

AD-A055 271

SYNECTICS CORP ROME N Y
RASTER PLOTTER SOFTWARE IMPROVEMENTS. (U)

F/G 9/2

MAR 78 V N CAVO

F30602-76-C-0440

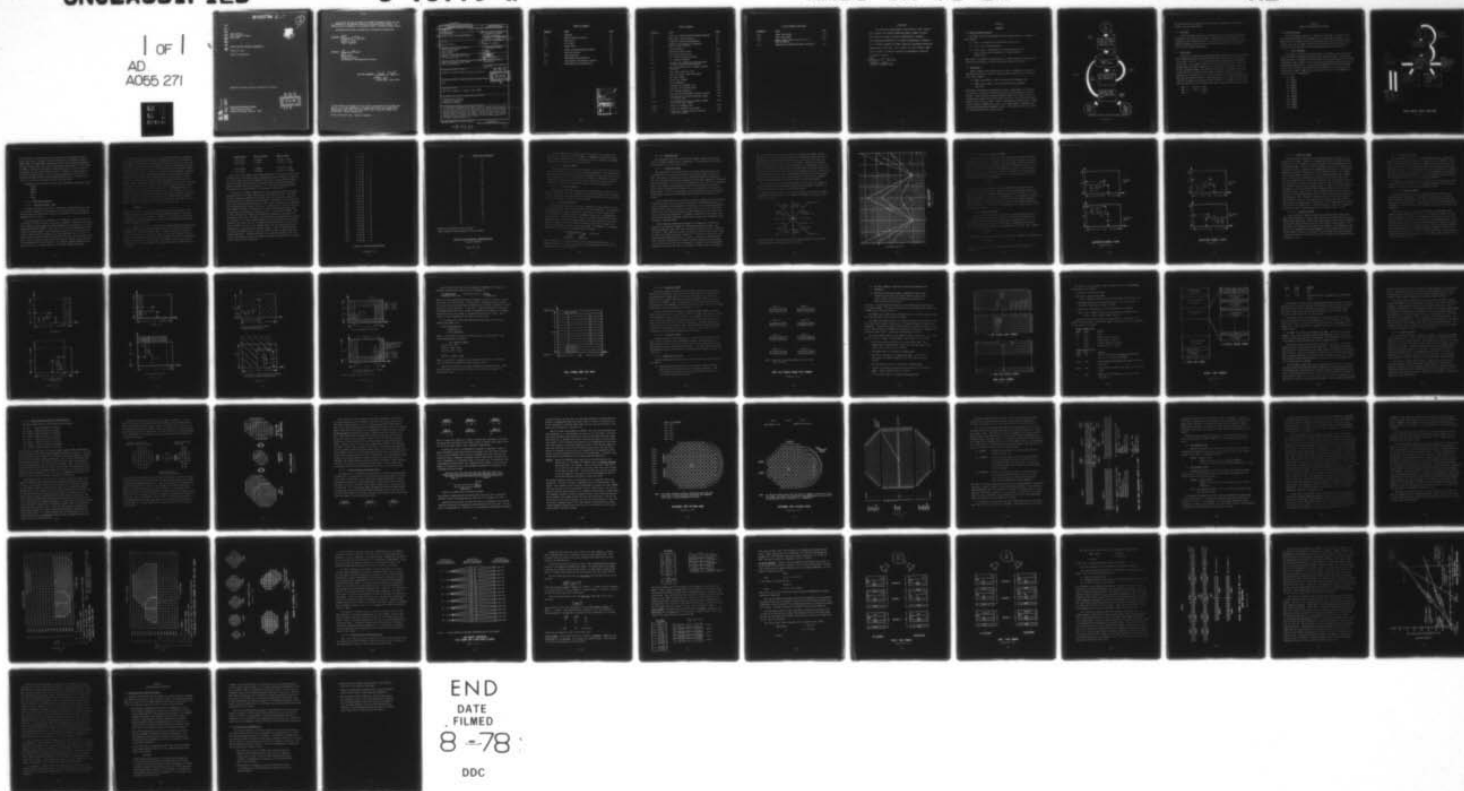
UNCLASSIFIED

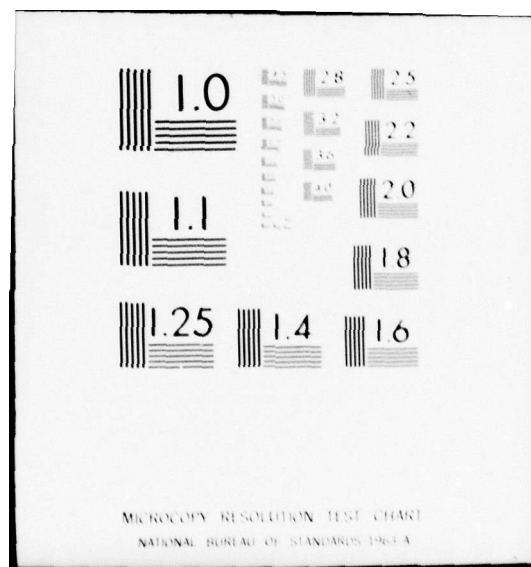
C-T0779-W

RADC-TR-78-64

NL

1 OF 1
AD
A055 271





FOR FURTHER TRAN

(2)
B.S.

AD A 055271

RADC-TR-78-64
Final Technical Report
March 1978



RASTER PLOTTER SOFTWARE IMPROVEMENTS

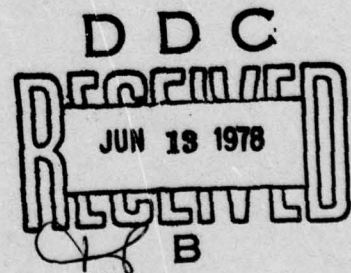
Vincent N. Cavo

Synectics Corporation

Approved for public release; distribution unlimited.

AD No. 1
DDC FILE COPY

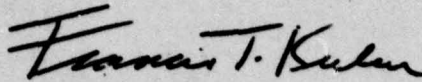
ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441



This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

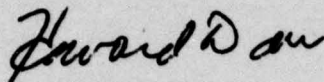
RADC-TR-78-64 has been reviewed and is approved for publication.

APPROVED:



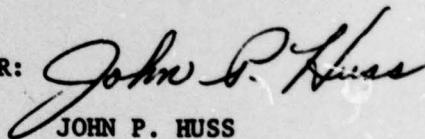
FRANCIS T. KULON
Project Engineer

APPROVED:



HOWARD DAVIS
Technical Director
Intelligence & Reconnaissance Division

FOR THE COMMANDER:



JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (IRRP) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-78-64	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) RASTER PLOTTER SOFTWARE IMPROVEMENTS	5. TYPE OF REPORT & PERIOD COVERED Final Technical Report Sept 76 - Sept 77	6. PERFORMING ORG. REPORT NUMBER CT0779-W
7. AUTHOR(s) Vincent N. Cavo	8. CONTRACT OR GRANT NUMBER(s) F30602-76-C-0440	9. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 64701B 43030327
10. PERFORMING ORGANIZATION NAME AND ADDRESS Synectics Corporation 310 E. Chestnut St Rome NY 13440	11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (IRRP) Griffiss AFB NY 13441	12. REPORT DATE March 1978
13. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Rome Air Development Center (IRRP) Griffiss AFB NY 13441	14. SECURITY CLASS. (of this report) UNCLASSIFIED	15. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) Same		
18. SUPPLEMENTARY NOTES RADC Project Engineer: Francis T. Kulon (IRRP)		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Scientific Data Reduction Lineal-Raster Conversion Automated Cartography		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report describes the software required to generate the drive tapes for the Raster Finishing Plotters being used by the Defense Mapping Agencies production centers. The data is supplied as lineal (vector) feature strings and the software described in this report converts the lineal data to raster formatted data of the proper line weights. The output tape of the conversion routines are used to produce the cartographic color separation negatives on the Raster Finishing Plotters.		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

447633

hc

DDC
RECEIVED
JUN 13 1978
B

Agency's

TABLE OF CONTENTS

<u>SECTION</u>	<u>TITLE</u>	<u>PAGE</u>
1	OVERVIEW	1-1
1.0	General Software Overview	1-1
1.1	Input Phase	1-1
1.2	Sort Phase	1-3
1.3	Output Phase	1-3
2	UNIVAC 1108 PRE-PROCESSING MODULE	2-1
2.1	Functional Design	2-1
2.1.1	Input Processor	2-1
3	CONCLUSIONS AND SUGGESTIONS	3-1
3.0	Incorporated Time Reduction Methods	3-1
3.1	Time Reduction Recommendations	3-2

ACCESSION		
NTIS	State Location	☒
DDC	DDC Number	☐
UNANNOUNCED		☐
JUSTIFICATION		
BY		
DISTRIBUTION/AVAILABILITY		
Dist.	AVAIL.	800/01
A		

LIST OF FIGURES

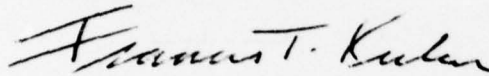
<u>FIGURE NO.</u>	<u>TITLE</u>	<u>PAGE</u>
1-1	UNIVAC-1108 RASTER PLOTTER SOFTWARE OVERVIEW	1-2
2-1	DMAAC/DMATC INPUT PROCESSOR	2-2
2-2	LINEWEIGHT INTEGER REPRESENTATION	2-6
2-3	APERTURE CODE/LINEWEIGHT CONVERSION TABLE (Present Default)	2-7
2-4	POINT REDUCTION	2-11
2-5	REFLECTION ACROSS X-AXIS	2-13
2-6	REFLECTION ACROSS Y-AXIS	2-14
2-7	90° COUNTER-CLOCKWISE ROTATION	2-17
2-8	90° CLOCKWISE ROTATION	2-18
2-9	JOG PLOT (90° COUNTER-CLOCKWISE ROTATION) ON ENTIRE PLOTTER PLOTTING SURFACE	2-19
2-10	PLOT WITH WINDOW AND 90° COUNTER-CLOCKWISE ROTATION	2-20
2-11	RUN Y-RANGE SORT FILE SPLIT	2-22
2-12	SORT FILE PACKED SEVEN BYTE FORMAT	2-24
2-13	MMS BLOCK FORMAT	2-26
2-14	OUTPUT TAPE FORMATS	2-28
2-15	DATA COMPRESSION	2-33
2-16	OCTAGONAL SPOT PATTERN (ODD)	2-37
2-17	OCTAGONAL SPOT PATTERN (EVEN)	2-38
2-18	SPOT PATTERN GENERATION	2-39
2-19	SCAN LINE WORD ARRANGEMENT CENTER X/Y POINT	2-41
2-20	ILLUSTRATION OF SLIDING (ROTATING) WINDOW FOR 16 SCAN LINES	2-45
2-21	ILLUSTRATION OF SLIDING (ROTATING) WINDOW FOR 16 SCAN LINES (LOWER)	2-46
2-22a	ARRAYS OF ELEMENTAL SPOTS	2-46
2-22b	OVERLAP OF ELEMENTAL SPOT ARRAYS	2-47
2-23	LINE-WEIGHT CONVERSION FROM 0.05MM AND 0.1MM FROM 0.025MM	2-49

LIST OF FIGURES (Continued)

<u>FIGURE NO.</u>	<u>TITLE</u>	<u>PAGE</u>
2-24	"EVEN" TAPE FORMAT	2-53
2-25	"ODD" TAPE FORMAT	2-54
2-26	BINARY FORMAT UNIVAC-1108 TO PDP 11/40 WORD CONVERSION	2-56
2-27	RELATIVE DATA COMPACTION METHOD COMPARISON	2-58

EVALUATION

The software improvements described in this report will enable the Defense Mapping Agency (DMA) to more efficiently produce various cartographic products. The automated generation of these cartographic products is in direct support of RADC technical program objectives described in TPO R2D. As a result of this effort, DMA will be able to supply required cartographic products to the Air Force.



