initial plan will be (a) reduction of backup costs, (b) increasing operational capability during recovery, (c) increasing confidence that build back capability exists to several generations of file for all forms of loss, (d) responding to increasing performance requirements under increasing loads, (e) achieving additional capabilities.

BACKUP PROCEDURES
Plan
7.12.2.

SECRET

Appendix 7.A.

USER LANGUAGE MANUAL

SECRET

SECRET

Appendix 7.A.

USER LANGUAGE MANUAL

7.A.1. PURPOSE OF MANUAL

Appendix 7.A. contains the preliminary specifications of the CHIVE language for querying and maintaining the CHIVE data base.

Briefly, Appendix 7.A. deals with:

- A description of the logical data structures (CHIVE files, records, phrases, and terms) which can be addressed and manipulated in the language.
- A brief description of the general capabilities of the language from the standpoint of the user (analyst). That is, a general description is given of what the language can do to assist the analyst in the solution of information processing problems.
- A specification of the rules (grammar and syntax) which will guide the analyst in correctly formulating his requests. The specification of rules constitutes a description of how expressions in the language can be formed to perform various EDP tasks.

The rules for specifying retrieval transactions are presented first. These are followed by the rules for file maintenance transactions.

To demonstrate the capabilities of the language and to illustrate the use of the rules, examples are stated

USER LANGUAGE MANUAL
Purpose of Manual
7.A.1.

SECRET

in ordinary English and are followed by a formulation of the problem in the CHIVE language.

7.A.2. CHIVE DATA ELEMENTS AND THEIR LOGICAL STRUCTURE*

The CHIVE language is designed for use with a generalized logical data structure which is user oriented: all CHIVE EDP records are similar in their logical structure to the structure of the document index record as it is developed by the analyst in the document indexing process. Moreover, the user can think in terms of a hierarchic structure of terms, phrases, and records in each file. The structure is identical for a file of document index records and for information files of any functional type (e.g., vocabulary lists, summary files, and unsynthesized files). As far as the user is concerned, the records in all of these files will be serially searched and scanned.

In order to exhibit this logical data structure, several logical levels of data can be distinguished and identified as follows.

*Logical data structures are distinguished from the physical data structures of the system; the latter term refers to the way data is organized and represented internally or "in the machine." The former term denotes the conceptual framework to be used in the analysis and specification of information problems in the CHIVE language.

USER LANGUAGE MANUAL
CHIVE Data Elements
7.A.2.

7.A.2.1. CHIVE Files

A CHIVE file is a named collection of records which can be processed as input to the CHIVE EDP System. This includes intermediate files that are dynamically developed by the user (information analyst) at execution time. These are temporary collections of records which are developed for subsequent processing in the same retrieval transaction.

7.A.2.2. CHIVE Records

Records can be described, first, in terms of two levels within a record: header and content levels; and secondly, in terms of phrase groups and phrases.

7.A.2.2.1. Header and Content Levels

The user can access and manipulate data at each of these levels within the record. The header phrase must include one or more record control terms which uniquely identify each record. For example, an information file might be controlled on PNO* header terms. In this case, each record in this information file would contain information on a unique personality. As another example,

*See Volume V for tag definitions.

USER LANGUAGE MANUAL
CHIVE Data Elements
7.A.2.2.1.

the record control for document index records is a combination of the accession number term and the item number term.

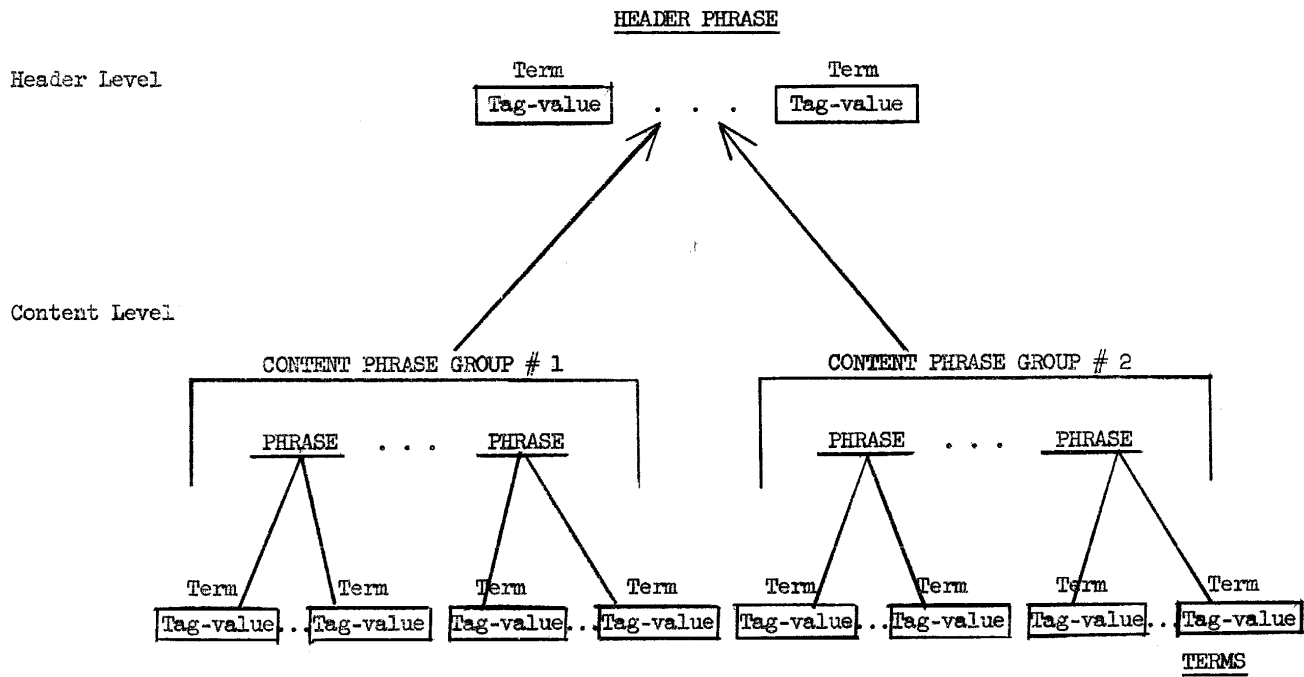
7.A.2.2.2. Phrase Groups and Phrases

The content level of each record is made up of content phrases which belong to phrase groups. Each content phrase logically links together a set of terms into some logically meaningful combinations. Figure 7.A-1 shows a sample CHIVE record with two phrase groups within the content level. A phrase group is made up of one or more content phrases, all of which may contain terms with a set of tags that is unique to that group. Each tag of a particular phrase group may occur in any phrase of that group. However, it cannot occur in a phrase of any other phrase group in the records of a particular file.

Figure 7.A-2 shows in more detail the header and content levels of the above sample CHIVE record. The content level contains phrases belonging to the two phrase groups. Phrase group #1 contains terms with tags CDY, CNC, and CSS. It contains three content phrases defined in this way. Phrase group #2 contains terms

USER LANGUAGE MANUAL
CHIVE Data Elements
7.A.2.2.2.

Figure 7.A-1
Logical Structure of CHIVE Records



SECRET

-196-

SECRET

Figure 7.A-2
SAMPLE CHIVE RECORD

		HEADER		PHRASE		TERMS			
Phrase Group #1	Phrase 1	CDY	Value	CNC	Value	CSS	Value	CNC	Value
	Phrase 2	CDY	Value	CNC	Value	CSS	Value		
	Phrase 3	CDY	Value	CNC	Value	CSS	Value		
Phrase Group #2	Phrase 1	OLC	Value	ONP	Value	ONR	Value		
	Phrase 2	OLC	Value	ONP	Value	ONR	Value		
	Phrase 3	OLC	Value	ONP	Value	ONR	Value		
	Phrase 4	OLC	Value	ONR	Value				

Content Level of CHIVE Record with Two Phrase Groups

SECRET

SECRET

with tags OLC, ONP, and ONR. It contains four content phrases in this way. Notice that the first phrase of phrase group #1 contains two CNC tags. Thus, although a tag cannot occur in more than one phrase group, it may occur several times within an individual phrase of its particular phrase group.

Notice also that in the last phrase of phrase group #2, no ONP tag occurs. Thus, although a phrase group is defined by a particular unique tag set, any tag of that set may or may not occur among the terms of a specific phrase.

It is for the above reasons that the content level of a document index record is said always to contain content phrases of one and only one phrase group. All phrases of a document index record belong to this one phrase group. This phrase group is defined by the set of all the tags in the document index language.

Figure 7.A-3 shows in detail the logical structure of a document index record*. Figure 7.A-4 shows a detail of the logical record structure of an information record.

*Field positions for header information might not, of course, correspond to those given in the examples.

USER LANGUAGE MANUAL
CHIVE Data Elements
7.A.2.2.2.

Figure 7.A-3

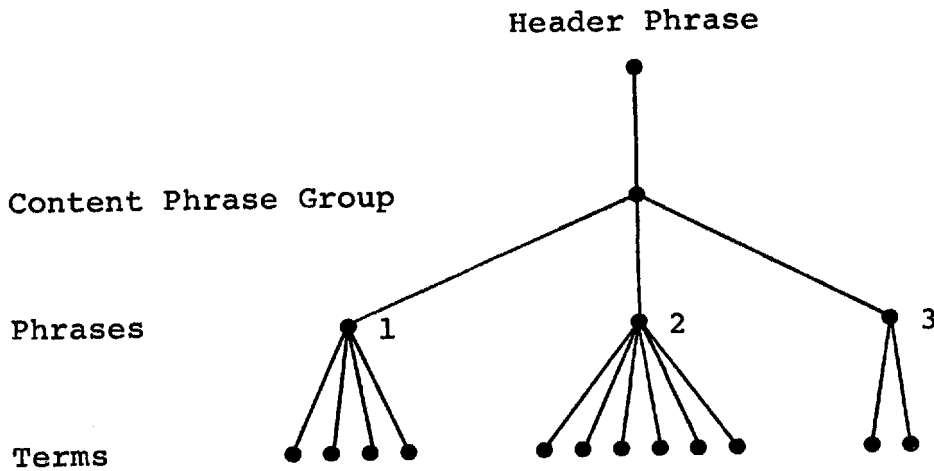
SCHMATIC AND TREE REPRESENTATION
OF
DOCUMENT INDEX RECORD

Record Control Term

Header Phrase	Tag 1	Value	Classification	
Phrase #1	LAC	1393	LAG	3206
	LAR	7971	LAF	67021
Phrase #2	PNO	[REDACTED]	PLC	1013
	PDT	062832	PEO	[REDACTED]
	[REDACTED]	PDD	1201049	
Phrase #3	PDL	090163	PDT	041264
			PA	

25X6
25X6
25X6

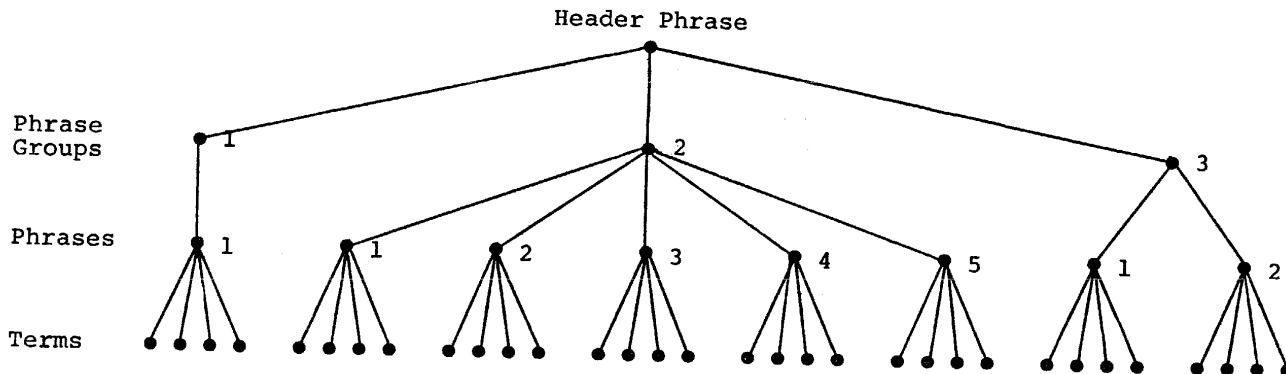
Content Phrase Group



SCHMATIC AND TREE REPRESENTATIONS
OF
INFORMATION RECORD

Header Phrase		Record Control	PNO	[REDACTED]		Classifi- cation	PLP	3449	PDP	0802-	25X6
Phrase Group 1	Phrase #1	PEO	273	PES	29	PED	7	PDE			
	Phrase #1	PFN	3113	PFM	3	PFO	27	PFH			
Phrase Group 2	Phrase #2	PFN	7612	PFM	3	PFO	22	PFH			
	Phrase #3	PFN	0174	PFM	3	PFO	22	PFH			
	Phrase #4	PFN	6920	PFM	3	PFO	22	PFH			
	Phrase #5	PFN	5391	PFM	3	PFO	27	PFH			
Phrase Group 3	Phrase #1	PJP	27	PHT	39	PHR	49	PHN			
	Phrase #2	PJP	39	PHT	94	PHR	77	PHN			

- 200 -



SECRET

Note that a phrase group might contain only one phrase member as in the example of Figure 7.A-4. Here, Phrase Group #1 has as its only member Phrase #1.

7.A.2.2.3. Terms

The lowest logical level of CHIVE data elements is the term. It consists of a tag and a value. The following parts of a term can be examined by the system and can thus be accessed in the language:

- The whole term including tag and value
- The tag alone
- The value alone

7.A.3. GENERAL ORGANIZATION OF CHIVE LANGUAGE

7.A.3.1. Broad Functional Capabilities

The CHIVE user language has broad functional capabilities in the two major areas of file maintenance and information retrieval. Maintenance capabilities consist of the unconditional and conditional modification of the data (tags and values) in the records, phrases or individual terms of CHIVE system data files. Retrieval

USER LANGUAGE MANUAL
General Organization
7.A.3.1.

SECRET

capabilities include:

- Accessing files and searching records to retrieve phrases and terms
- Modifying the retrieved data elements
- Sorting and reordering the records in a file
- Writing new system data files
- Printing output reports containing the retrieved data

The CHIVE language has been designed for use by the information analyst. There are several other important EDP functions performed by the file analyst which are not handled by this language. These functions are generally handled by IBM/360 assembler or compiler languages as described in Volume VII.

7.A.3.2. Job Specifications

The IBM Operating System/360 processes individual input entities called "jobs." Each CHIVE job can consist of one or more maintenance transactions, or one or more retrieval transactions. Following is an outline of the specifications which must be provided with each job:

Maintenance Job Specifications

- Requestor
- Editor
- Classification
- Type of job (retrieval or maintenance)

USER LANGUAGE MANUAL
General Organization
7.A.3.2.

- File Name

- Date

- Priority

Retrieval Job Specifications

- Requestor

- Customer (name and component)

- Classification (overall)

- Type of job (retrieval or maintenance)

- Indication of whether or not File Format Table control is to be exercised

- Date

- Priority

7.A.3.3. Input Coding Sheet

Figure 7.A-5 shows the coding sheet on which the user defines maintenance or retrieval transactions. The coding sheet contains test lines and command lines. In test lines the user specifies test conditions whose true or false results identify those records, phrases, or terms which are to be operated on by the commands of the CHIVE language. In command lines, the user specifies the various types of maintenance and retrieval commands in the CHIVE language. Many of these commands will reference test lines to determine the record, phrases or terms to be operated on.

USER LANGUAGE MANUAL
General Organization
7.A.3.3.

7.A.3.4. Test Conditions

Each test line specifies one logical test condition whose name is given in the TEST column as one of the letters: A, B, C, ..., Y. Each test condition has two possible results--true or false. There are two types of test conditions--simple and complex. A simple test condition is the basic logical expression written on one line. Complex test conditions reference other test conditions (either simple or complex), so that the true or false result of a complex test condition is a function of the true or false result of its constituent test conditions.

7.A.3.4.1. References to Test Conditions in the CRITERIA Field of Commands

The names, (A, B, C, ..., Y), of the test conditions are referenced in the three CRITERIA fields of command lines:

"R" = record level search criteria

"P" = phrase level search criteria

"T" = term level search criteria

CRITERIA test conditions which are true determine the operands of the CHIVE commands for the following purposes (see Table 7.A-1 and Section 7.A.3.5.):

- Determine those records (R), phrases (R and P) or terms (R, P and T) which are to be retrieved from a named input file (COPY, EXTRACT and EXPLODE commands)

- Determine those records (R), phrases (R and P) or terms (R, P and T) which are to be modified by the file maintenance or retrieval hit processing commands (APPEND, DELETE and REPLACE commands)
- Determine those records (R), phrases (R and P) or terms (R, P and T) which are the operands of computational and definition commands (TALLY, ACCUM and DEFINE commands)

The special letter "Z" in an R, P or T Search Criteria field indicates that all the records, phrases and terms respectively which are examined are qualified as operands for the command. "Z" is equivalent logically to specifying the name of a test condition which is always true.

The user can conceptually think that the true or false result of the CRITERIA test condition is evaluated for every record of the input file specified in the command.

The internal EDP processing may be such that the specified CRITERIA test conditions are not evaluated for every record, phrase or term. Indeed the use of devices such as directory files is aimed at avoiding the evaluation of test conditions on all records in a system data file. Nevertheless it is most convenient in using the CHIVE language to imagine that the R, P and T test conditions are exhaustively evaluated as described above.

7.A.3.4.2. Simple Test Conditions

The most common simple test condition is of the

SECRET

following kind:

TEST	TAG	OP	VALUES
A	TAGNAME	1	TERM VALUE

Expressed in ordinary English, simple test condition A is true if the record or phrase currently being examined has a term with the tag "TAGNAME" and has at least one value equal to "TERMVALUE." If no term can be found in the record or phrase with the specified tag and value, then test condition A is false with respect to that record or phrase.

Other types of simple test conditions can be specified by an appropriate entry in the OP field as explained in Section 7.A.4.2. These include other relational tests besides the equality test described above. For example, does the record or phrase currently being examined have a term with the tag "TAGNAME" and with a value that is less than or greater than the value "TERMVALUE" specified in the test condition. A span test condition would have two values (TERMVALUE 1 and TERMVALUE 2) specified. It would be true if the record or phrase currently being examined has a term with the tag "TAGNAME," and has a value which is

USER LANGUAGE MANUAL
 General Organization
 7.A.3.4.2.

SECRET

greater than the first test condition value TERMVALUE 1 but less than the second value TERMVALUE 2. A scan test condition would specify a combination of required characters and "don't care" characters in the VALUE field. A scan test would be true if the record or phrase currently being examined has a term with the tag "TAGNAME" and has a value which contains the specified string of required characters and don't care characters.

7.A.3.4.3. Complex Test Conditions

The true or false result of a complex test condition is a logical function of the true or false results of specified simple test conditions. Consider for example the following complex test condition:

TEST	TAG	OP	VALUES
H		3	G, F, Q, R

Expressed in ordinary English, complex test condition H is true if at least 3 of the four test conditions (G, F, Q, and R) are true anywhere in the record currently being examined. These four test conditions may be either simple test conditions or other complex test conditions. In general, the OP field of a complex test

USER LANGUAGE MANUAL
 General Organization
 7.A.3.4.3.

SECRET

condition contains a number M, while the VALUES field specifies the names of N test conditions. The complex test condition is true if M or more of the specified N test conditions are true. Section 7.A.4.3 describes complex test conditions in more detail.

7.A.3.4.4. Phrase Coordination of Test Conditions

The special asterisk character "*" can be inserted in the VALUES field of a complex test condition as shown below to indicate that the M of N test conditions must be true in the same phrase for each record or phrase being examined.

TEST	TAG	OP	VALUES
K		3	* G, F, Q, R

Expressed in ordinary English, complex test condition K is true only if at least 3 of the four test conditions (G, F, Q and R) are true for a given phrase in the record currently being examined. Test condition K must therefore logically be evaluated for each phrase in the record. Without the "*" test condition K would be evaluated only once across the entire record, with no

USER LANGUAGE MANUAL
 General Organization
 7.A.3.4.4.

SECRET

regard for phrase linkage. Test condition K is obviously more stringent than H, since H is true if three or more of the test conditions G, F, Q, and R are true anywhere in the record, and not necessarily in a single phrase.

7.A.3.5. Command Lines

Command lines contain the following eight fields:

COMMAND	LIMIT	CRITERIA	FILES	DATA
		R P T	IN OUT	

Table 7.A-1 summarizes the manner in which these fields must be coded for each command. There are three possibilities:

- "yes" indicates that the appropriate operand(s) must be entered in the field
- "no" indicates that no entry is required. Fields with "no" will simply be ignored
- "opt" indicates that the field is optional. A blank field is therefore as significant as a non-blank entry.

7.A.3.5.1. Summary Description of Command Lines

Commands are generally performed on a named Input File. The data elements operated on are specified by

USER LANGUAGE MANUAL
 General Organization
 7.A.3.5.1.

Table 7.A-1

Summary of Rules for COMMAND Lines

COMMAND	LIMIT	CRITERIA			FILES		DATA	
		R	P	T	IN	OUT		
COPY	Opt	yes	no	no	yes	yes	no	
	Opt	yes	yes	no	yes	yes	no	
		yes	yes	yes	yes	yes	yes	no
EXPLODE	Opt	yes	yes	no	yes	yes	no	
	Opt	yes	yes	yes	yes	yes	no	
EXTRACT	Opt	yes	no	no	yes	no	no	
	Opt	yes	yes	no	yes	no	no	
	Opt	yes	yes	yes	yes	no	no	
APPEND	Opt	yes	no	no	no	no	yes	Terms or \$TALLY n \$COMPUTE n \$ACCUM n \$DEFINE n
	Opt	yes	yes	no	no	no	yes	
REPLACE	Opt	yes	yes	no	no	no	yes	
	Opt	yes	yes	yes	no	no	yes	
DELETE	Opt	yes	no	no		no	no	
		yes	yes	no	no	no	no	
		yes	yes	yes	no	no	no	
TALLY n	Opt	yes	no	no	yes	no	no	
	Opt	yes	yes	no	yes	no	no	
	Opt	yes	yes	yes	yes	no	no	
DEFINE n	no	yes	yes	no	yes	no	no	
		yes	yes	yes	yes	no	no	
PRINT	no	no	no	no	yes	yes	yes	Sort key; report parameters
ACCUM n	Opt	yes	yes	yes	yes	yes	yes	\$TALLY n \$ACCUM n \$COMPUTE n
MERGER	Opt	no	no	no	yes	yes	no	
MERGE P	Opt	no	no	no	no	no	yes	
WRITE P	Opt	no	no	no	no	yes	no	
WRITER	OPT	no	no	no	no	yes	no	
COMPUTE n	Opt	no	no	no	yes	no	yes	

entering test condition names in the following Search Criteria fields:

- Records (R field only)
- Phrases (R and P fields)
- Terms (R, P and T fields)

A number may be placed in the Limit field which specifies the maximum number of times that the command is performed on the particular data elements of the input file. The results of the command will generally be written on a named Output File. Some commands require the specification of additional operands or parameters in the Data field.

7.A.3.6. FFT Control Concept

A retrieval job can be conducted in two basic modes:

- Under File Format Table control
- Without File Format Table control

FFT control is the normal mode. Under FFT control the Test Conditions, and the Value Fields of Commands can only reference tags which have been defined in the File Format Table of the named Input File. In addition the APPEND, DELETE and REPLACE commands are subject to certain restrictions under FFT control as described in Section 7.A.4.7.

USER LANGUAGE MANUAL
General Organization
7.A.3.6.

When a retrieval job is not under FFT control, any arbitrary tags can be referenced in the Test Conditions or the Data Fields of Commands. Most of the well defined restraints imposed by the FFT are relaxed.

7.A.4. LANGUAGE SPECIFICATIONS

7.A.4.1. Summary of Retrieval Commands

Many retrieval transactions require only the two basic retrieval commands--COPY and PRINT. The COPY command copies those records, phrases or terms which meet the search criteria. These data elements are copied from a named Input File. The PRINT command prints a named Input File (generally the COPY Output File) according to a named output report format. The EXPLODE command is similar to the COPY Command except that each phrase or term in the named Input File which meets the specified Search Criteria is copied as an individual record on the named Output File.

The COPY command copies retrieved data elements directly into a named output file. When it is necessary to edit or modify hit data elements, the EXTRACT and WRITE combination must be used. Included between these two commands are four possible commands--APPEND, DELETE,

REPLACE and MERGEP which modify the retrieved data.

The EXTRACT command extracts records, phrases and terms which meet the search criteria into a temporary work area. This extracting occurs one record at a time. A succeeding WRITE command will write out the modified contents of the work area into a named Output File. The APPEND command appends data which appears in the Data Field to the phrase(s) or term(s) in the work area which meet the Search Criteria. The DELETE command deletes data elements which meet the specified Search Criteria from the record in the temporary work area. The REPLACE command replaces the data elements in the work area record which meet the Search Criteria with data specified in the Value Field.

There are three computational commands--TALLY, ACCUM and COMPUTE--which perform computational operations on CHIVE data elements. TALLY counts the number of records, phrases or terms in a named Input File which meet the specified Search Criteria. ACCUM accumulates the sum of the numeric valued terms in a named Input File which meet the specified Search Criteria. COMPUTE performs arithmetic operations (add, subtract, multiply and divide) on CHIVE data elements and on user-defined values.

USER LANGUAGE MANUAL
Language Specifications
7.A.4.1.

SECRET

The MERGER command causes all records in a named input file, which contain a common term(s) to be physically co-located in a named output file. The MERGER command must be preceded by an EXTRACT command and hence it operates on the single extracted record in the temporary work area. All phrases within the record with a common value for a specified merge tag are merged into a single new phrase.

7.A.4.2. Simple Test Lines

A simple test line is made up of the following entries:

- TEST
- TAG
- OP
- VALUE(S)

7.A.4.2.1. Test Entry

Each test line is identified by letter names: A, B, C..., Y. Whenever a test condition is referenced in subsequent test or command lines, the letter name of the test line is used.

USER LANGUAGE MANUAL
Language Specifications
7.A.4.2.1.

SECRET

7.A.4.2.2. TAG Entry

This entry designates the tag or system value being tested. It can be any of the following:

- any tag defined in the File Format Table of the system data file. For the Document Index File these will be the tags of the indexing language.
- an arbitrary tag which is defined in the current job and which is referenced in an editing or computational command (when files are not under FFT control).
- A system generated value:

These include DEFINE n; TALLY n; ACCUM n.
See Sections 7.A.4.7.5., 7.A.4.8.1. and 7.A.4.8.2.

7.A.4.2.3. OP Entry

This entry can be either a numeric operator or a relational operator.

A numeric operator specifies the least number of values which must be found for the test to be satisfied. Also, if more than this number is found, the condition is met.

When a '1' appears with no value entry, the condition is counted as met if the tag specified in the tag entry is found with any value. Figure 7.A-7 shows the use of the OP numeric operator.

USER LANGUAGE MANUAL
Language Specifications
7.A.4.2.3.

TEST	TAG	OP	VALUE(S)
A	PNO	1	2341
B	PEO	2	271 & 278 & 279
C	CNC	2	29 & 30
D	CNC	1	

Figure 7.A-7

The Numeric OP Entry

Test line A - this simple test is satisfied if a PNO term with the value specified is found.

Test line B - this simple test is satisfied if PNO terms with two of the three values (271, 278 or 279) are found.

Test line C - this simple test is satisfied if CNC terms with both of the values are found.

Test line D - this simple test is satisfied if any CNC term is found.

Several relational operators may be entered. The values may be numeric or alphanumeric characters (see Figure 7.A.8.).

USER LANGUAGE MANUAL
Language Specifications
7.A.4.2.3.

TEST	TAG	OP	VALUE(S)
A	PDB	>	310705
B	PDN	<	290628
C	PDB	×	280401 & 300101
D	PNO	>	██████████
E	POH	N	DIRECTOR

25X6

Figure 7.A-8

The Relational OP Entry

> (greater than) - The simple test condition A is satisfied if a PDB term is found with a value greater than July 5, 1931.

< (less than) - The simple test condition B is satisfied if a PDN term is found with a value less than June 28, 1929.

X (span) - The simple test condition C is satisfied if a PDB term is found with a value greater than April 1, 1928 and less than January 1, 1930. Values must be separated by &. Only one & may appear on a line with this operator.

USER LANGUAGE MANUAL
 Language Specifications
 7.A.4.2.3.

The simple test condition D is satisfied if a PNO term is found whose value is greater than [REDACTED] 25X6 in the collating sequence of the IBM/360.

