DEPARTMENT OF MATHEMATICS
UNIVERSITY OF HOUSTON HOUSTON, TEXAS

NASA CR-
141622

(NASA-CR-141622) PROGRAM DOCUMENTATION OCM, HISTOG, PRINTUM (Houston Univ.) 37 p HC $3.75 CSCL 09B

N75-17981

Unclas
G3/61 10206

# PROGRAM DOCUMENTATION
# OCM, HISTOG, PRINTUM

BY JACK BRYANT

AUGUST 1974. REPORT #35



PREPARED FOR
EARTH OBSERVATION DIVISION, JSC
UNDER
CONTRACT NAS-9-12777

3801 CULLEN BLVD.
HOUSTON, TEXAS 77004

Program Documentation

OCM
HISTOG
PRINTUM

By

Jack Bryant
Texas A & M University
Consultant to
Mathematics Department
University of Houston
Houston, Texas

August 1974

Report #35
NASA Contract NAS-9-12777

COMPUTER PROGRAM DOCUMENTATION

ONE CHANNEL MAPS

PROGRAM OCM

By

Jack Bryant

August 1974

## PURPOSE

In dealing with large two-dimensional arrays (such as those representing digitized pictures), one difficult problem is to determine quickly and cheaply the result of an operation. For example, during the development of a program to filter digital data, the logic of how to handle the edges, loss of sync, dropped lines and so on is complicated enough to cause major debugging problems. A simple , even well known, program such as one to rectify or register data must be tested and debugged on the particular format of the data available. The whole process of developing software is slowed when a special display device must be used to view an intermediate stage of a program. It has long been recognized that a natural way of obtaining a quick (if crude) view of a large array is to use the high-speed printer of the computer with different symbols representing ranges in the data.

The problem attacked by program OCM (One Channel Map) has a more critical feature. In order to obtain training data for a classification algorithm, the exact (digital) coordinates of the boundaries of individual fields must be known. For this reason, even high quality hard photographic information may be useless. On the other hand, average quality programs (to produce maps on the printer) fail to display the data in sufficient detail to allow the user to locate (on the printout) fields corresponding to ground-truth. Program OCM fills the need for better quality reproduction of digital pictures on the high-speed printer.

## METHOD

Program OCM achieves the objective by printing an appropriate selection of characters eight lines per inch on blank (unlined) paper. The translation of the data is performed by a subroutine written by Mr. Dale Ruspino (at Texas A&M) which uses effectively the IBM System/360 "translate under mask" instruction to translate long strings in place, essentially instantaneously. Experimentally it was learned that, what ever the choice of symbols, only five levels of gray are unambiguously detected with a single print, and as the ribbon wears this number becomes four. With a new ribbon and overprinting once, with an additional translate to get the overprinted character, eight levels are detectable.

In order to use the translate subroutine, translate tables must be set up. The minimum and maximum significant values in the data must be specified. The program divides the data spread into eight equal (as possible) partitions, sets all lower values in the first partition, all higher in the eighth, and assigns symbols (suggested symbols are given in USAGE below) to each range. (The value 0 is assigned a special symbol.) The data is assumed to be a positive, in that low values represent dark areas; thus the first three symbols are overprinted to make them darker.

Program OCM then initializes a variable which breaks the data into one page size parts of not more than 126 columns each. (If more than this many columns are required, the program will return to this point, continuing until all data is printed.)

The program then initialized the associated variable IREF for direct access device 1 (set up by a DEFINE FILE statement and appropriate JCL), enters a loop to read, find the next row, translate and print and translate and overprint the row just read, until all rows are finished. If any columns remain, the program returns for more printing.

The translate program is so fast that no difference can be detected whether it is present or absent by looking at the run time.

DESCRIPTION OF PROGRAM VARIABLES

| | | |
|---|---|---|
| DUM(4) | LOGICAL*1 | Dummy vector equivalenced to IT - used in definition of translate table TS. |
| I,J,L | | Generic DO indicees |
| IC,ICP | | First and last column index currently being printed |
| ICE,ICS | | Last and first column index (read as data) to be printed |
| ICHAN | | Channel number (read as data) |
| IRE,IRS | | Last and first row index (read as data) |
| IREF | | Associated variable for unit 1 |
| IT | | Integer used as an index by filling the last byte via DUM and an equivalence statement. |
| IW | | Write parameter used in setting up translate table TR |
| MAX,MIN | | Maximum and minimum expected significant data values (read as data) |
| R(500) | LOBICAL*1 | Buffer into which each row is read. |
| SYMBOL(12) | LOGICAL*1 | Symbol table: first for 0, next eight first breakdown (i.e. for TR) next three overprinted over 2,3,4 (via TS) |
| TR(256) | LOGICAL*1 | Translate tables for subroutine |
| TS(256) | LOGICAL*1 | HP05 |
| X8 | | (Max-Min)/8.; used to accurately spread the data into eight parts. |

USAGE

Detailed deck set up and JCL: TAMU 360/65 with HASP

```
//NAME      JØB      (standard jobcard)
                        HASP CARDS:

/*CLASS          A              (0-110K)

/*FØRMS          14101010       (blank paper, 1 part, 8 lines/inch setup)

/*LINES/PAGE     0              (user controls spacing to new page)

/*RØUTE          PRINTER1       (selects a generally better operated printer)
                        EXEC CARD:

//        EXEC  FØRTG,REGIØN= 110K  (invokes FORTG cataloged procedure)
                        DD CARDS:

//SYSLIB   DD

//         DD

//         DD   DSNAME=USER.T405.EE.PRØBLIB(HP05),DISP=SHR
           (this JCL allows FORTRANG to retain its SYSLIB and adds HEX
           program HP05, resident on the described private disk)

//FT01F001 DD   UNIT= 2314,VØL=SER=JACK01,DSN=HILLCØ,

//         SPACE=(500,12000),DISP=(ØLD,KEEP),DCB=(RECFM=F,DSØRG=DA)
           (this DD card makes FØRTRAN UNIT 1 a direct access device)

//SØURCE   DD   *               (source program next)
           PROGRAM OCM DECK

//SYSIN    DD   *               (data next)

           DATA  (described next)

/*                              (end of file)
```

DATA DESCRIPTION:

SYMBOL FORMAT 12A1

ICHAN,IRS,IRE,ICS,ICE FORMAT 5I3

MIN,MAX FORMAT 2I3

repeat for as many pictures as may be desired; each picture thus takes two data cards.