FRANCIS T. KULON
Project Engineer/IRRP

SECTION 1

OVERVIEW

1.0 General Software Overview

The Raster Plotter Linear/Raster pre-processing software is composed of three phases:

- (1) Input - interpolation of lineal data center points
- (2) Sort - by Y,X and line weight
- (3) Output Phase - construction of octagonal patterns of 25 micron spots on each center-point to produce desired weight lines

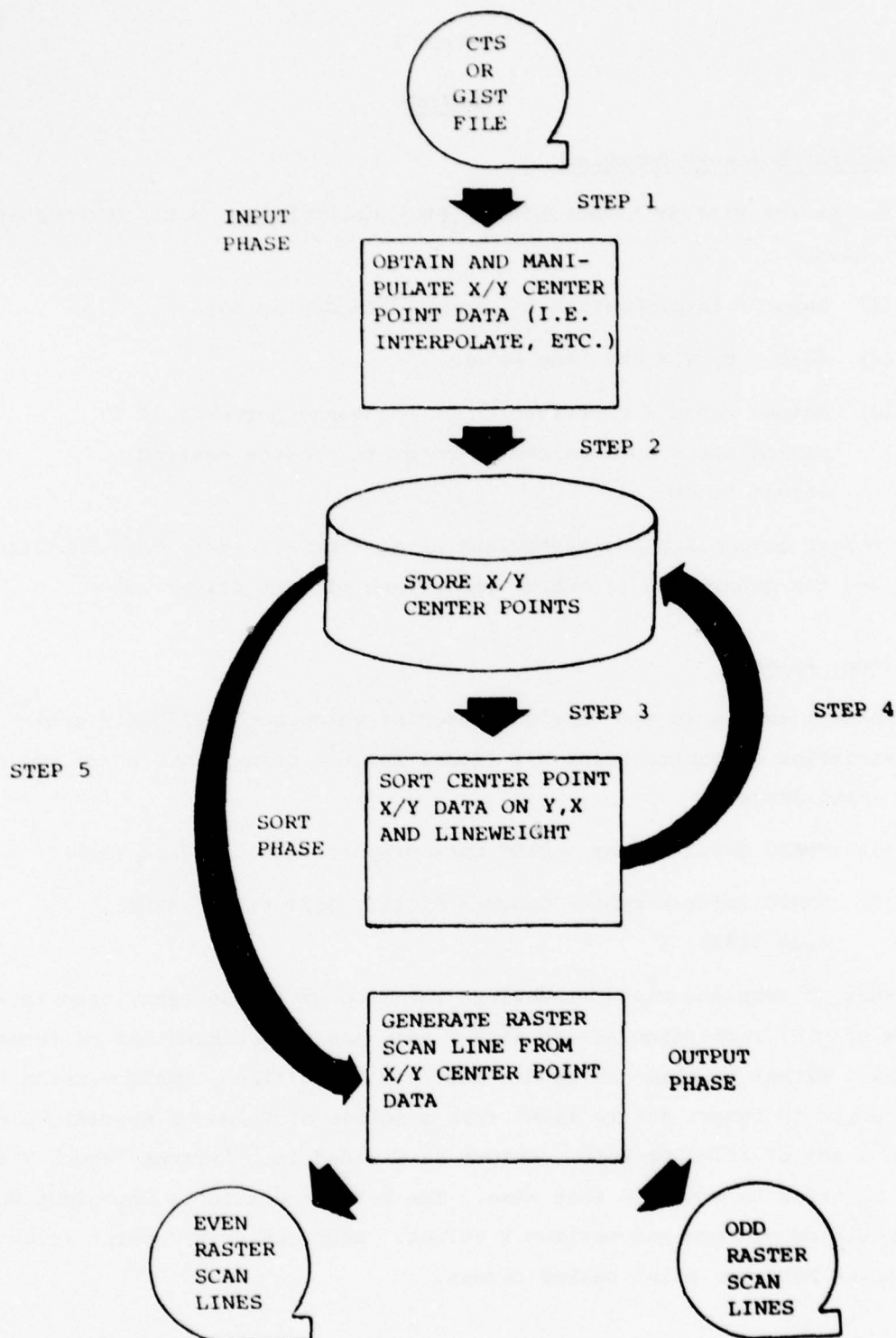
These phases accomplish the rasterization of lineal feature data submitted as input and the generation of output rasterized plotter driver tapes.

1.1 Input Phase

Phase I exists in two versions, each of which accomplishes a search for and extraction of center point X/Y lineal feature points for one of two possible input formats:

- (1) DMAAC Input Format - RADC Cartographic Test Standard (MMS)
- (2) DMATC Input Format - Concord Plotter GIST File (SHEREZ)
Code 51410

Both Phase I versions allow windowing, rotation of +90 degrees, translation to new origin, reflection across either axis and any combination of those options. Either version can handle multiple input files, DMATC version can be directed to ignore entire files from a series of files or specific blocks within a set of files or both. Output is divided into fifteen "equal Y range" files in order to minimize sort time. The Y range domain is dependent upon the specified minimum and maximum Y values. Each center X/Y point is output in a seven byte per point packed format.



UNIVAC-1108 RASTER PLOTTER SOFTWARE OVERVIEW

Figure No. 1-1

1-2

The Input Phase may be executed at any of three "resolutions" to trade-off accuracy for processing speed.

1.2 Sort Phase

A Univac supplied Cobol Sort is utilized to sort the fifteen data files generated by the Input Processor Phase. The data files are sorted on Y and increasing X and linewidth within Y. The sorted X/Y points and their corresponding linewidths are placed back into their original files in the seven byte per point packed format again.

1.3 Output Phase

The sorted X/Y points are unpacked dependent upon their scan line Y address. Then octagonal patterns are generated around all centerline points residing on the same Y scan line. Once a scan line is out of range of an incoming X/Y center point, (i.e. sixteen, eight or four scan lines away for 25, 50 and 100 microns output resolution respectively), the scan line is output to tape in one of two output formats, Run-length code or Binary. Run-length code format is more efficient in terms of tape usage and processing time up to spatial densities of 125 transitions/inch (5 transitions/mm) average, and binary form more efficient above that level. Phase III may be run at three resolutions to trade-off accuracy for processing speed if desired. For example, the CTS test data was processed in the following SUPs times at their respective resolutions.

RES = 1	4 hours	13 mins
RES = 2	1 hour	15 mins
RES = 4		16 mins

SECTION 2
UNIVAC 1108 PRE-PROCESSING MODULE

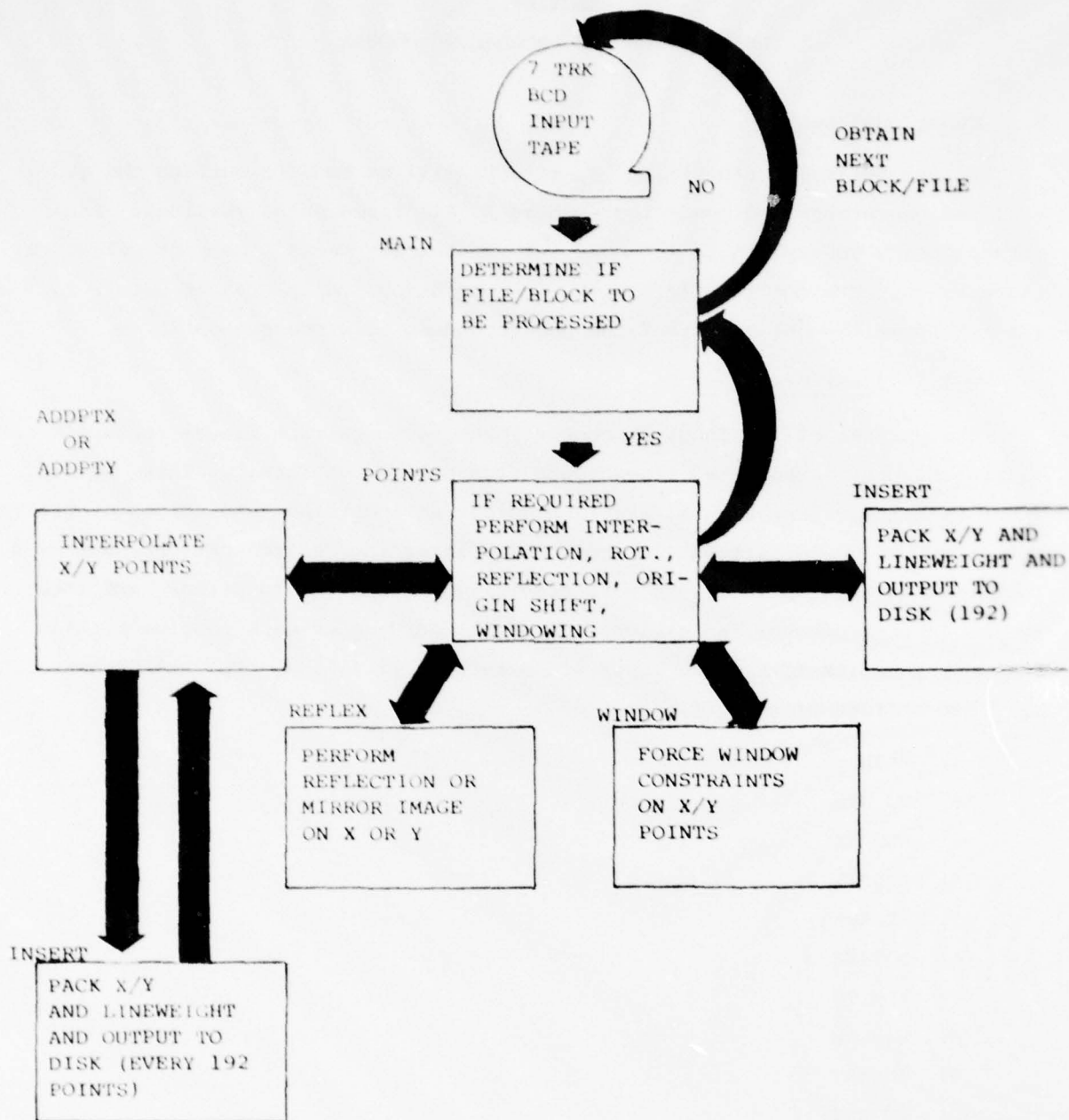
2.1 Functional Design

In the following discussion, an attempt will be made to surface the general design methods and logic implemented in all three phase versions. The purpose being to explain to the user the reasons why certain methods were used. Concepts that are not readily obvious will be discussed in greater detail so that the user may obtain a feel and understanding of processes going on.