N (negation) - The simple test condition E is satisfied if a POH term is found with a value other than "DIRECTOR." (If no POH tag appears, or if it appears with "DIRECTOR," the condition is not satisfied.) Thus, this operator requires that (a) the tag in the tag entry be present, (b) no value for that tag be equal to the value entered.

7.A.4.2.4. Value(s) Entry

Each value for which a test is made is entered here separated by &. One blank must appear on each side of the &. Additional blanks are treated as part of the value. Values can include the characters A-Z, 0-9 and CHIVE special symbols as described herein.

Other types of test conditions can be specified by appropriate special symbols in the value entry:

- Phrase coordination (*)
- Don't care characters (?, !)
- Tag occurrence (#)

USER LANGUAGE MANUAL
Language Specifications
7.A.4.2.4.

- Indirect value specification (\$ FILE)

Figure 7.A-19 in Section 7.A.4.3.5. shows the permissible combinations of the special value symbols with various operator field entries.

7.A.4.2.4.1. Phrase Coordination

Phrase coordination occurs whenever the values specified must be found or satisfied within the same phrase rather than merely within an entire record. The values must be preceded by an '*' (see line A, Figure 7.A-9).

When the special symbol, '*', is used in a simple test, the test line should specify that more than one value be matched. Lines B and C are not meaningful uses of the '*'.

TEST	TAG	OP	VALUE(S)
A	PNO	2	* 2174 & 2175
B	PNO	1	* 2174
C	PNO	1	* 2174 & 2175
D	PNO	2	* 2175 & 2176 & 2177

Figure 7.A-9

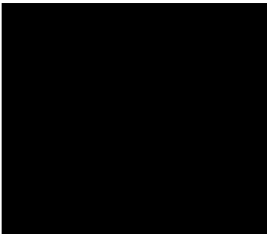
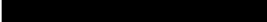
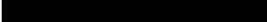
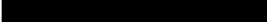
Value Entries with Phrase Coordination

USER LANGUAGE MANUAL
 Language Specifications
 7.A.4.2.4.1.

7.A.4.2.4.2. Don't Care Characters

There are two "don't care" characters available:

- ? says to ignore the single character appearing in the corresponding position in the value of the record term currently being examined.
- ! says to ignore a variable number of characters in the corresponding position in the value of a record term currently being examined. This ignoring stops when the character following the ! is encountered in the record term value.

TEST	TAG	OP	VALUE(S)
A	PNO	1	
B	PDB	1	
C	PNO	1	
D	LAC	1	
E	LAC	1	1??3!
F	PDT	X	??0564 & ??0665

25X6

Figure 7.A-10

Value Entries with DON'T CARE Characters

The test conditions in Figure 7.A-10 are satisfied i.e., they are true, as follows:

USER LANGUAGE MANUAL
 Language Specifications
 7.A.4.2.4.2.

SECRET

Test condition A--All characters, except the character in the position corresponding to the ?, match exactly.

Test condition B--All characters, except the four characters whose positions correspond with the four ?'s match exactly. There must be exactly six characters in the value being tested.

Test condition C--The first eight characters must match [REDACTED]. An exact match on the characters which follow [REDACTED] is not required. Any number of characters may follow [REDACTED].

Test condition D--The first three characters must match [REDACTED]. Any single character may follow [REDACTED] and there must be at least one such character. This character may be followed by any number of character(s), or by no characters at all. The last four characters must match PING exactly.

Test condition E--The first character must be a numeric 1 followed by any two (but only two) characters. The fourth character must be a numeric 3. The 3 may be followed by any number of any characters or by no characters at all.

USER LANGUAGE MANUAL
Language Specifications
7.A.4.2.4.2.

Test condition F--The values for the PDT tested must be such that (a) the last four characters are greater than 0564, and (b) the last four characters of the same value are less than 0665.

The don't care characters may not have an alphanumeric character on both sides when used with <, >, and X operators.

7.A.4.2.4.3. Tag Occurrence Symbol

The actual number of occurrences of the tag specified in the TAG entry must satisfy the relational condition specified in the OP entry with regard to the number "n" following the # symbol.

Examples of the tag occurrence symbol, "#" are shown in Figure 7.A-11.

TEST	TAG	OP	VALUE(S)
A	PNO	>	#1
B	PNO	<	#15
C	PNO	1	#4
D	PNO	N	#3
E	PNO	1	#3

Figure 7.A-11

Value Entries with Tag Occurrence Symbol

USER LANGUAGE MANUAL
Language Specifications
7.A.4.2.4.3.

- Test condition A--PNO must occur more than once.
- Test condition B--PNO must occur less than 15 times.
- Test condition C--PNO must occur exactly four times.
- Test condition D--PNO must occur any number of times other than three times
- Test condition E--PNO must occur three times in the same phrase.

7.A.4.2.4.4. \$ FILE Indirect Specification of Values

The special symbols, \$ FILE, specify that the characters immediately following these constitute the name of a file. The actual values will be obtained from the records of this file.

Examples of \$ FILE (name) are shown in Figure 7.A-13. This feature is used in enriched searching and is more fully explained in Section 7.A.5.1.

TEST	TAG	OP	VALUE(S)
A	PNO	1	\$ FILE OUT 1
B	PDB	>	\$ FILE OUT 2

Figure 7.A-13

Value Entry with \$ FILE (name)

USER LANGUAGE MANUAL
 Language Specifications
 7.A.4.2.4.4.

Test condition A--is satisfied if the record under consideration has at least one value equal to any of the PNO terms in File OUT 1.

Test condition B--is satisfied if the record under consideration has a PDB value greater than any one of the PDB values in File OUT 2.

Any file referenced by \$ FILE (name) in the VALUE field must exist in the system or must have been previously generated for the current transaction.

7.A.4.3. Complex Test Lines

A complex test line specifies a test condition made up of simple test conditions. It has no TAG entry. Value entries are always alphabetic names (A,B,...) of previously specified test lines.

7.A.4.3.1. Example of a Complex Test Line

25X6 [REDACTED] 25X6
25X6 [REDACTED] with an LAC of 2314 or 2715.

USER LANGUAGE MANUAL
Language Specifications
7.A.4.3.1.

TEST	TAG	OP	VALUE (S)	
A	PNO	2	[REDACTED]	25X6
B	LAC	1	2314 & 2715	
C		2	A & B	

Figure 7.A-14

Test Line (C) Made Up of Test Lines (A,B)

Test Line C is a complex test statement made up of the simple test statements on lines A and B. The '2' in the OP field indicates that both test A and test B must be true to satisfy test line C.

The Boolean expression equivalent to Test line C is: [REDACTED] AND (LAC 2314 or LAC 2715).

25X6

7.A.4.3.2. Phrase Coordination in a Complex Test Line

The phrase coordination symbol '*' may also be used as a value entry preceding the test line names.

As an example: identify the Organization Head for the [REDACTED]. The organization type number is 6700. Figure 7.A-15. shows the necessary entries.

25X6

USER LANGUAGE MANUAL
 Language Specifications
 7.A.4.3.2.

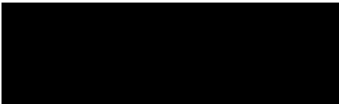

TEST	TAG	OP	VALUES
A	LAC	1	 25X6
B	ONO	1	
C	OTF	1	6700
D	POH	1	
E		4	*A & B & C & D

Figure 7.A-15

Phrase Coordination in a COMPLEX Test Line

Since the example calls for the occurrence of any POH in a phrase with conditions A, B, C, no value for line D is given and the numeric operator is a '1.' The condition defined on Line E, preceded by an * requires that the 4 conditions (A, B, C and D) be met within a phrase.

No complex test line can contain an * if it references a preceding test line which also contains an * (see Figure 7.A-16.).

USER LANGUAGE MANUAL
Language Specifications
7.A.4.3.2.

PERMISSABLE TEST LINES (Q & C)	TEST	TAG	OP	VALUES
		Q	PEO	2

PERMISSABLE TEST LINES (Q & C)	TEST	TAG	OP	VALUES
	A	PEO	2	27 & 28
	B	PNO	1	2784, 2794
	C			* A & B

ERRONEOUS TEST LINE (C)	TEST	TAG	OP	VALUE (S)
	A	PEO	2	*27 & 8 & 9
	B	PNO	2	2784, 2794
	C		2	* A & B

Figure 7.A-16

The Uses of '*' for Phrase Coordination

7.A.4.3.3. M of N Logic in a Complex Test Condition

Often, it may be useful to ask that at least two of three; or, two of four of the test conditions named be satisfied in a phrase or record.

As an example the user wishes to identify a conference attended by at least two, and possibly more than two of the PNO's listed on test line B. The conference date is August 31, 1964. Figure 7.A-17 shows the necessary test conditions.

TEST	TAG	OP	VALUE(S)
A	CDY	1	640831
B	PNO	2	2314 & 3124 & 4675 & 7124
C	CNC	1	
D		3	* A & B & C

Figure 7.A-17

Test Lines with M of N Logic

The entry in the OP column for line B is 2, indicating that the condition is met if at least two (and possibly more) of the PNO values appear.

7.A.4.3.4. Span Operator in a Complex Test Condition

The following example shows how a test line with the span operator is referenced in a complex test condition. Determine the phrases mentioning all PNO's who attended PEO 23, PEO 26, and PEO 28 during the first six months of the year 1932.

TEST	TAG	OP	VALUE
A	PEO	1	23, 26, 28
B	PNO	1	
C	PDE	X	311231 & 320701
D		3	* A & B & D

Figure 7.A-18

Example of Use of X Operator

7.A.4.3.5. Summary of Rules for Test Lines

The following matrix shows the permissible combinations of special symbols with Operator field entries.

Symbols	Operators:					
	X	N	1	Other number		
&	no	no	yes	no	yes	yes
?	opt	opt	opt	opt	opt	opt
!	no	no	no	opt	opt	opt
\$	opt	opt	no	opt	opt	no
#	opt	opt	opt	no	opt	no
*	no	no	no	no	no	yes

Figure 7.A-19

Summary of Test Line Rules

7.A.4.4. Basic Retrieval Commands: COPY, PRINT

Most retrieval transactions will require the use of only the two commands--COPY and PRINT.

COMMAND	LIMIT	CRITERIA			FILE		DATA
		R	P	T	IN	OUT	
COPY	(Opt)	v	v	v	v	v	
PRINT	(Opt)	-	-	-	v	v	v

Figure 7.A-20

COPY and PRINT Command Formats

7.A.4.4.1. COPY Command

The COPY command copies those records, phrases, or terms which meet the search criteria. Section 7.A.4.5. discusses the search CRITERIA entries in more detail and should be read in conjunction with the following description of the COPY and PRINT commands.

Rules:

1. Input and output files must be named.
2. Each record of the input file is examined to determine whether or not it contains a data element that meets the search criteria.
3. Record, phrase, and term criteria entries are required as follows:
 - To copy a term: record, phrase, and term (R, P, T) criteria entries are required.

- To copy a phrase: phrase and term (P, T) criteria entries are required.
- To copy a record: only the record (R) criterion entry is required.

4. The LIMIT entry is optional

5. No DATA entry is used.

The COPY command does not retrieve data into a temporary work area for editing or for computation. To perform post-retrieval operations other than printing, the EXTRACT command should be used.

The COPY command copies the data elements (records, phrases, or terms) specified by appropriate use of the Search Criteria entries. Each data element (R, P, or T) of the named input file that is specified as a Criteria entry is examined to determine that it meets the condition named by the Search Criteria entries. If all test conditions, so named are met for each data element specified, the data element of the lowest level specified, i.e., record (R), phrase (P), or term (T) is copied in the named output file as a record. Figure 7.A-21 shows examples of Search Criteria at the three levels.

USER LANGUAGE MANUAL
Language Specifications
7.A.4.4.1.

Figure 7.A-21

COPY and PRINT Command Lines

Record Level Search Criteria							
COMMAND	LIMIT	CRITERIA			FILE		DATA
		R	P	T	IN	OUT	
COPY		A			DIF	OUT 1	
PRINT					OUT 1	RCD	

EXAMPLE A

Phrase Level Search Criteria							
COMMAND	LIMIT	CRITERIA			FILE		DATA
		R	P	T	IN	OUT	
COPY	50	A	B		DIF	OUT 1	
PRINT					OUT 1	PHR	PNO

EXAMPLE B

Term Level Search Criteria							
COMMAND	LIMIT	CRITERIA			FILE		DATA
		R	P	T	IN	OUT	
COPY		Z	B	C	DIF	OUT	
PRINT					OUT	TER	PEO & PNO

EXAMPLE C

EXAMPLE A--COPY records satisfying the condition on test line A.

EXAMPLE B--COPY phrases satisfying the condition on test line B from records satisfying condition on test line A. The LIMIT entry limits the retrieval to a maximum of 50 hit phrases for output.

EXAMPLE C--COPY terms satisfying condition on test line C in those phrases satisfying condition on test line B. The Z entered for record criteria entry (R) indicates that phrases and terms from all records are examined for the test conditions in the phrase (P) and term (T) Criteria entries. (A test line could also be referenced in the R field if it is needed.)

7.A.4.4.2.PRINT COMMAND

The PRINT command prints a named Input File. PRINT designates a standard output format in the Output File entry. This command, in conjunction with the COPY command, constitute the basic retrieval command set.

Rules:

1. An Input File name must be specified.
2. The standard output format must be named as an OUT FILE entry.

USER LANGUAGE MANUAL
Language Specifications
7.A.4.4.2.

3. Sort keys can appear as DATA entries. Where multiple sort keys are specified, each key is separated by an &.

4. If there is more than one occurrence of a sort key in a record, that record is repeated in the output listing for each occurrence of individual sort key values.

5. The DATA field can contain additional parameters required for the named output format.

As an example, COPY and PRINT all records mentioning [REDACTED] in the same phrase. The necessary entries are shown in Figure 7.A-22 and 7.A-23.

25X6

TEST	TAG	OP	VALUE(S)
A	PNO	2	[REDACTED]
B	LAC	1	2314 & 2715
C			* A & B

25X6

Figure 7.A-22 - Test Line

COMMAND	LIMIT	CRITERIA			FILES		DATA
		R	P	T	IN	OUT	
COPY			C		DIF		OUT 1
PRINT					OUT 1		RCD

Figure 7.A-23

COMMAND lines for COPY and PRINT

SECRET

Figure 7.A-21 also shows several uses of the PRINT command. The first example shows the use of the PRINT command to print hit records. The named input file is OUT 1. The OUT FILE entry RCD, names a standard report format used for printing records. The second example shows the use of the PRINT command to print phrases. The OUT FILE entry,PHR, names a standard report format to print phrases. The DATA entry contains the sort key PNO. Major and intermediate sort keys, PEO and PNO, are shown in the third PRINT example of Figure 7.A-21.

7.A.4.5. Search CRITERIA and DEFINE Command

Figure 7.A-24 shows additional search criteria entries used in a COPY command depending upon the data element desired.

COMMAND	LIMIT	CRITERIA			FILES		DATA
		R	P	T	IN	OUT	
# 1 COPY		A			DIF	OUT 1	
# 2 COPY		A	B		DIF	OUT 2	
# 3 COPY		Z	B	C	DIF	OUT 3	

Figure 7.A-24

CRITERIA entries for Various Data Elements

Line # 1--COPY all records satisfying test condition on test line A.

USER LANGUAGE MANUAL
Language Specifications
7.A.4.5.

SECRET

Line # 2--COPY all phrases satisfying conditions on test line B from records satisfying conditions on test line A.

Line # 3--COPY all terms satisfying the condition defined on test line C in all phrases satisfying the condition on test line B. The Z entry in the record level (R) entry indicates that the terms from all records satisfying both the term and phrase level criteria are to be copied.

7.A.4.5.1. Example of Complex Search Criteria

"Copy all phrases with references to [REDACTED]

25X6

25X6

[REDACTED] if they occur in records referring to both of these personalities and a PAF reception somewhere in the

25X6

[REDACTED] between

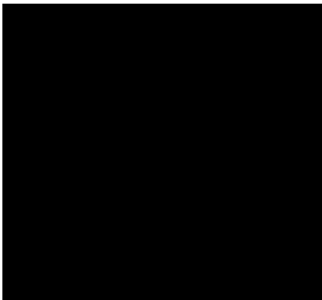
July 1955 and June 1964." The first figure, Figure 7.A-25, shows the CRITERIA entries necessary to copy the phrases on the two personalities. The CRITERIA phrase (P) entry is C.

Notice the complex test lines which refer to simple test lines: line C refers to A and B; line F refers to D and E. Notice, also, that line I is a complex test line that refers to both simple test lines, G, H, and other previously defined complex lines, C, F.

USER LANGUAGE MANUAL
Language Specifications
7.A.4.5.1.

Figure 7.A-25

Use of CRITERIA Entry

TEST	TAG	OP	VALUE (S)
A	PNO	1	
B	PNO	1	
C		2	
D	LAC	1	
E	LAC	1	
F		1	D & E
G	HDR	X	5507?? & 6406??
H	PAF	1	!RECEPTION!
I		4	C & F & G & H

25X6

COMMAND	LIMIT	CRITERIA			FILES		DATA
		R	P	T	IN	OUT	
COPY		I	C		DIF	OUT 1	
PRINT					OUT 1	PHR	

COMMAND	LIMIT	CRITERIA			FILES		DATA
		R	P	T	IN	OUT	
COPY		I	H		DIF	OUT	
PRINT					OUT 1	PHR	

Figure 7.A-26

Alternative use of CRITERIA Entry

Figure 7.A-26, shows the search CRITERIA entries for a problem similar to the example given above. However, in this example there is a change in the phrases to be copied. Previously, phrases containing the personalities named were copied. Here, the phrases containing a PAF of "reception" are to be copied. Thus, the phrase (P) entry is H. It is important to realize that in both examples, although the phrases copied differ, they are copied from records that satisfy identical test conditions. Accordingly, the record CRITERIA entry for both examples is I. (The test lines for both examples are identical.)

7.A.4.5.2. Define (m)

A data element (record or phrase) specified with appropriate CRITERIA entries can be uniquely identified by use of the DEFINE command.

USER LANGUAGE MANUAL
 Language Specifications
 7.A.4.5.2.

COMMAND	LIMIT	CRITERIA			FILE	DATA
		R	P	T	IN	
DEFINE 1		Z	E		File 1	
EXTRACT		Z			File 1	
APPEND		Z	G			\$ DEFINE 1
WRITEP						OUT 1

Figure 7.A-26A

DEFINE Command Line

The DEFINE command uniquely defines the data element specified in the CRITERIA entries for this command.

Rules:

1. CRITERIA entries must be specified as in EXTRACT and COPY commands.
2. The LIMIT entry is optional.
3. An Input File name must be specified.
4. "DEFINE" in the COMMAND entry is the name being assigned to the data element specified in the CRITERIA entries.

Subsequent references to this data element appear as DATA entries "DEFINE." In any such reference, the "DEFINE" name must be used.

5. Each DEFINE named must be unique for a retrieval transaction.

SECRET

6. A DEFINE name may be referenced as a Tag field entry in a test line by its assigned name.

Explanation

Those phrases which meet the condition specified on test line E and which occur in the named input file are named DEFINE 1 (see Figure 7.A-26A). Since the APPEND command has as its DATA operand "\$DEFINE 1," the phrases meeting the search CRITERIA for the DEFINE 1 command are appended to records extracted by the succeeding EXTRACT command in the example.

7.A.4.6. EXTRACT and WRITE

The COPY command copies retrieved data elements directly into the named output file. When it is necessary to edit or modify hit elements, the EXTRACT-WRITE combination must be used.

7.A.4.6.1. EXTRACT

Data elements specified in the CRITERIA entries are extracted into a work area for access and manipulation by subsequent commands.

USER LANGUAGE MANUAL
Language Specifications
7.A.4.6.1.

SECRET

COMMAND	LIMIT	CRITERIA			FILE		DATA
		R	P	T	IN	OUT	
TALLY 1		v	v				
EXTRACT	(opt)	v	v	v	DIF		
APPEND		v	v				\$TALLY1
WRITER						OUT 2	
PRINT					OUT 2	PHR	sort order key

Figure 7.A-27

Example of EXTRACT Command

Rules:

1. All search CRITERIA entries are required as in the COPY command.
2. An Input File must be named. No output file name is used.
3. The EXTRACT command may be followed only by APPEND, DELETE, REPLACE, MERGEP, WRITEP, or WRITER commands.
4. All EXTRACT commands within the scope of the same WRITER and WRITEP commands (see these commands below) must have the same input file names.
5. The LIMIT entry is optional.

Explanation

Phrases or terms which meet the search CRITERIA are extracted from a particular input record into a temporary work area as a single record. Hit records are similarly

copied into the work area for subsequent editing, computation, etc.

A succeeding WRITER or WRITEP command is necessary to copy what has been extracted into a named output file.

7.A.4.6.2. WRITER and WRITEP Commands

These commands write the contents of a work area containing extracted data elements into a named output file. The WRITER command writes out the contents of the work area as an individual record. The WRITEP command writes each phrase of a record from the work area as an individual record.

Rules: Figure 7.A-28 shows examples of proper and improper application of the following rules:

- An output file must be named by the WRITE command.
- Each WRITE command must be preceded by at least one EXTRACT command.
- No other commands may intervene between a WRITE and its preceding EXTRACT except: APPEND, DELETE, REPLACE, and MERGEP.
- The command sequence EXTRACT. . .EXTRACT. . . WRITE. . .WRITE is not permissible.

USER LANGUAGE MANUAL
Language Specifications
7.A.4.6.2.

permissible	EXTRACT APPEND WRITER
permissible	EXTRACT APPEND WRITER APPEND WRITER
not permissible	EXTRACT APPEND EXTRACT WRITER APPEND WRITER
not permissible	EXTRACT APPEND MERGER WRITER

Figure 7.A-28

Examples of WRITER COMMAND

Explanation

The WRITER command writes out the contents of the work area as records. Where the CRITERIA entry for the preceding EXTRACT specifies terms, all "hit" terms from each input record are written as a single record in the named output file. Where the CRITERIA entry for the preceding EXTRACT specifies phrases, all hit phrases from each input record are written as a single record in the

USER LANGUAGE MANUAL
Language Specifications
7.A.4.6.2.

named output file. Where the CRITERIA entry for the preceding EXTRACT command specifies records, each hit record of the input file is written in the named output file.

The WRITEP command writes out each phrase in the work area as a separate record.

7.A.4.7. EDITING Commands: APPEND, DELETE, REPLACE

The data EXTRACTed into a work area can be edited or modified by appending, deleting, replacing data elements extracted into this area.

7.A.4.7.1. APPEND Command

The APPEND command appends data which appears as a DATA entry to the data element in the work area which meets the search CRITERIA.

	COMMAND	LIMIT	CRITERIA			FILE		DATA
			R	P	T	IN	OUT	
	EXTRACT		C					(GRP) ONE
1	APPEND		Z	D				
	WRITE							
	EXTRACT		C					
2	APPEND		A					(PNO) 2314 (PEO)
3	APPEND		Z	B				\$Tally 10
	WRITE							

Figure 7.A-29

Examples of APPEND Command

Rules:

1. Data is always appended to the record which has been extracted into the work area by the EXTRACT command immediately preceding this command. APPEND must be preceded by an EXTRACT command, and it must be within the scope of a WRITE command.

2. Phrases or terms can be appended to hit records and terms appended to hit phrases. Thus, the search CRITERIA are record (R) or record and phrase, (R) and (P), entries only.

3. Data to be appended can be:

- Any term defined in the File Format Table. If the user is not operating under FFT control, then any user specified term (see 1 and 2, Figure 7.A.29).
- A \$TALLY, \$ COMPUTE \$ ACCUM, or \$ DEFINE. (see 3, Figure 7.A-29).

4. The LIMIT entry is optional.

Explanation

Data is added to those elements in the work area record which meet the search CRITERIA. Terms can be appended to phrases; or, phrases or terms can be appended to records (see Figures 7.A-30 and 7.A.-31).

USER LANGUAGE MANUAL
Language Specifications
7.A.4.7.1.

SECRET

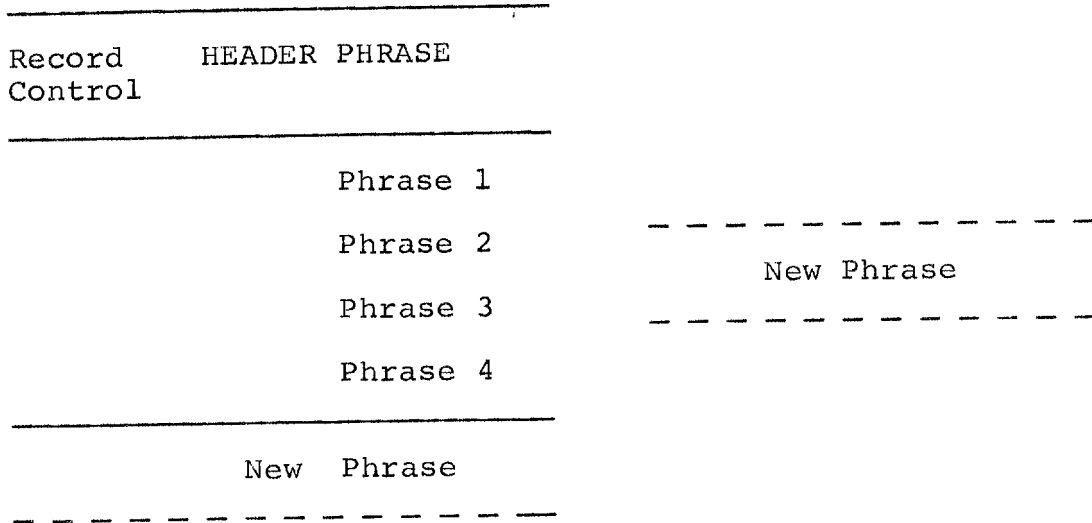


Figure 7.A-30

Phrase Appended to a Record

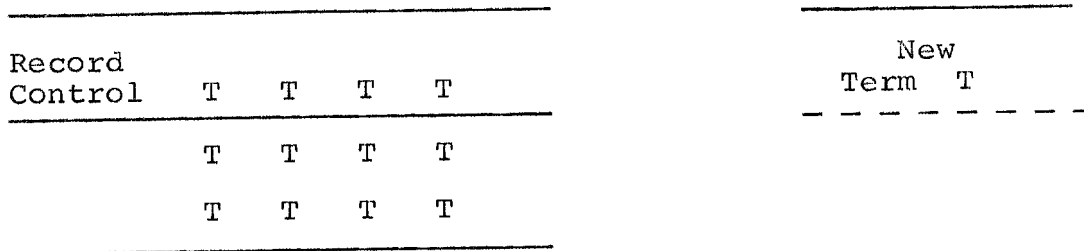


Figure 7.A-31

Term Appended to Phrase

Terms can be appended to phrases which meet the conditions of a referenced test line as in 1, Figure 7.A-29.

Phrases can be appended to records which meet the conditions of a reference test line as in 2, Figure 7.A-29

Data elements indirectly referenced may be appended as in 3, Figure 7.A-29. Here the result of a TALLY command, named TALLY 10, will be appended to phrases satisfying the condition on test line B for all records.

A complete record cannot be appended to the file with

7.A.4.7.2. DELETE Command

The DELETE command deletes data elements which meet the specified Search Criteria from the record in the temporary work area.

COMMAND	LIMIT	CRITERIA		
		R	P	T
1 DELETE		Z	A	C
2 DELETE		Z	B	
3 DELETE		A		

Figure 7.A-32

Examples of DELETE Command Lines

Rules:

1. The DELETE command operates on data extracted into a work area except for the record level. It must be preceded by an EXTRACT command, and it must be within the scope of a WRITE command.

2. Terms, phrases, or records can be deleted as in Figure 7.A-32, 1, 2, and 3 respectively.

3. The LIMIT entry is optional.

Explanation

Data which meets the search CRITERIA are deleted from the data element specified (see Figure 7.A-32, 1, 2, 3).

USER LANGUAGE MANUAL
Language Specifications
7.A.4.7.2.

- 1 - Terms which satisfy test line C in phrases which satisfy test line A are deleted from all records.
- 2 - Phrases which satisfy test line B are deleted from all records.
- 3 - Records which satisfy test line A are deleted from the file.

Only data elements in the work area record are operated on by the DELETE command.

With FFT control OFF, terms with any tag (including index language tags) can be deleted. With FFT control ON, only tags defined in the File Format Table may be deleted.

7.A.4.7.3. REPLACE Command

Data which appears as a DATA entry replaces the data element specified in the CRITERIA entries.