Subroutines required:

HP05 Assembly language program to perform translation.

CØLS Program to print column headings, which requires

TØDEC Program which furnishes the digital (base 10) representation of the column index.

Restrictions:

This program is highly machine dependent, even some what installation dependent. Clearly, however, the idea can be implemented, for instance on the 1108, using ȠTRAN and a translate instruction subroutine PRETTY (documented in the Lunar Orbiter Project) written in SLEWTH. The logic is trivial in any case.

The use of this program requires the data be written on a disk one line of 500 bytes each (125 words) per record unformatted. The data is expected to be lined up somewhat differently than it was on the tapes; there are 12000 records with the J-th row of channel I having index I + 24*(J-1). That is, row 1 channel 1 through row 1 channel 24 then row 2 channel 1 and so on. The value 0 represents no data and is printed differently.

Suggestions:

SYMBOL: A convenient selection of symbols to print is

U (for zero)

M overprinted with $

Ø overprinted with +

0 overprinted with -

*

=

+

-

blank

It has been determined that appropriate values for MAX and MIN for the display of field boundaries are about the top and bottom 7 to 10 percent on a cumulative distribution of all data values. Quite possibly other purposes will require different selections of MAX and MIN.

Run time:

Approximately 0.05 seconds per 126 character row desired. Thus, a run of 12 channels, cols 50 to 427 (requiring 3 126 col pages to print), rows 80 to 458 should take about 682.20 seconds; the actual run time was 689.40 seconds, including overhead charges. This is a lot of printing for which the overhead is high.

Lines output:

Specify two lines output for each line actually printed, plus an allowance for column headings and program and JCL listing.

REFERENCE INFORMATION INCLUDED:

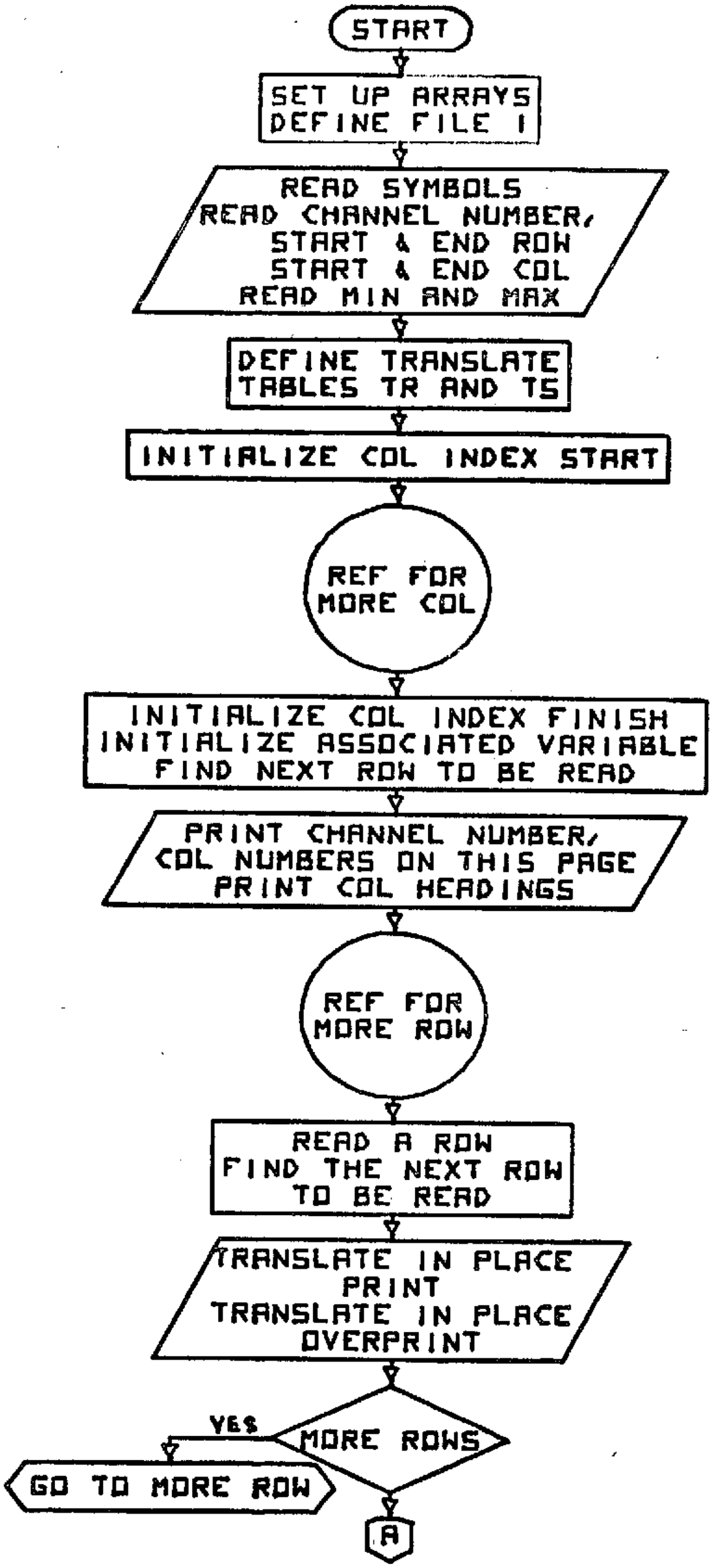
General flow chart

Detailed flow chart

Documentation of COLS

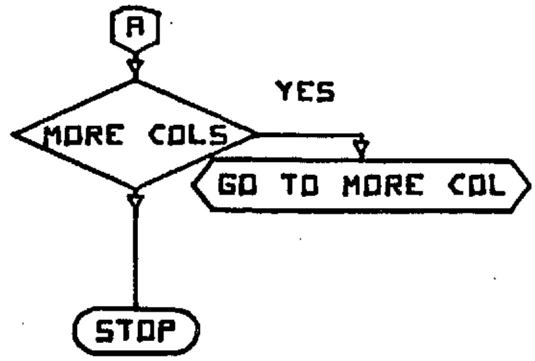
Comment on HP05

Sample run



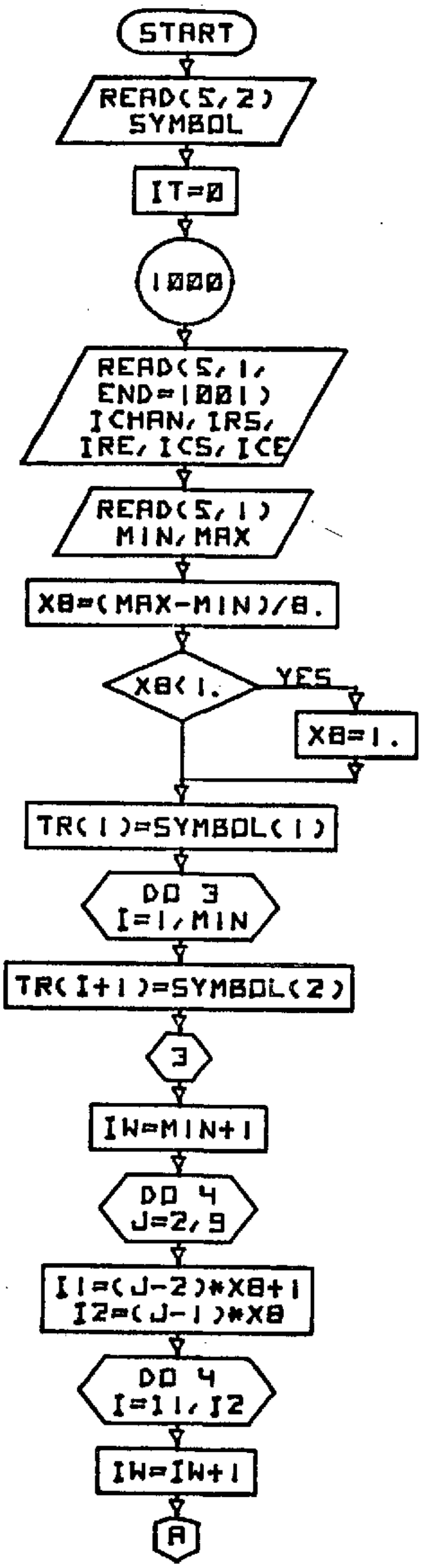



General flow chart - OCA



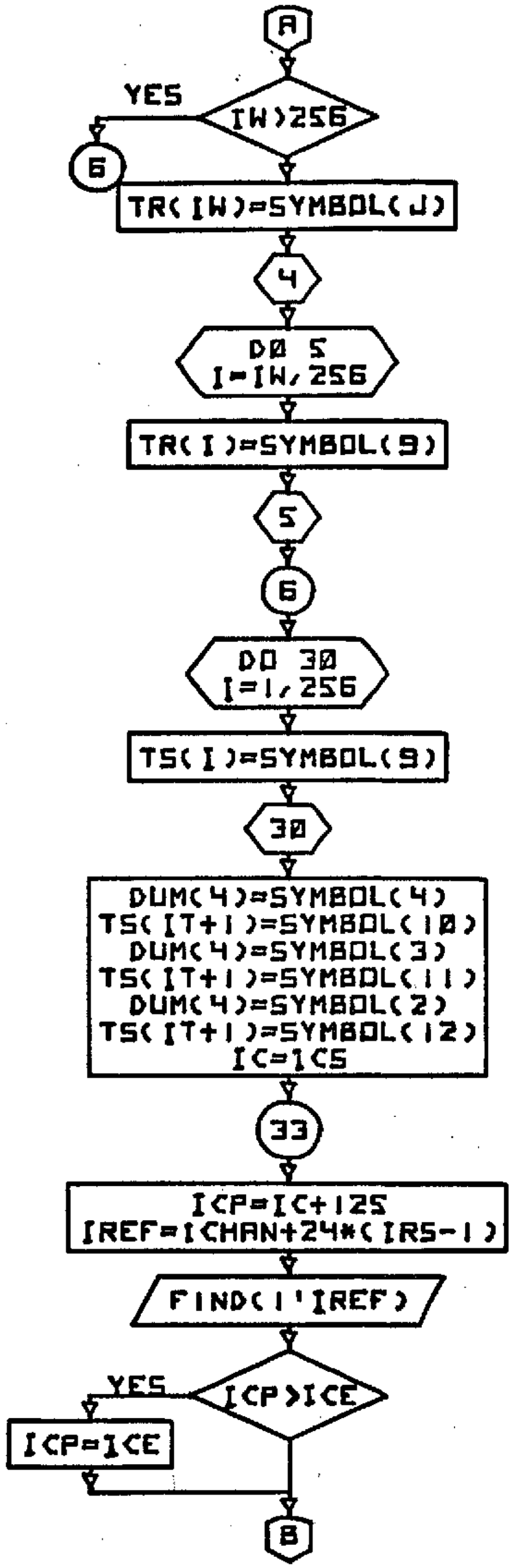



General flow chart - CCM-2



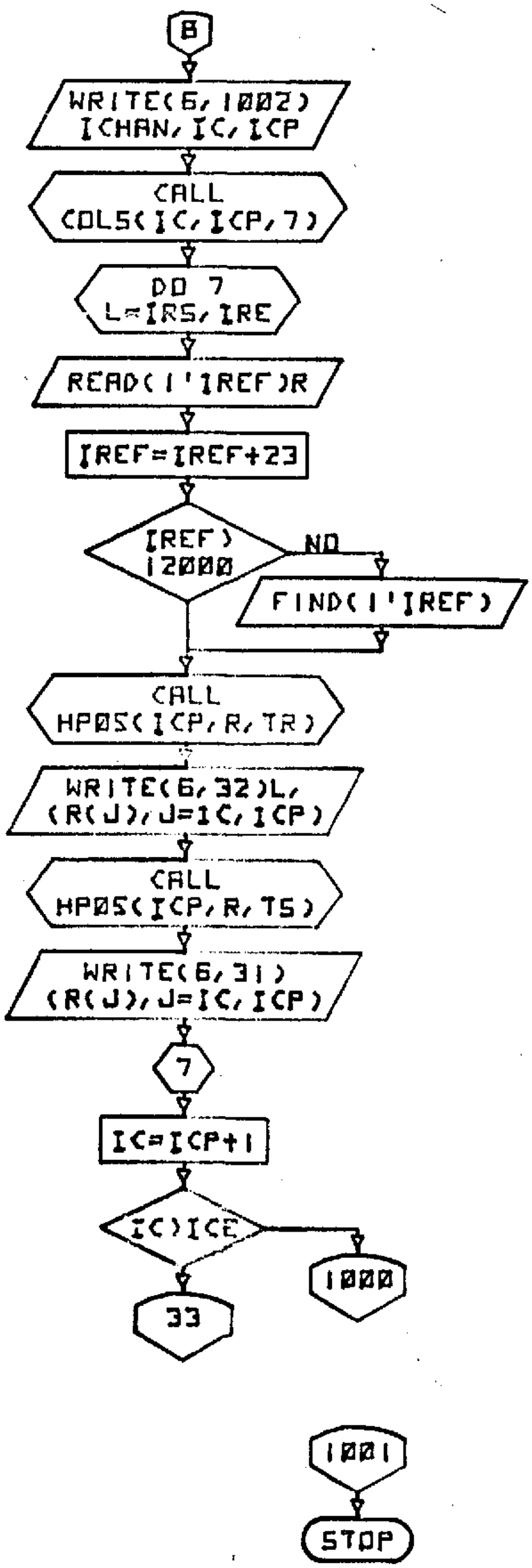



Detailed flow chart - OCM - 1






Detailed flow chart - CCM - 2






Detailed flow chart (cont. 3)

SUBROUTINE DOCUMENTATION: SUBROUTINE COLS

PURPOSE: Subroutine COLS prints column headings at the top of a page written vertically with leading zeros suppressed. It is used to greatly simplify the finding of the coordinates of field boundaries.

METHOD: In order to both speed the operation and simplify the code two arrays are employed. The full word integer vector FAST is equivalenced to a LOGICAL*1 array