2.1.1 Input Processor

The purpose of the Input Processor Phase is to extract lineal feature X/Y points and their associated lineweights from some input format. These points, along with their lineweights, are then output at specified resolutions of 0.025mm, 0.05mm and 0.1mm to fifteen (15) sort files after the desired options have been performed on these points. To more precisely explain the functions performed by the Input Processor, an explanation of the purpose of each routine within both versions (DMATC & DMAAC Input Processors) will follow. The DMAAC Phase I version consists of subroutines:

- (1) MAIN
- (2) POINTS
- (3) ADDPTX
- (4) ADDPTY
- (5) REFLEX
- (6) ROTATE
- (7) LIMITS
- (8) WINDOW
- (9) INSERT
- (10) PUTENT
- (11) NTRANS
- (12) MBYTES



DMAAC/DMATC INPUT PROCESSOR

Figure No. 2-1

All are Fortran routines except the last two, which are Assembler routines supplied by Univac. Because of slight differences in the NTRAN processor at DMATC, DMAAC, and SUNY/Albany (where most of the initial software development was done) and the need to be able to access more than the usual "10 file limit" of NTRAN, a "private" frozen version which allows up to 20 files is kept in the program file and used by Phases I and III. MBYTES is a short utility 6-bit-byte pack/unpack routine, which is considerably more efficient than an equivalent Fortran method of packing/unpacking buffer areas. Details of the functions of the other subroutines follow this subsection.

The DMATC version of Phase I requires five additional subroutines, namely

ASCBCD

NEWFIL

NWFEAT

EOF

BUILD

2.1.1.1 DMATC Input Processor

2.1.1.1.1 Mainline Program - Main

The mainline program will obtain a BCD (Binary Coded Decimal) input physical record, consisting of 1020 words with each BCD character occupying one word and right-justified within the word, and interpret the keyword commands embeded within the record.

The main purpose of this program is to 'recognize' and extract valid lineal feature X/Y points and their associated lineweights. These valid X/Y points occur immediately after each ON keyword command and are terminated by the next immediate OF keyword command. Also valid is the X/Y point immediately preceding the ON keyword command, which designates starting point of lineal feature string. Each X/Y point is defined by ten BCD characters, the first five of which represent the X coordinate value and the next five the Y coordinate value. Both X and Y coordinate values are absolute positive integer values representing thousandths of an inch increments. Any keyword commands embedded within these X/Y points is ignored (namely S the steering command designator).

Associated with each X/Y points is an aperture code which is designated by the last encountered aperture keyword command (BCD A). This aperture keyword command is immediately followed by two BCD digit characters, in the range 1 through 24, which represents the linewidth code to be associated with all following X/Y points up to the next encountered aperture code keyword command. This aperture code acts as a pointer to an Aperture Code/Linewidth conversion table such that an association between aperture code and linewidth value is established. The Aperture Code/Linewidth Conversion Table may be altered such that at anytime the aperture code may represent any of the valid linewidths 0.1mm - 0.8mm. For example, assume an aperture code of 02 is specified (designates contents of second element of Conversion Table), then this code could represent any of the valid linewidths 0.1 - 0.8mm depending upon the contents of the second element of the conversion table. That is, if on one job all X/Y points associated with aperture code 02 were 0.25mm thick (second element of conversion table contents was ten (10) designating ten elemental spots (0.025mm), the same X/Y points could be plotted at say 0.5mm if upon re-run of job the second elements of conversion table is set to:

$$\frac{0.5\text{mm}}{0.025\text{mm}} = 20$$

Should the user alter the Aperture Code/Linewidth conversion table, a re-compile and re-map to generate an Absolute executable element is necessary (see Appendix of Operator's Manual). Elements of the table altered by input via Namelist Statement assume their previous preset values upon each new execution of DMATC Input Phase.

It is important to note that the Aperture Code/Linewidth Conversion Table elements contain positive integer values in the range 4-32 which represents the number of successive elemental spots, 0.025mm in diameter, required to generate the necessary linewidths. These values do not represent the linewidths in thousandths of an inch units but instead the possible closest match within constraint of unit spot 0.025mm. For example, the closest the user could obtain to a four thousandths inch linewidth would be 0.1mm arrived at as follows:

<u>English Units</u>	<u>Metric (Actual)</u>	<u>Metric (Used)</u>
0.004 inches	0.1016mm	4*0.025 = 0.1mm
0.005 inches	0.127mm	5*0.025 = 0.125mm
⋮	⋮	⋮
0.032 inches	0.8128mm	32*0.025 = 0.8mm
0.033 inches	0.8382mm	33*0.025 = 0.825mm

Hence as in above example, the closest match for 0.004 inches would be four elemental spots. The whole integer representation may be obtained by dividing the actual metric value by the elemental spot size. Positive integer values outside the range 4-32 should not be specified within the Aperture Code/Line-weight Conversion Table (see Figure No. 2-2). The present default Aperture Code/Line-weight Conversion Table is as portrayed in Figure No. 2-3.

The only other keyword command which is not ignored and which influences processing flow is the block identification code denoted by 'BK'. The three BCD digit characters immediately following this code designates the present logical block that is presently being analyzed. In addition, if the three BCD digit characters following this code is '999', then an end-of-file is indicated. Two successive BK999's denote end-of-job. In light of this, each time the keyword command code 'BK' is encountered, the block identification number must be generated to determine the necessary action. If end-of-file (BK999), then the file count must be updated and variables initialized in preparation for processing of new file. Otherwise, the Files/Block Skip table must be searched to determine if this block within present file is to be processed or bypassed. Likewise, each time a new file is obtained, as denoted by first logical block after a BK999, the file is also checked against the Files/Block Skip Table to determine if to be processed or ignored. Again, it must be emphasized that two successive BK999's denote end-of-job and will result in Input Phase processing termination. Upon each encounter of a block identification code, the current file number, logical block number and physical record within this block will be output to the lineprinter.

(1)	0.025 mm	→	1
(2)	0.050 mm	→	2
(3)	0.025 mm	→	3
(4)	0.100 mm	→	4
(5)	0.125 mm	→	5
(6)	0.150 mm	→	6
(7)	0.175 mm	→	7
(8)	0.200 mm	→	8
(9)	0.225 mm	→	9
(10)	0.250 mm	→	10
(11)	0.275 mm	→	11
(12)	0.300 mm	→	12
(13)	0.325 mm	→	13
(14)	0.350 mm	→	14
(15)	0.375 mm	→	15
(16)	0.400 mm	→	16
(17)	0.425 mm	→	17
(18)	0.450 mm	→	18
(19)	0.475 mm	→	19
(20)	0.500 mm	→	20
(21)	0.525 mm	→	21
(22)	0.550 mm	→	22
(23)	0.575 mm	→	23
(24)	0.600 mm	→	24
(25)	0.625 mm	→	25
(26)	0.650 mm	→	26
(27)	0.675 mm	→	27
(28)	0.700 mm	→	28
(29)	0.725 mm	→	29
(30)	0.750 mm	→	30
(31)	0.775 mm	→	31
(32)	0.800 mm	→	32

LINEWEIGHT INTEGER REPRESENTATION

Figure No. 2-2

<u>CODE*</u>	<u>LINEWEIGHT CONNOTATION</u>
01	4
02	27
03	3
04	16
05	16
06	6
07	8
08	9
09	10
10	10
11	7
12	12
13	32
14	14
15	16
16	22
17	27
18	32
19	8
20	12
21	24
22	6
23	7
24	9

*Aperture code denotes index to buffer

APERA (Aperture Code/Lineweight Conversion Table

APERTURE CODE/LINEWEIGHT CONVERSION TABLE

(Present Default)

Figure No. 2-3

At job termination, accounting of all points within the fifteen sort data files is given along with a 'NORMAL' or 'ABNORMAL' termination message. The abnormal termination will be proceeded by error or diagnostic statements designating the abnormal condition encountered.

2.1.1.1.2 Subroutine NWFEAT

This subroutine will locate a new lineal X/Y point feature string when the successive invalid X/Y point skip count has been exhausted. This search will begin in present record and continue into successive records but will not cross block boundaries. That is, the search will terminate if BK block is located. In addition, should an aperture code be out of range (greater than 24) this routine will search for a new aperture code to the end of current file.

2.1.1.1.3 Subroutine NEWFIL

This routine is utilized to position the input tape to the next file or block. Thus resulting in the capability to skip over unwanted files or blocks. Also, if applicable, any error condition found before research for a new file or block is initiated is output to the line printer.

2.1.1.1.4 Subroutine BUILD

This subroutine generates the X and Y component X/Y point integer values of a point from the input 5 BCD characters (for each component). The generated 5 digit absolute positive integer values are placed in X and Y respectively. It is important to remember that these component values are in thousandths of an inch units. Upon exit from this routine, the values are converted to Metric millimeter units. The Metric component values are then converted to the whole number of elemental spots (0.025mm) displacements required to represent the point, by the division of Metric value by 0.025mm. For example, if X component was 5000 (5 inches) then the following would transpire:

$$\begin{array}{rcccl} 5000 & \text{MILS} & * & 0.0254\text{mm} & * & \frac{1}{0.025\text{mm}} & = & 5080 & \text{spots} \\ & \text{1 mil} & & & & \text{Spot} & & & \\ & \text{Metric} & & & & & & & \\ & \text{Conversion} & & & & \text{Conversion} & & & \end{array}$$

Each elemental spot is represented by one bit within the Output Phase. If a character other than a digit is found in X or Y positions an error flag is set and passed back to calling routine.

2.1.1.1.5 Subroutine EOF

This subroutine tests for an end-of-file (BK999) whenever a block identification keyword command code is encountered. If an end-of-file is located the flag IEOF is set to specify as such.

2.1.1.1.6 Subroutine POINTS

This subroutine is the focal point of the entire Input Processor since it performs all of the necessary functions on the input lineal feature X/Y data. The routine is entered each time a new lineal feature is encountered (ON) and exits whenever the next encountered valid keyword command is 'recognized'. Each lineal feature's X/Y points are 'generated' here by a call to the subroutine BUILD. The routine always processes lineal feature successive X/Y points in pairs such that determination of whether or not interpolation is necessary is made possible. Prior to interpolation, the user specified options of Rotation, Mirror Image, Origin Shift and Windowing are performed on the pairs of X/Y points.

