XIII. VECTOR MERGING AND VECTOR CORRELATIONS

T. Evslin, G. Hochgesang, and M. Lesk

1. Introduction

The present section is a description of link 8 in which documents looked-up in link 1, 3-7 are punched out. Additional vectors are read in from A2, the input tape, and A6, the permanent collection tape. The components of all document vectors are weighted according to component type. If request document correlations are to be performed, this is done here. If there is to be concept-concept or hierarchy expansion, the merged vectors are passed from link 8 to links 9 and 10 where the expansions take place. Link 10 passes control back to link 8 where the expanded vectors are then weighted and correlated. Most large-scale, logical control is handled by the link supervisor, SUPER 8, details of input-output, merging and stacking within the individual vectors are handled by MERVEC. Correlation is done by CORREL and DENOM; weighting by WEIGT; printing by WRITE; punching by PUNPUN; card input by SUPPLY; input from the permanent collection by SUPTAP; error messages by ERROR; and correlation output by CUTCOR. All input-output is handled through INCUT (see Penn-Jersey Transportation Study, EDP Procedure Description No. 29) except for the leaders which are written out through FORTRAN. CALLIO is used to call INOUT from FORTRAN programs.

2. The SUPER 8 Routine

A. Description

SUPER 8 is the main routine for link 8. It handles all logical decisions prior to and following the processing of a single vector. Within each vector, the operations are handled by MERVEC which calls the various utility routine.

Immediately upon execution of the link, SUPER 8 checks common location IAFT. If this indicator is "on," either concept-concept or hierarchy expansion or both have previously taken place, and all the requests plus some or all of the documents will have been written on a tape named in location NEXPT. All documents looked up in this run will already have been merged during a previous call to the link, which preceded the call to the expansion links. ISW is set to three.

If IAFT is off, ILN is checked to see if documents have been looked up in this run. If so, ISW is set to one and processing begins. If not, ISW is set to two. ISW controls input. If it equals one, documents looked-up in this run are merged in from A5 and A7. If it equals two, expanded vectors are brought in from NEXPT. If ISW equals three, a call is made for vectors from the input tape; and if ISW equals four, vectors are read in from A6, the permanent collection tape.

IRORD controls the types of vectors to be read in. If IRORD equals one, requests are to be read in. If IRORD is two, the request buffer has been filled, and requests are read in but ignored, except for printing and punching. If IRORD equals three, the *LIKE documents are read in; and if IRORD equals four, the *LIKE documents are read but are treated like *TEXT

documents and are not added to an already full request stock. Five is the highest possible value for IRORD, and implies that *TEXT documents are to be read in.

If text lookup has been performed in this run, ISW and IRORD both start at one. Requests are read in from A5 and A7 by successive calls to MERVEC. Each request vector is punched or printed if the user has so specified, and is either stacked in core if there is to be request-document correlation, or put out on B1 if there is to be further expansion. If more than 49 requests are found, or if the request buffer is filled, requests are only printed and punched but are not stacked or written out on B1, and IRORD is set to two.

When the first nonrequest is found on A5, ISW is set to three and repeated calls to CADIN, an entry point of MERVEC, bring in requests from A2 where they are stored in binary form, having been looked up and punched during a previous run. These requests are also printed if PRNVEC is on, and stacked or written out depending on whether correlation is to take place now or after expansion, respectively.

When a nonrequest is found on A2, SUPER 8 checks DOCTAP to see if a permanent collection tape has been mounted for this run. If it has, ISW is set to four and all the requests are read in from A6 by calls to the DOCIN entry point of MERVEC.

If there is no permanent collection tape, or if all the requests have been read in, IRORD is set to three if there is still room in the request buffer, or four otherwise. If further vectors looked up this run

are present, ISW is set to one again; otherwise it is set to three.

If IRORD is now three, *LIKE documents are read in sequentially from A5, A2, and A6 and stacked like requests. If IRORD equals four, *LIKE documents are read in the same sequence, but are treated as documents (see below). ISW is again stepped from one to three to four.

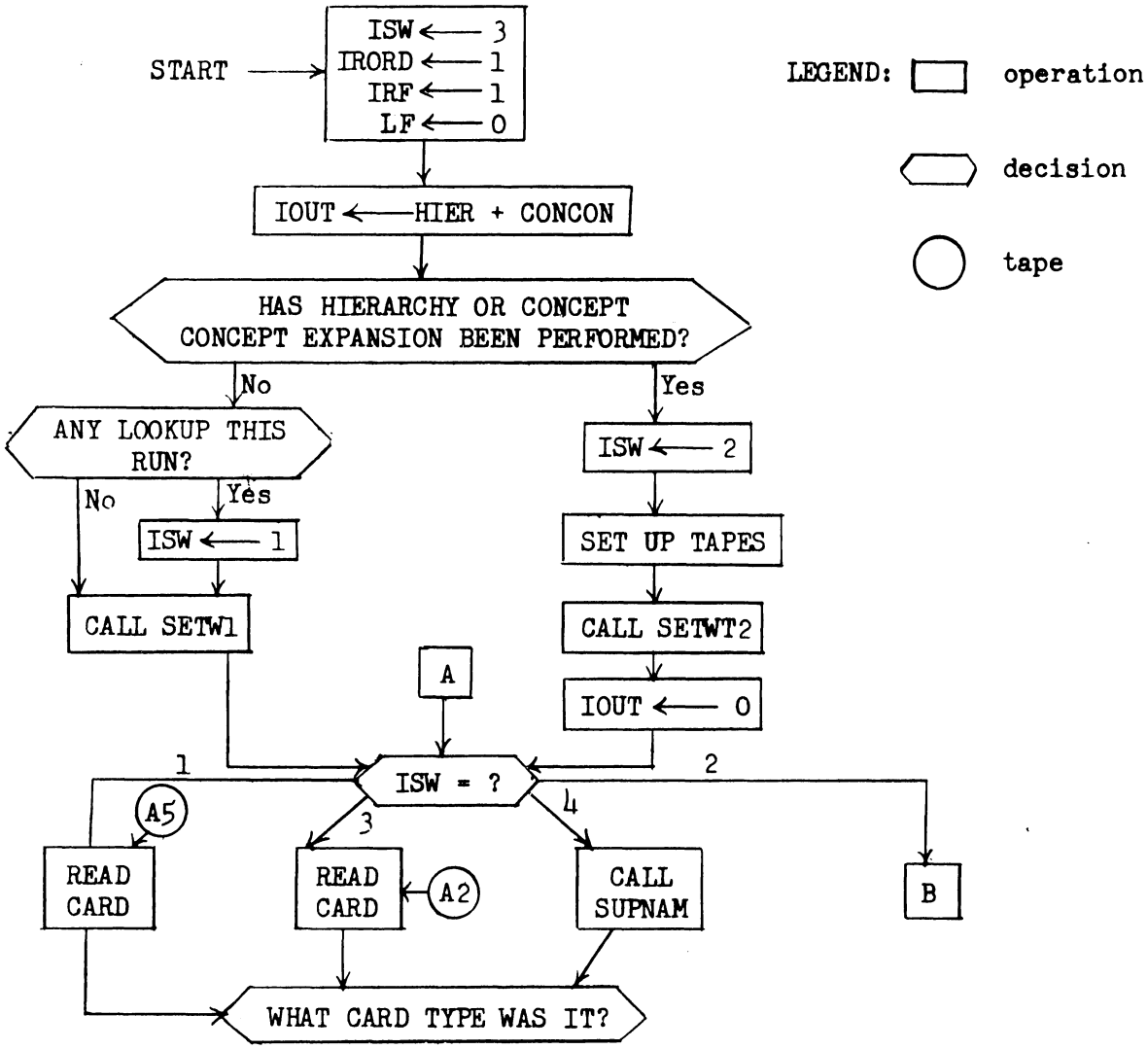
When all the *LIKE documents have been read in, ISW is set to five. If there is to be expansion by hierarchy only performed on requests, link 8 will call link 10 as soon as all the vectors looked up in this run have been read in, leaving some texts on A2 or A6 to be read in after requests will have been expanded.