PAGE. Initially FAST is filled with all blanks (40 Hex). Then a loop is entered which fills FAST with the digits of the column headings to be printed in the appropriate positions. Logic is included which suppresses leading zeros. The conversion to base ten is performed by subroutine TØDEC (which is self-explanatory). Then FAST is printed in one I/Ø statement.

USAGE:

Calling sequence: Call CØLS(I,J,N)

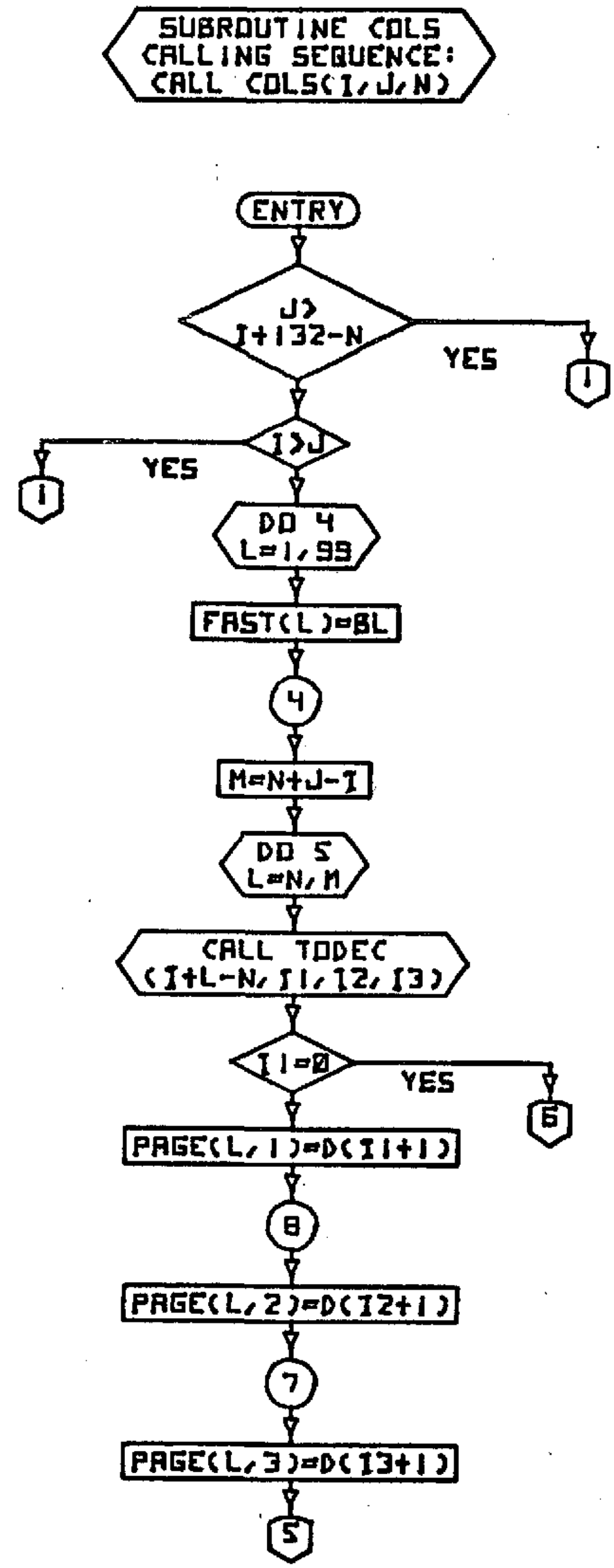
where I is the index of the first column to be printed

J is the index of the last

N is the printer position corresponding to the first column

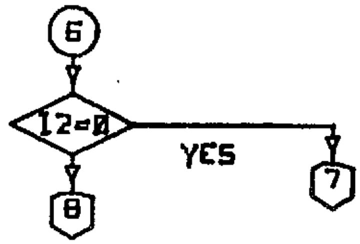
Restrictions: Obviously, $0 \le I \le J \le I + 132 - N$. A diagnostic to check this is included in the program. Also, TØDEC requires $J \le 999$.

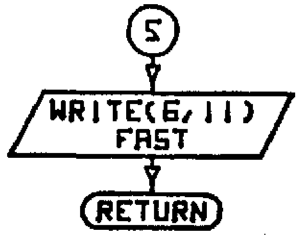
REFERENCE INFORMATION: Detailed flow chart, CØLS and TØDEC.