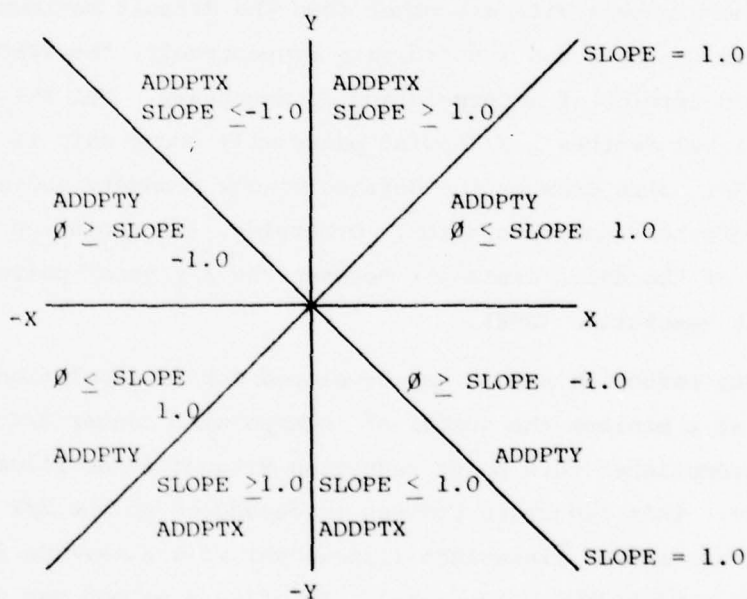
If the window limits are other than the default maximum size of 50800 spots, 71200 spots in the Y and X coordinate respectively, the WINDOW subroutine is called to determine if interpolation is necessary. For this case interpolation between lineal feature's X/Y point pairs will occur only if the line defined by the point pairs crosses the defined window boundary and either delta distance is greater than resolution. Otherwise, interpolation will always occur if either of the delta distances between the X/Y point pairs exceeds the specified point resolution (RES).

A data reduction method was developed for interpolation to reduce or maintain at a minimum the number of interpolated center X/Y points. The method accomplishes this point reduction without a sacrifice in line integrity or quality. This reduction process is dependent on the X/Y point pairs' slope and their associated linewidth (linewidth will always be identical for all points entering POINTS Subroutine). Briefly, a method was developed using table look-ups to determine the maximum long-axis center-to-center spacings of the octagonal array patterns permissible at any angle, for the linewidth desired, to insure no more than a one element (.025mm) change on the shorter

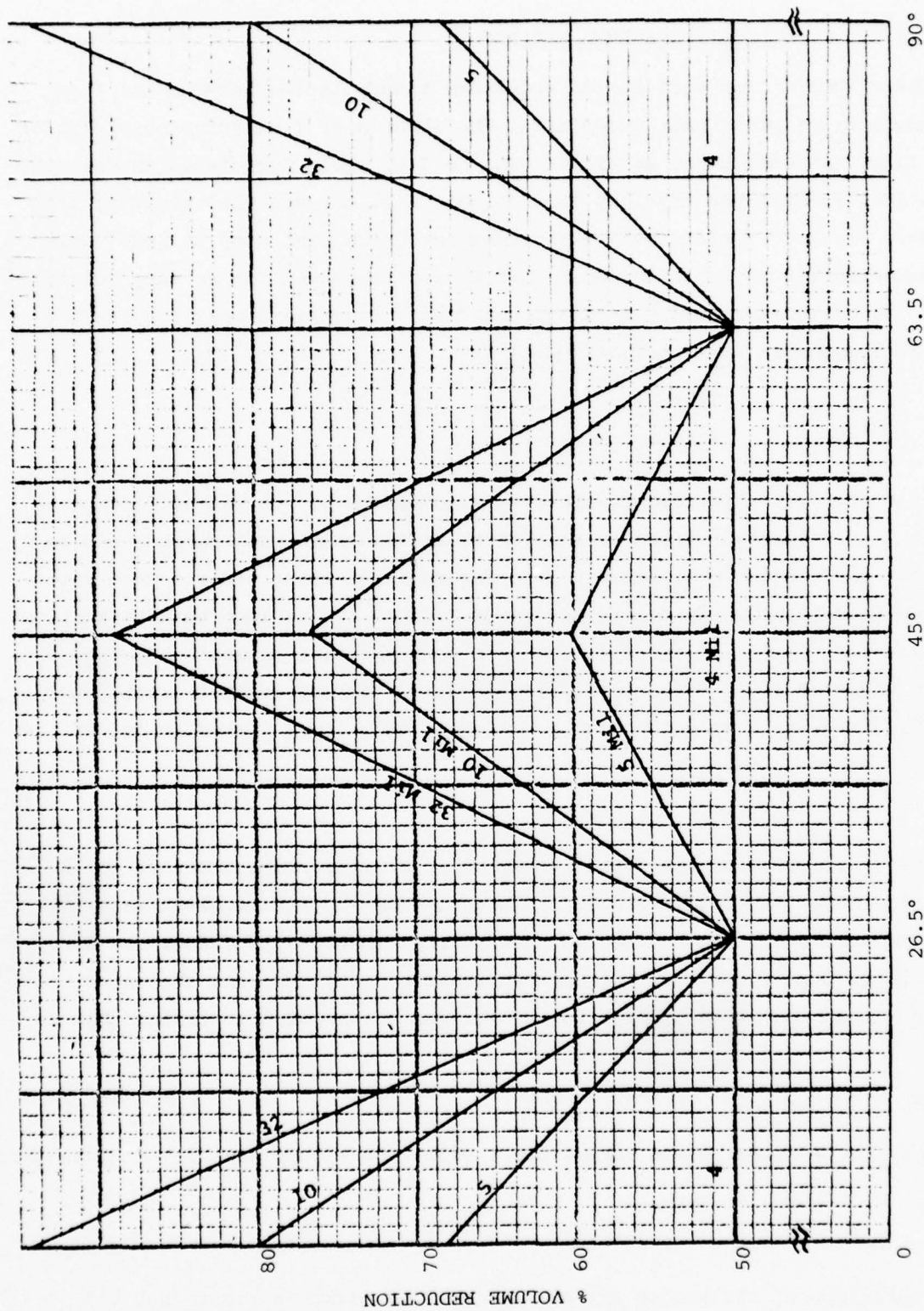
axis, as if the patterns had been placed at a constant one element spacing. This results in a savings of data volume to be handled by the Phase II (Sort) and Phase III (Output Process) operations of at least 50% in all cases and approaching 93% on 32 mil lines at 0 to 90 degrees. Figure No. 2-4 approximately illustrates the volume savings gained for a few linewidths as a function of direction angle. Note that the minimum (50%) level always occurs for a 4-mil line regardless of angle, and for all linewidths at 26.56° (Arc Tan 0.5). It is expected that a "typical" chart product with most feature lines falling in the 8 to 15 mil range at an even distribution of angles of line segments would average approximately 65% volume savings over the former method.

As a result of the point reduction method, a reduction of the number of center X/Y points being interpolated resulted and in addition a reduction in CPU processing time for Input, Sort and Output Phases.

Which of two subroutines is called to perform the interpolation (ADDPTX or ADDPTY) will depend upon the line segment's (defined by X/Y point pairs) slope. If the slope



value lies within a particular range on the above diagram, that subroutine will perform all interpolation for the X/Y pair.



ANGLE FROM HORIZONTAL
POINT REDUCTION

Figure No. 2-4

2.1.1.1.7 Subroutine ADDPTX and ADDPTY

These subroutines perform the necessary interpolation between the input line segment X/Y point pair endpoints. The input X or Y starting point (X1 or Y1) will be incremented by an element in the Step Table, generated in Points subroutine, and via the straight line formula will compute the resultant Y or X value. The X or Y will then be incremented by the next step in Step Table and the resultant X or Y obtained. This will be repeated for as many points as need to be interpolated. Upon encounter of a negative slope, the two X/Y endpoint components are interchanged.

$$X_1 \leftrightarrow X_2$$

$$Y_1 \leftrightarrow Y_2$$

such that Step Table increments are always added (never subtracted) from current X or Y thus resulting in ability to utilize same program code and in addition, maintaining a somewhat sorted interpolated point order. Once interpolation is complete, the points are swapped back. It is important to mention that interpolation of points will not go beyond the line segment endpoints. Once an endpoint is reached, interpolation will stop and control passed back to calling routine. As each point is interpolated, it is placed in one of the appropriate sort files.

2.1.1.1.8 Subroutine REFLEX

This subroutine accomplishes a reflection or mirror image across the X or Y axis of the particular area of interest. The routine is given an X/Y point and passes back the reflected point mirror imaged across X or Y. A mirror image across the X axis is accomplished by complementing the input point's Y component with respect to the maximum Y axis plotter limit. Namely,

$$Y = 50800 - Y$$

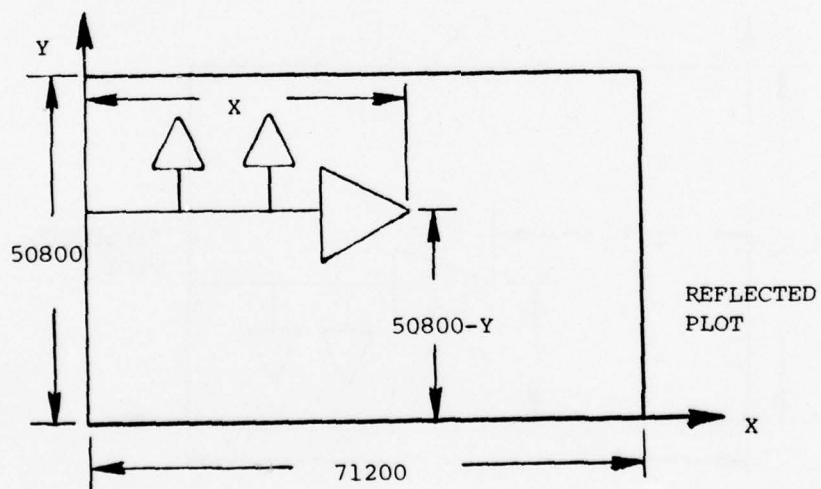
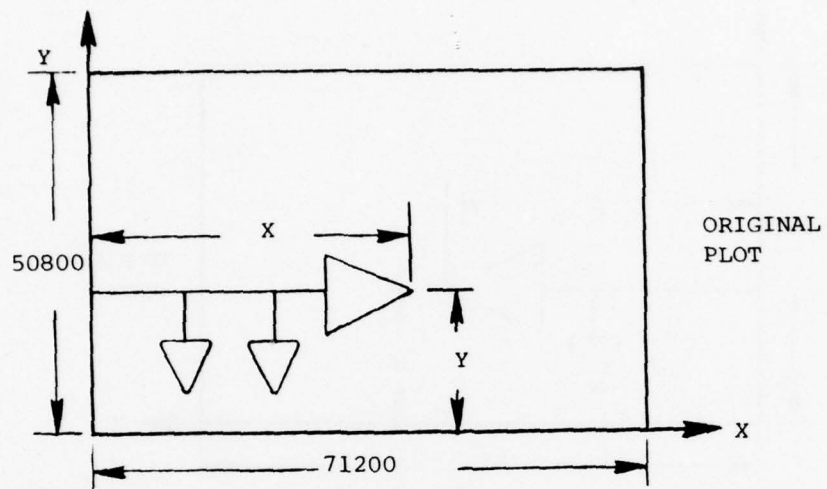
$$X = X$$