The *TEXT documents are read in and put out on B1 if they are to be expanded or if document-document or concept-concept expansion is to follow. *TEXT documents are not stacked but correlated socially with all requests as they pass through core. At the end of each *TEXT document, SUPER 8 calls ONTCOR to put out correlations of that document with each request if correlation is being performed at this time.

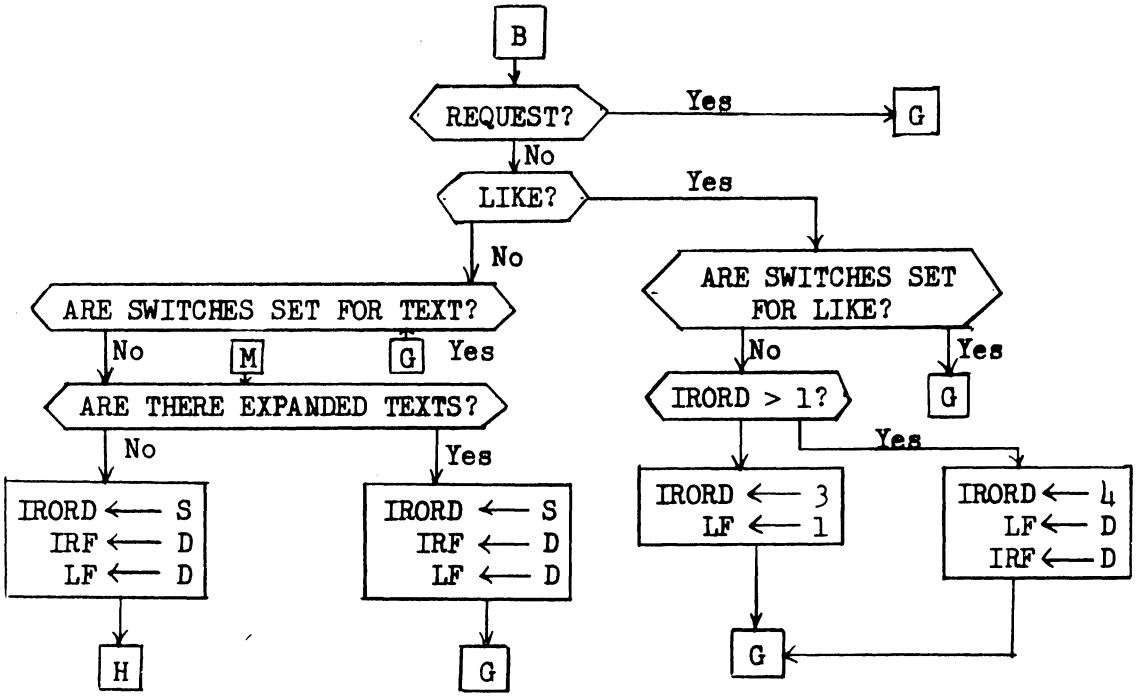
If expansion precedes link 8, ISW is set to two and repeated calls to AFTIN, another MERVEC entry point, are used to read in all expanded requests, as well as *LIKE documents, and *TEXT documents from NEXPT tape. If there are still texts on A2 and A6, ISW is set to three after NEXPT tape has been emptied, and SUPER 8 proceeds as above.

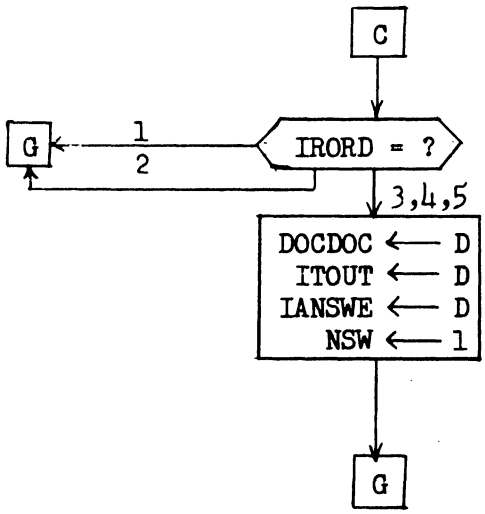
After all correlations are performed, SUPER 8 calls ENDEND which calculates the proper next chain link from the user-specified parameters in upper common.