COMMAND	LIMIT	CRITERIA			DATA
		R	P	T	
1	REPLACE	Z	A	C	(PNO) 2145
2	REPLACE	B	D		(PED) 293 & PNO 2171

Figure 7.A-33

REPLACE Command Lines

USER LANGUAGE MANUAL
 Language Specifications
 7.A.4.7.3.

Rules:

1. The new data appears as a DATA entry.
2. Record (R), phrase (P) and term (T) CRITERIA entries are required when terms are replaced.
3. Record (R) and phrase (P) entries are required when phrases are replaced.
4. This command must be preceded by an EXTRACT, and it must be within the scope of a WRITE command.
5. The LIMIT entry is optional for this command.
6. File control conventions apply as in the APPEND and DELETE commands.

Explanation

The new data is replaced into the same location as the term or phrase specified in the CRITERIA entries.

Data elements are replaced in Figure 7.A-33, 1 and 2, as follows;

- Terms specified on test line C are replaced by the term in the DATA entry. Replacements occur in all phrases specified on test line A.
- Phrases specified on test line D are replaced by the phrase appearing in the DATA entry. Replacements occur in all records on test line B.

7.A.4.7.4. EXPLODE Command

The EXPLODE command copies phrases or terms onto an output file as individual records.

USER LANGUAGE MANUAL
Language Specifications
7.A.4.7.4.

SECRET

	COMMAND	LIMIT	CRITERIA			FILES	
			R	P	T		
Phrase Explosion	EXPLODE	(Opt)	v	v		v	v
Term Explosion	EXPLODE	(Opt)	v	v	v	v	v

Figure 7.A-34

EXPLODE Command Lines

Rules:

1. Hit data elements are copied directly into a named output file as individual records.
2. An input and an output file must be named.
3. The LIMIT entry is optional.
4. CRITERIA entries are required.
5. Only phrases and terms may be "exploded."
6. CRITERIA entries must appear under either (R), (P), and (T) or under (R) and (P).

Explanation

If phrases are EXPLODED, those phrases meeting the search CRITERIA from each record of the named input file are copied in the named output file as individual records as in Figure 7.A-35.

USER LANGUAGE MANUAL
Language Specifications
7.A.4.7.4.

SECRET

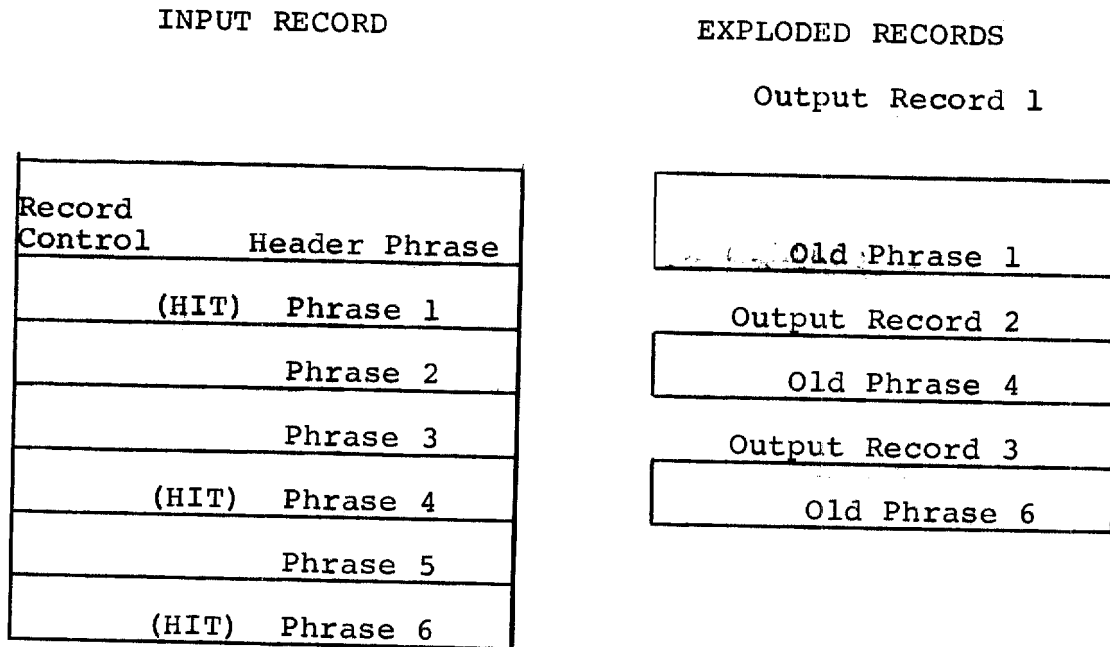


Figure 7.A-35

Exploded Hit Records

If term, (R), (P) and (T), CRITERIA are specified, those terms meeting the search criteria in each input record are copied as individual records in the named output file as shown in Figure 7.A-36.

7.A.4.8. Computational Commands: TALLY, ACCUM, COMPUTE

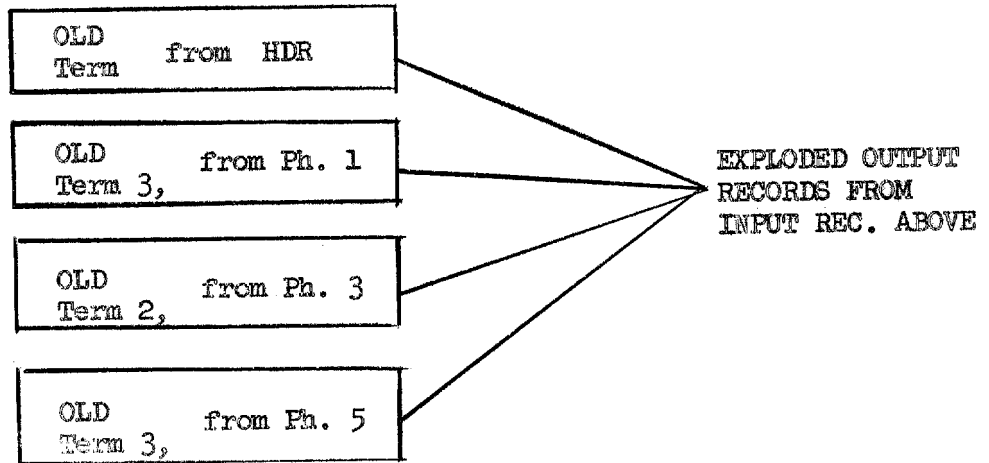
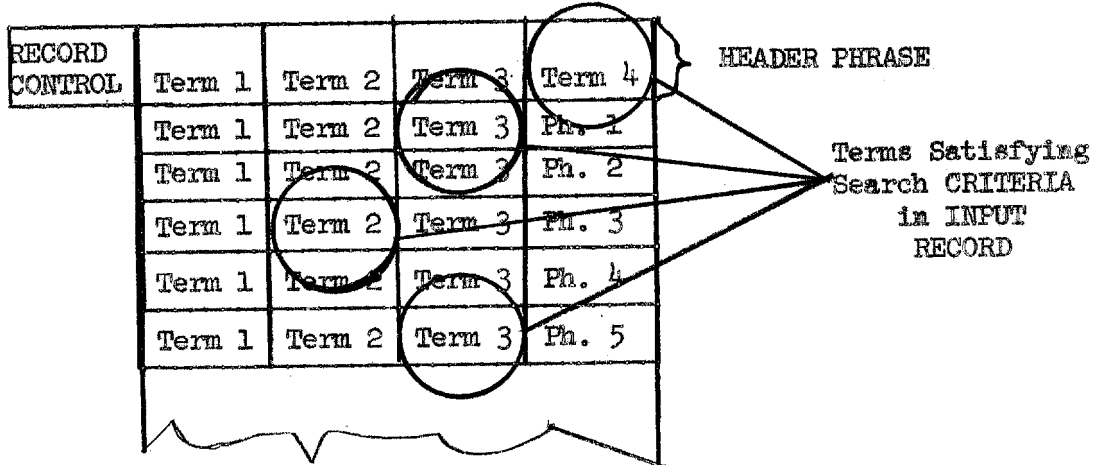
These commands are used in performing computational operations on CHIVE data elements:

TALLY - counts the number of times a data element (record, phrase, term) satisfies the specified CRITERIA test condition(s) within a specified logical structure.

SECRET

Figure 7.A-36

EXPLODE Terms Command



SECRET

ACCUM - accumulates a sum of the values of terms which satisfy the specified CRITERIA test condition(s).

COMPUTE (-) performs arithmetic operations on CHIVE data elements and on user defined values.

7.A.4.8.1. TALLY Command

The TALLY command increments a counter by one for each occurrence of a specified term (or terms) within a phrase; a specified phrase (or phrases) within a record; or a specified record (or records) within a named input file.

Tallying of:	COMMAND	LIMIT	CRITERIA			FILE
			R	P	T	
terms in records	TALLY 1	(Opt)	v	v	v	v
phrases in records	TALLY 2	"	v	v		v
records in file	TALLY 3	"	v			v

Figure 7.A-38

TALLY Command Lines

Rules and Explanation:

Search CRITERIA entries are required as follows:

- to TALLY occurrence of phrases which satisfy a named test line within a record: record (R) and phrase (P) CRITERIA entries are required.
- To TALLY occurrences of terms within a record: record (R), phrase (P) and term (T) CRITERIA entries are required.

USER LANGUAGE MANUAL
Language Specifications
7.A.4.8.1.

- To TALLY records which satisfy a named test line in the file: only a record (R) CRITERIA entry is required. In this case, the TALLY is accumulative throughout the job. Its contents are printed automatically.

A unique arbitrary number is assigned to the TALLY command as in Figure 7.A-38. The number of occurrences of the data element specified in the criteria entry is saved in a location with this name "TALLY." The number of occurrences can be subsequently referenced by use of the assigned name. "TALLY" names must be assigned a number which is unique for the retrieval transaction in which it appears.

Example: In all records of the named file, TALLY the occurrences of phrases which satisfy the test condition on test line C. Append the result of the TALLY operation in each extracted record (see Figure 7.A-39).

Since the contents of TALLY 1 are being appended to the record for later output, "TALLY 1" appears as a DATA entry preceded by \$.

USER LANGUAGE MANUAL
Language Specifications
7.A.4.8.1.

Test	TAG	OP	VALUE (s)
A	PNO	1	[REDACTED]
B	PEO	1	249
C		2	* A & B

25X6

COMMAND	LIMIT	CRITERIA			FILE		DATA
		R	P	T	IN	OUT	
TALLY 1		Z	C		DIF		
EXTRACT		C			DIF		
APPEND		Z					\$TALLY 1
WRITE						OUT 1	
PRINT					OUT 1	RCD	

Figure 7.A-39

TALLY and APPEND Result for Output

The assigned name "TALLY n" can also be referenced as a test line entry by its assigned test line name. It must be defined as a tag in a test line for each transaction in which it is referenced

7.A.4.8.2. ACCUM n Command

This command accumulates a sum of values found in:

- Records
- The results of TALLY and COMPUTE operations
- The result of another ACCUM operation

Rules:

1. Search CRITERIA entries are required always to the term (T) level, i.e., record (R), phrase (P) and term (T) CRITERIA entries are required.
2. Input file name must be entered.
3. Output file name is not entered.
4. Each ACCUM Command entry in a given retrieval transaction must be given a number "n" which is unique for that transaction.
5. A \$ACCUM n name can appear as a DATA entry in the APPEND or REPLACE commands.

Explanation

The values of terms that satisfy search CRITERIA entries are summed and stored in a location with the name ACCUM n.

In Figure 7.A-40 sums are accumulated for each input record of all NOO values ("number of organizational affiliations") that occur in a phrase with PNO [REDACTED] 25X6
The values thus accumulated for each input record are defined as ACCUM 1. This accumulated sum is printed automatically.

Values can be also accumulated across the file for TALLY and COMPUTE n values. They must be defined as tags in test lines as in Figure 7.A-41. Here a TALLY

USER LANGUAGE MANUAL
Language Specifications
7.A.4.8.2.

TEST	TAG	OP	VALUE (S)
A	PEO	1	77
B	PNO	1	2374
C			* A & B
D	TALLY 1	1	

COMMAND	LIMIT	CRITERIA			FILE		DATA
		R	P	T	IN	OUT	
TALLY 1		Z	C		DIF		
ACCUM 1		Z	Z	D	DIF		

Figure 7.A-41

Command Lines for Accumulating TALLY Result

is made in each input record of all phrases with PEO 77 and PNO 2374. (TALLY 1 command line.) ACCUM 1 produces a sum for all values of TALLY 1 defined on Test Line D.

TALLY 1 contains the number of hit phrases for each input record. The ACCUM 1 command line produces a total of occurrences of these hit phrases for the entire file.

7.A.4.8.3. COMPUTE n Command

This command performs arithmetic operations on numeric valued terms; values of TALLY and ACCUM counters; values of other COMPUTE's; and user defined constants.

SECRET

TEST	TAG	OP	VALUE (s)
A	NOO	1	
B	PNO	1	██████████
C			* A & B

25X6

COMMAND	LIMIT	CRITERIA			FILE		DATA
		R	P	T	IN	OUT	
ACCUM 1		Z	C	A	DIF		

Figure 7.A-40

Command Lines for Accumulating Across a File

Rules and Explanation:

Each COMPUTE command of a retrieval transaction must be assigned a number n which is unique for that transaction.

Operators and operands appear as DATA entries (see Figure 7.A-42).

COMMAND	LIMIT	CRITERIA	FILE	DATA			
		R	P	T	IN	OUT	
COMPUTE 1							Tag1+Tag2-Tag3
COMPUTE 2							ACCUM1*ACCUM2/TALLY7
COMPUTE 3							Tag1*Tag2/Tag3-Tag4
COMPUTE 4							Tag1*Tag2-Tag3
COMPUTE 5							Tag4/COMPUTE4

Figure 7.A-42

COMPUTE Command Lines

USER LANGUAGE MANUAL
Language Specifications
7.A.4.8.3.

SECRET

Operators are:

FIRST ORDER * multiply / divide

SECOND ORDER + add - subtract

The order in the list above is the order in which these operations will be performed for each COMPUTE command. Multiply and divide are performed first; then add and subtract. The operations performed in Figure 7.A-42 are performed by the system as follows:

COMPUTE 1: Add Tag 1 to Tag 2. Subtract Tag 3 from result.

COMPUTE 2: Multiply ACCUM 1 by ACCUM 2. Divide product by TALLY 7.

COMPUTE 3: $\frac{\text{Tag 1} \times \text{Tag 2}}{\text{Tag 3}} - \text{Tag 4}$

COMPUTE 4: Multiply Tag 1 by Tag 2. Subtract Tag 3 from product.

COMPUTE 5: Tag 4 divided by result of COMPUTE 4.

The COMPUTE command operates on one input record at a time when its operands are values of tags. In the first command line of Figure 7.A-42, COMPUTE 1 is computed and stored for each record. If operands cannot be uniquely defined, the system regards this as an error condition.

USER LANGUAGE MANUAL
Language Specifications
7.A.4.8.3.

SECRET

Operands can be:

- Defined tags
- Values for counters, TALLY (n)
- Accumulated sums, ACCUM (n)
- Previously generated COMPUTE n results
- Defined constants

Each COMPUTE command must be assigned a number, n, which is unique for the retrieval transaction.

The result is always stored in the name COMPUTE n, and it must be referenced by this name.

A COMPUTE n name can be referenced as a DATA entry in APPEND and REPLACE commands. It must be preceded by \$. (See Figure 7.A-43.)

COMMAND	LIMIT	CRITERIA		
		R	P	T
APPEND		Z	C	\$COMPUTE 10

Figure 7.A-43

COMPUTE n Referenced in an APPEND Command Line

The result stored in COMPUTE 10 is appended to each phrase which satisfies the CRITERIA entries for the APPEND command.

USER LANGUAGE MANUAL
Language Specifications
7.A.4.8.3.

SECRET

7.A.5. INTER-RECORD AND INFERENTIAL LOGIC


This section describes those commands which perform functions that require inter-record and inferential logic.

7.A.5.1. Enriched Searching

Requests may entail inter-record logic against the file. These requests often require two passes against the entire file and the formulation of two integral query requests by the user. The enriched query feature of the CHIVE language enables the user to formulate requests of this sort in a single retrieval transaction.

This class of requests retrieves all information related to a logically determinable set of data elements which cannot, however, be specified explicitly at request time. For example, a request calls for biographical information on personalities who have studied at the

25X6

 since 1960. The PNO's involved in this request cannot be specified at request time. Satisfaction of the request normally requires two passes to:

1. Determine the names (PNO's) of personalities attending the INSTITUTE since 1960, and then
2. Copy all biographical information on each of the PNO values which were generated in the first pass.

USER LANGUAGE MANUAL
Inter-record & Inferential Logic
7.A.5.1.

SECRET

The first search (see Figure 7.A-44) copies all PNO's appearing in phrases mentioning the Institute (394); the POE in these phrases will be later than December 31, 1959 (1960).

This search will generate the values for all PNO's to be used in the secondary search. The second search then re-examines all records for phrases satisfying the condition defined on test line F. Test line F is satisfied if and only if a PNO value from the input (DIF) file matches a PNO value in the file OUT 1. The special characters, "\$ FILE" are a requirement; they inform the system that OUT 1 is the name of a file and not the value for the tag entered on that line.

USER LANGUAGE MANUAL
Inter-record & Inferential Logic
7.A.5.1.

SECRET

TEST	TAG	OP	VALUE (s)
A	LAC	1	2147
B	PEO	1	394
C	PDE	>	591231
D	PNO	1	
E		4	*A & B & C & D
F	PNO	1	\$FILE OUT 1

COMMAND	LIMIT	CRITERIA			FILES		DATA
		R	P	T	IN	OUT	
COPY		Z	E	D	DIF	OUT 1	
COPY		Z	F		DIF	OUT 2	
PRINT					OUT 2	PHR	PNO

Figure 7.A-44
 Enriched Querying

7.A.5.2. MERGER and MERGE P Commands

7.A.5.2.1. MERGER Command

The MERGER command causes all records, in a named input file, which contain a common data element to be co-located in a named output file. The user can request a merge on a single tag or on multiple merge key tags.

SECRET

COMMAND	LIMIT	CRITERIA			FILES		DATA
		R	P	T	IN	OUT	
MERGER	-	-	-	-	X	X	PNO,

COMMAND	LIMIT	CRITERIA			FILES		DATA
		R	P	T	IN	OUT	
MERGER	-	-	-	-	X	X	PNO, & CNC,

Figure 7.A-45

MERGER with Single and with Multiple Merge Keys

Rules:

1. Input and output files must be named.
2. A merge key must be specified as a DATA entry. All records in which a common value for this key tag is found co-located.
3. No CRITERIA entries appear with this command.
4. The MERGER command does not appear within the scope of a WRITE command since this command initiates a file operation.

Explanation

Those records with a common value for the merge key are co-located in a named output file.

In Figure 7.A-46 records 1 and 2 contain the common value, 20, for the merge key CNC. These records are

USER LANGUAGE MANUAL
 Inter-record & Inferential Logic
 7.A.5.2.1.

SECRET

Figure 7.A-46

Input Records Before MERGER Command

RECORD CONTROL

CNQ 1100	(Tag - Value)	HEADER	(Tag - Value)	(Tag - Value)
-------------	------------------	--------	------------------	------------------

Phrase 1	LAC 27	(CNC 20)	CSS 17		
----------	--------	----------	--------	--	--

Phrase 2	LAC 32	CNC 14	CSS 22		
----------	--------	--------	--------	--	--

⋮
⋮
⋮

⋮
⋮
⋮

⋮
⋮
⋮

Phrase n	LAC 39	CNC 19	CSS 64		
----------	--------	--------	--------	--	--

RECORD NO. 1

RECORD CONTROL

CNQ 1820	(Tag - Value)	HEADER	(Tag - Value)	(Tag - Value)
-------------	------------------	--------	------------------	------------------

Phrase 1	LAC 29	CNC 26	CSS 16		
----------	--------	--------	--------	--	--

Phrase 2	LAC 14	(CNC 20)	CSS 41		
----------	--------	----------	--------	--	--

⋮
⋮
⋮

⋮
⋮
⋮

⋮
⋮
⋮

Phrase n	LAC 13	CNC 23	CSS 16		
----------	--------	--------	--------	--	--

RECORD NO. 2

merged into a single record as in Figure 7.A-47. The new record control for the merged record is CNZ 20. Records 1 and 3 will also be merged into a single record since they both possess a CNC14. Each header of the old records is retained with the new merged records.

If different values of the merge key are found in exactly the same records, those records will be merged into a logical record for each merge key value. Where a value for the merge key occurs in only one record of the named input file, that record is also copied in the named output file. Thus in Record 3 of Figure 7.A-47, if it is assumed that CNC 17 occurs only in that record of the file, then Record 3 is copied in the named output file as an individual record. (Also, records 1 or 2 are copied in the output file as individual records if any other CNC's occur only in either record.)

Where multiple keys are specified, those records with common values for the combination of merge keys are merged into a single record. In Figure 7.A-48, records 1, 2, and 3 have the common pair CNC 1 and PNO9 and they are merged as in Figure 7.A-49.

For each combination of values which appears in more than one record, a new record is created that is made up of each record in which that combination of common values appeared. When there is more than one

USER LANGUAGE MANUAL
Inter-record & Inferential Logic
7.A.5.2.1.

Figure 7.A-47

MERGED RECORD with Common CNC 20

RECORD CONTROL

CNQ 1920	(TAG - VALUE)	(TAG - VALUE)	HEADER	(TAG - VALUE)	(TAG - VALUE)
Phrase 1	LAC 40	CNC 17	CSS 17	CTC 6	
Phrase 2	LAC 70	CNC 16	CSS 22	CTC 4	
	:		:		:
Phrase n	LAC 39	CNC 14	CSS 20	CTC 3	

RECORD 3

NEW RECORD CONTROL

CNZ 20	(Tag - Value)	HEADER	(Old Record 1)	(Tag - Value)
		HEADER	(Old Record 2)	(Tag - Value)
LAC 27	CNC 20	CSS 17	PHRASE 1	
LAC 32	CNC 14	CSS 22	PHRASE 2	
LAC 39	CNC 19	CSS 64	PHRASE 3	
LAC 29	CNC 26	CSS 16	PHRASE 4	
LAC 14	CNC 20	CSS 41	PHRASE 5	
LAC 13	CNC 23	CSS 16	PHRASE 6	
	:		:	:
	:		:	:
LAC 30	CNC 14	CSS 16	PHRASE n	

Figure 7.A-48

Input Records with Common Values
for Multiple Merge Key PNO and CNC

CNC 1	T-V	T-V	HEADER	T-V	T-V	T-V
	PNO 9	CNC 1	PAF 4	PCM 3	PED 7	PHRASE 1
	PNO 8	CDY 65	PHF 4	PHN 0	PJP 29	PHRASE 2

RECORD 1

CNC 1	T-V	T-V	HEADER		T-V	T-V
	CNC 1	PNO 9	CSS 9	OLP 6	ONP 6	PHRASE 1
	PNO 8	CDY 65	PLP 64	POH 4	RMO 69	PHRASE 2

RECORD 2

CNC 1	T-V	T-V	HEADER		T-V	T-V
	LAC 6	LAN 4	PNO 7	PNO 9	CNC 1	PHRASE 1
	LAC 7	CNC 4	CSS 22	PNO 8	CDY 65	PHRASE 2

RECORD 3

Figure 7.A-49

Merged Record with Common Multiple Merge Key

CNZ 1	PNZ 9	HEADER (Old Record 1)				
		HEADER (Old Record 2)				
		HEADER (Old Record 3)				
	PNO 9	CNC 1	PAF 4	PCM 3	PED 7	PHRASE 1
	PNO 8	CDY 65	PHF 4	PHN 0	PJP 29	PHRASE 2
	CNC 1	PNO 9	CSS 9	OLP 6	ONP 6	PHRASE 3
	PNO 8	CDY 65	PLP 64	POH 4	RMO 69	PHRASE 4
LAC 6	LAN 4	PNO 7	PNO 9	CNC 1	PHRASE 5	
LAC 7	CNC 4	PNO 8	CDY 65	CDY 65	PHRASE 6	

SECRET

pair of common values in exactly the same combination of records, those records are merged multiple times into new records.

When a combination of multiple merge key values appears uniquely in a record, that record is copied on the named output file.

An example is given below to illustrate the use of the MERGER command for "inferential searching."

Problem: The user seeks information on an unidentified scientific conference which took place in MOSCOW on or later than July 5, 1963. The names of several outstanding scientific personalities who attended the conference are known. (Again, the identity of the conference itself is unknown.)

It is assumed that the results of an earlier request transaction have shown that the information desired is not contained in any single record of the CHIVE files. Thus the inter-record logic performed with the MERGER command is required for this transaction.

USER LANGUAGE MANUAL
Inter-record & Inferential Logic
7.A.5.2.1.

SECRET

TEST	TAG	OP	VALUE(s)
A	LAC	1	9993, MOSCOW
B	CDY	>	630705
C	CNC	1	
D	PNO	1	2344 & 2344 & 2346 & 2347 & 2348
E		4	*A & B & C & D
F	PNO	3	2344 & 2345 & 2346 & 2347 & 2348

COMMAND	LIMIT	CRITERIA			FILES		DATA
		R	P	T	IN	OUT	
COPY		E			DIF	OUT 1	
MERGER					OUT 1	OUT 2	CNC
COPY		F			OUT 2	OUT 3	
PRINT					OUT 3	RCD	

Figure 7.A-50

Test and Command Lines for Inferential MERGER Logic

Each command in Figure 7.A-50 is explained as follows:

COPY - All records are copied which mention a conference in Moscow later than July 5, 1963 and at least one of the known personalities. The input file is the Document Index Records File (DIF). The output file is named OUT 1.

USER LANGUAGE MANUAL
Inter-record & Inferential Logic
7.A.5.2.1.

MERGER - Using the OUT 1 file as input, all records with a common CNC are merged.

All records from file OUT 1 with common values for the CNC tag have been merged into single logical records and put into file OUT 2. It is now possible to search each of these merged records in file OUT 2 for mention of the five known personalities.

COPY - Any record in file OUT 2 mentioning three or more of the five known personalities is copied on file OUT 3. Ideally, the search will retrieve at least one record mentioning all of the known personalities. And, if each of the personalities has been indexed with the CNC that is sought, at least one record mentioning all of them and that CNC will be retrieved. The numeric operator '3' is used in the event that two of the personalities might not have been indexed in any document index records with the CNC sought.

All hit records are then PRINTed in a standard format for records.

7.A.5.2.2. MERGEP Command

All phrases within a record with a common value for the merge key are combined or merged into a single new

USER LANGUAGE MANUAL
Inter-record & Inferential Logic
7.A.5.2.2.

phrase. Phrases can be merged on a single merge key or on multiple merge keys. (Examples of both cases are given in Figure 7.A-51.)

COMMAND	LIMIT	CRITERIA			FILE IN	FILE OUT	DATA
		R	P	T			
MERGE P	-	-	-	-	-	-	PNO

COMMAND	LIMIT	CRITERIA			FILE IN	FILE OUT	DATA
		R	P	T			
MERGE P	-	-	-	-	-	-	PNO & PEO

Figure 7.A-51

MERGE P with Single and Multiple Merge Keys

Rules:

1. Input and output files are not named.
2. A merge key must be specified as a DATA entry. All phrases in which a common value for this key are found are merged into a single phrase. Multiple merge keys may be specified. They must be separated by "&".
3. No LIMIT or CRITERIA entries are used.
4. This command must be preceded by an EXTRACT command, and it must be within the scope of a WRITE command. The command operates on data which is retrieved by an extract command, i.e., on data which is in the work area.

USER LANGUAGE MANUAL
Inter-record & Inferential Logic
7.A.5.2.2.

Explanation

All phrases within a record with common values for the merge key are merged into a single new phrase.

RECORD CONTROL	HEADER PHRASE					
	PNO 1	PEO 7	PNO 2	PEO 4	PNO 3	Phr.1
	PNO 1	PEO 8	PNO 2	PEO 9	PNO 3	Phr.2
	PNO 1	PEO 10	PNO 4	PEO 6	PNO 5	Phr.3
	PNO 6	PEO 1	PNO -	PEO -	PNO -	Phr.4
	PNO 7	PEO 14	PNO -	PEO -	PEO -	Phr.5
	PNO -	PEO -	PNO -	PNO -	PEO -	Phr.6
	PNO -	PEO -	PNO -	PNO -	PEO -	Phr.7

Figure 7.A-52

Phrases with Common Values for Merge Keys

In figure 7.A-53, Phrases 1, 2, and 3 have a common PNO value (1). Phrases 1 and 2 also have common PNO values (2) and (3).