Similarly for a mirror image across the Y axis, the X component is complemented.

$$X = 71200 - X$$

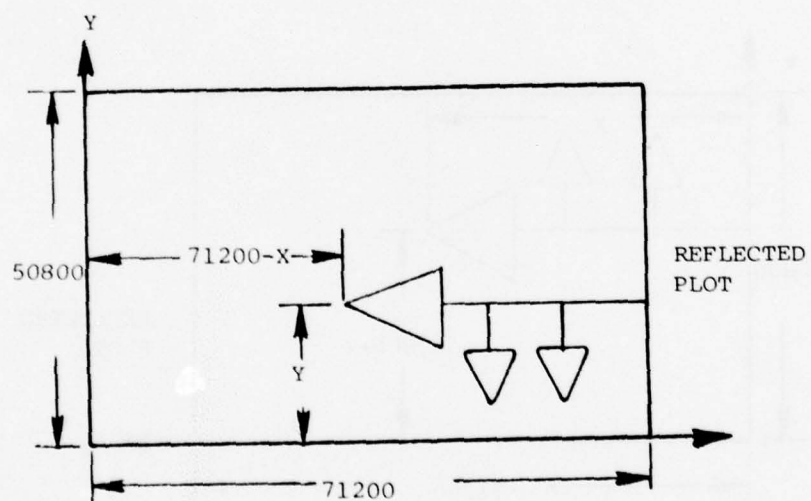
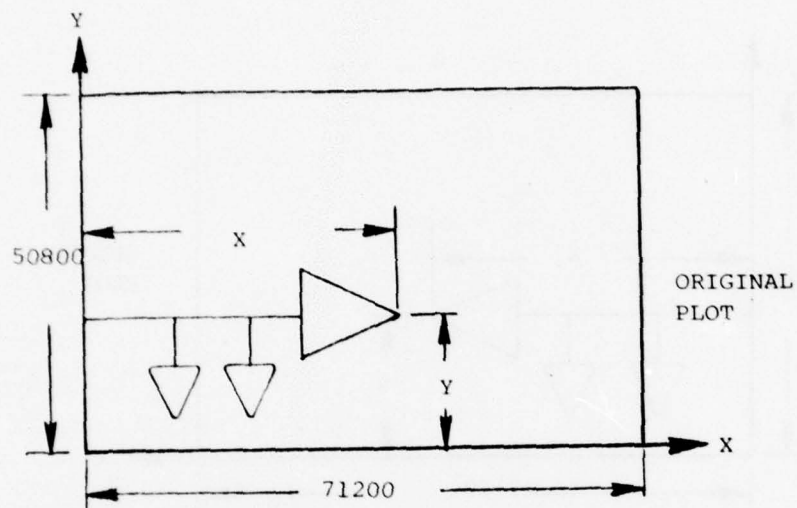
$$Y = Y$$

An illustration of reflection across both axis is given in Figure No. 2-5 and 2-6.



REFLECTION ACROSS X-AXIS

Figure No. 2-5



REFLECTION ACROSS Y-AXIS

Figure No. 2-6

2.1.1.1.9 Subroutine ROTATE

This subroutine accomplishes a $\pm 90^\circ$ rotation about the origin of a particular plot or image. Rotation is accomplished by first performing a translation on the X or Y component of the input point and then swapping the X and Y coordinate values. For a 90° counter-clockwise rotation the Y component of the input X/Y point is translated or complemented with respect to the plotter's maximum Y-axis limit (along plotter carriage 50800) and then X and Y components are interchanged. Likewise, for a 90° clockwise rotation the X component of input point is translated with respect to the plotter X-axis limit (plotter circumference of 71200) before components are interchanged. Rotation like reflection, can only be performed relative to the entire plotter chart size (178cm X 127cm) of 71200 elemental spots (0.025mm) on the X-axis which is circumference of plotter drum and 50800 elemental spots on Y-axis or along carriage of drum. Charts smaller than the maximum size will be oriented as required on the maximum plotting surface in case of rotation and/or reflection (see Figure No. 2-9). Otherwise, the plotter origin will coincide with the specified plot's origin. It is important to note that upon $+90^\circ$ rotation, that any X/Y lineal data beyond X-axis limit of 50800 (approximately 50 inches) will be lost (see shaded areas in following figures). Upon -90° (clockwise) rotation, all data in approximately first 20 inches of chart is lost.

2.1.1.1.10 Subroutine LIMITS

This routine will adjust the specified or default window limits XMIN, XMAX, YMIN and YMAX if a rotation and/or reflection is specified by the user. Window limits specified, if other than default, must not take into consideration any rotation and/or reflection the user will request of the software. This adjustment will automatically be made by the software. The adjustment is necessary since windowing is performed after the lineal feature points have been rotated and/or reflected. The order of lineal feature X/Y point processing requires that rotation and/or reflection and interpolation be performed before windowing of data.

2.1.1.1.11 Subroutine Window

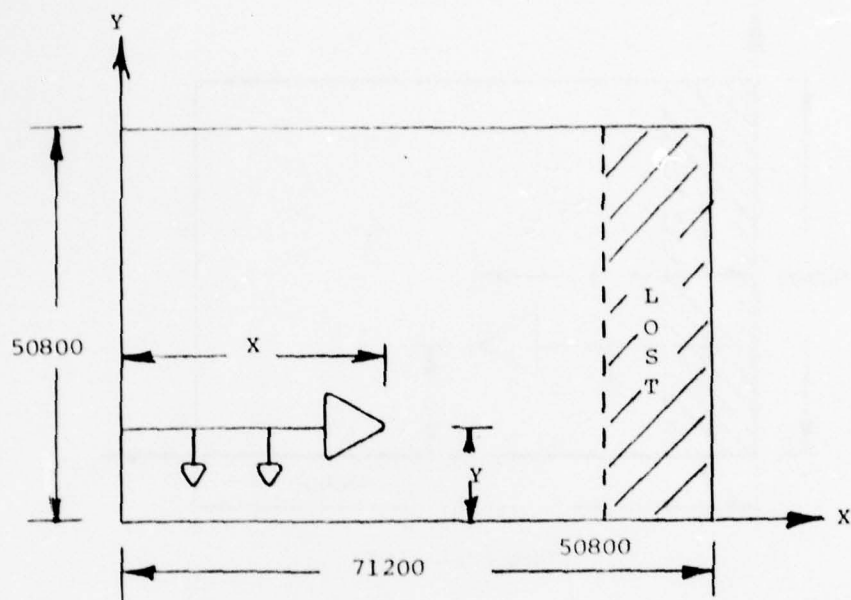
This subroutine is utilized to determine whether or not interpolation between input line segment endpoints is required. A flag (INTFLG) is set to one (1) if interpolation is necessary, otherwise flag set to zero (0). The routine is entered only if window limits other than the default values are specified by the user. To determine whether or not interpolation is required the routine analyzes the line defined by the input endpoints for intersection with the specified window boundaries. If line intersects window boundaries, then interpolation flag is set (=1) otherwise cleared (0).

In addition, if the subroutine is called with the dispatch flag cleared (IPATH=0) the routine will test first endpoint conformity to window limits. Following are Figures illustrating the windowing option.

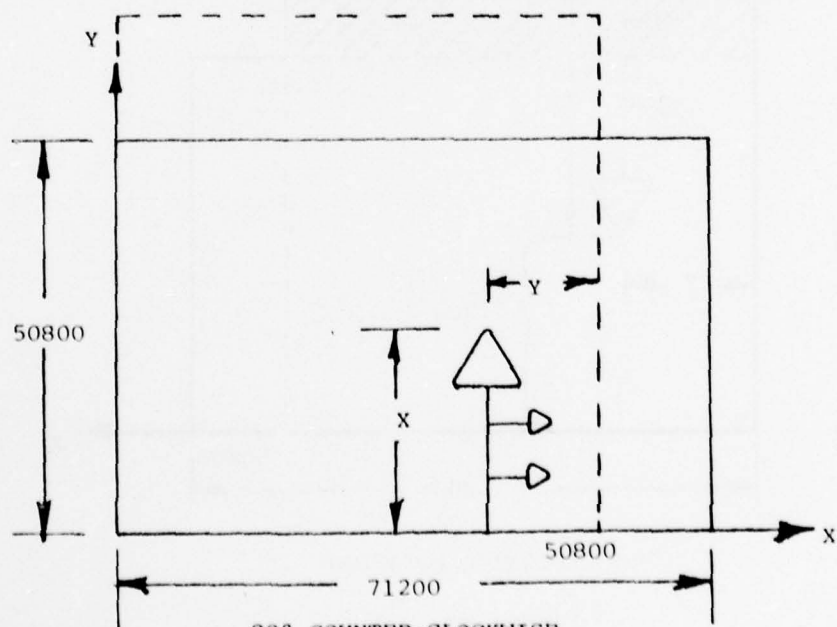
2.1.1.1.12 Subroutine INSERT

This routine splits the run's Y coordinate range into fifteen (15) separate bands of equal width. The purpose of this split is to create fifteen separate and independent sort data files which may be sorted faster than one very large sort file. Once these data files are sorted they may be implemented in the Output Phase in the order 1-15 sort file (see Univac 1108 Sort/Merge).

When the X/Y point is passed to this routine, it determines which Y range or band it belongs in. By a call to the subroutine PUTENT, the point and its linewidth is stored in the appropriate section for that file in a temporary buffer. This file's point count is also incremented by one. When the point count for any of the fifteen files reaches 192, meaning one whole mass storage sector can be filled (224 words = 192 X/Y point and linewidth), the appropriate section of the Temporary buffer (LARGE) is written to its associate sort file (one of fifteen). The 224 words limit is used since the system ideally buffers mass storage I/O at 224 words, which equates to eight sectors (28 words per sector).