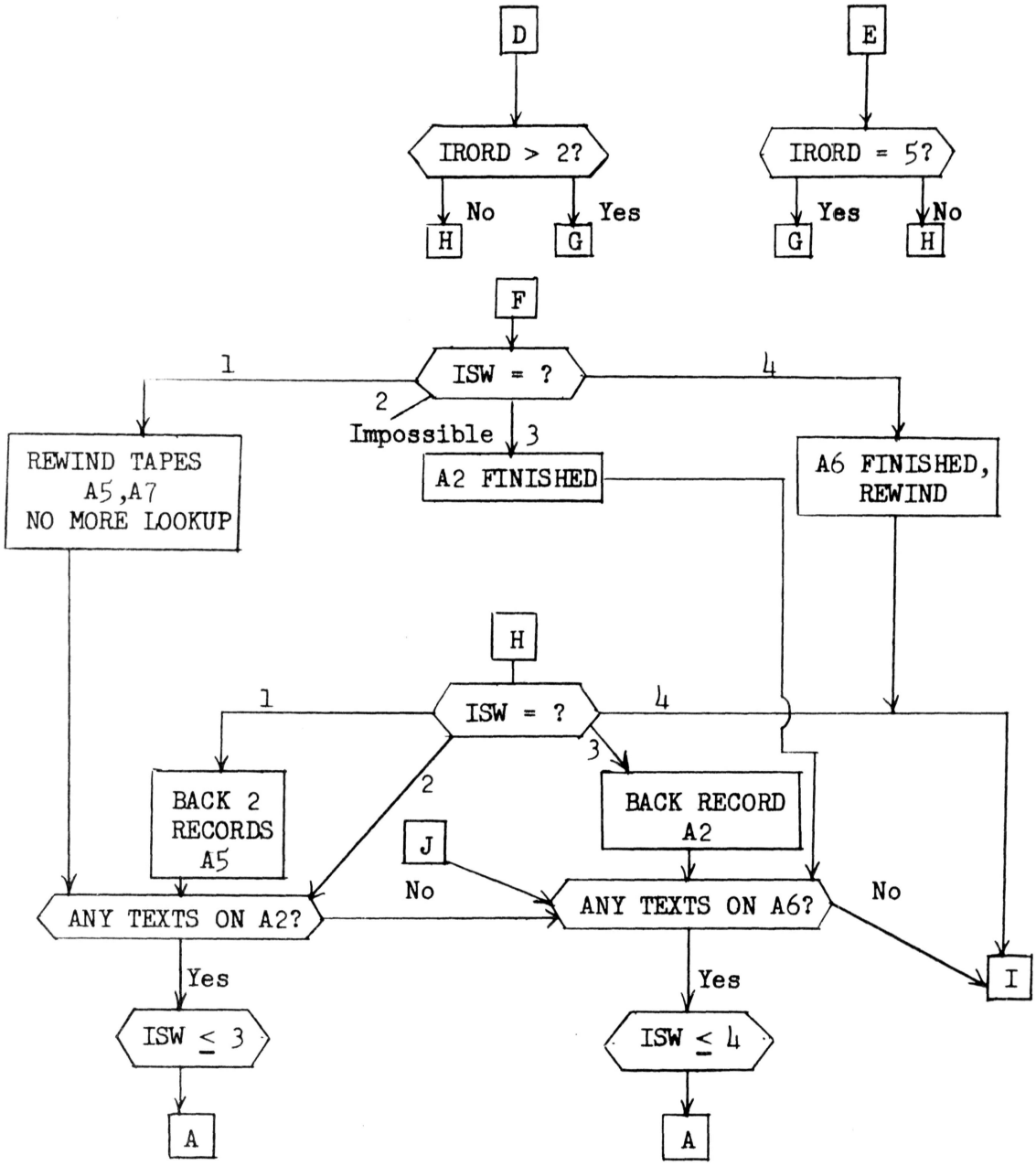
B. SUPER 8 Flowchart

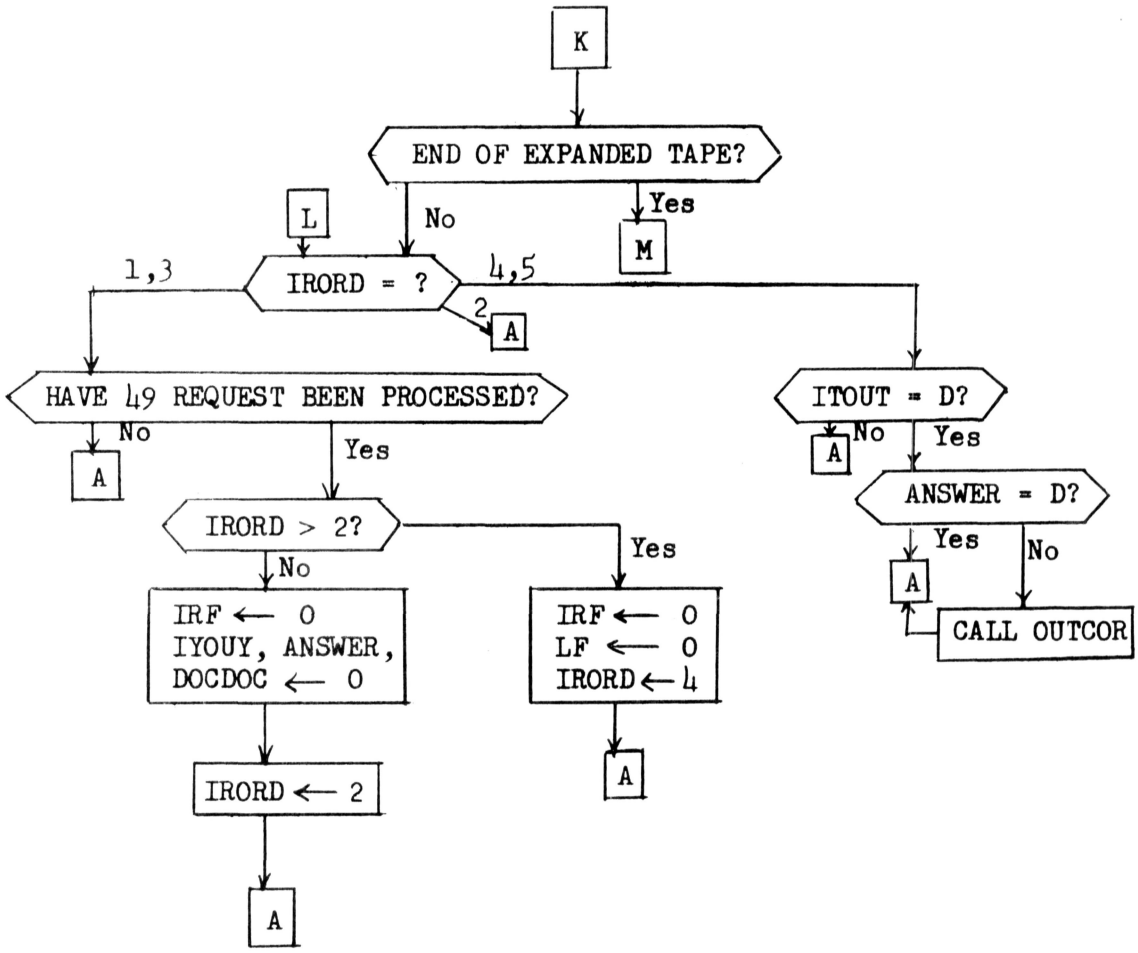


*FIND → C
*LIST → C
*LIKE → D
*TEXT → E
*ONLY → A
*STOP → F
EOF → F


B
REQUEST?
Yes
G
No
LIKE?
Yes
No
ARE SWITCHES SET FOR TEXT?
No
M
G
Yes
ARE THERE EXPANDED TEXTS?
No
Yes
IRORD ← S
IRF ← D
LF ← D
IRORD ← S
IRF ← D
LF ← D
H
G
ARE SWITCHES SET FOR LIKE?
No
Yes
IRORD > 1?
G
Yes
IRORD ← 3
LF ← 1
IRORD ← 4
LF ← D
IRF ← D
G