If more than one merge key value is common to exactly the same phrases, those phrases will be merged into a new single phrase only once. This convention is different from the MERGER convention for different merge key values occurring in exactly the same records. The matrix in

USER LANGUAGE MANUAL
 Inter-record & Inferential Logic
 7.A.5.2.2.

Figure 7.A-53 shows that PNO values 2 and 3 are common to exactly the same phrases, namely phrases 1 and 2. Thus, only one new phrase is created from phrases 2 and 3 below, even though both values 2 and 3 are found in these phrases.

	Ph 1	Ph 2	Ph 3	Ph 4	Ph 5
1	x	x	x		
2	x	x			
3	x	x			
4			x		
5			x		
6				x	
7					x

PNO Values

Figure 7.A-53

Matrix of VALUES for Phrases 1-5

Every unique combination of phrases containing a common value is merged into a single, new phrase.

The application of the above rules to the record in Figure 7.A-53 results in the following phrase merging operations:

- Phrases 1, 2, and 3 into a single phrase for PNO 1, as shown in Figure 7.A-54.

USER LANGUAGE MANUAL
 Inter-record & Inferential Logic
 7.A.5.2.2.

- Phrases 1 and 2 into a single phrase for PNO 2 since it occurs in a unique combination of phrases.
- No new phrase for PNO 3 (applying 2 above.) since PNO 3 appears in the same combination as PNO 2.

PNO 1	PEO 7	PEO 8	PEO 10	PNO 2	PNO 4	PEO 4
PEO 9	PEO 6	PNO 3	PNO 5			

Figure 7.A-54

New Phrase Merged from Phrases 1, 2, and 3

When a value for the merge key appears in only one phrase, that phrase is copied into the new record. The phrase is inserted for every different value that occurs uniquely in that phrase.

Thus, phrases 3, 4 and 5 are inserted in the new record as individual phrases since they contain PNO's 4, 6, and 7 respectively. Notice that Phrase 3 is not inserted redundantly for the occurrence of PNO 5.

All other phrases in the record are inserted in the new record.

Multiple merge keys may be specified as DATA entries. Phrases within a record are merged into a new phrase if, and only if, they contain the same values for the combination of merge keys.

USER LANGUAGE MANUAL
 Inter-record & Inferential Logic
 7.A.5.2.2.

Figure 7.A-55

New Record after MERGEP Execution

RECORD CONTROL	HEADER										
Merged Ph. 1	PNO 1	PEO 7	PEO 8	PEO 10	PNO 2	PNO 4	PEO 4	PEO 9	PEO 6	PNO 3	PNO 5
Merged Ph. 2	PNO 1	PEO 7	PEO 8	PNO 2	PEO 4	PEO 9	PNO 3	-	-	-	-
Ph. 3	PNO 1	PEO 10	PNO 4	PNO 6	PNO 5	-	-	-	-	-	-
Ph. 4	PNO 6	PEO 1	PNO -	PEO -	PNO -	-	-	-	-	-	-
Ph. 5	PNO 7	PEO 14	PNO -	PEO -	PEO -	-	-	-	-	-	-
Ph. 6	PNO -	PEO -	PNO -	PNO -	PEO -	-	-	-	-	-	-
Ph. 7	PNO -	PEO -	PNO -	PNO -	PEO -	-	-	-	-	-	-

SECRET

SECRET

Phrases 1 and 2 in Figure 7.A-56 contain common values for the merge keys PNO and PEO, namely, PNO 1 and PEO 7. Phrases 1 and 2 will therefore be merged into a single new phrase. Phrases 2 and 3 contain common value 3 for merge key PNO, but since the values for the second merge key, PEO, differ, they are not merged.

All other points above apply except that a common value for both merge keys must be present.

7.A.6. FILE MAINTENANCE

Two types of file maintenance transactions are distinguished in the CHIVE System:

- Conditional file maintenance--changes to CHIVE files involving data elements which apply to logically specifiable classes of data elements. (Specification of these changes involves the use of the CHIVE language in a way which is similar to its use for retrieval transactions.)
- Unconditional file maintenance--changes to CHIVE data elements which can be specified through the use of unique control terms: record control, phrase control, tag name, or tag name-value.

7.A.6.1. Conditional Maintenance

Conditional maintenance is accomplished through the use of APPEND, DELETE, and REPLACE commands.

The Rules and Explanations for each of these commands are identical to those given in Section 7.8. for the same commands of the query language, i.e., for retrieval transactions.

USER LANGUAGE MANUAL
File Maintenance
7.A.6.1.

Figure 7.A-56

MERGE Phrases for Multiple Merge Keys

RECORD CONTROL	HEADER PHRASE			
	PNO 1	PEO 7	PNO 2	Ph. 1
	PNO 1	PEO 7	PNO 3	Ph. 2
	PNO 2	PEO 2	PNO 3	Ph. 3
	PNO 3	PEO 4	PNO 5	Ph. 4
	PNO -	PEO -	PNO -	Ph. 5
	PNO -	PEO -	PNO -	Ph. 6

7.A.6.1.1. APPEND Command

This command adds the data which appears as a DATA entry to the data element that meets the search CRITERIA.

COMMAND	LIMIT	CRITERIA			DATA
		R	P	T	
1 APPEND		Z	C		PNO 7641
2 APPEND		A			PNO 7294 & PEO 7396

Figure 7.A-57

Command Lines for File Maintenance APPEND

Rules:

1. Search CRITERIA for records (R) and phrases (P) are required to APPEND terms. To APPEND phrases to records, only record (R) CRITERIA entries are necessary.
2. File input and output entries are not required. The file being maintained is named in the MAINTENANCE JOB SPECIFICATIONS (see Section 7.A.3.).
3. All data to be appended must appear as DATA entries
4. The LIMIT entry is optional.
5. Records cannot be appended to the file with this command.

Explanation

Figure 7.A-57 shows two examples of APPEND commands:
 (1) APPEND adds "PNO 7641" to phrases which meet the phrase CRITERIA entry, i.e., the condition specified on

test line C, in all records. (2) APPEND adds the phrase "PNO 7294 PEO 7396" to all records which meet the record CRITERIA entry A.

7.A.6.1.2. DELETE Command

Terms, records, or phrases are deleted from the file.

COMMAND	LIMIT	CRITERIA			FILES	DATA
		R	P	T		
1 DELETE		A	Z	C		
2 DELETE		Z	B			
3 DELETE		D				

Figure 7.A-58

Command Lines for File Maintenance DELETE

Rules:

1. CRITERIA entries are required as follows:
 - (R) - to DELETE records specified
 - (R), (P) - to DELETE specified phrases from specified records.
 - (R), (P), (T) - To DELETE specified terms from specified phrases in specified records.
2. No DATA entries appear.
3. Other entries are the same as the APPEND command.

Explanation

Figure 7.A-58 shows examples of three DELETE commands:

- (1) DELETE - terms which meet conditions on test line C

are deleted in all phrases of records which meet condition specified on test line A. (2) DELETE - phrases specified on test line B are deleted from all records. (3) DELETE - records which meet conditions specified on test line D are deleted from the file.

7.A.6.1.3. REPLACE Command

Data elements in the DATA entry REPLACE those data elements which meet the search CRITERIA.

COMMAND	LIMIT	CRITERIA			FILE	DATA
		R	P	T		
1 REPLACE		A	B	C		PEO 17
2 REPLACE		A	B			PNO 2617 & PEO 99

Figure 7.A-59

Command Lines for File Maintenance REPLACE

Rules:

1. Search CRITERIA entries are required as follows
 - (R), (P) - to replace a phrase
 - (R), (P), (T) - to replace a term
2. Records cannot be replaced.
3. Other rules are identical to the rules for the APPEND command.

USER LANGUAGE MANUAL
File Maintenance
7.A.6.1.3.

Explanation

Figure 7.A-59 shows examples of two REPLACE commands:

(1) REPLACE - the term in the DATA entry replaces all terms which meet term, phrase, and record CRITERIA entries. (2) REPLACE - the phrase in the DATA entry replaces all phrases which meet phrase and record CRITERIA entries.

7.A.6.2. Unconditional Maintenance

A special form will be provided to specify maintenance to be performed when a uniquely specifiable element of a file is addressed. This form will include (among other things) on each line:

- the unique record to be addressed (by record control terms)
- the unique phrase to be addressed (by phrase control data)
- the unique term to be addressed (either the tag name or a tag-value combination)
- the operation to be performed (append, delete, replace)
- the data to be inserted or appended.

USER LANGUAGE MANUAL
File Maintenance
7.A.6.2.

SECRET

Appendix 7.B.

INPUT TRANSCRIPTION STUDY

One of the major problems of the CHIVE system is that of reading the indexing records and other data into the computer system for subsequent processing and file maintenance operations. Four categories of systems were evaluated for this application: (a) punched cards, (b) punched paper tape, (c) cathode-ray tube/keyboard (CRT/KB) equipments, and (d) page readers. The results of that evaluation are described in the sections that follow, using the above headings.

A preliminary oral presentation of the major portion of the material included in this report was made to OCS and CHIVE staff members on 19 November 1964. Immediately following that presentation, a decision was made jointly by the OCS and the CHIVE staffs to utilize the page reader methods for the indexing operation, and to give further attention to that specific area. Accordingly, supplementary investigations were made (primarily by phone) to provide a broader base of information in anticipation of a page reader procurement program.

INPUT TRANSCRIPTION
7.B.

SECRET

The fundamental assumptions on which the evaluations were based are specified in Section 7.B.1, and details concerning operational aspects and the cost and manpower requirements of each of the candidate methods are included in Section 7.B.2. The results of the study are summarized in Section 7.B.3., and a breakdown of the costs for each of the systems configurations is provided in CHIVE/R-2-65.

7.B.1. FUNDAMENTAL ASSUMPTIONS

The various candidate input methods differ somewhat in detail, but there are certain basic procedures and factors that would be common to all of the methods.

The following assumptions were made for all evaluations:

- The index records are provided in a format (indexer work sheet) that is to be used for direct transcription into the data-input system.
- The index records are transcribed with keyboard equipment (where proofing is required, hard copy is also prepared during this operation).
- Proofing is made with reference to the original indexer work sheet.
- Corrections are made as necessary against the proofing information.
- The index records are transcribed off-line onto magnetic tape for subsequent computer processing.

INPUT TRANSCRIPTION
Fundamental Assumptions
7.B.1.

SECRET

SECRET

- All equipment operating speeds are de-rated to 75% of the corresponding nominal values, unless estimates of effective rates were available from other sources.
- Equipment costs were determined both for purchase (including maintenance), based on five-year amortization at zero percent interest rate and for lease. The lower of the two costs is used for the evaluations in each case.

Cost information is provided graphically in Section 7.B.2. in terms of the program cost per year, based on a variable number of index items processed. Salaries for experienced typists are assumed to be \$5170 per year, salaries for keypunch operators \$4650 per year, and salaries for special equipment operators \$6000 per year for a single-shift operation of 264 days per year.* Costs for quantity items such as cards, paper, etc., are included, but in no case are overhead costs, or costs that are believed to be common to all candidates incorporated into the estimates.**

The cost and the manpower requirements for the transcription of the index records are very significant. Accordingly, the effective production rates are correspondingly critical. Based on information obtained

* C.F., CHIVE/R-2-64

** A cost breakdown for each system is provided in CHIVE/R-2-65.

INPUT TRANSCRIPTION
Fundamental Assumptions
7.B.1.

SECRET

SECRET

from various sources,* the production rates assumed for the evaluations were as follows:**

Typewriter--10,000 characters per hour

Paper-tape punches and IBM 826--9000 characters per hour

Keypunch--8000 characters per hour.

In all evaluations, it was assumed that there are 1000 characters per indexing record, and that only upper case characters are utilized.

On visits made to the Defense Documentation Center (DDC) and to the National Library of Medicine (NLM) immediately

* The IBM Typewriter Division specifies a typical typewriting rate of 10,200 characters/hr; personnel implementing a new library indexing operation at Lockheed, Palo Alto have determined that production rates slightly in excess of 10,000 characters/hr. are achieved by operators working for periods of two hours on the IBM 826 keyboard; in a report entitled "Application of Print Readers to the Needs of Intelligence Agencies," Study-147, March 1964, the Institute for Defense Analysis reports that an experienced keypunch operator can punch about 500 verified cards/day. If it is assumed that there are 80 characters punched per card, and that the verifier rate is 40% faster than the keypunch rate, the production of 500 verified cards/day would correspond to a keypunch rate of approximately 8000 characters/hour. Continuing OCR/MD tests have also indicated keypunch rates of approximately 8000 characters/hour.

**In all evaluations, costs were calculated and are presented graphically for typewriter/keypunch speeds varying over a range of approximately \pm 12% from these nominal values.

INPUT TRANSCRIPTION
Fundamental Assumptions
7.B.1.

SECRET

SECRET

prior to the oral presentation at the Agency, information was obtained on production rates achieved at those organizations for operations where the keyboard operator is not provided complete information from a single work sheet, but is instead required to extract material to be transcribed from both the work sheet and the original document or journal. For these cases, the rates were reported to be approximately only 3500 characters per hour. Certainly for direct transcription from a single work sheet, the average rate achieved must be much higher than this figure, but the problem of predicting a reasonable transcription rate should be considered further. Better estimates could be obtained with an in-house test run for several days with a representative group of typists and keyboard operators working on representative input records.

The proofing operation is reported to constitute a very significant time requirement at Lockheed, at DDC, and at NLM where the hard copy produced during the transcription process is checked directly against the original. Based on the Lockheed figures, it was specified for our evaluations that the proofing operation requires 2.8 minutes

INPUT TRANSCRIPTION
Fundamental Assumptions
7.B.1.

-289-

SECRET

SECRET

per indexing record.* The (normalized) times required for proofing at DDC and at NLM were reported to be greater than that required at Lockheed by approximately the same factor as the ratios of the times required for the corresponding transcription operations (approximately 2.6:1).

The greatest variable of all the operations in the various inputting systems is that of the correction time required. For example, in the paper tape and the punched card cases where normal correction procedures involve scanning the original tape or cards to locate the specific items that must be corrected, the time requirement is severe. Conversely, with the cathode-ray tube/keyboard (CRT/KB) technique, the text is immediately available on the display tube for processing, and the time required to achieve the necessary corrections is minimized. For the evaluations, the correction time requirement was assumed to vary over the range from 2.8 to 0.8 minutes per indexing record. The correction times are specified individually for each input system considered in the

* This specification applies to all systems except the punched card case where the IBM 059 Verifier is used. For this case, initial OCR/MD test data is used.

INPUT TRANSCRIPTION
Fundamental Assumptions
7.B.1.

SECRET

SECRET

following sections of this report.* It was assumed here that there are two errors per item. This assumption was based on the statement by the IBM Typewriter Division that an experienced typists makes one error in approximately 500 characters, and on the statement by the DDC personnel that their typists typically make two errors in approximately 80 percent to 85 percent of the 1000-character items transcribed (NLM specifies that errors occur in only 27% of the items produced by their Flexowriter operators, but these items are of approximately only 250 characters in length. Complete statistical data are not available from NLM).

As noted previously, the decision was made after the preliminary oral presentation to plan on the use of page readers. Coupled with this decision are new operational aspects of the input problem that are to be considered for a slightly modified mode of operation:

- The output of the page reader is to be coupled directly to the master data processing system by the executive control system, rather than being transferred off-line to magnetic tape. This eliminates the need for a separate magnetic tape system associated with the page reader, but adds the need for some interface equipment.

*The primary reference data utilized is that provided by the Lockheed staff in charge of implementing the Library indexing and transcription operation.

INPUT TRANSCRIPTION
Fundamental Assumptions
7.B.1.

SECRET

SECRET

- Whereas the original page-reader evaluations were based on single-font equipments, any equipment to be considered for future evaluations should have the capability of reading two or more fonts.
- In those cases where an indexer may elect to use a typewriter in the preparation of the index work sheets, it should be possible that these work sheets may be used directly as the input document to the page reader system, thereby eliminating the need for a separate typing operation. It is even possible that for the long term, with a suitable education program, all of the indexers can be induced to prepare their work sheets with a typewriter in a form suitable for direct scanning by the page reader. The additional cost of the extra time that would probably be required by the indexers for the preparation of suitable typewritten work sheets would necessarily be compared to the cost of the typewriter transcription process that has thus far been assumed to be necessary. However, it would seem to be an area in which some additional cost and manpower savings might be realized.

In Section 7.B.2.4., in which the page reader operation is considered, the costs and the manpower requirements are specified primarily for the original mode of operation. However, cost data are provided also for the on-line mode of operation with the one currently-available, production page reader capable of operation on a multi-font basis.

7.B.2. EVALUATIONS OF THE CANDIDATE INPUT SYSTEMS

7.B.2.1. Punched Card Systems

Punched card systems of all shapes and sizes have been in operation essentially since the beginning of automatic

INPUT TRANSCRIPTION
Evaluations
7.B.2.1.

SECRET

SECRET

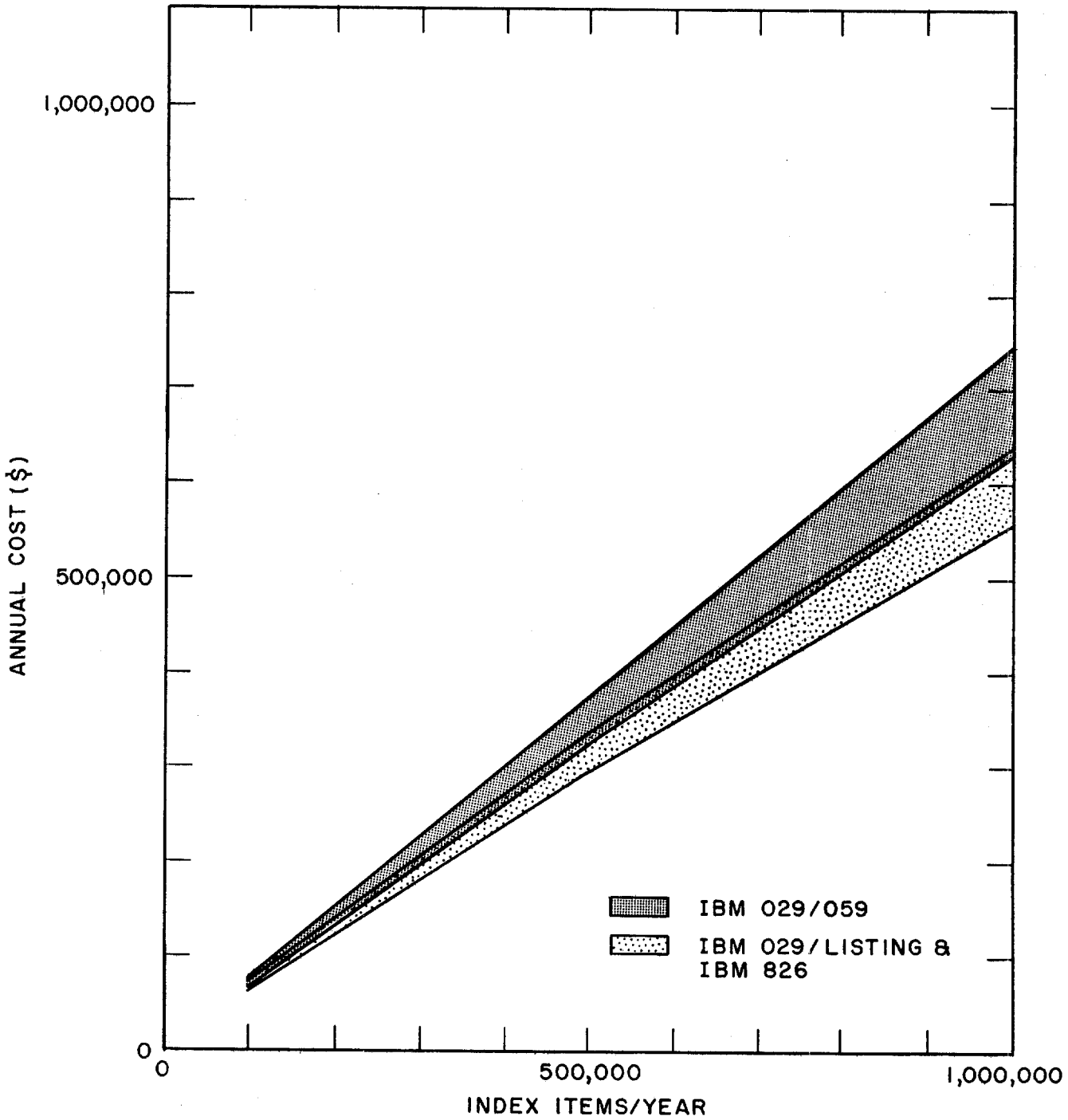
data-processing systems. Accordingly, a detailed discussion of the punched card systems as possibly applied to the CHIVE data input operation is not warranted in this report. It will suffice to only review briefly the three configurations of punched card systems that were considered in the project studies. These three systems are:

- A keypunch/verifier configuration. It is to be noted that the costs considered in this evaluation are those of the recently announced solid-state models of the keypunch and verifier (IBM 029/059 equipments) instead of the familiar IBM 026/056 equipments. This caused only a slight percentage increase in the overall cost of the system.
- A keypunch/listing configuration. In this case a keypunch operator punches the cards in the normal fashion, but uses a listing for a visual proofing.
- An IBM 826 configuration. In this case hard copy is prepared on a typewriter electrically coupled to a companion keypunch unit, on which cards are simultaneously punched. (This is the configuration utilized very successfully by the Lockheed staff responsible for implementing a new document cataloging and retrieval system in their library.)

The summary cost information for the three punched card systems is shown in Fig. 7.B-1. Operation of the keypunch/verifier configuration is the most expensive of the three--primarily because more operators and more pieces of equipment are required (c.f., the comparative tabulation in Table 7.B-2, Section 7.B.3., in which the number of typists or key punch operators required is listed). The

INPUT TRANSCRIPTION
Evaluations
7.B.2.1.

SECRET



OPERATING COSTS
PUNCHED CARD SYSTEMS

FIGURE 7.B-1

SECRET

SECRET

other two configurations are approximately equivalent in terms of operating cost, as indicated in Figure 7.B-1.

As stated in Section 7.B.1., the proof time that was utilized in all of the cost evaluations is 2.8 minutes per indexing record, except for the keypunch/verifier equipment configuration. In this case, the proof was specified to be 6.4 minutes per record, this being based on initial estimates of keypunch verifying rates.

Regarding the correction operation, the keypunch/listing operation was specified to require a relatively large amount of time for corrections--2.6 minutes per item. This was based on estimates of the time required for the relatively large number of steps involved. In particular, since efficient listing operations require that the cards from many items be run through the lister in large batches, a considerable amount of time is subsequently required for scanning for, and retrieval of, the specific cards for which correction cards must be made. For the keypunch/verifier (029/059) and the IBM 826 operation, procedures can be arranged whereby the cards for a specific item can be retained by the operator(s) until corrections are made. Accordingly, the time

INPUT TRANSCRIPTION
Evaluations
7.B.2.1.

SECRET

required for corrections can be minimized to a reasonable extent. For these two equipment configurations, it was specified for the evaluations that a correction time of 1.6 minutes per item is required.

For the listing operation and for the conversion from punched cards to magnetic tape, two modes of operation were assumed. For relatively low rates of production, rates quoted by service bureaus were utilized in determining costs. However, for relatively high production rates where cost analysis indicated justification for purchase of suitable equipment, costs were calculated on that basis. In the latter case, purchase of a Univac Model 1004-2 was assumed for listing operations, and purchase of CDC equipment was assumed for the card-to-magnetic-tape conversion (c.f., CHIVE/R-2-65). In both cases, the cost of operators for special machines was included.

The number of keypunch operators (or typists) and proofing clerks required for the three punched card systems for production of from 100,000 to 1,000,000 items per year varies from 8 to 129. A complete comparative tabulation is provided in Section 7.B.3. (Table 7.B-2).

INPUT TRANSCRIPTION
Evaluations
7.B.2.1.

SECRET

SECRET

7.B.2.2. Punched Paper Tape Systems

Punched paper tape equipments have been developed to a relatively high degree of sophistication and reliability during the last few years. In some cases, manufacturers have incorporated IBM Selectric typewriters as the basic element of their equipments to increase the operating ease and efficiency. From the automatic control standpoint various manufacturers have incorporated program control features into their equipments whereby special paper tape loops punched with specific program control codes can be used to greatly facilitate repetitive modes of operation. Six types of units were investigated in this study.*

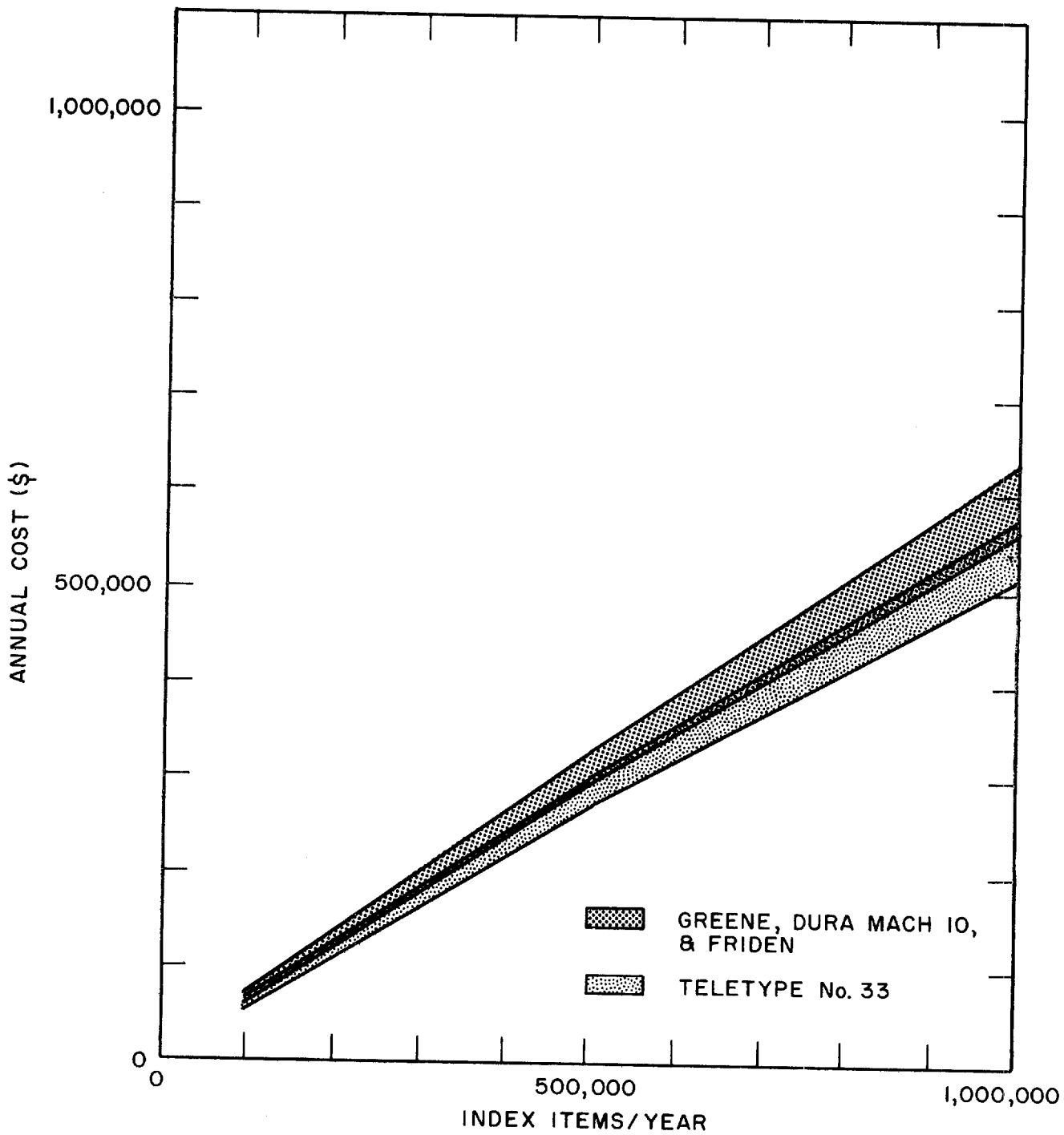
The results of the cost evaluations for the punched paper systems are shown in Figs. 7.B-2 and 3. For all such equipments considered, the cost variations are relatively

*The equipments investigated were the Dura Mach 10, Friden Flexowriter, the Greene Letteriter, the Royal SDW 5000, the Teletype Model 33, and the Teletype Model 35. Additional units such as the Kleinschmidt Model 321, and the Remington Rand Sychrotape units are available, but were not evaluated. In the case of the Sychrotape, Remington Rand has sold all inventory and production rights to Radio Electronics, Inc., New York City, and this latter company is withholding further production until sales trends of their new magnetic tape typewriter units are evaluated.