ORIGINAL PLOT



90° COUNTER-CLOCKWISE
ROTATION

Figure No. 2-7

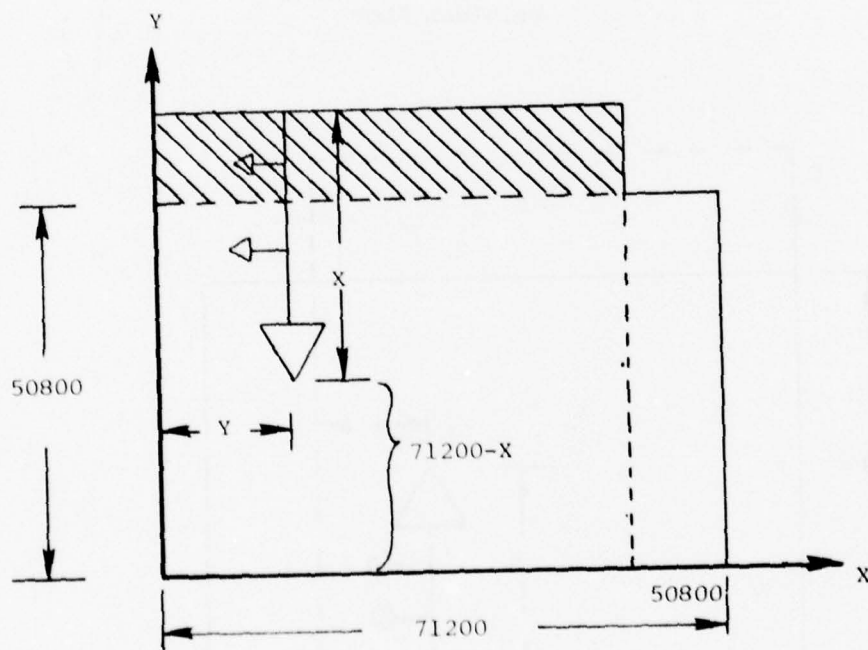
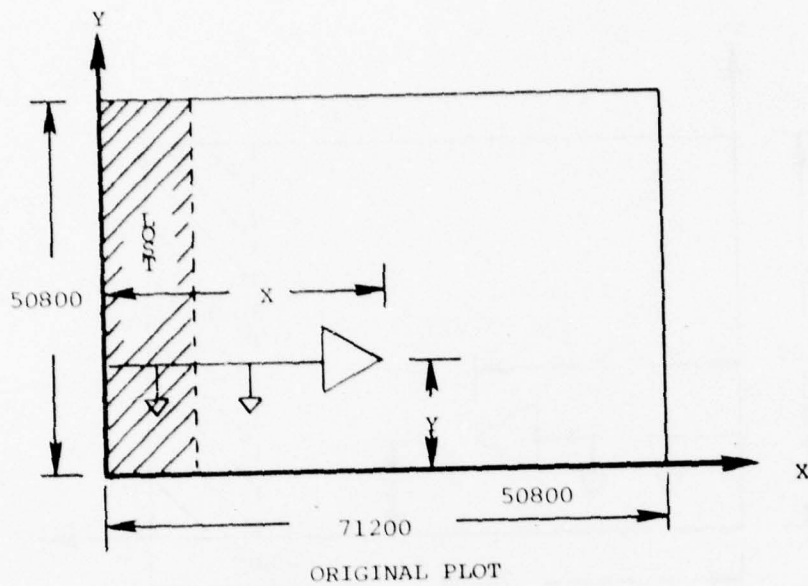
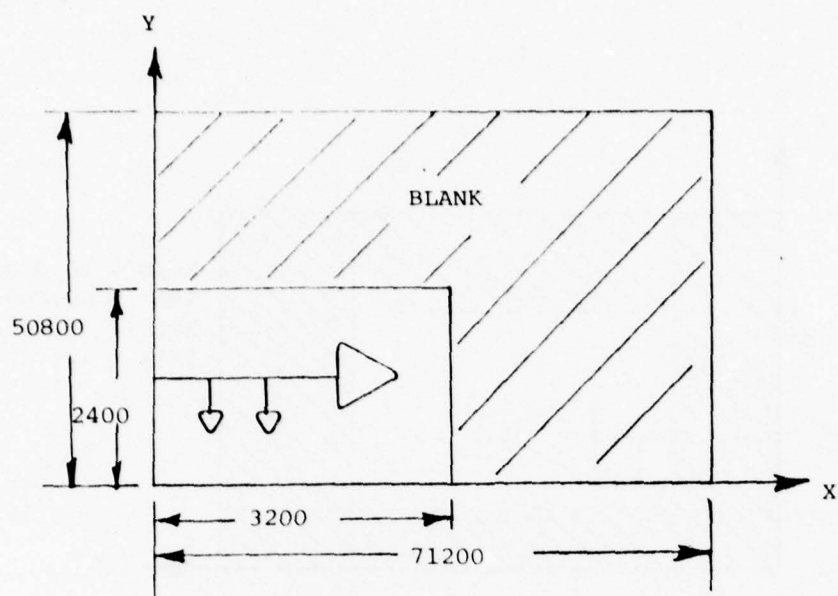
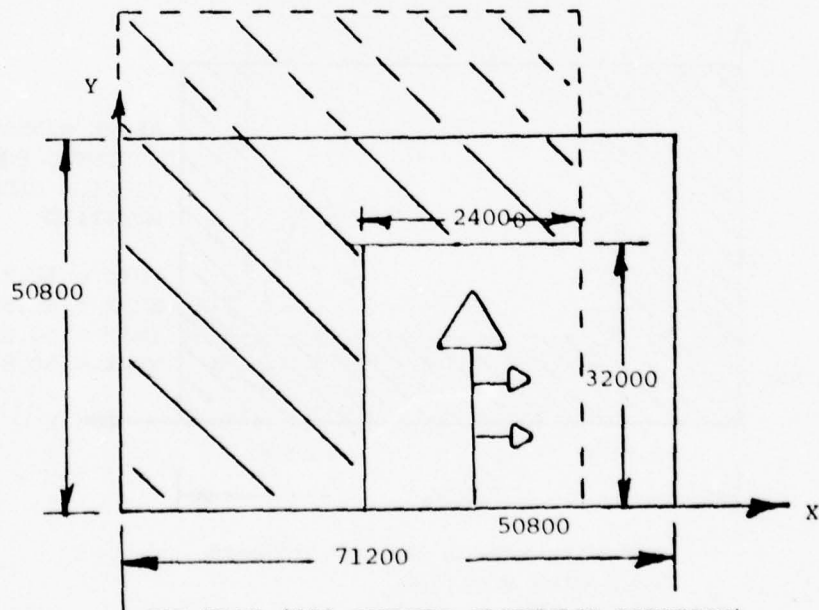


Figure No. 2-8

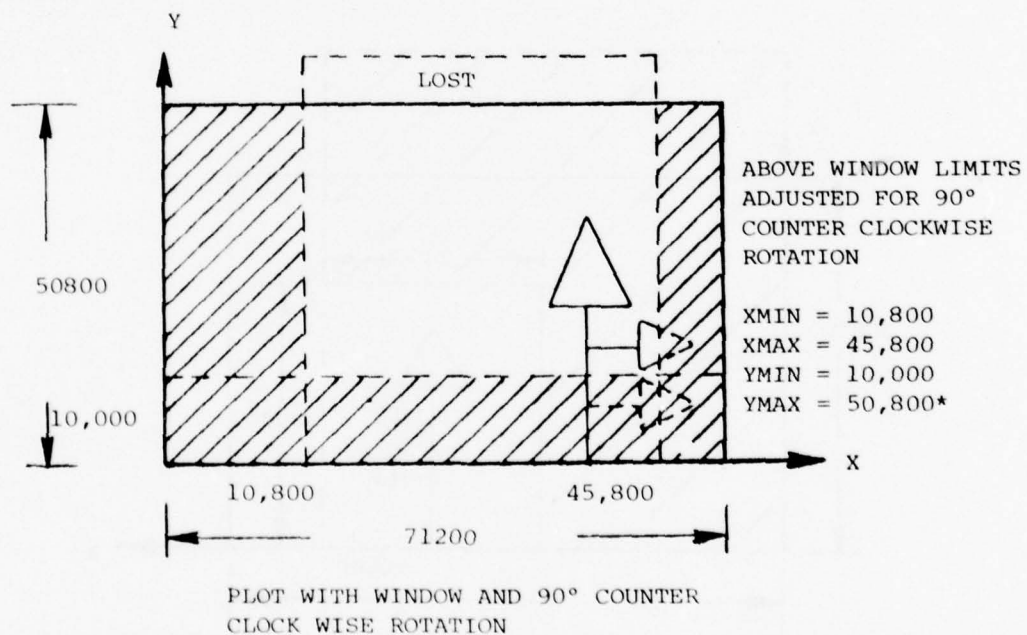
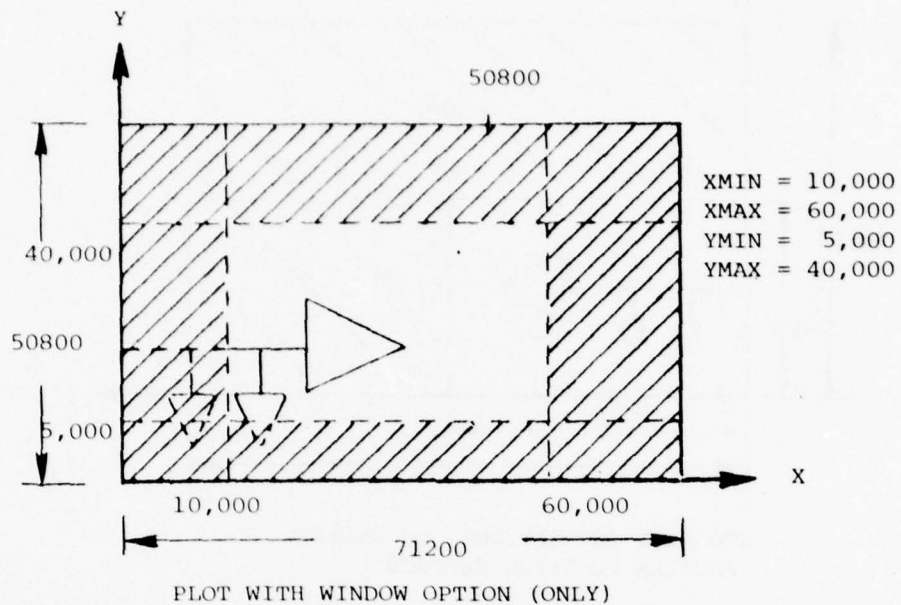


JOG PLOT (NO OPTIONS) ON ENTIRE
PLOTTER PLOTTING SURFACE



JOG PLOT (90° COUNTER-CLOCKWISE ROTATION)
ON ENTIRE PLOTTER PLOTTING SURFACE

Figure No. 2-9



* = LIMIT EXCEEDED 50800 SO SET TO 50800

Figure No. 2-10

Each of the fifteen sort files is assigned a maximum of 500 tracks. As presently assigned each track will hold a maximum of:

$$\frac{1792 \text{ WORDS/TRACK}}{224(192 \text{ pts}-8 \text{ sectors})} \quad *192 \text{ pts} * 500 \text{ tracks} = \frac{768,000}{\text{points/file}}$$

Hence, the total point and associated lineweight content of the entire fifteen sort files is $11.52 * 10^6$ points. The Sort Phase core memory and mass storage files space allocations have been optimumly set up to handle a maximum of $768 * 10^3$ points and their lineweights. If X/Y point Y range distribution is such that the maximum track size for one or more Y bands must be increased to accommodate additional points, an increase in optimum sorting time will result. The increase in time is dependent on the total file point count.