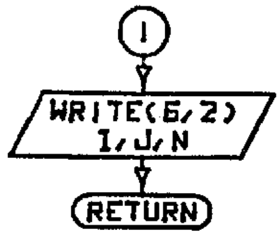





Detailed flow chart - COLS




FORMAT CODES:

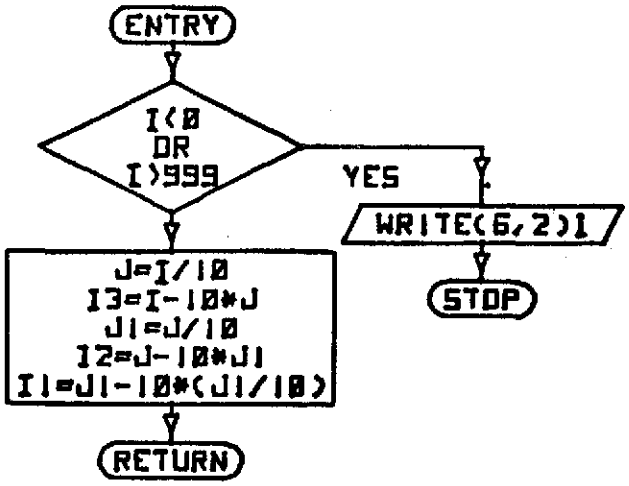
```
11=('1',33A4/' ',33A4/' ',33A4/)
 2=(' UNACCEPTABLE INPUT TO COLS:',3I10/'1')
```

PROGRAM VARIABLES D AND BL ARE INITIALIZED IN TYPE DECLARATION STATEMENTS. VARIABLES PAGE AND FAST ARE EQUIVALENCED.



D. tailed flow chart - COLS

SUBROUTINE TODEC
CALLING SEQUENCE:
CALL TODEC(I,I1,I2,I3)




FORMAT STATEMENT REFERENCED:

2=(' UNACCEPTABLE INPUT TO TODEC -- FATAL ERROR',I10)



Detailed flow chart - TODEC

SUBROUTINE COMMENT: SUBROUTINE HP05

Authors: Dale Ruspino and Bruce Marion, July 1974.

Calling Sequence: CALL HP05(I,R,TR)

where

I is the length in bytes of that portion of the vector R to be translated;

R is a vector of arbitrary length (within the maximum dimension restriction of FORTRAN) to be translated in place;

TR is the vector of length 256 (bytes) containing the translate table.

Method: Using the 360 assembly language instruction TM, the vector R is broken down into 255 length pieces, translated (ie the bytes are replaced by others according to what is contained in TR) until all I bytes have been processed. For example, suppose

I is 2, R is (in Hex)

01 0D and TR starts out

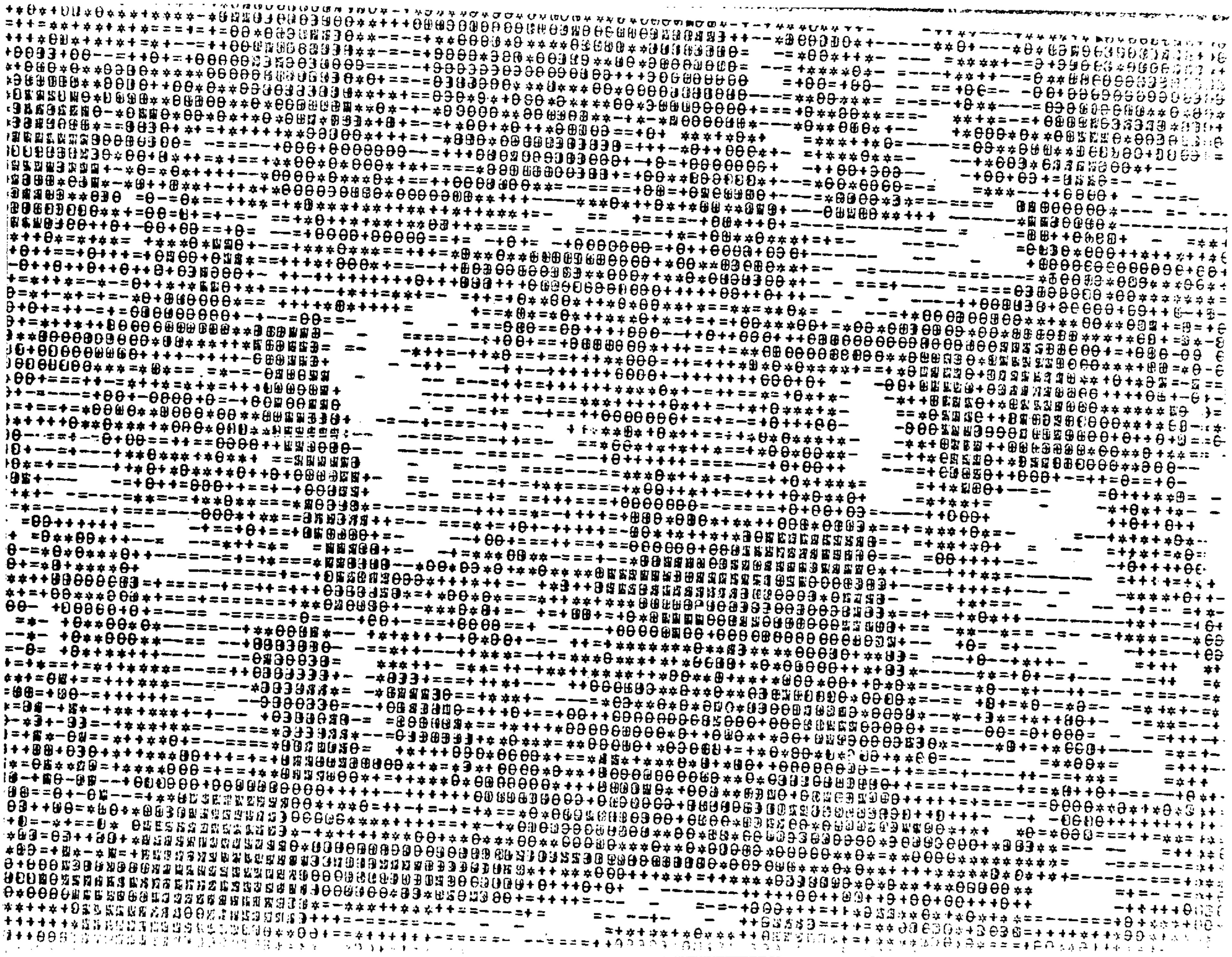
CC 17 DD 40 40 40 40 40 40 40 40 CC 22 13 ...