INPUT TRANSCRIPTION
Evaluations
7.B.2.2.

SECRET

SECRET



OPERATING COSTS
PAPER TAPE SYSTEMS

FIGURE 7.B-2

SECRET

SECRET

7.B.2.2. Punched Paper Tape Systems

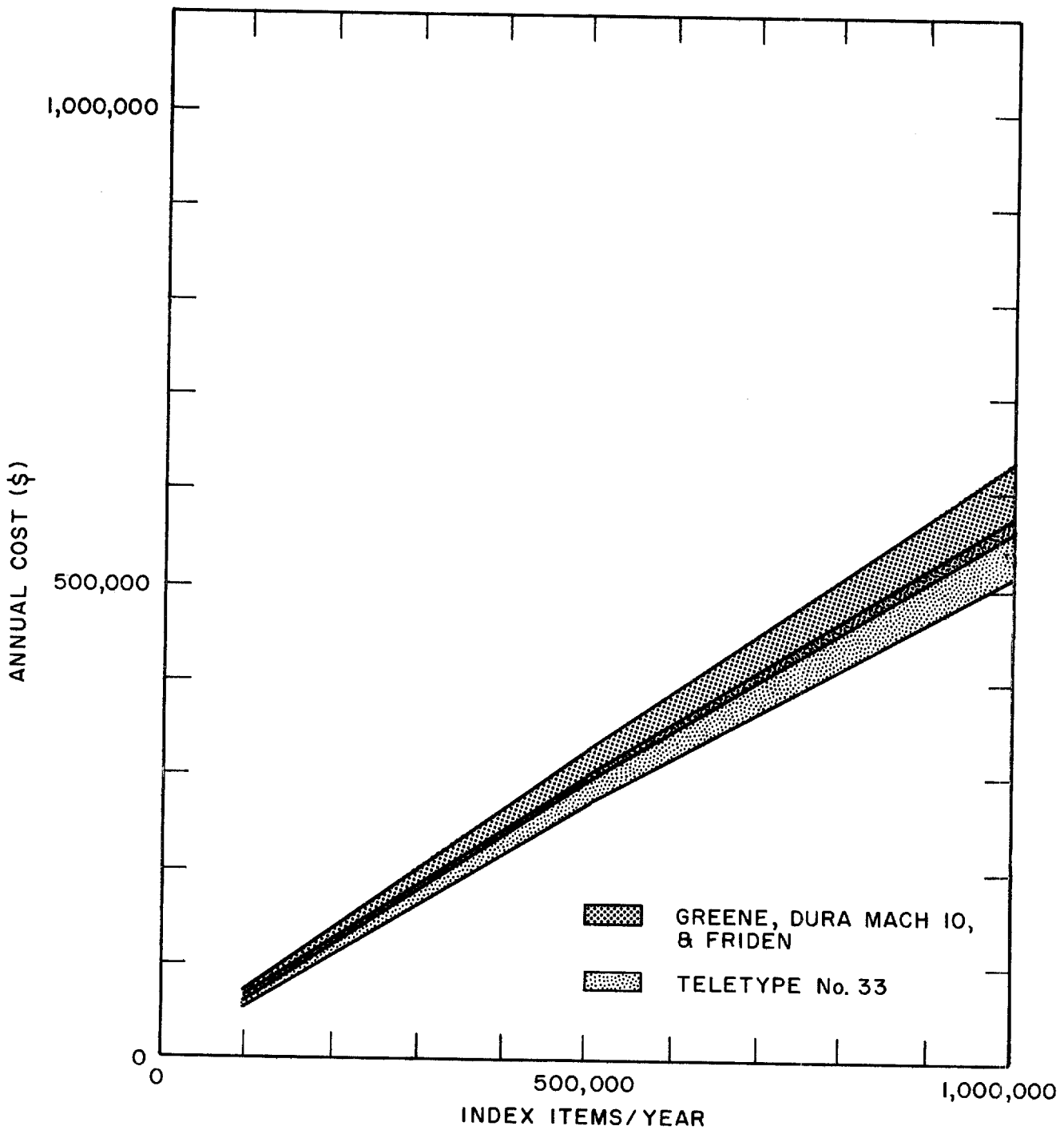
Punched paper tape equipments have been developed to a relatively high degree of sophistication and reliability during the last few years. In some cases, manufacturers have incorporated IBM Selectric typewriters as the basic element of their equipments to increase the operating ease and efficiency. From the automatic control standpoint various manufacturers have incorporated program control features into their equipments whereby special paper tape loops punched with specific program control codes can be used to greatly facilitate repetitive modes of operation. Six types of units were investigated in this study.*

The results of the cost evaluations for the punched paper systems are shown in Figs. 7.B-2 and 3. For all such equipments considered, the cost variations are relatively

*The equipments investigated were the Dura Mach 10, Friden Flexowriter, the Greene Letteriter, the Royal SDW 5000, the Teletype Model 33, and the Teletype Model 35. Additional units such as the Kleinschmidt Model 321, and the Remington Rand Sychrotape units are available, but were not evaluated. In the case of the Sychrotape, Remington Rand has sold all inventory and production rights to Radio Electronics, Inc., New York City, and this latter company is withholding further production until sales trends of their new magnetic tape typewriter units are evaluated.

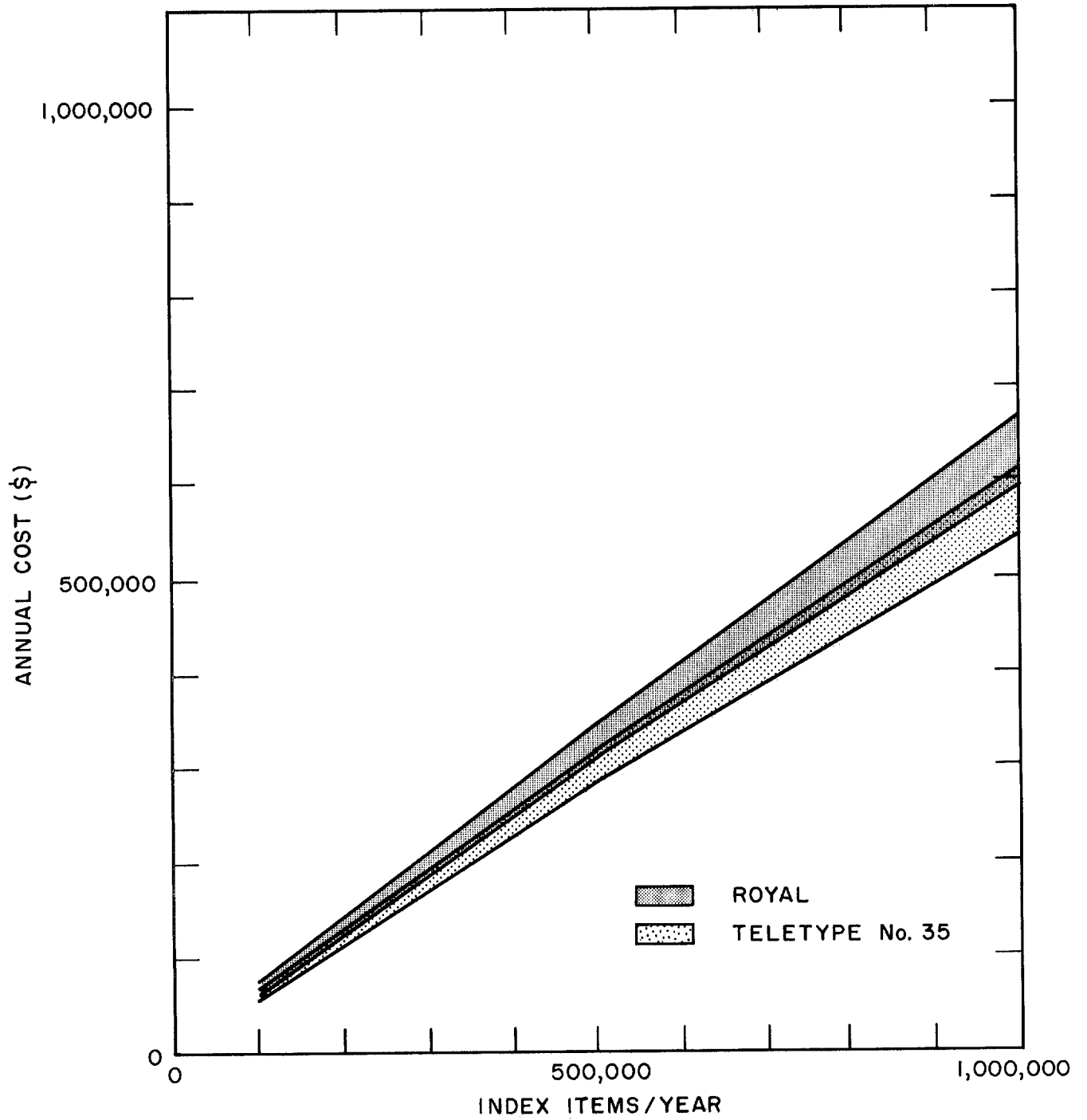
INPUT TRANSCRIPTION
Evaluations 7.
7.B.2.2.

SECRET



OPERATING COSTS
PAPER TAPE SYSTEMS

FIGURE 7.B-2



OPERATING COSTS
PAPER TAPE SYSTEMS

FIGURE 7.B-3

small--approximately 10 to 15 percent. This is due primarily to the fact that the major costs are the salaries of the typists, thereby subordinating the effects of the variable equipment costs, even though the various units range in purchase price from \$652 to \$4,000.

Large-scale data-entry installations utilizing punched paper tape equipments such as those visited at the Defense Documentation Center (DDC) and the National Library of Medicine (NLM) have been in operation for a significant period of time, and satisfactory operational results are reported. However, it appears that the use of punched paper tape systems is declining rather than expanding. The administrative personnel in charge of several such installations stated that they are considering a change-over to some data-input system other than the punched paper tape system because they believe that improvements in the overall efficiency of operation can be achieved. In addition, discussions with the data-processing personnel at a large title insurance company produced the information that they consider punched paper tape to be operationally unsuitable. In their opinion, the problem of making corrections is a difficult one, and in addition, merely the problem of handling very

INPUT TRANSCRIPTION
Evaluations
7.B.2.2.

SECRET

large amounts of paper tape in either fan-fold or roll form is too cumbersome to be considered feasible. Accordingly, they have decided to utilize page readers rather than punched paper tape equipments.

The correction times specified for use in these evaluations of the punched paper tape systems is 2.8 minutes per index item. It is to be noted that this is the largest time requirement specified for any of the systems considered. This specification was based primarily on the problems of achieving corrections of errors that are made at random points throughout the material. In the straightforward correction method such as that used at DDC, it is necessary to utilize the initial tape for automatic re-typing and re-punching of the item to the point where the first correction must be made--at this point the operator stops the automatic operation of the equipment, and types in the correction material. The operator then again places the machine into the automatic operation mode until the next error is encountered, or until re-typing and re-punching of the item is completed. Perhaps some improvement can be achieved by utilizing a more sophisticated semi-automatic correction technique such as that employed at NLM, but

INPUT TRANSCRIPTION
Evaluations
7.B.2.2.

SECRET

unfortunately, adequate information concerning time requirements for the techniques utilized there are not now available. The procedure at NLM is to prepare a special correction tape rather than using the straightforward method of preparing a new, corrected tape of the index item. The correction tapes must contain full identification of the item, the specific designation of the location within the index item where the correction(s) must be made, and the designation of the nature and content of the correction(s) to be made. At NLM, the correction tapes are then spliced into the original tapes containing the index items, and all information is then read into the data-processing system at one time. The actual corrections are then made automatically by the computer. The computer is also programmed to check the text material for specific types of errors such as those which might have been made in specification of descriptors, etc., and a printout of these additional errors is made for use in preparation of a second set of correction tapes. This additional correction information is then punched into tape, and entered into and processed by the computer, but no final check of the data entered is made.

INPUT TRANSCRIPTION
Evaluations
7.B.2.2.

SECRET

For the conversion from paper tape to magnetic tape,* the use of a service bureau at standard rates was assumed for relatively low rates of production. However, for higher production rates where the service bureau costs could prove to be excessive, purchase of the necessary conversion equipment was assumed (c.f., CHIVE/R-2-65). The cost of an operator for the conversion equipment was also incorporated into the cost analysis.

For the transcription, proofing, and correction of from 100,000 to 1,000,000 index sheets per year, the number of typists and proofing clerks required ranges from 9 to 103. A complete tabulation is provided in Section 7.B.3. (Table 7.B-2).

7.B.2.3. The Cathode-Ray Tube/Keyboard (CRT/KB) System

A greatly expanded utilization of the cathode-ray tube has been made for information display and manipulation during the past few years. Of the earlier special

*Both IBM and Radio Electronics, Inc., have recently introduced keyboard equipments for data storage directly on magnetic tapes. However, these units were not considered extensively in this study. Neither unit produces a tape that can be utilized on computer tape transports, and it is believed that the operational limitations and disadvantages of punched paper tape systems would be common to these new magnetic tape systems.

INPUT TRANSCRIPTION
Evaluations
7.B.2.3.

SECRET

applications of cathode-ray tubes, perhaps one of the most notable was the SAGE system where the cathode-ray tube is used as the display medium in conjunction with radar tracking of aircraft. This system incorporated character and symbol generation, and light pen techniques experimentally as early as 1954. Since that time, cathode-ray displays have been used in relatively sophisticated form in conjunction with computers of various manufacturers to expedite entry and analysis of numeric or alphabetical data stored in memory. It is only natural that techniques and equipments of this general nature should ultimately find widespread application in the general fields of data entry into, and data retrieval from, data-processing systems.* Equipments produced by six corporations have been

*There are at least six such CRT/KB systems reported to be installed for commercial use at this date. Also, smaller desk sets have been marketed for specialized uses such as stock price and activity quotations.

INPUT TRANSCRIPTION
Evaluations
7.B.2.3.

SECRET

evaluated in connection with the study of the CHIVE data-input problem.*

The basic configuration of the CRT/KB system is very simple. Fundamentally, it consists merely of a typewriter-like keyboard with which the operator enters data into the system. As he types, the characters are "instantaneously" displayed on the face of the oscilloscope, and are simultaneously stored in a temporary buffer register. Generally, some form of electronic pointer or cursor is provided to designate on the face of the display tube the position at which the next character shall be displayed or revised. Ideally, the central data-processing unit is not required in connection with any of this initial data-entry process associated with the CRT/KB unit. It is only when the operator has completed typing all of the data involved, and has checked and corrected it as required, that the central processing unit is involved in any way. At this

*These corporations are Burroughs, Data Display Inc., IBM, Information Products Corporation, Philco, and Teleregister. However, it should be noted that the Burroughs and Philco equipments are not production items, and that the corresponding pricing information is not firm. Other companies are also producing equipments capable of character generation and oscilloscopic display. Raytheon, Inc. has recently announced a very sophisticated system designed primarily for military applications. Raytheon plans to produce a less expensive version for commercial applications, but is not yet prepared to provide any technical specifications, or cost and delivery schedules.

INPUT TRANSCRIPTION
Evaluations
7.B.2.3.

SECRET

SECRET

time, the operator presses some form of a data-entry key, thereby indicating to the system that, as appropriate, the central data-processing unit should be interrupted for acceptance or processing of the data stored in the CRT/KB buffer.

One of the major features of the CRT/KB system is the facility with which corrections can be made. In all of the equipments considered in this study, simple "substitution" types of corrections can be made merely by moving the display pointer until it designates the first character to be corrected, and then typing the correct characters so that the operator can check them directly, and the corresponding changes are simultaneously made in the buffer storage. An even greater facility is realized in those cases where insertion of characters, words, or even sentences must be achieved at some point within the body of text that has already been entered into the CRT/KB system. In this case, the operator needs only to designate the point of insertion, press some type of "Insert" key, and then type the characters that are to be entered. The text material following that which is being inserted is automatically advanced--both on the face of the display tube and in the buffer storage. This facility is incorporated within the CRT/KB units produced

INPUT TRANSCRIPTION
Evaluations
7.B.2.3.

by some of the manufacturers,* but in other cases it would necessarily be provided in conjunction with access to and special programming for the central data-processing unit.

The cost information for the CRT/KB units is provided in Figures 7.B-4 and 5. It is to be noted that the cost for a system based on the IBM 2250 equipments is very much greater than that of any of the other units. This IBM equipment is much more sophisticated than is required for the CHIVE input problem, but it was considered desirable to include it, nevertheless, for comparative purposes. The unit is capable of as many as 3,848 character positions** on the face of the tube, and provides the capability of graphical display and light-pen operations.***

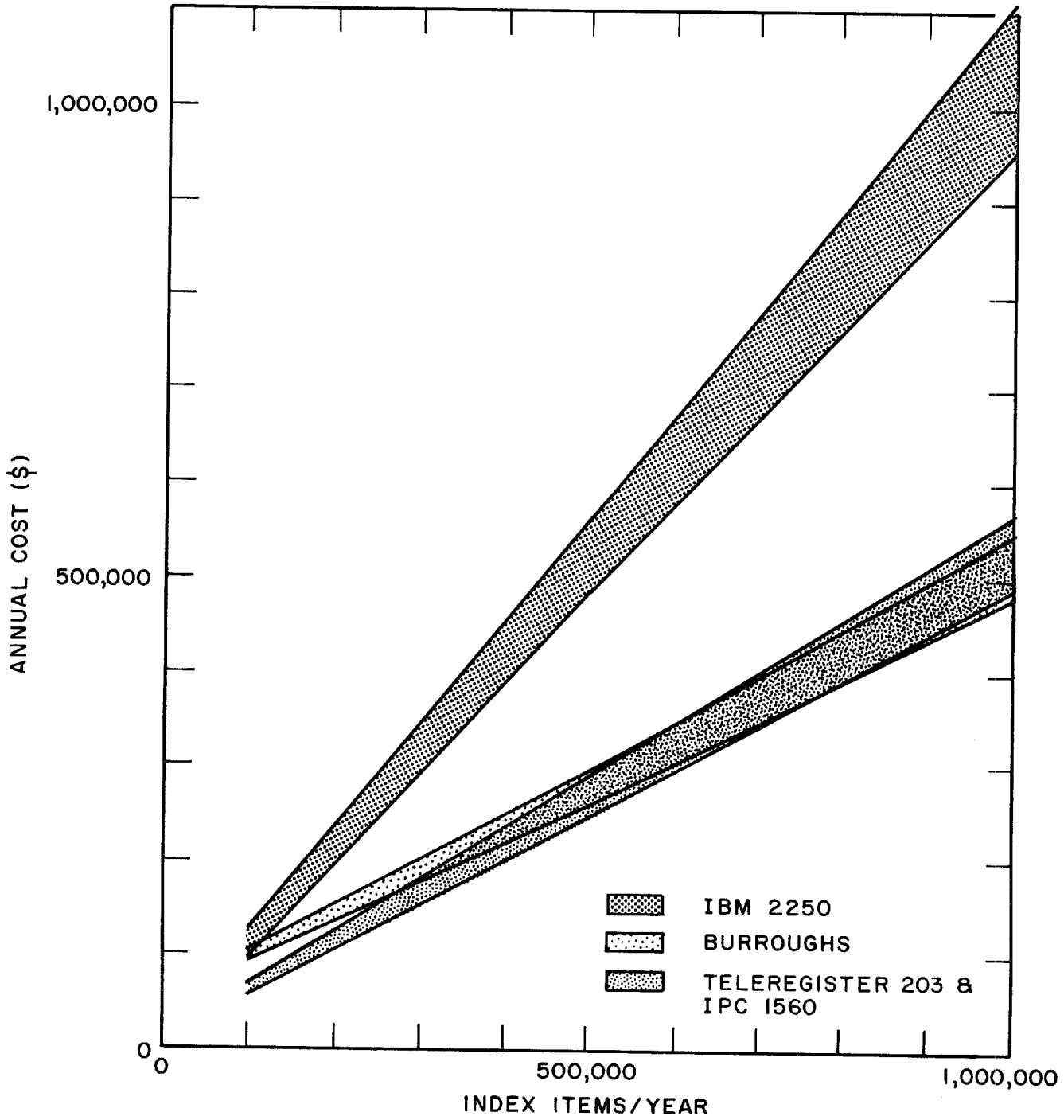
*The insertion capability is provided in the Burroughs, IBM and Philco equipments.

**However, where as many as eight display stations are connected to one control unit, the number of characters per station which can be simultaneously displayed may be limited to as few as approximately 1,000 characters. IBM has not yet made information of this type generally available.

***IBM also provides the Model 1015 type of unit, but this unit is designed primarily for use in data inquiry applications. As many as 60 Model 1015 inquiry stations can be operated in conjunction with one control unit (Type 1016), but only one station can be active at any one time. Consequently, it would not be suitable for a data-entry application of the type considered in the study. Also, the display tube is of the storage type; this would probably introduce some minor complications in connection with correction of errors.

INPUT TRANSCRIPTION
Evaluations
7.B.2.3.

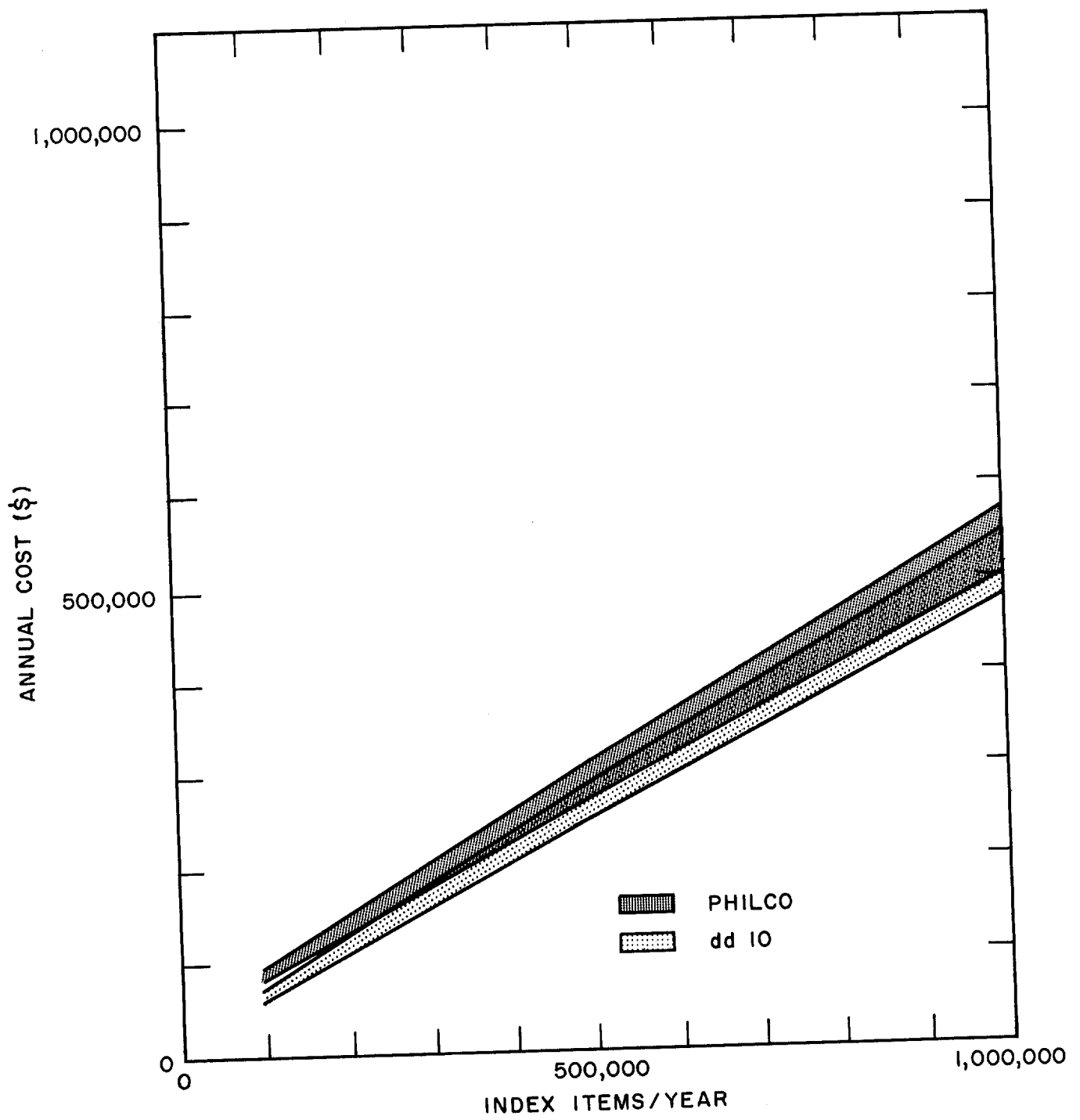
SECRET



OPERATING COSTS
CRT/KB SYSTEMS

FIGURE 7.B-4

SECRET



OPERATING COSTS
CRT/KB SYSTEMS

FIGURE 7.B-5

Of the units manufactured by the other corporations, the costs of processing one million index items per year do not differ by a significant amount. However, for processing of 100,000 items per year, the cost of the Burroughs and the Philco equipment is substantially greater. This is because of the fact that those system designs are based on large central control units which are required regardless of the number of data entry and display stations connected into the system--as contrasted to designs on a modular control-unit basis.

For the entry, proofing, and correction of from 100,000 to 1,000,000 index sheets per year, the number of typists required ranges from 7 to 81. A complete tabulation is provided in Section 7.B.3. (Table 7.B-2). This constitutes the lowest typist requirement of all of the systems considered, because of two factors. First, typists (as compared to slower keypunch operators) are utilized for the entry of the data into the system. Secondly, evaluations were based on the specification that the correction time required by the CRT/KB systems is shorter than it is for any of the other systems considered. In this case, a correction time of 0.8 minutes per indexing record was specified.

INPUT TRANSCRIPTION
Evaluations
7.B.2.3.

SECRET

7.B.2.4. Page Reader Systems

Commercially produced optical character readers have been available for approximately ten years. However, it has been only in the last two or three years that such readers have become relatively sophisticated, configured to read full pages of printed information (as well as single lines in rigidly controlled locations on the document), and capable of providing a reading accuracy equal to or better than one reject per thousand characters and one substitution per 10,000 characters (with relatively high-quality typewritten material).* Improvements in accuracy have been due not only to improved techniques of character recognition, but also, to a significant extent, to the cooperation of typewriter manufacturers in exercising better control of the characters produced by their typewriters. Previously, the control of the alignment and the detail of the characters had been inadequate.

During the past few months, the character reading situation has reportedly been improved even further.

*Higher accuracies are generally obtained with characters produced with high-quality printing equipments. Printing of this type is termed "pre-printed" material, and typewritten characters fall into the "post-printed" category.

INPUT TRANSCRIPTION
Evaluations
7.B.2.4.

SECRET

SECRET

Unfortunately, these reports have not yet been backed up by publications, but they were provided by reliable sources and are therefore considered to be of significance. In brief, it was stated that during recent acceptance tests of a commercial page reader, there were no substitutions whatsoever, and the reject rate was so low that tabulation of rejects was discontinued in deference to the operational aspects of the tests. These tests were made with relatively high-quality print control, but it seems reasonable to infer that reading of printing of somewhat lower quality should be possible at a moderate sacrifice of accuracy.

At the present time, only two companies are offering page readers for sale that can be considered to be production equipments. These companies are Farrington Electronics and Recognition Equipment. Farrington has delivered at least ten page-reading units capable of reading a single font--the Self-Check font--and Recognition Equipment has delivered four page-reading units capable of reading such fonts as IBM 407E, IBM 1403, and E13B. In addition, Farrington has delivered approximately 90 document readers, and Recognition Equipment has delivered at least one.

INPUT TRANSCRIPTION
Evaluations
7.B.2.4.