C
G
1
2
IRORD = ?
3,4,5
DOCDOC ← D
ITOUT ← D
IANSWE ← D
NSW ← 1
G


D
IRORD > 2?
No
Yes
H
G
E
IRORD = 5?
Yes
No
G
H
F
ISW = ?
1
2
Impossible
3
4
REWIND TAPES
A5, A7
NO MORE LOOKUP
A2 FINISHED
A6 FINISHED,
REWIND
H
ISW = ?
1
2
3
4
BACK 2
RECORDS
A5
BACK RECORD
A2
J
No
ANY TEXTS ON A2?
ANY TEXTS ON A6?
No
Yes
Yes
I
ISW ≤ 3
ISW ≤ 4
A
A


K
END OF EXPANDED TAPE?
No
Yes
L
M
1,3
IRORD = ?
4,5
2
A
HAVE 49 REQUEST BEEN PROCESSED?
No
Yes
A
IRORD > 2?
Yes
No
IRF ← 0
IYOUY, ANSWER,
DOCDOC ← 0
IRORD ← 2
A
IRF ← 0
LF ← 0
IRORD ← 4
A
ITOUT = D?
No
Yes
A
ANSWER = D?
Yes
No
A
CALL OUTCOR

3. The MERVEC Routine

A. Description

Subroutine MERVEC handles the merging, stacking, and output processing of each vector. It also calls the appropriate subroutine for reading tape or cards, and for weighting, printing, and punching vectors. MERVEC uses five entry points:

MERVEC	read in documents looked up this run from A5 and A7.
AFTIN	read in documents expanded by CONCON or HIER.
CADIN	read in previously looked up documents from A2, the input tape.
DOCIN	read in previously looked up documents from A6, the collection tape.
FINFIN	flush buffer and put end of file mark on B1, vectors to be expanded.

(a) Main Logic

The first four entry points of MERVEC perform the same function and differ only in their mode of input. MERVEC and AFTIN cause concept numbers to be returned sequentially from a buffer periodically filled by MERVEC through calls to INOUT. CADIN causes concept numbers to be returned by calls to external subroutine SUPPLY, while AFTIN uses SUPTAP for the same purpose. Only when entry point MERVEC is used will documents be punched, since the other enty points are reading either documents looked up and already punched in preceding runs, or expanded vectors which may

not be punched. Entry point MERVEC is the only one which has to deal with a separate tape of syntax-produced concept numbers.

Concepts are picked up sequentially in the main loop of the program and internal subroutine USE is called. The documents of the words which contain the concept numbers are compared until a new concept number is found. The new concept is saved and internal subroutine EUSE is called to take care of the old concept. Then the loop begins again with the new concept. A vector is terminated when internal subroutine GET1 returns a zero concept number. When entry point MERVEC is being used and SYNTAX is "on," GET2 must be called by the main loop to provide sequential syntactically-produced concepts for merging into the vector.

Before exiting, MERVEC informs the PUNCH and WRITE routines that a vector has been terminated by calling ENDPUN and WFIN. If the current text was being written out for document-document expansion, internal subroutine DEND is called to flush the buffer.

MERVEC returns a FORTRAN integer in the third parameter of its calling sequence. This integer is one if all processing was performed according to plan; two if a request or *LIKE document did not fit into the stack; and greater than two for an I/O transmission error.

(b) Internal Subroutine

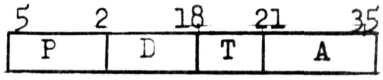
USE calls WRITE if PRNVEC is on; it calls WEIGT if LOGVEC is not on, and calls PUNPUN if the vector is to be punched. Use then returns.

EUSE calls WWGT if PRNVEC is on, calls TOUT if the vector is to be put out for expansion, DOUT if the vector is to be put out for document-

document correlation, and stacks the concept if a correlation is to be perfomed immediately.

TOUT	writes vectors out on B1 for concept-concept or hierarchy expansion.
DOUT	puts vectors out for document-document correlation
DEND	flushes out the DOUT buffer.
TEND	flushes out the TOUT buffer and puts an end of file mark on B1.

(c) Formats



36 Bit Word

Before expansion, the decrement contains a fifteen-bit concept number, the address contains a weight, and the tag and prefix are specified as fcllows:

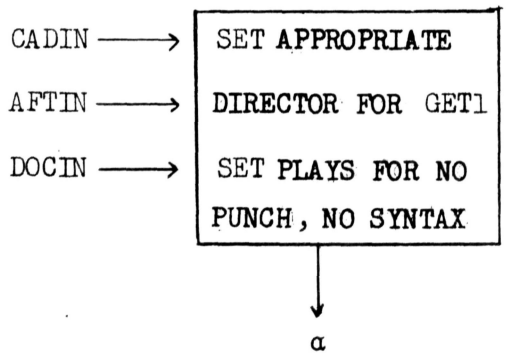
Prefix =	Tag=	Concept derived from:
0	0	word stem from body
0	1	syntactic phrase from body
0	2	statistical phrase from body
0	4	word stem from title
0	5	syntactic phrase from body
0	6	statistical phrase from body
4	0	cluster from body
4	4	cluster from title
5	0	author
6	0	journal
7	0	bibliographic entry