After the call to HP05, R is (in Hex) 1713. HP05 attains its speed by the use of this parallel translate instruction and by not saving registers which are not disturbed and not checking the input for errors. Indeed, HP05 has proved to be at least as fast as a dummy program (in FORTRAN)

```
SUBROUTINE     HP05(I,R,TR)
LOGICAL*1    R(1),TR(1)
RETURN
END
```

used while debugging the main program.

Reference information: Listing



COMPUTER PROGRAM DOCUMENTATION

HISTOGRAMS

Program HISTOG

by

Jack Bryant

August 1974

## PURPOSE

In drawing one channel maps on the high-speed printer, one needs to set the maximum and minimum expected significant data values. It turns out that for the 24 channels we have to work with these values vary wildly; for example channel 22 has its 5 to 95 percent cumulative distribution values 27 and 62, whereas channel 24 (on the same pass) has the same values 20 and 31. Program HISTOG accumulates, prints and punches complete histograms of all 24 channels. The card output is available in easy to read form for further analysis.

## METHOD

The program collects data stored on a direct access device and prepares histograms. Since the information is packed by rows, it must be unpacked before it can be used as integer information. This function is performed in FORTRAN by filling the fourth byte of a dummy logical vector (LOGICAL*1) which is alligned with a full word integer using an EQUIVALENCE statement.

## DESCRIPTION OF PROGRAM VARIABLES:

| | | |
|---|---|---|
| H(128) | INTEGER*4 | Used to accumulate histograms |
| I,J,JJ,JP | | Generic DO loop parameters |
| IF | | Used in FIND instruction on direct access device unit 1 |
| IR | | Associated variable for unit 1 |
| L(4) | LOGICAL*1 | L and M are assigned the same storage |
| M | INTEGER*4 | location; used to unpack information |
| R(500) | LOGICAL*1 | Read buffer-contains row most recently read. |

USAGE

Detailed deck setup and JCL:TAMU 360/65 with HASP

```
//NAME      JOB  (standard jobcard)
/*CLASS        A
//    EXEC     FØRTG,REGIØN=110 K
//FT01F001     DD     UNIT=2314,VØL=SER=JACK01,
//    DSN=HILLCØ,SPACE=(500,12000),DISP=(ØLD,KEEP),
//    DCB=(RECFM=F,DSØRG=DA)
//SØURCE       DD     *
      HISTØG  PRØGRAM DECK
//SYSIN        DD     *
/*
```

(For more detailed comments on JCL, see the documentation of program OCM.)

Description of output: The main output is the card output.

(The same information is printed, however.) Each channel produces 15 cards, as follows:

```
CHANNEL-SEQUENCE NUMBER              (I5)
ONE SPACE                            (1X)
H(1),...,H(9)                        (9I8)
           repeat CHANNEL-SEQUENCE NUMBER, 1X
H(10),...,H(18)                      (9I8)
```

The fifteenth (last) card for each channel contains the CHANNEL-SEQUENCE NUMBER, 1X and H(127), H(128).

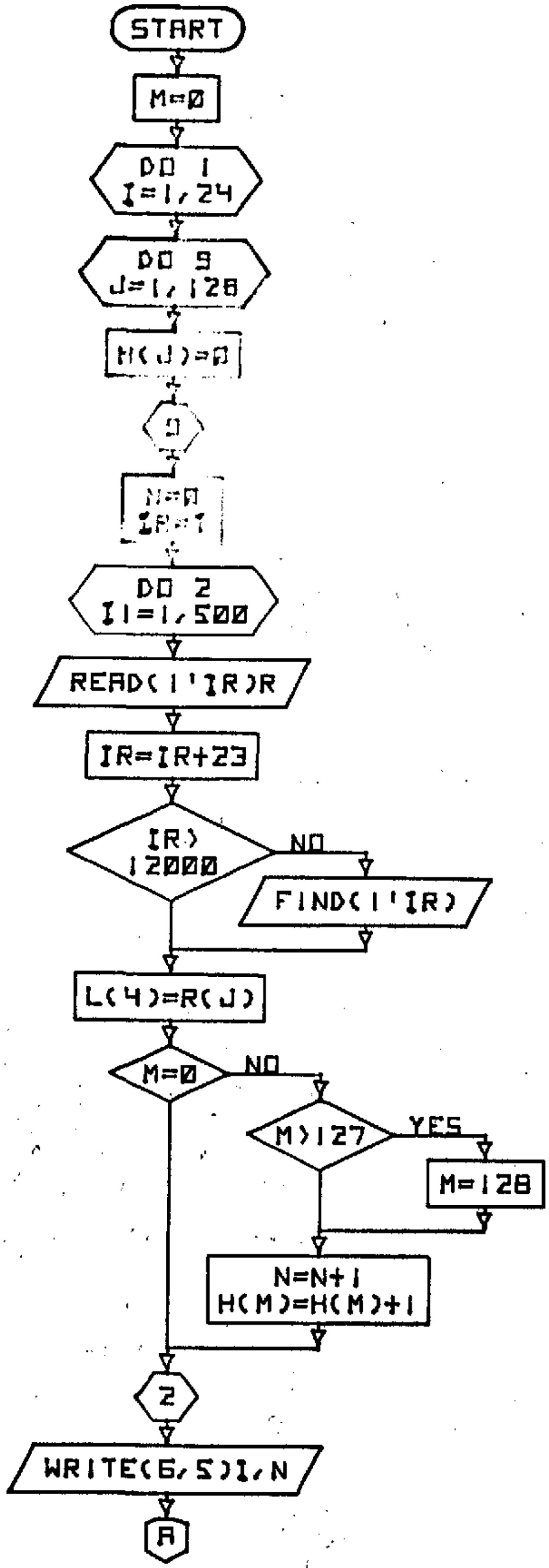
Restrictions: The data value 0 is ignored altogether. Values over 127 (none were found) are accumulated in H(128).