Two other companies--Rabinow Electronics and Philco Corporation--are scheduling deliveries of their new types of page readers during the first or second quarter of 1965. The Rabinow equipment is to be capable of reading only the ASA font, but the company personnel claim the capability of producing a multi-font machine on special order. The Philco machine is to be capable of reading several fonts--the maximum number being of the order of approximately 16. However, it must be noted that this machine is not really a multi-font machine capable of reading intermixed fonts without some sort of control being exercised. Specifically, each field of text printed with characters of a single type of font must be designated by means of plus signs, or the machine must be programmed to shift the recognition programs for arbitrarily specified, pre-designated fields of text, there being only one type of font per field.*

In addition to the current or imminent producers of page readers as enumerated above, GE, IBM, and RCA appear to have plans for marketing commercial page readers in the near future. However, no significant amount of

*There is one exception--more than one font can be read simultaneously in a given field provided that corresponding characters of all such fonts have identical aspect ratios.

INPUT TRANSCRIPTION
Evaluations
7.B.2.4.

information concerning machine characteristics, and costs and delivery schedules, is available for evaluation (as of December, 1964). In a final category are Sylvania, National Cash Register, Burroughs, and Link--all of which have character reader programs under way, but no announced or implied intention of producing a "standard" type of page reader within the foreseeable future.*

It is not appropriate in this report to attempt to provide detailed information concerning the basic features and limitations of each of the candidate page readers.** Investigations and evaluations in depth would be more appropriate during a procurement phase of the CHIVE program. Further considerations are limited here to the cost and manpower aspects of the anticipated CHIVE indexing program.

Cost information for single-font machines is provided for off-line page-reader systems incorporating the Farrington reader, the Rabinow reader, and the

* However, the Sylvania personnel have indicated that they would be glad to receive any request for bid that may be issued in conjunction with procurement of a page reader.

**The SRI project personnel have on file brochures and contact reports concerning the salient details of the page readers that have been developed to a reasonable degree at this time.

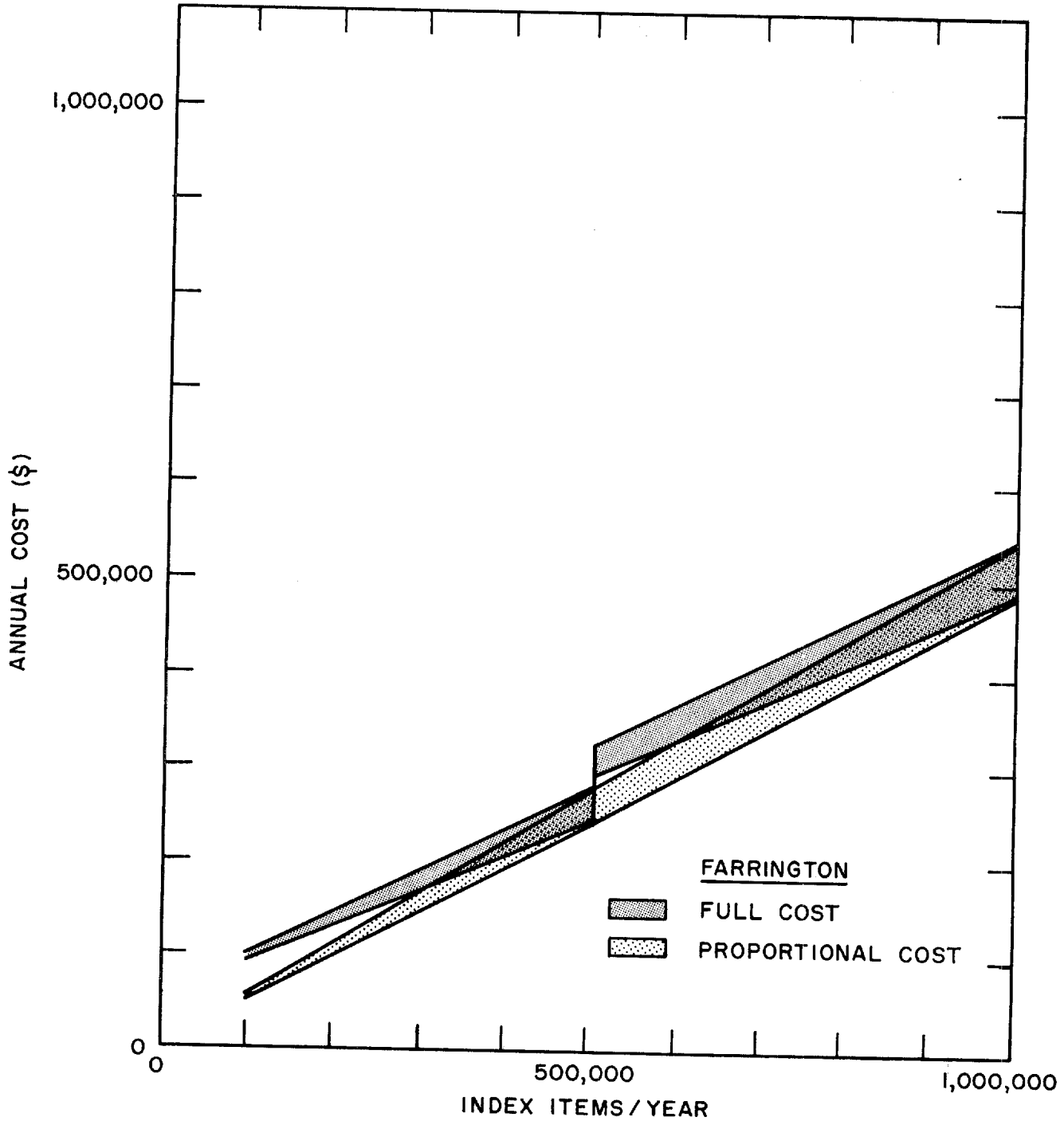
INPUT TRANSCRIPTION
Evaluations
7.B.2.4.

Recognition Equipment reader in Figures 7.B-6, 7, and 8, respectively. In each case, cost information is provided for two sets of conditions. The upper band is required to pay the full cost of the page-reading equipment. The lower band corresponds to the situation where it is assumed that the CHIVE operation is required to pay only the proportional page reader cost, based on the fraction of the 8-hour shift that is actually utilized for the CHIVE operation.

In all cases, a text density of 60% of the maximum possible is assumed, and it is further assumed that each page reader is operated at an effective speed de-rated to 75% of its nominal capability. In the cases of both the Rabinow and the Recognition Equipment systems, costing was based on inclusion of 8000 words of memory in the small computer associated with the page readers. Basic page-reading operations can be achieved with 4000 words, or even with 2000 words, but it is believed that at least 8000 words of memory are required in order to be able to provide a reasonable amount of on-line pre-editing. In the case of the Farrington machine, no computer memory is provided--all programming and control are achieved by means of a plug-board arrangement.

INPUT TRANSCRIPTION
Evaluations
7.B.2.4.

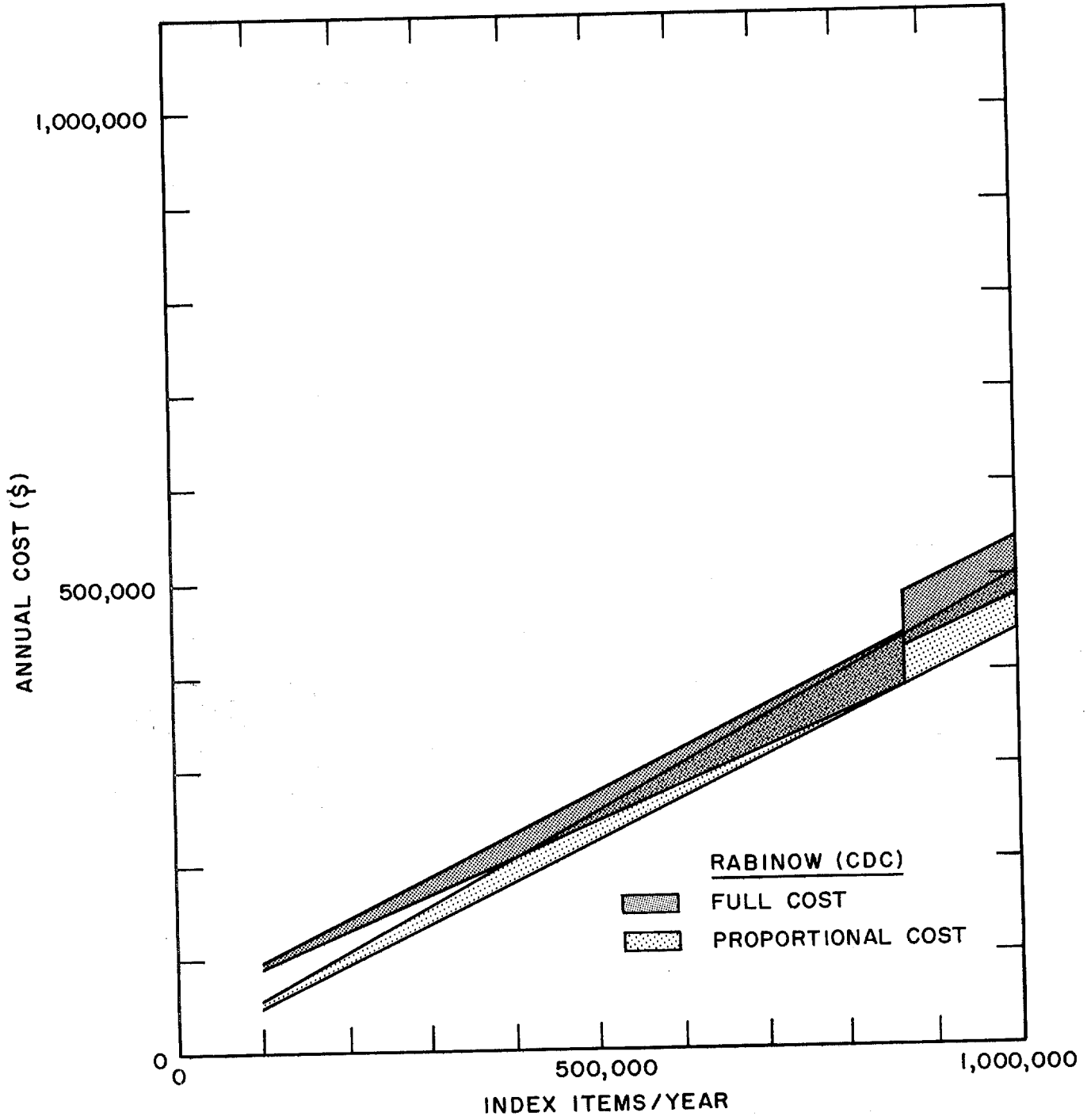
SECRET



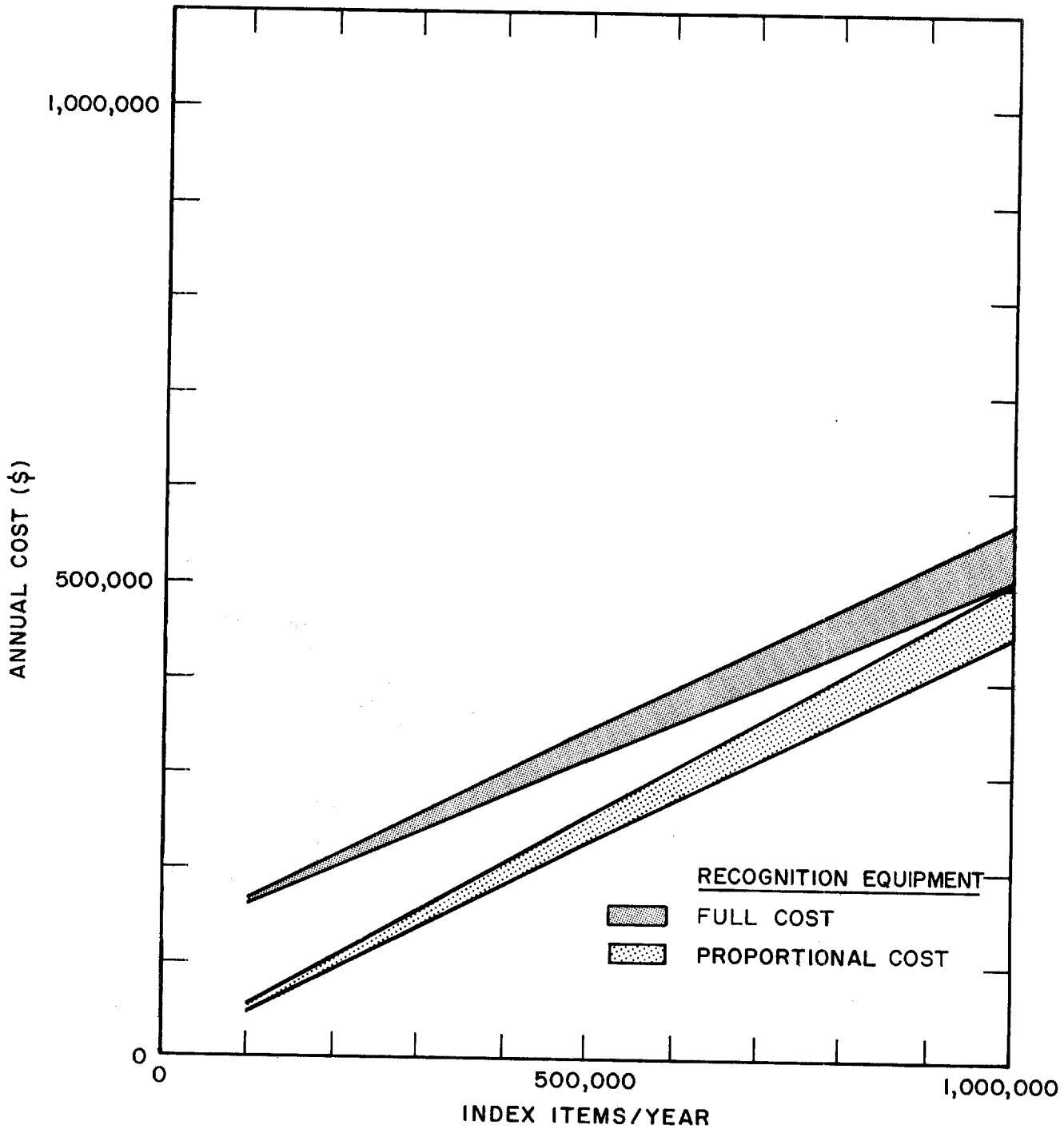
OPERATING COSTS
FARRINGTON PAGE READING SYSTEM

FIGURE 7.B-6

SECRET



OPERATING COSTS
RABINOW (CDC) PAGE READING SYSTEM
FIGURE 7.B-7



OPERATING COSTS
RECOGNITION EQUIPMENT PAGE READING SYSTEM

FIGURE 7.B-8

The discontinuities of the upper cost bands of the systems based on the Farrington and the Rabinow readers (Figures 7.B-6 and 7) are caused by the fact that one reader is not sufficiently fast to process as many as one million items per year. Accordingly, the discontinuities represent the added costs of an additional reader in each of the two cases.

The correction time required for the page-reader operations was specified as 1.0 minute per item, which is well toward the lower end of the range for all of the systems considered. This specification was based on the assumptions that simple errors can easily be deleted through the use of white over-paint (Snopake, etc.) or adhesive strips in preparation for over-typing, and that more extensive errors can be accommodated by computer-programmed processing of coded correction entries at the end of the item.

The costs indicated in Figures 7.B-6, 7, and 8 were all determined for off-line operation with single-font page reader equipments. For the Recognition Equipment reader, a configuration designed for on-line operation with five fonts would cost approximately \$8292 additional

INPUT TRANSCRIPTION
Evaluations
7.B.2.4.

SECRET

per year.* No figure supplementing Figure 7.B-8 has been made because this additional cost would not cause a significant shift of the cost bands of that figure. The greatest upward shift would be only approximately 5 percent at the 100,000-items-per-year end of the upper band (full cost support by the index operation), tapering down to only a 1.5 percent increase at the 1,000,000-items-per-year end of that band. For the lower band (proportional cost support by the CHIVE operation), the upward shift would be only approximately 0.9 percent. However, the costs for the on-line, five-font configuration of the Recognition Equipment unit are included in Section 7.B.3. (Table 7.B-1).

It is not feasible at this time to provide graphs of the costs that could be anticipated for systems based on page readers procured from IBM RCA, GE, or any of

*This increase is due (a) to an added cost of \$9492 per year for the additional equipment required to increase from 60 to 240 characters the number of patterns that can be recognized, (b) to the added cost of approximately \$4,240 per year for interface equipment, and (c) to a decreased cost of \$5440 made possible by the elimination of the requirement for the off-line magnetic tape transport system. (It must be noted that the interface cost is that specified for operation in conjunction with an IBM 7090 computer. Recognition Equipment does not as yet have information for the IBM 360 computers on which to base cost specifications).

INPUT TRANSCRIPTION
Evaluations
7.B.2.4.

SECRET

the other potential suppliers, because no reliable specifications of the costs of their equipments have yet been made available. It can only be stated that estimates of the purchase cost of equipments obtained from these manufacturers range from \$500,000 to \$1,000,000. For this increased cost range, and using the upper band of Figure 7.B-8 as a reference, an upward shift of approximately \$10,000 to \$110,000 (per year) would occur.* The cost increases for the lower curve (proportional support by the indexing operation) would of course be a function of the reading speed of the readers under consideration.

For the transcription, proofing, and correction of from 100,000 to 1,000,000 index sheets per year, the number of typists and proofing clerks required ranges from 8 to 83. A complete tabulation is provided in Section 7.B.3. (Table 7.B-2).

7.B.3. SUMMARY

The results of the cost evaluations of the four major types of data input systems considered in this

*These shifts in the plot are based on a purchase cost of approximately \$450,000 for the Recognition Equipment page reader.

INPUT TRANSCRIPTION
Summary
7.B.3.

SECRET

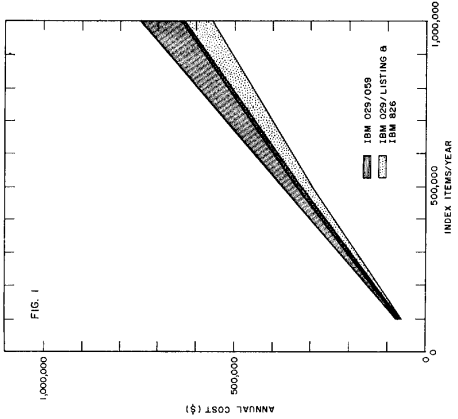
study are combined graphically in Figure 7.B-9, and in tabular form in Table 7.B-1 to facilitate direct comparisons. Of prime significance is the fact that with one exception,* the predicted costs of all systems for processing of 1,000,000 indexing records per year are of the same order of magnitude in the range from approximately \$440,000 to \$750,000. This is due primarily to two factors. First, the major cost for all systems is the salaries of the typists (or keypunch operators) and the proofing clerks. Consequently, the influence of large variations in equipment costs is subordinated--even though these costs differ in some cases by an order of magnitude.** Secondly, the more expensive equipments in general seem to be associated with the faster, more efficient operations. Because of the preponderant salary costs, even a relatively

* The exception is the IBM 2250 Display Unit in the CRT/KB category (Figure 7.B-4). As stated in Section 7.B.2.3., this equipment is more sophisticated than required for the CHIVE indexing application, but it was nevertheless included for reference purposes.

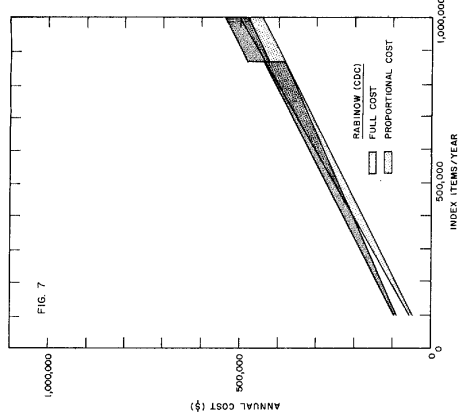
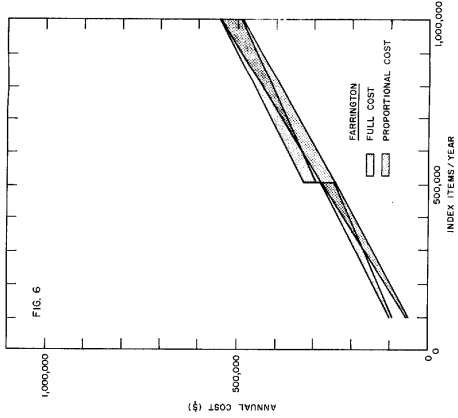
**For example, the Teleregister CRT/KB units cost approximately \$1,900 per unit per year, as compared to the Teletype Model 33 paper tape punch at approximately \$170 per unit per year.

SECRET
 INPUT TRANSCRIPTION
 Summary
 7.B.3.

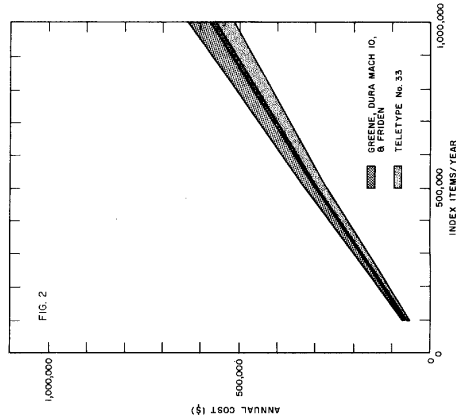
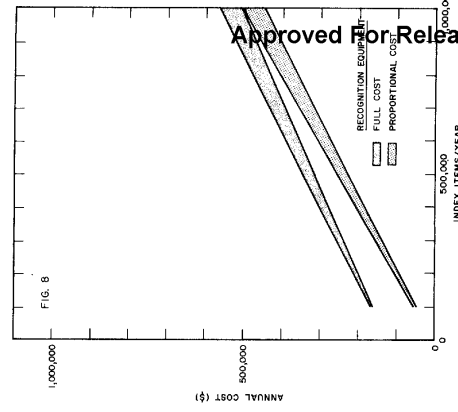
SECRET
 INPUT TRANSCRIPTION
 Summary
 7.B.3.



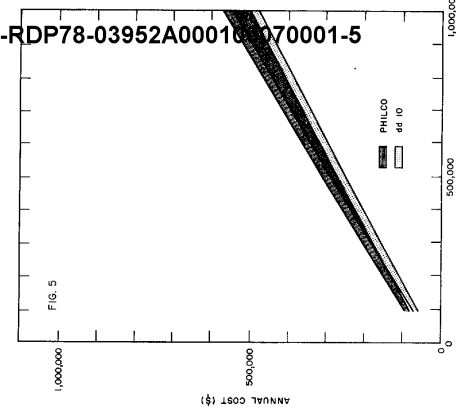
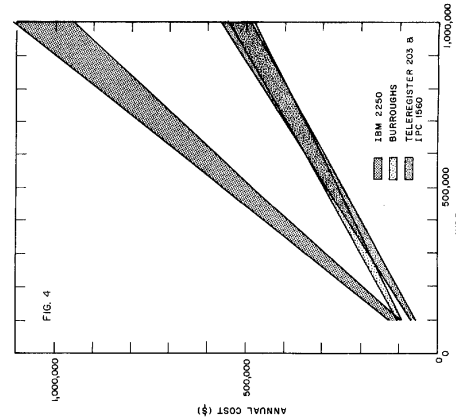
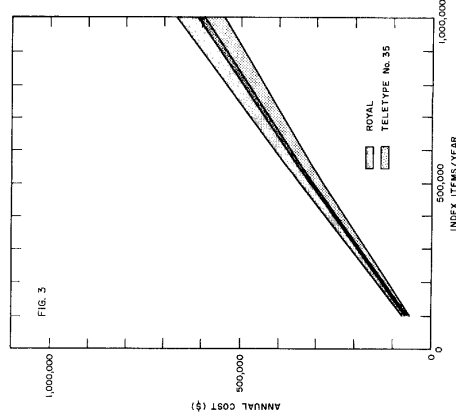
PUNCHED CARD SYSTEMS



PAGE READING SYSTEMS



PUNCHED PAPER TAPE SYSTEMS



CRT /KB SYSTEMS

FIGURE 7.B-9 COMPARATIVE OPERATING COSTS OF CANDIDATE INPUT SYSTEMS (off-line)

COST SUMMARY
 OF
 CANDIDATE INPUT SYSTEMS

System		Yearly Cost (Thousands of Dollars)							
		100,000 Items/Yr.		500,000 Items/Yr.		1,000,000 Items/Yr.			
		Min.	Max.	Min.	Max.	Min.	Max.		
PUNCHED CARDS	IBM 029/059	67.1	77.8	327	385	632	748		
	IBM 029/Listing	66.6	71.8	303	341	570	647		
	IBM 826	57.0	70.0	284	327	550	632		
PUNCHED TAPE PAPER	Dura Mach 10	58.7	70.5	286	322	549	621		
	Flexowriter	58.3	70.1	284	320	546	617		
	Greene Letterriter	58.7	70.5	286	322	549	621		
	Royal SDW 5000	60.8	73.2	296	334	570	645		
	Teletype #33	54.6	65.3	266	298	510	574		
	Teletype #35	57.0	68.3	277	312	533	601		
CRT/KB	Burroughs *	90.5	105	259	297	481	549		
	dd 10	60.6	73.6	246	285	477	548		
	IBM 2250	97.3	129	485	565	956	1,110		
	IPC 1560	53.1	67.8	247	288	487	565		
	Philco *	82.3	95.5	267	307	501	573		
	Telerigister 203	50.4	64.2	250	294	500	577		
PAGE READER	Off-line single-font	Farrington **	Full Cost	95.4	101	247	279	488	546
			Prop. Cost	53.3	58.7	246	278	486	544
		Rabinow **	Full Cost	91.3	96.7	243	275	479	538
		(CDC)	Prop. Cost	48.6	54.0	222	254	444	502
	Recognition Equipment	Full Cost	170	175	321	353	516	574	
		Prop. Cost	49.4	54.8	226	258	453	511	
	On-line 5-font	Recognition	Full Cost	178	184	330	361	524	583
		Equipment	Prop. Cost	49.9	55.2	228	260	457	515

* Not a production item -- pricing is approximate.

small advantage in speed of transcription or error correction can offset a substantial disadvantage in equipment costs.

The uniformity in costs is not so pronounced in all cases for lower production rates, particularly at 100,000 indexing records per year. This non-uniformity is due to relatively large, "non-modular" equipment costs which are not averaged out significantly at the lower production rates. The most significant cases of this type are the systems based on Burroughs and Philco CRT/KB units, and the systems based on page readers where full (equipment) costs are assumed to be borne by the indexing operation.

The number of typists (or keypunch operators) and proofing clerks required for the various systems are tabulated in Table 7.B-2. For the extreme cases, the predicted numbers range from 70 to 129 typists and clerks for production of 1,000,000 indexing records per year. These numbers represent very substantial staffing problems. It is recommended that more extensive and continuing in-house tests be made under conditions as representative of anticipated operating conditions as possible to validate the initial timing assumptions and to uncover ways to improve the operation.

INPUT TRANSCRIPTION
Summary
7.B.3.

Table 7.B-2

SUMMARY OF REQUIREMENTS FOR TYPISTS/KEYPUNCH
OPERATORS AND PROOFING CLERKS FOR CANDIDATE
INPUT SYSTEMS

SYSTEM		Number of Typists (or Key punch Operators) & Proofing Clerks					
		100,000 Items/Yr.		500,000 Items/Yr.		1,000,000 Items/Yr.	
		Min.	Max.	Min.	Max.	Min.	Max.
PUNCHED CARDS	IBM 029/059	11	13	54	65	107	129
	IBM 029/Listing	10	11	47	54	94	108
	IBM 826	8	10	40	47	80	93
Punched Paper Tape		9	11	46	52	91	103
CRT/KB		7	9	35	41	70	81
Page Reader		8	9	36	42	72	83

SECRET

--326--

SECRET

Many measures might be used to make a relative comparison of the four approaches chosen for this analysis. Some of these measures are noted in Table 7.B-3. Early in the analysis it was suggested that cost was the main factor, with staff size another important consideration. For this reason, most of the attention was directed to these two factors.

In any event, since the personnel costs are preponderant, and since providing working space and facilities for the personnel also constitutes a significant problem, all reasonable efforts should be expended to improve operating efficiency in order to minimize the personnel requirements. In this connection, the decision by the OCS and the CHIVE staffs to plan for use of a page-reading system offers the possibility of some gains in this direction with the hope of processing many or all of the indexers' work sheets directly on the page reader, thus eliminating the transcription process in part, or perhaps even completely (c.f., Section 7.B.1.). If feasible to any significant extent, this could conceivably result in very appreciable reductions in costs, space, and manpower requirements.

INPUT TRANSCRIPTION
Summary
7.B.3.