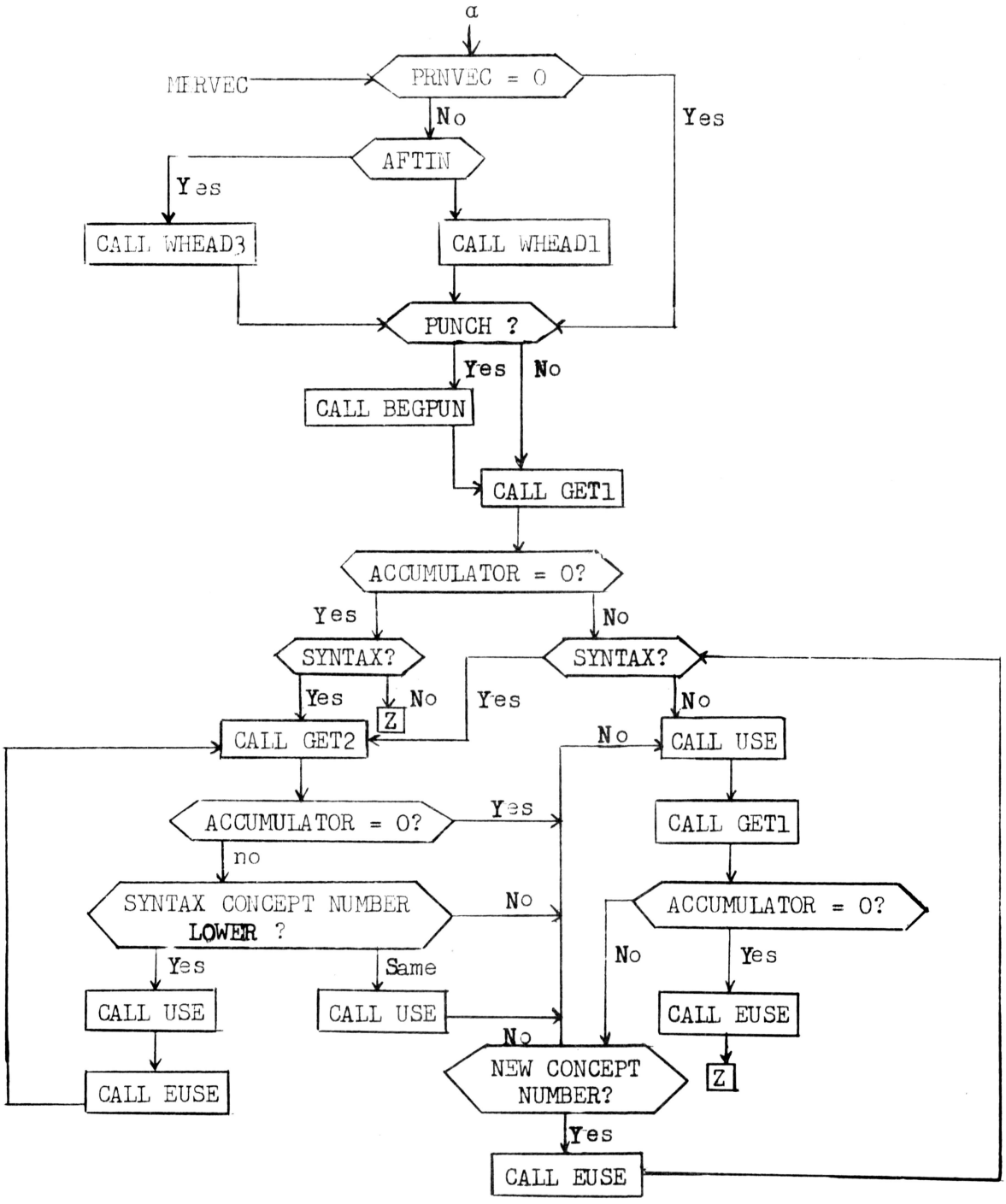
After expansion, the entire left half of the word contains an 18-bit concept number; the address contains a weight; and the tags are specified as follows:

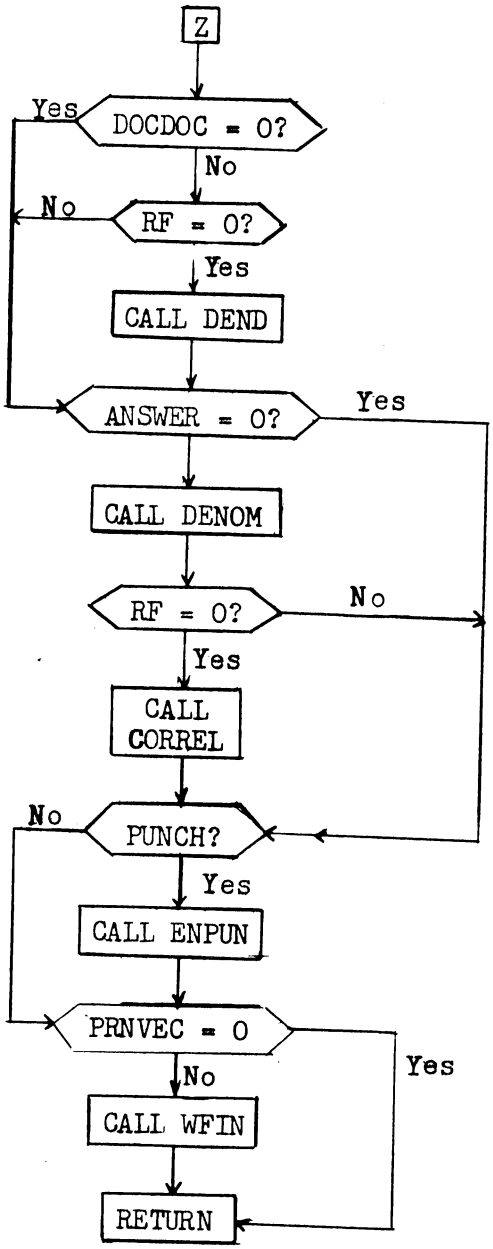
Tag	Concept from:
0	original vector
1	concept-concept expansion
2	hierarchy parent
3	hierarchy brother
4	hierarchy leaf
5	hierarchy cross-reference