As an example of Y range band computation assume the run Y limits are YMIN=0, YMAX=50800, then

$$\begin{aligned} \text{ISPLIT} &= (\text{YMAX}-\text{YMIN}) / 15 + 1 \\ &= (50800 - 0) / 15 + 1 \\ &= 3387 \text{ Y-UNITS/FILE} \end{aligned}$$

That is, the fifteen sort files will each contain a Y value range of 3387 values maximum split as follows:

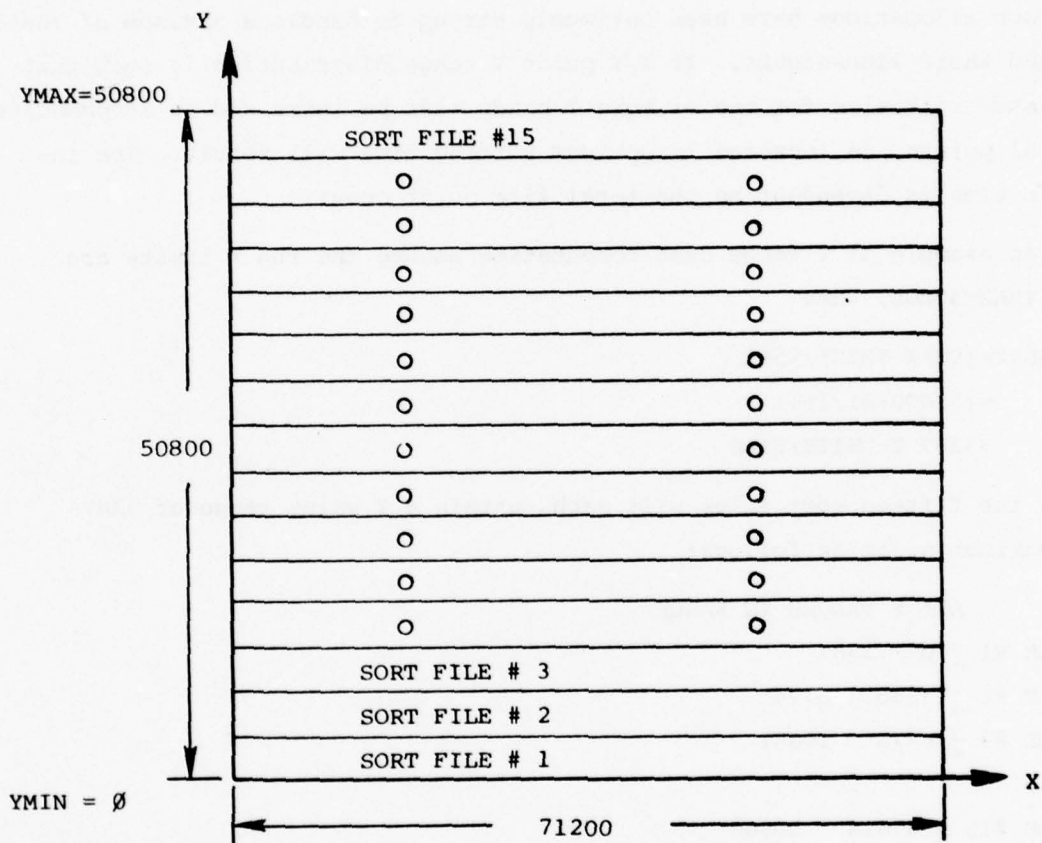
```

      ALL Y VALUES IN RANGE
FILE #1 → 0 → 3387
FILE #2 → 3388 → 6774
FILE #3 → 6775 → 10061
      :
      :
FILE #15 → 47414 → 50800

```

Thus, if a point's Y component value is 6800 the point and its lineweight will be placed in the third file (see Figure No. 2-11).

When all the points for the Input Processor have been processed, a word which designates an end-of-file (all bits on) is written to each of the fifteen sort files.



RUN Y-RANGE SORT FILE SPLIT

Figure No. 2-11

2.1.1.1.13 Subroutine PUTENT

The routine packs the X/Y point and lineweight into the next available seven bytes (6 bits/byte), three bytes for X and Y and one for lineweight, of the appropriate 224 word portion of the temporary buffer, with respect to point's sort file number association (point Y coordinate value's association with sort file number). This is accomplished via a Univac 1108 Library supplied function called MBYTES. Each point is packed in the order of Y-coordinate, X-coordinate and then lineweight (see Figure No. 2-12.)

Once all 224 words of a portion of temporary buffer (LARGE) associated with a particular sort file has been filled (192 Y/X/LW), this block is written to the appropriate mass storage sort file. Since NTRAN is used for I/O, no 'green' words are supplied by system. Each sector of 28 words will fully contain 24 X/Y points and their associated lineweight. Note that three bytes (6 bit/byte) for X and Y values is more than sufficient to represent maximum values of 71200 and 50800 respectively. In fact, 17 and 16 bits for X and Y would have sufficed however cumbersome to work with.

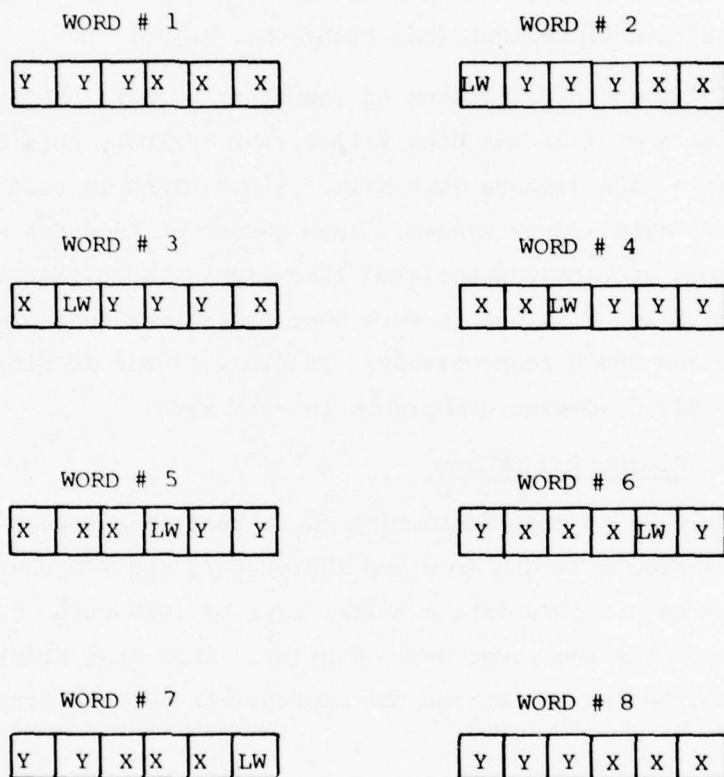
2.1.1.1.14 Subroutine ASCBCD

The main purpose of this subroutine is to read into core a physical BCD tape record of variable length (maximum 170 words), six BCD characters per word, and unpack each record into a buffer area of 1020 words such that each BCD character occupies the lower byte (6 bits). This then simplifies mainline processing by not having to extract the appropriate BCD character before testing for it.

2.1.1.2 DMAAC Input Processor

The main differences between the two Input Processor versions other than the format

- (1) Mainline program conversion of all X/Y points from GE floating point format to Univac 1108 positive integer components. (MMS X/Y points are in 0.0001 (mil unit hence must be divided by ten (10) before conversion to metric).



NOTE: THE PACKED SEVEN BYTE FORMAT IS CYCLIC EVERY SEVEN (7) WORDS

SORT FILE PACKED SEVEN BYTE FORMAT

Figure No. 2-12

- (2) X/Y point component values don't need to be generated from characters.
- (3) The Points subroutine receives a maximum of fifteen data points at any one time, which were extracted from a data logical record. The Points subroutine is entered each time a data record is located.

Following is a more detailed explanation of the DMAAC Input Processor mainline program (MAIN). The other subroutines are not mentioned since identical to DMATC Input Processor routines.

The DMAAC input format will be discussed first, to provide clearer insight into the description of the mainline program.

The DMAAC tape format consists of physical records of 298 words each. The file begins with a header label record of 14 words followed by an end-of-file (hardware). Following this header label record are N physical records of 298 words. These records are themselves followed by an end-of-file (hardware) and a trailing label record. A more detailed description follows:

The overall MMS32 magnetic tape format is shown in Figure No. 2-13. Each word is 36 bits in length. The breakdown of the figure is as follows:

- o The first fourteen words consist of a GE Header label. The Header Label is fixed-content, except for the Tape Number field which is set to the Job No. assigned.
- o The Header Label is followed by an End of File.
- o The EOF is followed by "n" Physical Records. Each record is 298_{10} words in length (except the last, which may or may not be full).
- o The last physical record is followed by an End of File.
- o The EOF is followed by fourteen words consisting of a GE Trailer Label. (Same as Header Label in content).
- o The GE Trailer Label is followed by an End of File.

0	2	3	5	6	8	9	11	12	14	15	17	18	20	21	23	24	26	27	29	30	32	33	35	
ZERO																						0		
ZERO																						1		
ZERO																						2		
ZERO																						3		
ZERO																			R	R	4			
ZERO														F	F	F	F	5						
ZERO														S	S	S	S	6						
ZERO														C	C	C	C	7						
T1				T2				T3				T4				T5				T6				8
T91				T92				T93				T94				T95				T96				23
X MINIMUM																						24		
X MAXIMUM																						25		
Y MINIMUM																						26		
Y MAXIMUM																						27		
X FIRST																						28		
Y FIRST																						29		
X LAST																						30		
Y LAST																						31		

0	14	35
# (X,Y) COORDINATE PAIRS	ZERO	0
	ZERO	1
	X1	2
	Y1	3
	X2	4
	Y2	5
	X15	30
	Y15	31

MMS BLOCK FORMAT

The Physical Record format is shown in Figure No. 2-14. The breakdown of the figure is as follows:

- o The first word of each record:

BITS 0-17: record serial number

18-35: number of words in the record excluding itself (297_{10} max.)

- o The remaining part of the record consists of nine logical (max.) blocks or records each 32 words in length.