Table 7.B-3
Approved For Release 2000/05/08 : CIA-RDP78-03952A000100070001-5
 Comparison of Alternative Input Transcription Methods

Factor	Punched Cards	Punched Paper Tape	CRT/KB	Optical Page Reader
Cost for Final System (annual cost in dollars)*	550-748,000	510-645,000	477-577,000	457-583,000
Staff for Final System**	80-129	91-103	70-81	72-83
Floor Space for Final System (sq. ft.)***	3645-6210	3640-4160	2800-3240	3180-3660
Time Lag to get final indexing record into computer system	Moderate	Moderate	Very small	Small
Ease of Error Notation & Correction	Moderate	Moderate	Very good	Good
Compatibility with current staff, equipment, & procedures	Good	Some Difference	Some Difference	Some Difference
Possible Maintenance problems	Considerable Maint. experience exists for these equipments.....		Relatively little maint. experience exists for these equipments.....	
Space required for Materials Storage (in addition to space noted above)	Some	Some	None	Some
Need Special Controls on print quality	None	None	None	Definitely
Training Required by Transcription Clerk	About the same for all systems.....			
Capability for Immediate Recall of Earlier Records for Review or Manipulation	Moderate	Moderate	Very good	Good
Possibility of Indexers Eventually Entering Information without Paper Transcription	Poor	Poor	Very good	Good
Special Problems	Display limited to 500-1000 char. at one time. Possible flicker or resolution (human factors) problems.			

* See Table I in CHIVE R-2-65

** See Table II

*** See Table III

SECRET

SECRET

SECRET

Planning to use a page reader suggests a number of follow-on tasks. An indication of the activities that such a decision dictates is provided by the list in Table 7.B-4. One of the principal tasks noted is that of choosing a specific page reader model. Some tentative specifications are set forth in Table 7.B-5. At the time these specifications are completely detailed, other models than those herein considered may appear as likely candidates.

INPUT TRANSCRIPTION
Summary
7.B.3.

SECRET

Subsequent Tasks for OCR and OCS After Choosing the Page Reader Approach
(not necessarily in sequential order)

- A. Make arrangements with other potential users to share the page reader facility.
 - 1. Agreement on type fonts (choice of specific fonts, and their sizes, and the number of characters required).
 - 2. Agreement on location of equipment (convenience, power, refrigeration, noise, structural considerations).
 - 3. Initial time scheduling.
 - 4. Possible need by other users for peripheral page reader features (off-line magnetic tape transport, printer, multiple page hoppers).
 - 5. Initiate study of other OCR application's within broad CHIVE context (e.g., reading library requests or acquisition forms).
- B. Establish the Basic System Specification (see Table 7.B-5).
 - 1. Equipment.
 - 2. Environment.
 - 3. Delivery and maintenance.
- C. Choose the specific page reader model.
- D. Establish Document Format.
 - 1. Forms design.
 - 2. Determination of forms suppliers.
 - 3. Procurement and test of forms.
- E. Establish Control of Typing Quality.
 - 1. Specification, selection, and procurement of special typewriters.
 - 2. Specification of materials (paper, ribbons, correction material).
- F. Establish Tolerable Error Levels.
 - 1. Determine relative criticality of various elements of the index record.
 - 2. Determine what error checking or proofing procedures (e.g., use of check digits or table look-up verification for selected elements) are reasonable.
- G. Prepare instruction manual for typists.
- H. Perform computer programming of the page reader control functions.

Table 7.B-5

TENTATIVE PAGE READER SPECIFICATIONS - PROJECT CHIVE

A. Fonts

1. Need a capability of handling two or more fonts with 60-character sets as follows:
 - a. A standard IBM electric typewriter font (Elite), 10 or 12 character per inch and 6 or 8 lines per inch. Upper case alphabetic characters will be used.
 - b. One NPL font compatible with the IBM 1403 printer in size and spacing; up to 132 characters per line.
2. Require a reader which can handle one font per page and can change from one font to another without manual intervention (a special symbol may be employed to signal a change in font).

B. Reading Speeds

1. Reading speeds can be specified only for throughput for the system. Eventual requirement is for 1,000,000 records of 1000 characters each plus special query requirements. This volume is to be read in a normal year of 52 weeks, 40 hours per week. This requires a system reading speed of 150 or more characters per second throughput with a page throughput of 10-15 pages per minute; each page to contain an average of 1000 characters. This is a total system capability--one or more readers.

SECRET

C. General Features

1. Page sizes possible:
 - a. Normal page sizes from 3 x 5 to 8.5 x 14 inches
 - b. Normal IBM 1403 printer paper - 10 x 14 inches (up to 132 char/line).
2. Possibility of intermixing different size pages in the input hopper.
3. Need for some means of segregating rejected pages from error-free pages in the output stacking station (or stations).
4. Input stacking station should have a capacity of 50 or more pages with a capability of adding to the stack while the reader is running. Output stacking station(s) should have a capacity of at least 200 sheets contingent on the relative position of the input and output stacking stations.
5. Need a capability of marking rejected lines on the rejected page.
6. Equipment should be capable of ignoring at least one background color so that pre-lined forms may be used.
7. The reader should be sensitive to mark sensing.
8. Capability for handling normal business paper stock.

D. Characteristics of Page Reader's Control Processor--
If System Acquired has a Control Processor

1. Need 16,000 or more 8 bit characters of core storage or the equivalent
2. Interfaced for on-line operation with IBM 360 with a controlled interrupt capability (parity and character length compatibility is implied).

SECRET

SECRET

3. Ability on the part of the processor to signal the system to mark the lines (and offset stack rejected document) or give the line and character count of an unrecognizable character.
4. Ability to send documents to related output stacker(s) under program control.
5. Processor should be able to signal the equipment to rescan a line.
6. Ability to do some simple formatting and editing of information from the document before it is transferred to the IBM 360.
7. Software package for the control processor should be compatible with Operating System/360.

E. Operation Guarantees

1. Delivery Schedule. Delivery required to interface with the IBM 360 installation schedule of February - April 1966.
2. Maintenance
 - a. If equipment purchased:

Either (1) complete maintenance training for CIA employees and on-call (4 hrs.) engineering service,

or (2) contract maintenance including on-call (4 hrs.) engineering maintenance.
 - b. If equipment rented. Full preventative and engineering maintenance.
3. Performance. No more than 10 rejects per 100,000 input characters from normally prepared cleanly typed documents. No more than 10 characters substituted per 100,000 input characters from normally prepared cleanly typed documents.

SECRET

F. Environmental Conditions

To meet normal government specifications.

G. Spare Equipment Required

1. If purchased. Recommended list of spare parts and costs (unless provided for in maintenance contract) derived from a mean-time-to-failure study of equipment components.
2. If rented. Sufficient spare parts to allow normal maintenance replacement within 4 hours.

Appendix 7.C.

EDP OUTPUT FORMATS

The standing report definitions which will be provided initially have been enumerated in this appendix. In order to demonstrate the makeup of these reports, the layouts of representative types have been prepared.

7.C.1. DOCUMENT INDEX FILE REPORTS

Figure 7.C-1, Document Identification and Term Report, consisting of three parts, shows the type of report which may be generated as the result of a search of the Document Index Record file.

The cover page, which may actually be a back cover since it contains information extracted from each of the records in the reported file, lists the document access number for all document reports, in the order of their selection. As a summary sheet it can be used by the requestor to demarcate all of those documents which he wishes retrieved for more detailed examination.

The first page presents, for subsequent reference, the specifications upon which the search was based. The

OUTPUT FORMATS
File Reports
7.C.1.

Figure 7.C-1

DOCUMENT ID-TERM REPORT-COVER PAGE

** SECURITY CLASSIFICATION **

DATE _____ DOCUMENT INDEX SEARCH _____ PAGE COVER _____
* * *

REQUESTOR NAME: _____.

REQUESTOR LOCATION: _____.

REQUEST NUMBER: _____.

DOCUMENTS SATISFYING THIS REQUEST:

X _____ X	X _____ X	X _____ X	X _____ X
X _____ X	X _____ X	X _____ X	X _____ X
X _____ X	X _____ X	X _____ X	X _____ X
X _____ X	X _____ X	X _____ X	X _____ X
X _____ X	X _____ X	X _____ X	X _____ X

COVER
PAGE

** SECURITY CLASSIFICATION **

SECRET
- 336 -

SECRET

Figure 7.C-1

DOCUMENT ID - TERM REPORT - FIRST PAGE

* * SECURITY CLASSIFICATION * *

DATE _____

DOCUMENT INDEX SEARCH

PAGE 1

* * *

REQUEST NUMBER _____.

SEARCH LOGIC:

JOB RETRIEVE: x _____ x
 x _____ x

TESTS: x _____ x
 x _____ x

PROCESS: x _____ x
 x _____ x
 x _____ x

END JOB: x _____ x

FIRST PAGE

* * SECURITY CLASSIFICATION * *

-337-
SECRET

SECRET

Figure 7.C-1

DOCUMENT ID - TERM REPORT - BODY PAGE

** SECURITY CLASSIFICATION **

DATE _____.

DOCUMENT INDEX SEARCH

PAGE 2 → N

SECRET

-338-

<u>ACCESSION NUMBER</u>	<u>SERIES ID NUMBER</u>	<u>TERMS</u>	
		<u>(TAG: VALUE)</u>	
X _____ X	X _____ X	X _____ X	X _____ X
		X _____ X	X _____ X
		X _____ X	X _____ X
		X _____ X	X _____ X
X _____ X	X _____ X	X _____ X	X _____ X
		X _____ X	X _____ X
X _____ X	X _____ X	X _____ X	X _____ X
		X _____ X	X _____ X

SECRET

DOC ID N
TERMS
REPORT

** SECURITY CLASSIFICATION **

SECRET

remaining pages highlight the access number and identification of each of the documents whose index records meet the retrieval criteria and it then presents selected terms appearing in that index record.

Figure 7.C-2, Abbreviated Header and Contents Report, will use the cover and first pages as they appear in 7.C-1 and the content pages as they appear in 7.C-2. The content pages will contain selected items out of the document index record header phrase and the entire content of the remainder of that record.

Figure 7.C-3, Full Header and Contents Report will use the cover and first pages as they appear in 7.C-1 and the content pages as they appear in 7.C-3. The entire content of both the header phrase and the remainder of the document index record will be printed.

Figure 7.C-4, KWIC - Document Index Titles will also use the cover and first pages as shown in 7.C-1. The remainder of the report will present only the document accession number and a fixed number N of characters of the document title. The whole title will be permuted cyclically so that a key word appears, left-justified, at the left double column shown in the figure, and the N characters which result from that permutation

OUTPUT FORMATS
File Reports
7.C.1.

SECRET

Figure 7.C-2

ABBREVIATED HEADER - CONTENTS REPORT

** SECURITY CLASSIFICATION **

DATE _____

DOCUMENT INDEX SEARCH

PAGE 2 → N

ACCESSION NUMBER	DISSEMINATION CONTROL	TYPE CODE	CLASS CODE	DOCUMENT SERIES NO	REPORT SOURCE	PUBLICATION DATE	INFORMATION DATE
x _____ x	x _____ x	x x	x x	x _____ x	x x	x _____ x	x _____ x

TITLE: x _____ x

PHRASE 1: x _____ x _____ x _____ x _____ x _____ x _____ x

x _____ x _____ x _____ x _____ x _____ x _____ x

PHRASE 2: x _____ x _____ x _____ x _____ x _____ x _____ x

x _____ x _____ x _____ x _____ x _____ x _____ x

x _____ x _____ x _____ x _____ x _____ x _____ x

x _____ x _____ x _____ x _____ x _____ x _____ x

PHRASE N: x _____ x _____ x _____ x

ABBR HEADER N CONTENTS REPORT

** SECURITY CLASSIFICATION **

SECRET -340-

SECRET

Figure 7.C-3

FULL HEADER - CONTENTS REPORT

** SECURITY CLASSIFICATION **

DATE	DOCUMENT INDEX SEARCH										PAGE 2→N
ACCESSION NUMBER	DISSEMINATION CONTROL		TYPE CODE	CLASS CODE	DOCUMENT SERIES NC	REPORT SOURCE	PUBLICATION DATE	INFORMATION DATE			
x	x	x	x	x	x	x	x	x	x	x	
EVALUATION	TOTAL PAGES	ITEM PAGES	VOL	ED	ISSUE	COUNTRIES	CROSS REF	CITED REF	SENDER LOC	RECEIVER LOC	
x	x	x	x	x	x	x	x	x	x	x	
LANGUAGE	TRANSLATING AGENCY		PUBLISHER		PUBLISHER LOCATION		EDITOR	AUTHOR(s)			
x	x	x	x	x	x	x	x	x	x	x	
TITLE:	x										
	x										
	x										
PHRASE 1:	x	x	x	x	x	x	x	x	x	x	
PHRASE N:	x	x	x	x	x	x					

FULL HEADER N
CONTENTS
REPORT

** SECURITY CLASSIFICATION **

SECRET

SECRET

-341-

Figure 7.C-4

KWIC TITLE REPORT

** SECURITY CLASSIFICATION **

DATE _____.

KWIC - DOCUMENT INDEX TITLES

PAGE _____

TITLE		ACCESSION	NO
X		X	X
X		X	X
X		X	X
X		X	X
X		X	X
X		X	X
X		X	X
X		X	X
X		X	X
X		X	X

KWIC
TITLE

** SECURITY CLASSIFICATION **

-342-
SECRET

SECRET

will be printed. Titles longer than N characters may be truncated at both ends and those shorter than N will be padded out to N characters with blanks between the original ending and beginning characters of the title. There will be as many recurrent permutations of the title appearing as lines as there are key words in the title. Normally the lines will be sorted on the basis of the characters appearing between the two double columns shown in the diagram, which for sorting purposes will be treated as a single key field.

Figure 7.C-5, Map Catalog Card, will be printed on a fanfold of 3 x 5 inch cards for use by the Map Library. Condensed cover and terminating cards may be specified similar to 7.C-1 but the content pages will appear as in 7.C-5.

7.C.2. VOCABULARY CONTROL FILE REPORTS

Figure 7.C-6 is an example of the type of report which may be prepared to present vocabulary control information about organization and facilities. The information retained in the Master Organization and Facility Information File will be printed a record at a time. Where such information has been inserted in the

OUTPUT FORMATS
Vocabulary Control
7.C.2.

Figure 7.C-5

MAP CATALOG CARD

* AREA *	* SCALE *	* CLASSIF *	* ML AREA CODE *
* SUBJECT *	* SUBJECT NO *	* ACC NO *	
TITLE _____	DATE _____	EDITION _____	
PUBLISHER _____	PUB NO _____		
SOURCE _____			
* DESCRIPTION *			
ADDED ENTRIES _____			
MAP CATALOG CARD			

SECRET

- 344 -

SECRET

Figure 7.C-6

ORGANIZATION/FACILITY CONTROL REPORT

** SECURITY CLASSIFICATION **

ORGANIZATION/FACILITY CONTROL REPORT

DATE _____ PAGE _____

ORGANIZATION NAME _____>x

② ASSIGNED NAME: x _____>x

③ FOREIGN NAME: x _____>x

④ VARIANT NAME: x _____>x

⑤ PREVIOUS NAME: x _____>x

⑥ NAME ABBR: x _____ x, TYPE O/F: x _____ x, OF #: x _____ x

TEL CODE: x _____ x, DI #: x _____ x

GR NO'S: FIB- x _____ x, SR- x _____ x, COMOR- x _____ x, BR- x _____ x, BE- x _____ x, NPIC- x _____ x

⑦ ADDRESS: x _____ x x _____ x x _____ x
x _____ x x _____ x x _____ x

⑧ PARENT ORG: x _____>x

⑨ REMARKS: x _____>x
x _____>x
x _____>x

SOURCE CITATIONS: ① x _____ x, ② x _____ x, ③ x _____ x, ④ x _____ x, ⑤ x _____ x
x _____>x

SECRET -345-

SECRET

file, cross referencing of sources for the data printed will be listed in summary for each record.

7.C.3. SUMMARY DATA FILE REPORTS

Figure 7.C-7 presents the cover page for a Leader Appearance Report, describing the basis for the retrieval of the information reported, and a content page which includes for each leader cited the cross references of the sources with the information listed.

Figure 7.C-8, which will be accompanied by a cover page such as appears in 7.C-7, shows the content pages for such a report where no cross referencing information about sources is provided.

7.C.4. UNSYNTHEZIZED FILE REPORT

Figure 7.C-9, Foreign Travel Report, shows a cover page describing the basis for the retrieval of records for the file reported, and the content pages showing selected information about personalities who have been reported to have travelled to places of interest. Since the information is unsynthesized and unevaluated all sources used will be cited. The report may contain redundant, or incorrect information but can be used in developing evaluated items about the travel of these individuals.

OUTPUT FORMATS
Unsynthesized File
7.C.4.

Figure 7.C-7

LEADER APPEARANCE - SOURCE CITATION REPORT - COVER PAGE

* * SECURITY CLASSIFICATION * *

DATE _____	<u>LEADER APPEARANCE REPORT</u>	PAGE _____	COVER _____
REQUESTOR NAME _____	* * *	PERIOD OF REPORT _____	
REQUESTOR LOCATION _____		x _____	x TO x _____ x
REQUEST NUMBER _____			
PRIMARY SEARCH CRITERIA _____			
<u>SEARCH LOGIC:</u>			
JOB RETRIEVE:	x _____		x
	x _____		x
TEST:	x _____		x
	x _____		x
PROCESS:	x _____		x
	x _____		x
END JOB:	x _____ x		

COVER PAGE

* * SECURITY CLASSIFICATION * *

SECRET

-347-

SECRET

Figure 7.C-7

LEADER APPEARANCE - SOURCE CITATION REPORT - BODY PAGE

** SECURITY CLASSIFICATION **

DATE _____ LEADER APPEARANCE REPORT PAGE 1 → N

NAME	TITLE	CITIZENSHIP	APPEARANCE			
			DATE	CITY	COUNTRY	FUNCTION
x _____	x _____	x _____	x _____	x _____	x _____	x _____
			① x _____	x x _____	x x _____	x x _____
			② x _____	x x _____	x x _____	x x _____
			③ x _____	x x _____	x x _____	x x _____
SOURCE: ① x _____	x _____	② x _____	③ x _____	x _____		
x _____	x _____	x _____	x _____	x _____	x _____	x _____
			① x _____	x x _____	x x _____	x x _____
			② x _____	x x _____	x x _____	x x _____
SOURCE: ① x _____	x _____	② x _____				
x _____	x _____	x _____	x _____	x _____	x _____	x _____
			① x _____	x x _____	x x _____	x x _____
SOURCE: ① x _____	x _____					

-348-
SECRET

SECRET

LEADER
APPEARANCE
REPORT
WITH SOURCES

** SECURITY CLASSIFICATION **

Figure 7.C-8

CONDENSED LEADER APPEARANCE REPORT

** SECURITY CLASSIFICATION **

LEADER APPEARANCE REPORT

* * *

PAGE 1-N

DATE _____

NAME	TITLE	CITIZENSHIP	APPEARANCE			
			DATE	CITY	COUNTRY	FUNCTION
X _____	X _____	X _____	X _____	X _____	X _____	X _____
X _____	X _____	X _____	X _____	X _____	X _____	X _____
X _____	X _____	X _____	X _____	X _____	X _____	X _____
X _____	X _____	X _____	X _____	X _____	X _____	X _____
X _____	X _____	X _____	X _____	X _____	X _____	X _____
X _____	X _____	X _____	X _____	X _____	X _____	X _____

-349-
SECRET

SECRET

LEADER
APPEARANCE
REPORT
W/O SOURCES

** SECURITY CLASSIFICATION **

Figure 7.C-9

INDIVIDUAL TRAVEL TO FOREIGN COUNTRIES REPORT - COVER PAGE

** SECURITY CLASSIFICATION **

DATE _____.

FOREIGN TRAVEL REPORT

PAGE _____ COVER _____

* * *

REQUESTOR NAME _____.

REQUESTOR LOCATION _____.

REQUEST NUMBER _____.

PRIMARY SEARCH CRITERIA _____.

SEARCH LOGIC:

JOB RETRIEVE:	x	_____	x
	x	_____	x
TEST:	x	_____	x
	x	_____	x
PROCESS:	x	_____	x
	x	_____	x
END JOB:	x	_____	x

COVER
PAGE

** SECURITY CLASSIFICATION **

SECRET

SECRET

Figure 7.C-9

INDIVIDUAL TRAVEL TO FOREIGN COUNTRIES REPORT - BODY PAGE

* * SECURITY CLASSIFICATION * *

DATE _____	<u>FOREIGN TRAVEL REPORT</u>						PAGE _____
			*	*	*		
<u>PERSONALITY</u>	<u>OCCUPATION</u>	<u>DATE OF TVL</u>	<u>REASON</u>	<u>TRAVEL TO</u>	<u>SOURCE</u>		
x _____	x _____	x _____	x _____	x _____	x _____	x _____	
		x _____	x _____	x _____	x _____	x _____	
		x _____	x _____	x _____	x _____	x _____	
x _____	x _____	x _____	x _____	x _____	x _____	x _____	
x _____	x _____	x _____	x _____	x _____	x _____	x _____	
		x _____	x _____	x _____	x _____	x _____	
		x _____	x _____	x _____	x _____	x _____	
x _____	x _____	x _____	x _____	x _____	x _____	x _____	

-351-
SECRET

SECRET

INDIVIDUALS N
TRAVEL
REPORT

* * SECURITY CLASSIFICATION * *

Appendix 7.D.

SYSTEM MACROS USED IN OPERATIONAL PROGRAMS

This section is an introduction to many of the macros of OS/360 that are used by the operational programs. It gives the name and a brief description of the macro, but assumes the reader has read Chapter 7.5. of this report.

The departure macros are used when the processing of a task passes from one operational program to another operational program. The details for these macros are discussed in CHIVE/N-8-64.

Data set control macros are used to find and keep track of data sets as well as to open and close data sets. Data set accessing macros are used for reading and writing records of data sets. These macros are discussed in Section 7.5.7. of this report and in CHIVE/N-1-65.

Storage allocation macros are used to request and release core storage. These are also discussed in Section 7.5.6. of this report and in CHIVE/N-9-64.

Preliminary versions of the macros in this report are discussed in CHIVE/N-5-64.

SYSTEM MACROS
7.D.

SECRET

7.D.1. DEPARTURE MACROS

Departure macros vary depending on the task structure used. The task structure can be:

- Simple: one load module (a set of programs always loaded together) which contains all the code required for task execution.
- Overlay: similar to above except that a preplanned overlay pattern in the load module is followed so that all the code required at any time is in core.
- Dynamic serial: more than one load module is called during the course of program execution. Transfers between load modules go through the Executive Control Program.
- Dynamic Parallel: an asynchronously operating sub-program (lower level program) is established as a task and follows normal task rules.

7.D.1.1. Direct Transfers (CALL)

Direct transfers are transfers between programs in the same load module. The linkage editor (part of the program loader) sets up the transfers when the programs in the load module are loaded.

The direct transfer macro is "CALL, A" where A is the name of the subprogram being transferred to. When subprogram A desires to transfer back to the calling program it does so by means of a "RETURN" macro. Neither transfer goes through the Executive Control Program. Methods

SYSTEM MACROS
Departure Macros
7.D.1.1.

SECRET

SECRET

of passing parameters can include linkage in specified registers or bytes relative to the "Call A" macro.

In the overlay structure transfers are the same except that a segment load "SEGLD" macro is given to start the loading of a segment and segment wait macro "SEGWT" is given to insure the segment has been loaded before direct transfers to the segment are started.

7.D.1.2. Indirect Transfers Within the Same Task

These transfers are used when transferring to a program in a different load module in the same task. These transfers must go through the Executive Control Program.

7.D.1.2.1. LINK Macro

"LINK" is used to pass control from one load module to another when a "RETURN" is expected. Both the "LINK" and "RETURN" go through the Executive Control Program.

The form of the macro is "LINK, A" where A is the symbolic name of a lower level program in a different load module that is being transferred to. Parameters cannot be passed relative to the "LINK" macro.

7.D.1.2.2. XCTL Macro

"XCTL" is the macro used to pass control from one load module to another when the program being transferred

SYSTEM MACROS
Departure Macros
7.D.1.2.2.

SECRET

SECRET

to is on the same level. This macro is the same as "LINK" except that A is on the same level and no "RETURN" is given to the calling program.

7.D.1.2.3. LOAD Macro

"LOAD" can be used instead of "LINK" to bring a program into core and then the transfer and return can be direct. "LOAD" is used when a task must make frequent use of the same load module. The "LOAD, A" macro performs the function of finding program A, allocating space, fetching, and telling the calling program where A is located, but not transferring to A. Program A is placed in a reserved status so that it will not be overlaid. The reserved status can be cancelled by a "DELETE, A" macro.

After A is loaded, transfers between it and the calling program are direct by means of a BALR instruction and a "RETURN" macro.

7.D.1.3. Indirect Transfer for Parallel Processing (ATTACH)

The "ATTACH" macro creates a new task in the system that can proceed in parallel with the creating task. "ATTACH" should be used only when a significant amount of overlap between the two tasks can be achieved. The parent task (the task giving the ATTACH) can give the

SYSTEM MACROS
Departure Macros
7.D.1.3.

SECRET

SECRET

sub-task (the task created by the ATTACH) a higher or lower priority than its own to help create an advantageous overlap of operations. The parent task can check for completion of the sub-task with the "WAIT" macro.

7.D.1.4. Other Departure Macros

Some departure macros not covered above involve a task terminating itself and a task waiting for completion of IO or another task. A task can normally or abnormally terminate itself. When a task terminates itself, it also terminates all its subtasks, etc.

When a task cannot proceed further before completion of some order it has given (IO command or subtask), it issues a "WAIT". The Executive Control Program will not return CPU control to this task until its "WAIT" is satisfied.

7.D.2. DATA SET CONTROL MACROS

The "OPEN" macro is used to get a data set ready for accessing. This must be done before any data is read or written. The only data sets which a program does not need to "OPEN" are supervisor controlled data sets such as the catalog, the system library, and the data sets used for roll-out. These data sets are referred to indirectly by the problem program.

SYSTEM MACROS
Data Set Control
7.D.2.

SECRET

"OPEN" will fill in all DCB (Data Control Block) fields and establish the interconnections to the supervisor programs that will handle all IO for this data set. These interconnections included buffering and accessing routines built to handle requests on this data set.

The "CLOSE" macro restores the DCB to its original form prior to "OPEN" and frees the storage of the various lists and routines built up by the supervisor to handle this data set.

7.D.3. DATA SET ACCESSING MACROS

When a program opens a data set, it specifies the accessing method it will use. The OPEN macro sets up accessing routines and buffering routines to handle this accessing method. These accessing methods are discussed in Section 7.5.7. The user's request for IO goes to these accessing and buffering routines, which communicate the need for IO to the supervisor by a system macro that references a prewritten channel program. The types of channel programs which the OPEN will set up for the following accessing methods are listed below.

<u>Accessing Method</u>	<u>Macros</u>
BSAM	Read, Write
QSAM	Get, Put

SYSTEM MACROS
Data Set Accessing
7.D.3.

BISAM	Read, Write
QISAM	Setl, Get, Put
BPAM	Read, Write, Find, Stow, Bldl

- FIND: initializes the sequential accessing of a particular named member of the data set. It locates the member desired. Subsequent READ macros retrieve the physical records that make up the member.
- STOW: is used to change the directory entries of the data set.
- BLDL: reads control information from the directory based on a list of names. This control information can be saved in core so subsequent FIND macros can use it.
- READ: for BISAM is a request to read a physical record with the key specified from the data set.
- READ: for other access methods is a request for the next physical record in the data set.
- WRITE: for BISAM is a request for a physical record to be written on a data set. The key specified will determine the order of the record in the data set.
- WRITE: for other access methods is a request for a physical record to be written in sequence on the data set.
- GET: is the request for a logical record, it retrieves the next record in sequence from the data set.
- PUT: is the request to write a logical record as the next in sequence logical record of the data set.
- SETL: is the macro that specifies a new starting point for accessing a data set in sequence.

SYSTEM MACROS
Data Set Accessing
7.D.3.

7.D.4. STORAGE ALLOCATION MACROS

7.D.4.1. GET MAIN Macro

GET MAIN is the operational programmer's macro request for core storage. The amounts of space allocated by type of request in OS/360 are:

	<u>One Block Request</u>	<u>Multiple Block Request</u>
Non-Conditional must have specified amount	Block of size specified	Each block of a specified size
Conditional - Special return if request cannot be met	Block of size specified	Each block of a specified size
Variable - Special return if minimum length is not available	Largest block available within specified limit	---

7.D.4.2. FREEMAIN Macro

FREEMAIN is the macro instruction to dynamically release main storage so that it becomes available to system for future use. The return procedure is checked for recognizable errors. If the return frees a 2K byte block, it is returned to overall free storage. A check is made to see if the 2K block could fulfill a pending roll-out or is needed for roll-in.