B. MERVEC Flowchart

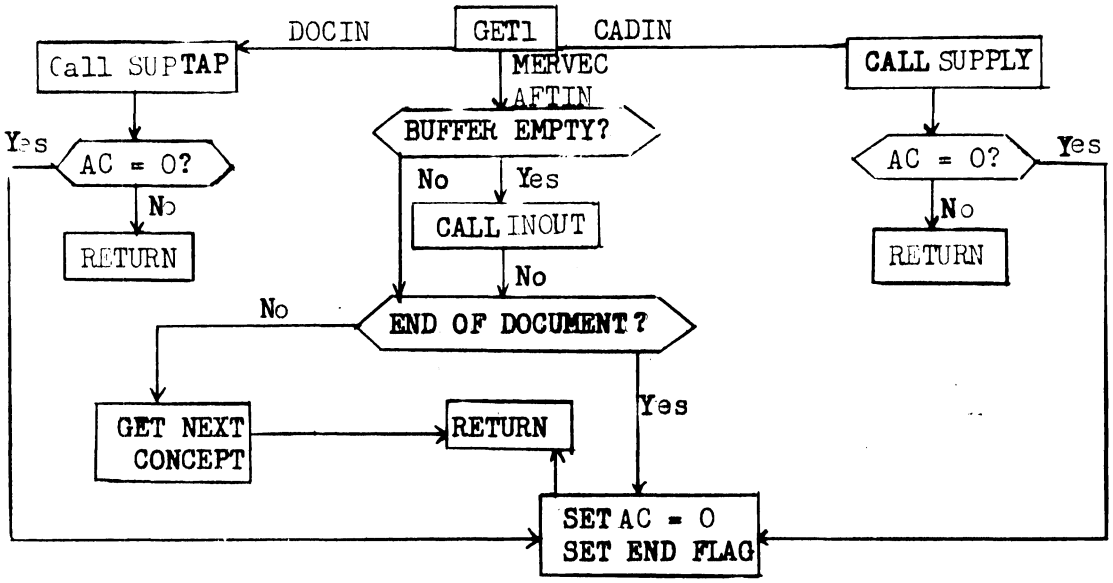
(a) Main Logic

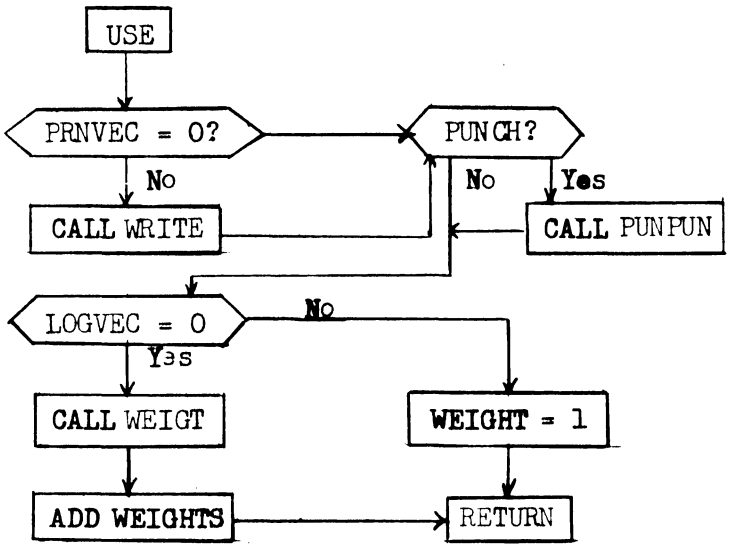


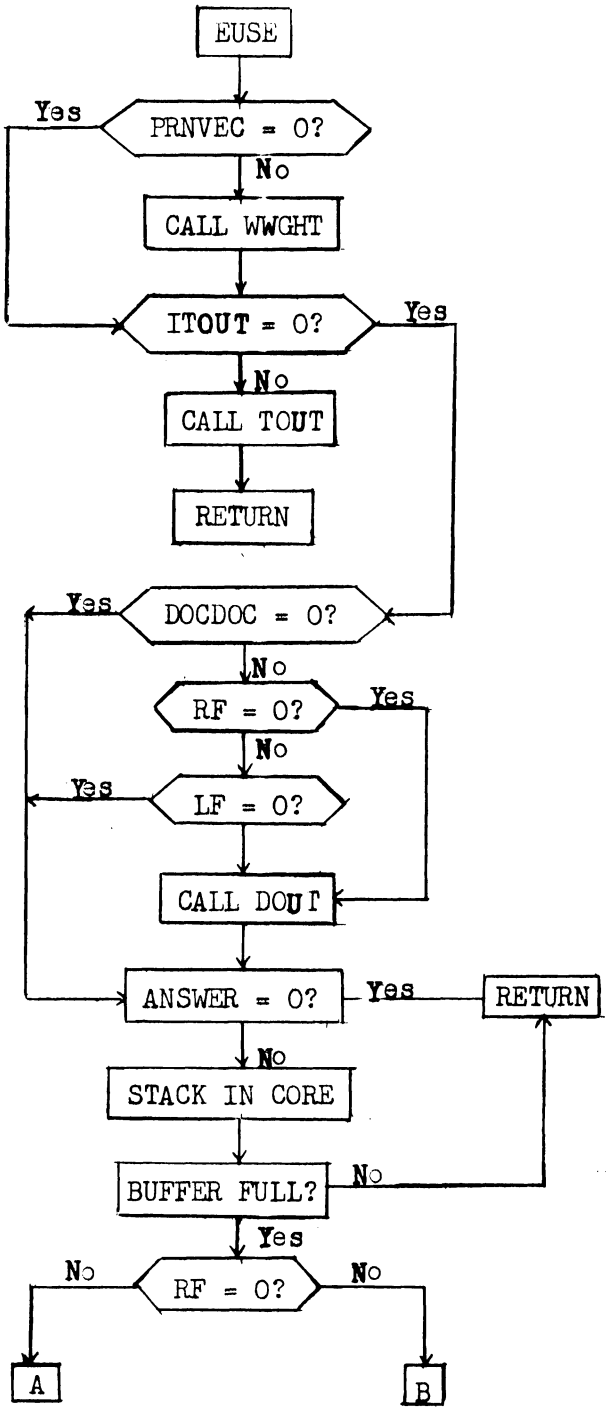

a
MERVEC
PRNVEC = 0
No
Yes
AFTIN
Yes
CALL WHEAD3
CALL WHEAD1
PUNCH ?
Yes
No
CALL BEGPUN
CALL GET1
ACCUMULATOR = 0?
Yes
No
SYNTAX?
SYNTAX?
Yes
No
Z
Yes
No
CALL GET2
CALL USE
No
ACCUMULATOR = 0?
Yes
CALL GET1
no
SYNTAX CONCEPT NUMBER LOWER ?
No
ACCUMULATOR = 0?
No
Yes
Yes
Same
CALL USE
CALL USE
CALL EUSE
No
NEW CONCEPT NUMBER?
Z
CALL EUSE
Yes
CALL EUSE

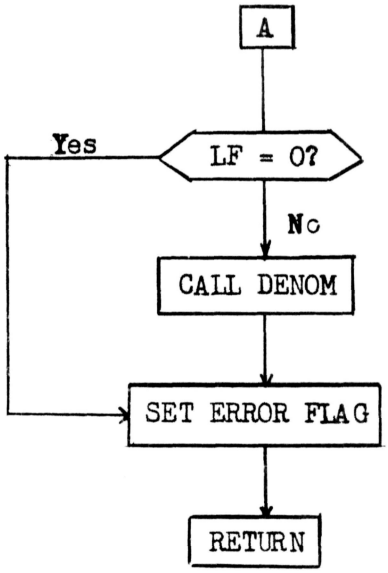

Z
DOCDOC = 0?
Yes
No
RF = 0?
No
Yes
CALL DEND
ANSWER = 0?
Yes
CALL DENOM
RF = 0?
No
Yes
CALL CORREL
PUNCH?
No
Yes
CALL ENPUN
PRNVEC = 0
Yes
No
CALL WFIN
RETURN