Run time: Slightly less than six minutes are required for all 24 channels; this amounts to about 60 μs for each value unpacked and accumulated, counting all overhead. The overhead associated with 12000 reads is, however, considerable; probably less than a third of the time shown on the printout is spent doing calculations.

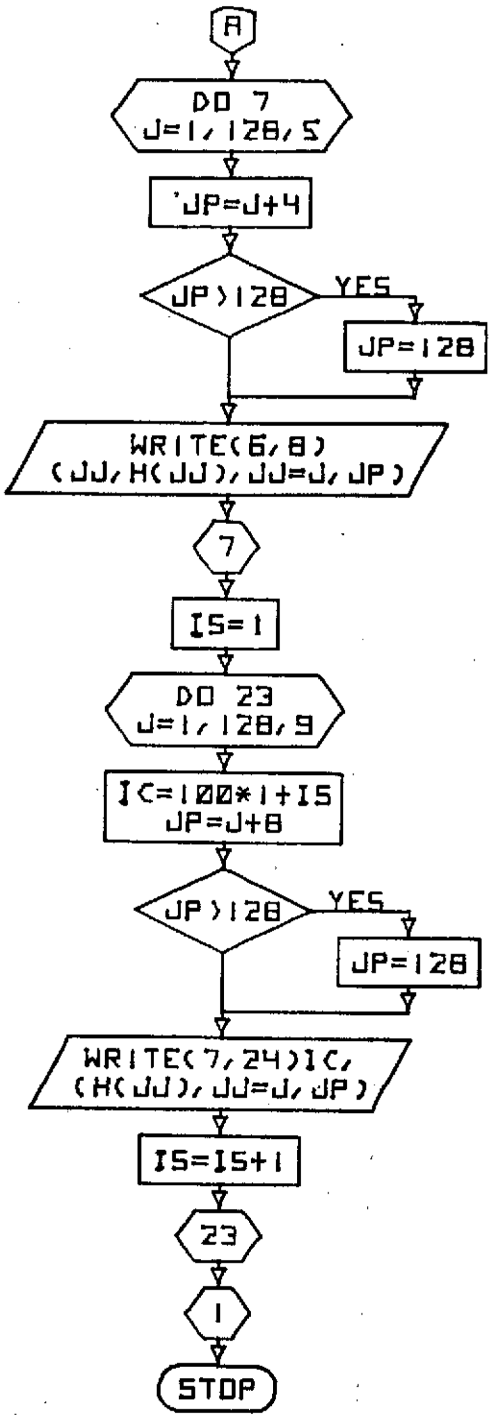
REFERENCE INFORMATION INCLUDED

Detailed flow chart

Sample run




Detailed flow chart 415706 1




Detailed flow chart HISTOG - 2

COMPUTER PROGRAM DOCUMENTATION

# PRINT HISTOGRAMS AND CALCULATE CUMULATIVE DISTRIBUTION

BY

Jack Bryant

August 1974

## PURPOSE

Program HISTOG accumulates raw data (histograms) from the 24 channels stored on the disk; what is needed, however, are estimates of the minimum and maximum significant data values present as input to program OCM. Program PRINTUM calculates the cumulative distributions and also prints histograms (on the printer). Only the 5 to 95 percent distributions are printed.

## METHOD

Program PRINTUM reads the channel number and histogram vector H, blanks the LOGICAL *1 array PAGE (126,127), determines the maximum value M present in the data, fills page with '*' out to the properly scaled index H(I)*126/M, and then prints the histogram. The total number of values accumulated is counted and a cumulative distribution is taken of H (starting with value 1); those percentages lying between 5 and 95 percent are printed. The program returns for more data until an EØF is encountered.

## DESCRIPTION OF PROGRAM VARIABLES

Note: all are INTEGER*4 unless otherwise indicated.

| | |
|---|---|
| A | DO loop index |
| B LOGICAL*1 | Blank |
| C | Channel number; read along with H |
| H(127) | Vector - the raw histogram |
| I | Generic DO loop index; then used to print descending |
| J | Generic DO loop index and parameter |
| JJ,JP | Generic Do loop index and parameter |
| M | Maximum value found in H |

| | | |
|---|---|---|
| PAGE(126,127) | LOGICAL*1 | Array allowing print of histograms |
| R | | Percent printed |
| S | | DO loop index; then the grand total of values in H. |
| ST | LOGICAL*1 | '*' |
| T | | Cumulative sums of H |
| V | | DO loop index. |

USAGE

Detailed deck setup and JCL: TAMU 360/65 with HASP

```
//JØBNAME        JØB     (standard job card)
/*CLASS             A
/*LINES/PAGE        0
//      EXEC        FØRTG,REGIØN= 110K
//SØURCE            DD    *
       PRINTUM PRØGRAM DECK
//SYSIN             DD    *
       DATA - THE CARD ØUTPUT ØF HISTØG
/*
```

Data description:

The card output of HISTØG; specifically, each card contains the channel number (I3) a sequence number which is skipped (3X) and data for reading into H (9I8).

Lines Output:

Approximately 4000 lines for 24 channels.