SYSTEM MACROS
Storage Allocation
7.D.4.2.

SECRET

Appendix 7.E.

HARDWARE UNIT CHARACTERISTICS

In this appendix the hardware characteristics are divided into CPU, storage, and IO hardware characteristics. The IO hardware characteristics are further divided into channels, bulk storage I/O devices, and I/O devices used for communication between the user and the system.

7.E.1. CENTRAL PROCESSING UNIT

The major parts of the /360 CPU which the user is interested in and the difference between the various models of the /360 are discussed in this section.

7.E.1.1. Characteristics That Differ Between Models

The difference between the models of the /360 is storage capacity and speed, and the amount of I/O which the model storage speed can support. Another difference is the speed of the circuits.

These differences in speed and I/O capability greatly affect over-all performance; none affects the appearance of the design to the programmer, nor does it affect the results of any program that does not depend on time.

Capacity of main core storage varies from 8,192 to 524,288 bytes, depending on the system model. A

HARDWARE CHARACTERISTICS
Central Processing Unit
7.E.1.1.

SECRET

SECRET

byte consists of eight bits plus a parity (check) bit, and is the smallest addressable unit in the IBM System/360. Storage capacities are always given in bytes. The main storage capacity, speed, and access width for the models of the 360 is shown below.

Main Storage							
Capacity (bytes)	System Model						
8,192	C30						
16,384	D30	D40					
32,768	E30	E40					
65,536	F30	F40	F50				
131,072		G40	G50	G60			
262,144		H40	H50	H60	H62	H70	
524,288				I60	I62	I70	
Bytes per Access	1	2	4	8	8	8	
Main Storage Speed	2 us	2.5 us	2 us	2 us	1 us	1 us	

A channel directs the flow of information between input/output devices and main storage. It relieves the central processing unit of the task of communicating directly with input/output devices and permits processing of data to proceed concurrently with input/output devices.

In general, I/O operations on a channel are overlapped with processing, and all channels can operate simultaneously, provided only that the processing unit's data rate capabilities are not exceeded. Nominal data rates for the

HARDWARE CHARACTERISTICS
 Central Processing Unit
 7.E.1.1.

SECRET

SECRET

selector channels range from 250 thousand bytes to 1.3 million bytes per second, depending on the system model and the channel options selected.

The only exception to the foregoing is a special high-speed selector channel available for Model 50, which operates in non-overlap mode.

The number of selector channels available on the various models and their maximum data rates are shown in the following table.

Channels	/360 Model					
	30	40	50	60	62	70
Maximum number	2	2	3	6	6	6
Sel Ch #1 (250 KC)	X	X				
Sel Ch #2 (250 KC)	X	X				
Sel Ch #1 (400 KC)		X	X			
Sel Ch #2 (400 KC)		X	X			
Sel Ch #3 (400 KC)			X			
Sel Ch #1 (1.3 MC)			X	X	X	X
Sel Ch #2,3,4,5,6 (1.3 MC)				X	X	X

A multiplexor channel can be substituted for one of the selector channels.

7.E.1.2. 16 General Registers

The CPU can address information in 16 general registers. The general registers can be used as index registers, in

HARDWARE CHARACTERISTICS
Central Processing Unit
7.E.1.2.

SECRET

address arithmetic and indexing, and as accumulators in fixed-point arithmetic and logical operations. The registers have a capacity of one word (32 bits). The general registers are identified by numbers 0-15 and are selected by a four-bit field in the instruction called the R field.

For some operations, two adjacent registers can be coupled together, providing a two-word capacity. In these operations, the addressed register contains the high-order operand bits and must have an even address, while the implied register, containing the low-order operand bits, has the next higher address.

7.E.1.3. 4 Floating-Point Registers

Four floating-point registers are available for floating-point operations. These registers are two words (64 bits) in length and can contain either a short (one word) or a long (two word) floating-point operand. A short operand occupies the high-order bits of a floating-point register. The low-order portion of the register is ignored and remains unchanged in short-precision arithmetic. The floating-point registers are identified by the numbers 0, 2, 4, and 6. The operation code determines which type of register is to be used in an operation.

HARDWARE CHARACTERISTICS
Central Processing Unit
7.E.1.3.

SECRET

7.E.1.4. Instruction Sets

System/360 is a general-purpose system designed to be tailored for commercial, scientific, communications, or control applications. A Standard instruction set provides the basic computing function of the system. To this set a decimal feature may be added to provide a commercial instruction set or a floating-point feature may be added to provide a scientific instruction set. When the storage protection feature is added to the commercial and scientific features, a Universal set is obtained.

7.E.1.5. Types of Arithmetic--Fixed Point, Decimal, Floating Point

Fixed Point

The fixed-point instruction set performs binary arithmetic on operands serving as addresses, index quantities, and counts, as well as fixed-point data. In general, both operands are signed and 32 bits long. Negative quantities are held in two's-complement form. One operand is always in one of the 16 general registers; the other operand may be in main storage or in a general register.

The instruction set provides for loading, adding, subtracting, comparing, multiplying, dividing, and

HARDWARE CHARACTERISTICS
Central Processing Unit
7.E.1.5.

SECRET

storing, as well as for the sign control, radix conversion, and shifting of fixed-point operands. The entire instruction set is included in the standard instruction set.

The condition code is set as a result of all sign-control add, subtract, compare, and shift operations, and can be checked by logical instructions.

Decimal

Decimal arithmetic operates on data in the packed format. In this format, two decimal digits are placed in one eight-bit byte. Since data are often communicated to or from external devices in the zoned format (which has one digit in an eight-bit byte), the necessary format-conversion operations are also provided in this instruction group.

Data are interpreted as integers, right-aligned in their fields. They are kept in true notation with a sign in the low-order eight-bit byte.

Processing takes place right to left between main-storage locations. All decimal arithmetic instructions use a two-address format. Each address specifies the leftmost byte of an operand. Associated with this address is a length field, indicating the number of additional bytes that the operand extends beyond the first byte.

HARDWARE CHARACTERISTICS
Central Processing Unit
7.E.1.5.

SECRET

The decimal arithmetic instruction set provides for adding, subtracting, comparing, multiplying, and dividing, as well as the format conversion of variable-length operands. Most decimal instructions are part of the decimal feature.

The condition code is set as a result of all add-type and comparison operations and can be checked by logical instructions.

Floating-Point Arithmetic

The purpose of the floating-point instruction set is to perform calculations using operands with a wide range of magnitude and yielding results scaled to preserve precision.

A floating-point number consists of a signed exponent and a signed fraction. The quantity expressed by this number is the product of the fraction and the number 16 raised to the power of the exponent. The exponent is expressed in excess 64 binary notation; the fraction is expressed as a hexadecimal number having a radix point to the left of the high-order digit.

To avoid unnecessary storing and loading operations for results and operands, four floating-point registers are provided. The floating-point instruction set pro-

HARDWARE CHARACTERISTICS
Central Processing Unit
7.E.1.5.

SECRET

vides for loading, adding, subtracting, comparing, multiplying, dividing, and storing, as well as the sign control of short or long operands. On the other hand, long operands provide greater accuracy of computation. Operations may be either register to register or storage to register. All floating-point instructions and registers are part of the floating-point feature.

To preserve maximum precision, addition, subtraction, multiplication, and division are performed with normalized results. Addition and subtraction may also be performed with unnormalized results. Normalized and unnormalized operands may be used in any floating-point operation.

The condition code is set as a result of all sign control, add, subtract, and compare operations.

7.E.1.6. Interrupt Structure

An interrupt system alerts the system to an exceptional condition, the end of an I/O operation, program errors, machine errors, etc., and sends the system to the appropriate routine following the detection of such an event. The system must have, in effect, the ability to pause to answer the telephone and then to resume the interrupted work.

HARDWARE REQUIREMENTS
Central Processing Unit
7.E.1.6.

SECRET

The interruption system makes possible the operation of a system in a non-stop environment and greatly aids the efficient use of I/O equipment. The desire to make the interruption procedure as short and simple as possible means that the method of switching between the interrupted program and the program that services the interruption must be quite efficient. It operates as follows:

The complete status of the System/360 is held in eight bytes of information. This status information, which consists of the instruction counter, condition code, storage protection key, etc., is saved when an interruption occurs, all the status information, together with an identification of the cause of the interruption, is combined into a double (eight-byte) word called the program status word (PSW).

The PSW is stored at a fixed location, the address of which depends on the type of interruption. The system then automatically fetches a new PSW from a different fixed location, the address of which is also dependent on the type of interruption. Each class of interruption has two fixed locations in main storage: one to receive the old PSW when the interruption occurs, and the other to supply the new PSW that governs the servicing of that class of interruption.

HARDWARE REQUIREMENTS
Central Processing Unit
7.E.1.6.

SECRET

After the interruption has been serviced, a single instruction uses the stored PSW to reset the processing unit to the status it had before the interruption.

The interruption system separates interruptions into five classes:

Program interruptions are caused by various kinds of programming errors; the exact type of error is displayed in the old PSW.

Supervisor Call interruptions are caused by the program issuing an instruction to turn over control to the supervisor (a type of master control program). The exact reason for the call is shown in the old PSW.

External interruptions are caused by either an external device requiring attention or by the timer going to zero. (The external device may be a signal from any device communicating via the direct control feature.)

Machine Check interruptions are caused by the machine-checking circuits detecting a machine error. The system is automatically switched to a diagnostic procedure.

I/O interruptions are caused by an I/O unit ending an operation or otherwise needing attention. Identifications of the device and channel causing the interruption are

HARDWARE REQUIREMENTS
Central Processing Unit
7.E.1.6.

SECRET

stored in the old PSW; in addition, the status of the device and channel is stored in a fixed location.

Some of the program interruptions, all external interruptions, and all I/O interruptions can be masked (ignored); when this is done, the external and I/O interruptions are held pending and taken at a later time (governed by the control program).

During the execution of an instruction, several interruptive events may occur simultaneously. When this occurs, the competing interruptions are serviced in a fixed order of priority.

7.E.2. 2361 CORE STORAGE

This directly addressable core storage may be added to an IBM System/360. Model 50, 60, 62 or 70. These units are not input/output units. This low-cost auxiliary bulk storage may be used as an extension of main storage, as shared storage for two systems, or as intermediate storage between main storage and input/output devices. It facilitates real-time, multiprogrammed, and communications processing, as well as multiprocessing. This storage is addressed contiguously with main storage. Speed of this storage is 8 microseconds. The number of bytes obtained per storage access, and all other features

HARDWARE REQUIREMENTS
2361 Core Storage
7.E.2.

SECRET

of large-capacity storage except cycle time, are the same as those of the main storage of the system.

The model 1 contains 1,048,576 bytes of storage, and the Model 2 contains 2,097,152 bytes.

A maximum of four Model 2's may be added to the System/360, Models 60, 62 and 70, to provide 8,388,608 positions of directly addressable core storage. A maximum of two Model 2's may be added to the System/360 Model 50 to provide 4,194,304 positions of directly addressable core storage.

Direct addressing of this large storage is accomplished by having the effective address of an instruction determined by either the 24 bit contents of a general register or the sum of a 12 bit displacement field and the 24 bit contents of up to two general registers.

7.E.3. I/O CHANNELS

Each System/360 channel directs the flow of information between main storage and those input/output devices the channel controls. The channel relieves the processing unit of the tasks of communicating directly with I/O and permits data processing to proceed concurrently with input/output operations. Each channel has facilities for:

HARDWARE REQUIREMENTS
I/O Channels
7.E.3.

SECRET

- Accepting I/O instructions from the CPU
- Addressing devices specified by I/O instructions
- Fetching channel control information from main storage
- Decoding control information
- Testing control information for validity
- Executing control information
- Providing control signals to the I/O interface
- Accepting control-response signals from the I/O interface
- Buffering data transfers
- Checking parity of bytes transferred
- Counting the number of bytes transferred
- Accepting status information from I/O devices
- Maintaining channel-status information
- Sending requested status information to main storage
- Sequencing interruption requests from I/O devices
- Signaling interruptions to the CPU

The System/360 has two types of channels: multiplexor and selector. The channel facility necessary to sustain an operation with an I/O device is called a subchannel. The selector channel has one subchannel; the multiplexor channel has multiple subchannels.

HARDWARE REQUIREMENTS
I/O Channels
7.E.3.

SECRET

7.E.3.1. Selector Channels

Selector channels are normally used where high-speed devices (for example, magnetic tape, files and drums) are attached to a system. Selector channels transmit bursts of data to or from a single input/output device at a time. The bursts extend over the whole block of data or, when command chaining is specified, over the whole sequence of blocks.

The channel facilities required for sustaining an input/output operation are termed a "subchannel." The selector channels have only one subchannel and operate only one data transmission at a time. Other input/output devices attached to the channel can in the meantime execute operations that do not involve communications with the channel.

7.E.3.2. Multiplexor Channel

The multiplexor channel separates the operations of high-speed devices from those of lower-speed devices. Operations on the channel are in two modes: a "multiplex" mode for lower data rates, and a "burst" mode for the higher.

In the multiplex mode, the single data path of the channel can be time-shared by a large number of low-speed

HARDWARE REQUIREMENTS
I/O Channels
7.E.3.2.

SECRET

I/O devices operating simultaneously; the channel receives and sends data to them on demand. When operating in the burst mode, however, a single I/O device captures the multiplexor channel and does not relinquish it from the time it is selected until the last byte is serviced.

Examples of low-speed devices that can operate simultaneously on the multiplexor channel are:

- Printers
- Card Punches
- Card Readers
- Terminals

Examples of I/O devices that operate in the burst mode are tape units and disk, drum, or data cell storage.

7.E.4. BULK STORAGE DEVICES AND ASSOCIATED CONTROL UNITS

I/O devices used for bulk system storage include tape (2400, 7330), disks (1302 and 2311), and data cells (2321).

7.E.4.1. Tape Units 2401, 2402, 2403, 2404

The 2400 series tapes are 9-track magnetic tapes, specially designed for use on the System/360. All tape units can operate in the read-backward mode. Recording density is 800 bytes per inch; inter-record gap

HARDWARE REQUIREMENTS
Storage Devices/Control Units
7.E.4.1.

is .6 inches. Single track errors are corrected automatically on the reread of the record. Each 2401, 2402, 2403, 2404 is offered in three models with the following performance.

	Model 1	Model 2	Model 3
Data Rate	22,500 bytes per sec.	45,000 bytes per sec.	90,000 bytes per sec.
2401	Magnetic Tape Unit A single drive requiring connection to a control unit.		
2402	Magnetic Tape Unit Two independent drives housed in a single cabinet requiring connection to a control unit		
2403	Magnetic Tape Unit and Control A single drive which has a self-contained control unit that allows reading or writing of magnetic tape on a single channel.		
2404	Magnetic Tape Unit and Control A single drive which contains a control unit that allows simultaneous reading and writing on two channels.		

Each tape control can control as many as eight tape drives. The simultaneous control attaches to two channels in a manner that permits a read-type operation on one tape drive to be overlapped with a simultaneous write, write tape mark, or erase tape operation on another drive.

HARDWARE REQUIREMENTS
Storage Devices/Control Units
7.E.4.1.

SECRET

Controls attach to a selector of multiplexor channel and operate in the burst mode.

Tape controls are also available in two stand-alone versions:

2803

Tape Control Unit
Allows read or write operations on a single channel

2804

Tape Control Unit
Allows simultaneous reading and writing on two channels.

Any model of tape drive can be controlled by any tape control. Characteristics of the three tape drive models are shown below.

<u>Characteristic</u>	<u>Model 1</u>	<u>Model 2</u>	<u>Model 3</u>
Data Rate (kb/kd)*	22.5/45	45/90	90/180
Tape Speed (ips)	28.125	56.25	112.5
Density (bpi)	800	800	800
Nominal Inter-record Gap (inches)	.6	.6	.6
Nominal Inter-record Gap Time (millsec)	21.2	10.6	5.3
Rewind Time (minutes)	3.2	2.6	1
Rewind Unload Time (minutes)	2.2	2.2	1
Tape Width (inches)	.5	.5	.5
Number of Tracks	9	9	9

* kb = thousands of bytes per second
kd = thousands of decimal digits per second
ips = inches per second
bpi = bytes per inch

HARDWARE REQUIREMENTS
Storage Devices/Control Units
7.E.4.1.

SECRET

7.E.4.2. 2816 Switching Unit

With the 16-drive control feature installed on the appropriate tape controls and the addition of two IBM 2816 Switching Units, a maximum of 16 tape drives may be controlled in common by a maximum of four tape controls. The minimum is control of four tape drives by two tape controls; any combination between the minimum and maximum is available. For more than eight drives, the second switching unit is required.

Model 1 of the switching unit is used for switching tape drives in the 2400 series. The drives in the tape-switching pool may be located in any model of 2401, 2402, or 2403. The tape controls involved may be in the IBM 2403 Magnetic Tape Unit and Control or the 2803 Tape Control.

If the 16-drive control feature is not installed on a tape or Hypertape control, it can control only the first eight drives in the tape or Hypertape pool.

7.E.4.3. 1302 Disk Storage Unit

The IBM 1302 Disk Storage can record and retrieve data either at random or sequentially, the random capability permits immediate access to specific areas of information without the need to sequentially examine all

HARDWARE REQUIREMENTS
Storage Devices/Control Units
7.E.4.3.

the data in the file. The speed of the access enables the user to maintain up-to-the-second data files and make frequent retrievals of the data stored.

The 1302 Model N 1 contains one disk storage module; the Model N 2 contains two disk storage modules. Capacity per module is 112.14 million bytes (or 224.28 million packed decimal digits and signs). Eight accesses (four modules) may be connected to the System/360 through an IBM 2841 Storage Control Unit, for on-line random access to 448.56 million bytes (or 897.12 million packed decimal digits and signs) per 2841 attached. Each module has two access mechanisms with each access accessing 250 cylinders. A special feature on the control unit permits eight more access mechanisms, with four additional modules of disk storage, to be addressed by that control unit.

The recording medium consists of thin, magnetically-coated metal disks that spin at 1,790 revolutions per minute. The transfer rate between the 1302 and the processing unit is 156 thousand bytes per second.

For accessing the 250 cylinders accessed by one access mechanism are divided into five groups:

1st group has 6 areas of 10 cylinders each.

2nd group has 4 areas of 10 cylinders each.

HARDWARE REQUIREMENTS
Storage Devices/Control Units
7.E.4.3.

3rd group has 6 areas of 10 cylinders each.

4th group has 4 areas of 10 cylinders each.

5th group has 5 areas of 10 cylinders each.

To access a track in:

- Same cylinder is 0 ms
- Another cylinder in same 10 cylinder area is 50 ms.
- Another area in same group is 120 ms.
- Another group is 180 ms.

This gives an average access time of 165 ms.

7.E.4.4. 2311 Disk Storage Drive

The IBM 2311 Disk Storage Drive provides random access storage for 7.25 million bytes (or 14.5 million packed decimal digits and signs) on a single disk pack. Eight disk storage drives can be attached to each IBM 2841 Storage Control Unit, for a total on-line capacity of 58 million bytes per 2841. In addition, unlimited storage capacity is possible because the disk pack in each drive can be easily removed and replaced with another in less than 1 minute.

The disk pack weighs only 10 pounds; each has six 14-inch disks, mounted 1/2 inch apart on a vertical shaft. The inside ten disk surfaces are used for recording data and the outermost two surfaces are pro-

HARDWARE REQUIREMENTS
Storage Devices/Control Units
7.E.4.4.

SECRET

tective plates. When the disk pack is installed in the disk storage, information is written on or read from the disk surfaces by magnetic read/write heads, mounted in pairs between each two disks on a movable comb-like access mechanism.

The data rate of the IBM 2311 is 156 thousand bytes per second. Sequential track-to-track access time is 30 milliseconds. The maximum seek time is 145 milliseconds, and the average seek time is 85 milliseconds. The average rotational delay is 12.5 milliseconds.

7.E.4.5. 2321 Data Cell Drive

Each 2321 Data Cell Drive can hold 400 million bytes (or 800 million packed decimal digits and signs) of on-line data. Eight 2321's may be attached to a 2841 Storage Control Unit, for a total on-line capacity of 3.2 billion bytes or 6.4 million pack decimal digits and signs per 2841 control unit. The data rate is 55 thousand bytes per second.

Each data cell drive contains from one to ten data cells, each having a capacity of 40 million bytes. The data cells are all removable and interchangeable, permitting an open ended capacity for libraries of data cells.

HARDWARE REQUIREMENTS
Storage Devices/Control Units
7.E.4.5.

SECRET

~~SECRET~~
CONFIDENTIAL

The storage medium is a strip of magnetic tape 2 1/4 inches wide by 13 inches long. Each data cell contains 200 of these strips, divided into 20 subcells of 10 strips each. A rotary positioning system aligns the selected subcell beneath the access station. As shown below, access time to a strip varies from 175 milliseconds to 600 milliseconds, depending on the number of subcells the array must move.

	Number of Subcells Array Must Move			
	0	1	50	100 (max.)
No strip on drum	175 ms	250 ms	350 ms	400 ms
Previously addressed strip on drum	375 ms	450 ms	550 ms	600 ms

7.E.4.6. 2841 Storage Control

The IBM 2841 Storage Control Unit provides the capability of attaching serial-by-bit types of random access storage devices to the IBM System/360. Adapters are available for the 2841 to control these storage units:

- IBM 1302 Disk Storage Models N1, N2
- IBM 2311 Disk Storage Drive
- IBM 2321 Data Cell Drive

HARDWARE REQUIREMENTS
 Storage Devices/Control Units
 7.E.4.6.

CONFIDENTIAL

~~SECRET~~
CONFIDENTIAL

Adapters can be mixed to provide 2841 control of up to eight access mechanisms, in any combination, on the above units.

ACCESS MECHANISMS PER FILE

1302 N1	Two Access Mechanisms (one module)
1302 N2	Four Access Mechanisms (two modules)
2311	One Access Mechanism
2321	One Access Mechanism

FILES PER 2841

(One of the following, when adapters are not intermixed)

1302 N1	4
1302 N2	2
2311	8
2321	8

The 2841 interprets and executes all orders from the system, and checks the validity of the data transferred to and from the storage devices.

7.E.5. SYSTEM COMMUNICATION DEVICES

Various IO devices are needed for convenient communication between the user and computer system. Some of these IO devices available on the 360 are discussed in this section.

HARDWARE REQUIREMENTS
Communication Devices
7.E.5.

~~SECRET~~

CONFIDENTIAL

~~SECRET~~
CONFIDENTIAL

7.E.5.1. 1442 Card Read Punch

The IBM 1442 Card Read Punch Model N1 uses a single common card path for reading and punching, and reads and punches cards serially by card column. Serial card feeding, past a read and then a punch station, makes it possible for the program to read data from a card, then hold the card at the next consecutive card column, and then punch data (such as the results of a calculation) into the same card, all during a single card pass.

The IBM 1442 Model N1 reads or punches cards at the following maximum rates:

Reading - 400 cards per minute

Punching - 160 card columns per second

The 1442 is an I/O device typically operated on the multiplexor channel. The control circuits are within the 1442.

The 1442 Model N1 reads and punches an extended card code that permits punching any of 256 different combinations of holes in each card column. The optional card image feature permits the low-order six bits of bytes read from the processing unit to be punched into the upper six or lower six rows of a card, enabling 160 bytes to be placed in the card. On reading, the information is read column by column and transmitted to

HARDWARE REQUIREMENTS
Communication Devices
7.E.5.1.

~~SECRET~~
CONFIDENTIAL

~~SECRET~~

~~CONFIDENTIAL~~

the processing unit byte by byte, with two high-order bits (6 and 7) being set to zero. Interspersed cards punched in either the normal or the card image mode may be read.

The read hopper holds 1,200 cards. Card movement is from the hopper to one of the two stackers (pockets). The stackers hold 1,200 cards each, and cards can be removed from either stacker without stopping the machine.

7.E.5.2. 2201 Printer

The IBM 2201 Printer Model 3 printer has 132 print positions and a speed of 1,100 lines per minute.

In the standard character set, there is a regular distribution of the 48 characters in five identical groups. In the optional preferred character set, the 48 characters are distributed irregularly, with more of some and fewer of others than in the standard character set. This produces print rates of three levels, depending on the characters printed:

- 1400 lines per minute for the 10 numeric digits and ., *-
- 950 lines per minute for the 26 letters and %\$/&
- 500 lines per minute for the characters + @ # +

HARDWARE REQUIREMENTS
Communication Devices
7.E.5.2.

~~SECRET~~

~~CONFIDENTIAL~~

When characters from different levels are inter-mixed, the speed becomes variable at rates between the levels.

The 2201 is controlled, buffered, and attached to the System/360 by an IBM 2821 Control Unit Model 1 or 2. One or two additional 2201 printers can be serviced by a 2821 Model 3, with a 2821 Model 1 or 2 as a prerequisite.

7.E.5.3. 1052 Console Typewriter

The IBM 1052 Printer-Keyboard Model 1, part of the 1050 Data Communication System, can be attached to the system for communication between remote locations and the system, and the operator and the system.

The keyboard and the printing are electrically and mechanically independent of each other. The keyboard can be used for system input, and the printer accepts computer output. A single keyboard depression causes an I/O interruption, and a double keyboard depression signals the end of data entry.

The 1052 prints at 14.8 characters per second. It has replaceable printing head and a typewriter style keyboard.

7.E.5.4. 1050 Data Communication System

The IBM 1510 Data Communication System is a multi-purpose office-oriented TELE-PROCESSING system. It can

HARDWARE REQUIREMENTS
Communication Devices

7.E.5.4.

~~CONFIDENTIAL~~

~~SECRET~~ CONFIDENTIAL

be operated over leased communications lines (through the IBM 2701 Data Adapter Unit or 2702 Transmission Control) to connect the System/360 to remote locations.

The versatility of the 1050 system is furthered by its variety of configurations. The choice of configurations ranges from a receiver-printer to a configuration that includes:

- Manual keyboard entry
- Printed page output
- Punched-card input and output
- Paper-tape input and output
- Edge-punched-document input and output

The component type numbers and names are:

- 1051 Control Unit
- 1052 Printer-Keyboard
- 1053 Printer
- 1054 Paper Tape Punch
- 1055 Paper Tape Punch
- 1056 Card Reader
- 1057 Card Punch
- 1058 Printing Card Punch

In addition to providing on-line transmission at 14.8 characters per second, the 1050 can also simultaneously

HARDWARE REQUIREMENTS
Communication Devices
7.E.5.4.

~~SECRET~~

CONFIDENTIAL

~~SECRET~~ CONFIDENTIAL

prepare and record off-line data. Transmission is checked for accuracy.

The IBM 1051 Control Unit is required in all configurations and contains the power supply, code translator, data channels, and control circuitry for the 1050 system.

7.E.5.5. 1015 Inquiry Display Terminal

This device and its IBM 1016 Control Unit make it possible for an operator to see records in main or auxiliary storage as needed, such as upon customer demand. The information is written at a rate of up to 650 characters per second, about 40 times faster than that produced for an operator by means of typeout. Once written, the information is retained for as long as desired.

The viewing area of this cathode ray tube has the capacity for 30 lines of 40 characters each, for a total of up to 1,200 characters at a time. Horizontal spacing is ten characters to the inch; vertical spacing, eight lines per inch. The characters standard for display are: all 36 alphanumeric characters plus 23 special characters.

Plotting is available optionally. Plotting resolution is approximately 20 points per inch horizontally and 15 vertically. The plotting matrix is 60 x 80.

HARDWARE REQUIREMENTS
Communication Devices
7.E.5.5.

CONFIDENTIAL

~~SECRET~~

~~SECRET~~ CONFIDENTIAL

Four special characters are used for plotting. A line advance suppress feature is also incorporated into the plotting option to allow the display of more than one plotting point in a given character space.

Up to ten 101 Model 1's can be attached to the 1016 Control Unit. Each Model 1 allows the attachment of five 1015 Model 2's providing a maximum of sixty 1015 Inquiry Display Terminals (ten 1015 Model 1's and fifty 1015 Model 2's).

HARDWARE REQUIREMENTS
Communication Devices
7.E.5.5.

~~SECRET~~ CONFIDENTIAL

~~SECRET~~

~~CONFIDENTIAL~~

~~SECRET~~

~~CONFIDENTIAL~~