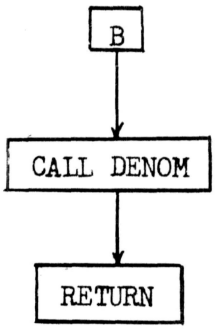
C. Internal Subroutine






EUSE
PRNVEC = 0?
Yes
No
CALL WWGHT
ITOUT = 0?
Yes
No
CALL TOUT
RETURN
DOCDOC = 0?
Yes
No
RF = 0?
Yes
No
LF = 0?
Yes
CALL DOUT
ANSWER = 0?
Yes
RETURN
No
STACK IN CORE
BUFFER FULL?
No
Yes
RF = 0?
No
No
A
B


A
Yes
LF = 0?
No
CALL DENOM
SET ERROR FLAG
RETURN


B
CALL DENOM
RETURN

4. The WRITE Routine

A. Description

WRITE handles the printing of vectors on the output tape, A3. WRITE has seven entry points, all of which are called by a standard FORTRAN call.

The first four entry points to WRITE are WHEAD1, WHEAD2, WHEAD3, and WHEAD4. These entry points initialize the succeeding routines and must be used before calling WRITE for the first time and before calling WRITE after calling WFIN. The purpose of WHEAD1, WHEAD2, WHEAD3, and WHEAD4, is to set up the proper page heading for the type of vector being printed. These headings are specified as follows:

Entry point	Resulting Page Heading				
WHEAD1	CONCEPT NO	STEM BODY TITLE	STAT PHR BODY TITLE	SYN PHR BODY TITLE	TOT WGT
WHEAD2	CLUSTER NO	OCCURRENCES BODY TITLE	WEIGHT		
WHEAD3	CONCEPT NO	STEM CONCON	ROOT LEAF	CROSS	TOT WGT
WHEAD4	CLUSTER NO	OCCURRENCES BODY TITLE	WEIGHT		

After setting up the heading, calls may be made to the entry point WRITE. WRITE causes concept numbers and weights to be listed in the proper

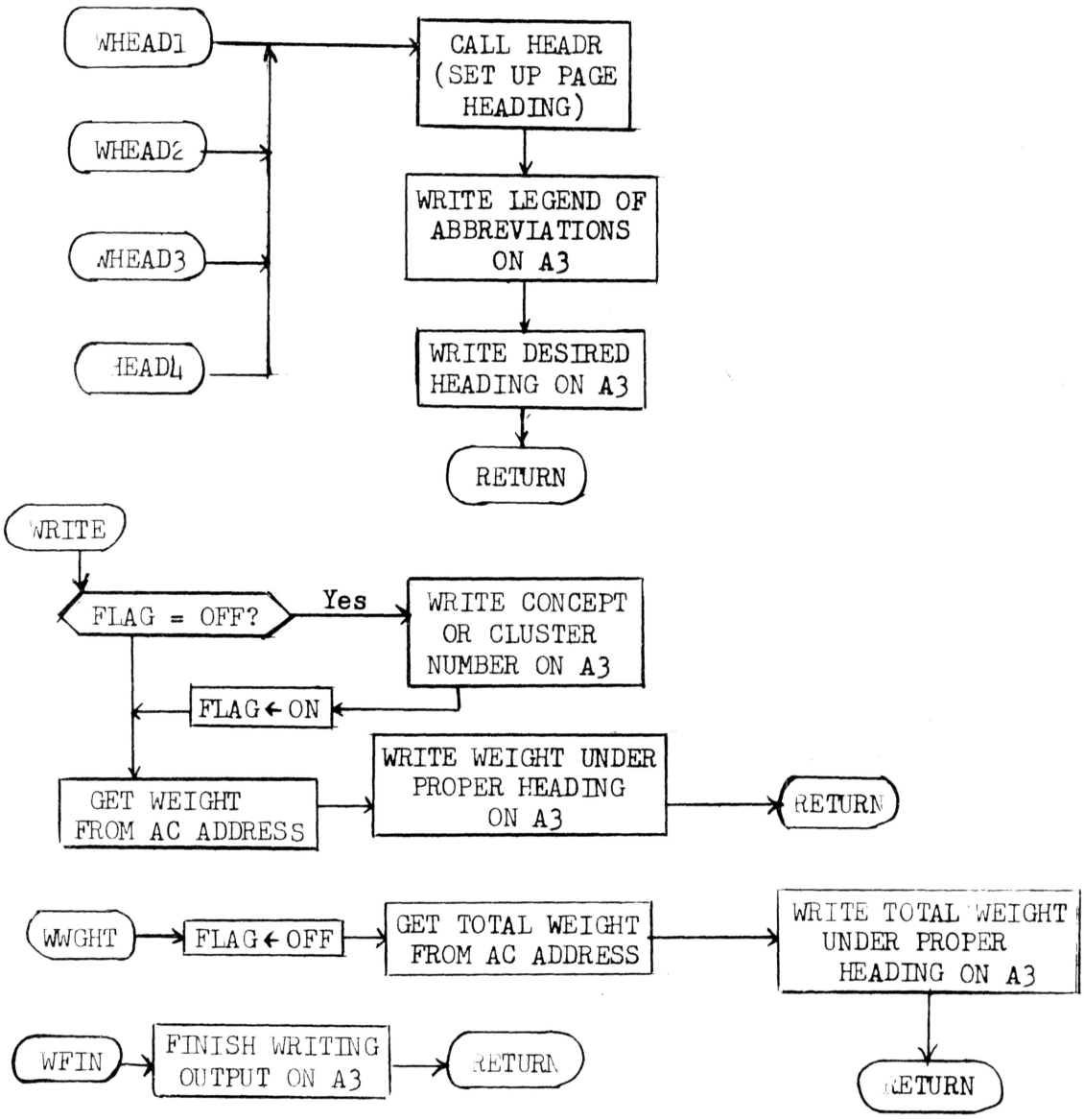
column under the heading previously selected. It finds these numbers and weights in the accumulator (AC) expects them to have the formats specified above. When first called WRITE assumes the AC address to contain a concept or cluster number with a weight in the AC decrement. All succeeding calls to WRITE are assumed to refer to this concept or cluster number until a call is made to WWGHT. The weights from the AC address are inserted in the proper column under the selected heading according to the prefix and tag codes shown in the tables of Part 3 of this section.

A call to WWGHT prepares WRITE to accept a new concept or cluster number. The total weight for the old concept or cluster number is taken from the AC address and inserted under the heading. The next call to WRITE will start a new line in the output table.