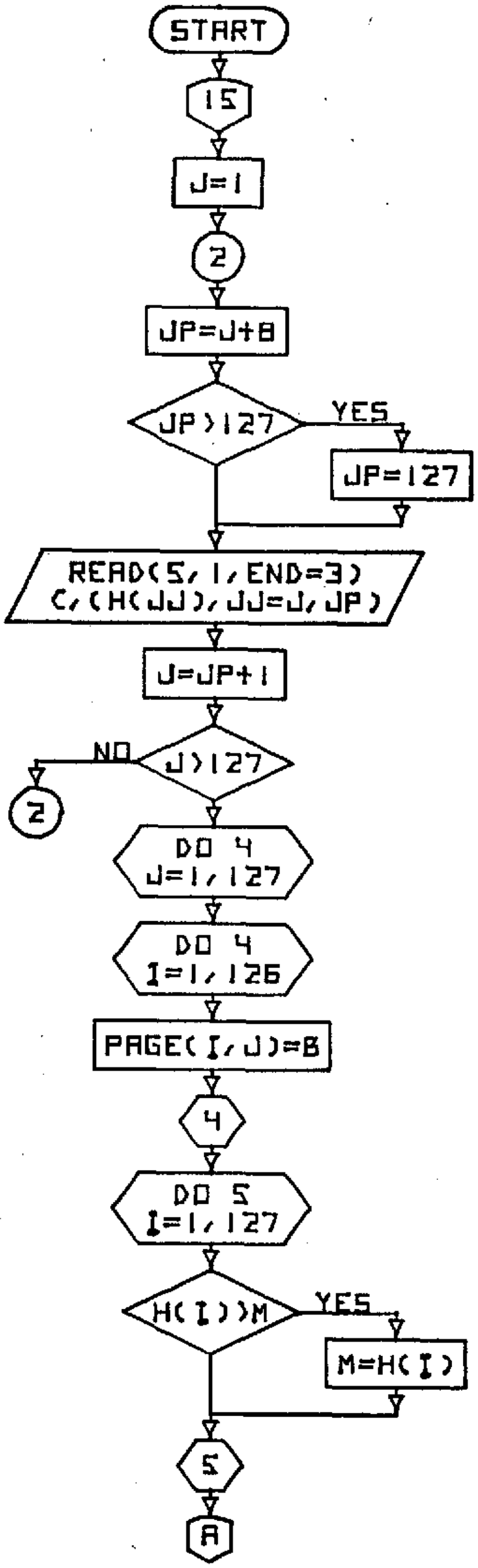
Execution time:

Approximately 3.7 seconds per channel plus overhead; a total of 1.63 minutes for 24 channels.

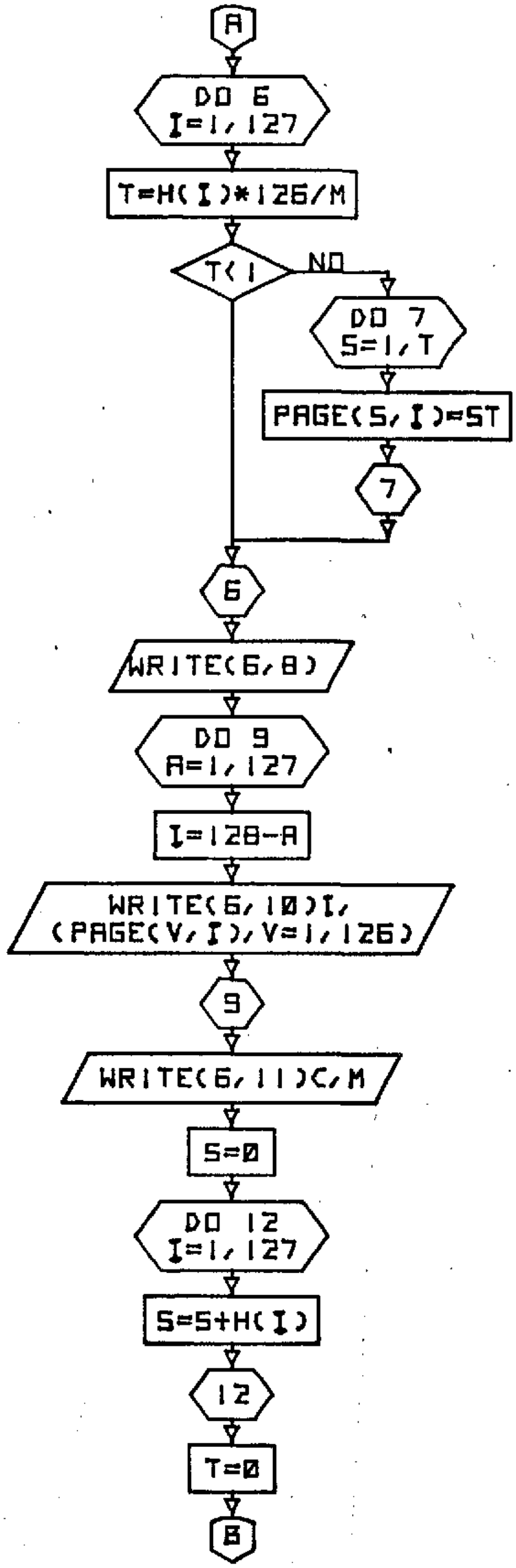
REFERENCE INFORMATION INCLUDED:

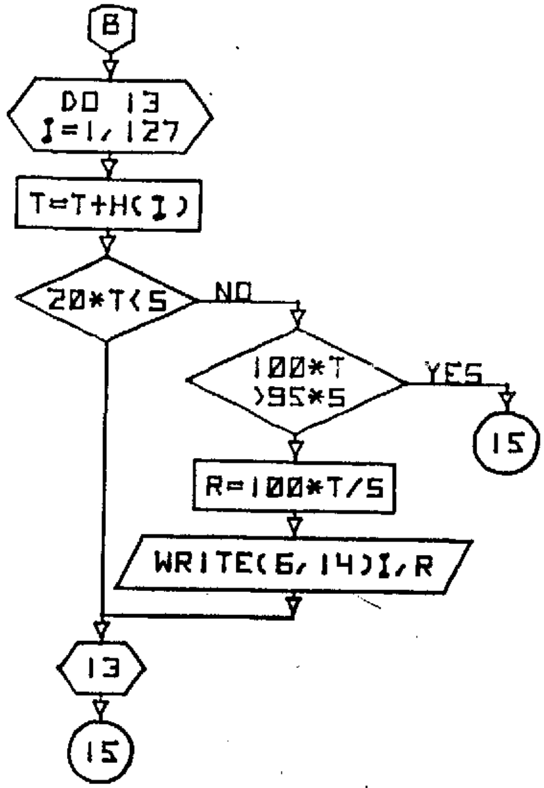
Detailed flow chart

Sample run




Detailed flow chart - PRINSUM - 1


A
DO 6
I=1,127
T=H(I)*126/M
T<1
NO
DO 7
S=1,T
PAGE(S,I)=ST
7
6
WRITE(6,8)
DO 9
A=1,127
I=128-A
WRITE(6,10)I,
(PAGE(V,I),V=1,126)
9
WRITE(6,11)C,M
S=0
DO 12
I=1,127
S=S+H(I)
12
T=0
B






Detail flow chart