The last entry point to the WRITE subroutine is WFIN. This call informs WRITE to finish its output transmission and the last call to the WRITE subroutine must always be made to WFIN. WFIN must also be called before changing from one heading to another.

WRITE uses the subroutines HEADR, BCDDEC, BCDADD, and INOUT.

B. WRITE Flowchart



5. The SUPPLY and SUPTAP Subroutines

A. Description

Documents which are not submitted as English texts are read-in by these routines. Two types of preprocessed documents are acceptable, cards from tape A2, and a document tape on A6.

Cards from A2 are read by SUPPLY. SUPPLY reads concept vectors from decks punched in earlier runs, and supplies them to MERVEC. The identification associated with the document is written onto a tape for use by the REPORT program (link 13). This tape is usually A4, but if the supervisor executes a call BEGSUP (ITAPE), all identification for *FINDS and *LIKES is placed on ITAPE instead. The actual concept vectors are obtained by calling SUPPLYF, which returns one word containing a concept number and a weight in standard MERVEC format. At the end of a document, SUPPLY returns a word of zeros as a sentinel. The identification written out is as follows:

If ANSWER is SHORT,	none
If ANSWER is MEDIUM,	the *FIND, *LIKE, or *TEXT card
If ANSWER is LONG ,	the complete citation (*TEXT card, author, title, and journal)

Another entry point, IFSUPF, returns a nonzero value if SUPPLY has written any identification on a tape other than A4; otherwise it returns zero.

SUPTAP is similar to SUPPLY except that it reads concept vectors from A6, a previously mounted document collection tape. This tape is blocked in 250-word records. SUPTAP has the following entry points:

(1) SUPNAM (ARG) returns the leader card for the document to ARG.

(2) SUFTAPF looks exactly like SUPPLYF to MERVEC.

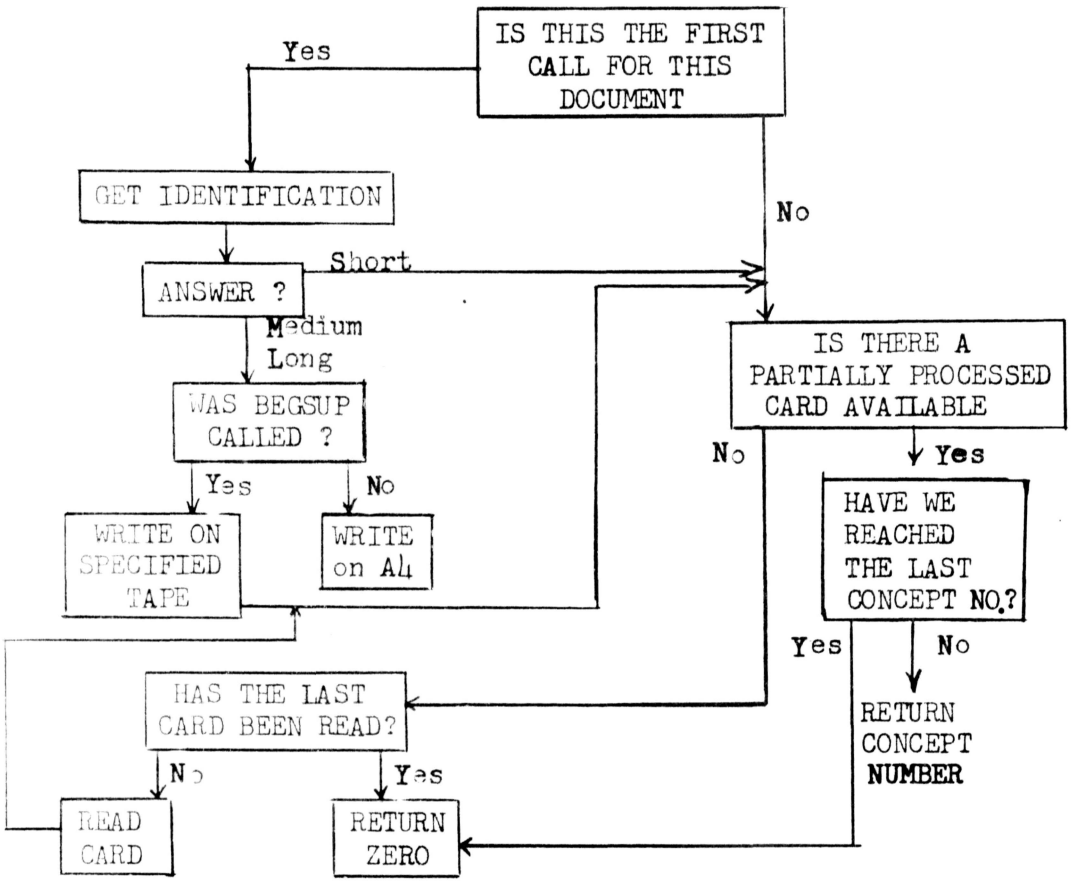
(3) IFSUPF, is the same as in SUPPLY.

(4) BEGSUP is exactly as in SUPPLY, identification is handled in the same way.

The document collection tape is written by MAKTAP, an independent program. MAKTAP requires as data input, the binary decks if the documents to be written on the document tape (A6). Several sets of binary decks may be written on the same tape. They are separated by cards with *NAME in columns 1-6 and the 12-character name of the collection in columns 7-18.

To use the document tape, the SETFLX specifications DOCTAP should be used. The first card after the SETFLX cards should have *FILE in columns 1-6 and either a decimal integer or a collection name in columns 7-18. If an integer n is given the nth collection is read. If a collection name is given, that collection is read. If the card is omitted, the first collection is read.

B. SUPPLY Flowchart



6. The ERROR Routine

ERROR is a simple program which prints out a number transmitted to it as part of the message "ERROR WAS DETECTED. NUMBER WAS." If the number is less than 0-equal to ten, ERROR returns; otherwise it calls DUMP.

ERROR NUMBER	MEANING
1	(unassigned)
2	illegal control card
3	too many requests, remainder will be ignored
4	request out of order, ignored
5	*LIKE will be treated as *TEXT only
6	request overflowed buffer, ignored
7	*LIKE overflowed buffer, treated as *TEXT only
8	*LIKE out of order, treated as *TEXT
9	(unassigned)
10	(unassigned)
11	can't write document-document tape, MERVEC
12	can't write B1, MERVEC
13	can't read expanded tape, MERVEC

ERROR NUMBER	MEANING
14	can't read syntactic tape, A7, MERVEC
15-20	(unassigned)
21	can't write B1, MERVEC
22	can't read A5, SUPER 8
23	can't read A2, SUPER 8
24	can't read B1, SUPER 8
25	can't write A4, SUPER 8
26	can't write correlation tape CUTCOR