- o Each block is preceded by one word containing the following:

BITS 0-17: number of words in the block (40_8) excluding itself

18-35: specific report number (1021_8)

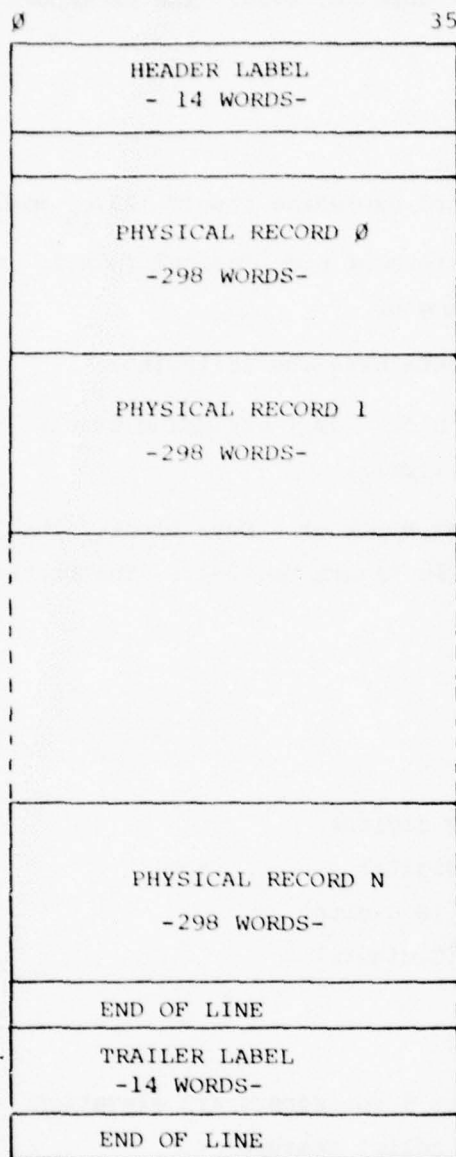
Each Logical Block may be either a Header Block or a Data Block. The format for the two types of blocks are shown in Figure No. 2-13. The breakdown of each block is as follows:

Feature Header Block

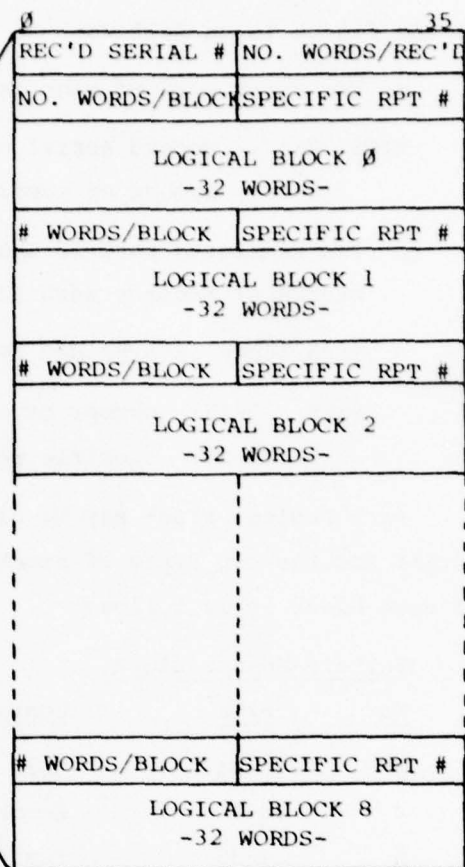
<u>Word</u>	<u>Bits</u>	<u>Content</u>
0-3	0-34	Zero
4	30-35	Record Content (R digits)
5	24-35	Feature Type (F digits)
6	24-35	Feature Sub-Type (S digits)
7	24-35	Feature Control (C digits)

Feature Header Block

<u>Word</u>	<u>Bits</u>	<u>Content</u>
8-23	0-35	Text Fields (first 6 characters are elevation value if it is a relief feature)
24-27	0-35	Feature Bounding Rectangle (X,Y pairs in GE floating point)
28-31	0-35	Start and End of Feature (X,Y pairs in GE floating point)
0	0-14	Number of (X,Y) coordinate pairs in the block (15_{10} max.)



A. MMS TAPE FORMAT



B. PHYSICAL RECORD FORMAT

OUTPUT TAPE FORMATS

Figure No. 2-14

<u>Word</u>	<u>Bits</u>	<u>Content</u>
0	15-35	Zero
1	0-35	Zero
2-31	0-35	Alternating X and Y coordinate data (GE floating point) .

The sequence of logical blocks deserves some special attention. The first logical block of the first physical record must be a JCRS Header Block and the last logical block of the last physical record must be a JCRE Header Block. Both these header blocks contain all ZEROS except for Word 4 which contain 11_8 and 13_8 respectively for the JCRS and JCRE.

Each feature header block precedes one or more data blocks. The data blocks may contain up to 15 X,Y coordinates at 0.1 mil resolution or a single discrete X,Y point. In the latter case, each data block is preceded by its header block. Note that each X,Y coordinate consists of two GE floating point numbers (words).

The DMAAC Input format is described as follows. The user first chooses number of files to be processed, window limits (optional), any origin shift rotation, mirror image and the run resolution. This information is specified in the Namelist data cards which are input at execution time.

Once these options are specified the records are read into a processing buffer the size of one physical record (298 words). The word preceding each logical record is used to extract the number of words contained in the following logical record. Once this information is obtained, a test of the first word is made to determine whether the logical record is a header or data record. Should this word be non-zero then processing of a data record is initialized, otherwise a header record is assumed.

For the Header record case, the only processing that occurs is a test of the fifth word of record for the JCRS or JCRE records. Should an Octal 13 be encountered within lower six bits of the word, processing termination will result; a JCRE record being the case. Since an octal 11 indicates a new file, if encountered the number of files specified to be processed is decremented by one. When the file count is zero processing terminates. The only other data obtained

from this header record is the line-weight for the following X/Y points feature string. This occurs in the lower six bits of the seventh word of the record. It is important to note that one or more header records may proceed a data record(s), however data records may not precede a header record(s).

In processing data records, the first word of the logical data records contains the number of X/Y points pairs to be found in the record. This value is used as a loop argument to process the designated X/Y points. Each X/Y coordinate is extracted in its binary form, which is in GE floating point format, and converted to integer X or Y coordinates. This value is then loaded into the appropriate X or Y buffer area. Once all of the feature data's X/Y points are converted, the subroutine POINTS is called to process these points. Any rotation, mirror imaging, etc., is accomplished in this subroutine. The feature data processing is now complete, and a new logical record is processed. Once the whole block (9 logical records) has been processed a new block (298 words) is read into processing buffer and processing continues.

2.1.1.3 Sorting Procedure

Once the input processing is complete, the data is sorted using a UNIVAC Utility ANSI Cobol Sort. Each point's 7-byte packed X/Y coordinate and line-weight are used as a single sort key. The Y-value occurs in bytes 1-3, X-value in bytes 4-6, and the line weight located in the byte 7 (see Output Phase for description of format). The sort thus operates on a seven character field, where each character represents a 6 bit byte. Fifteen data files may be sorted either sequentially, in ascending order from file 1 to file 15, or randomly, if more than one file at a time is to be sorted. The particular sorting method is optional, depending on the program data cards runstream.

Each data file is made small enough to speed up the sort, but at the same time, capable to hold a sufficient number of points. The files as presently designed will hold a maximum of 768 K points. The reason for such large data files is due to the non-uniformity of the data points, since the probability that dense areas exist on the source is very large, ruling out any general X/Y point distribution.

2.1.1.4 Output Processor Functional Considerations

The Output Phase exists in the following eight versions

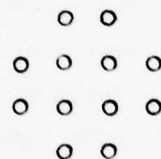
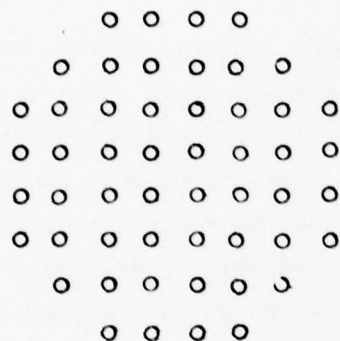
- (1) P3 - output resolution 25 microns
- (2) P3/R2 - output resolution 50 microns
- (3) P3/R4 - output resolution 100 microns
- (4) P3/HW - output resolution 25 microns
- (5) FP3 - output resolution 25 microns
- (6) FP3/R2 - output resolution 50 microns
- (7) FP3/R4 - output resolution 100 microns
- (8) FP3/HW - output resolution 25 microns

each of which is comprised of the routines MAIN, RLCBIN, MBYTES, NTRANS where the latter two are the same Assembly language subroutines in Phase I. The first four versions have one section of code (about 135 Fortran Statements) replaced by a call to an Assembly language subroutine, CBR, to produce Run-Length-Code form of output at about eight times faster than the Fortran code utilized in the last four versions (FP3, FP3/R2, FP3/R4, FP3/HW). Thus, the first four listed Output Phase versions (P3, P3/R2, P3/R4, P3/HW) are extremely more time efficient, due to use of Assembler CBR routine, than their counter-parts FP3, FP3/R2, FP3/R4 and FP3/HW. As a result, it is recommended that only the versions P3, P3/R2, P3/R4 and P3/HW be considered for execution.

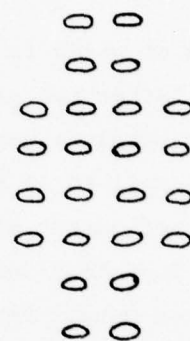
Half-width versions (valid for all X coordinate values less than 35.04 inches), denoted by HW, run much more quickly than full width versions (all X values $\leq$ 70.08 inches) because scan line images can be processed completely in core buffers instead of requiring two disc files to handle line weights greater than 16 mils. Best run times (at output resolution of 25 microns) can be obtained using either HW versions or by using only linewidths less or equal 16 mils in input collection with full width versions P3 and FP3. The 50 and 100 micron output resolution versions will process the full chart size (50 X 70.08 inches) for maximum linewidth of 32 mils (input collection) totally within core due to reduction of linewidths by Output Phase to one-half and one-fourth their original values. Thus, only the versions P3 and FP3 will require mass storage scan line files when input collection linewidths are greater than 16 mils. "F" prefix versions are all Fortran which run 5 to 8 times slower than the corresponding versions using an Assembly language subroutine to pack up the RLC output.

Resolution in Phase III is not quite the same as in Phase I in that in Phase I the resolution is effectively the "minimum center-to-center spacing of points defining a feature" which results in a slight loss of detail at 50 and 100 micron resolution specified. Output resolutions of 50 and 100 microns are effectively over-all "compressions" or scale reductions in both axes which can be compensated or corrected back to true scale by the plotter hardware in a cruder representation.

Section of 8-MIL Line
At True 25 Micron Resolution

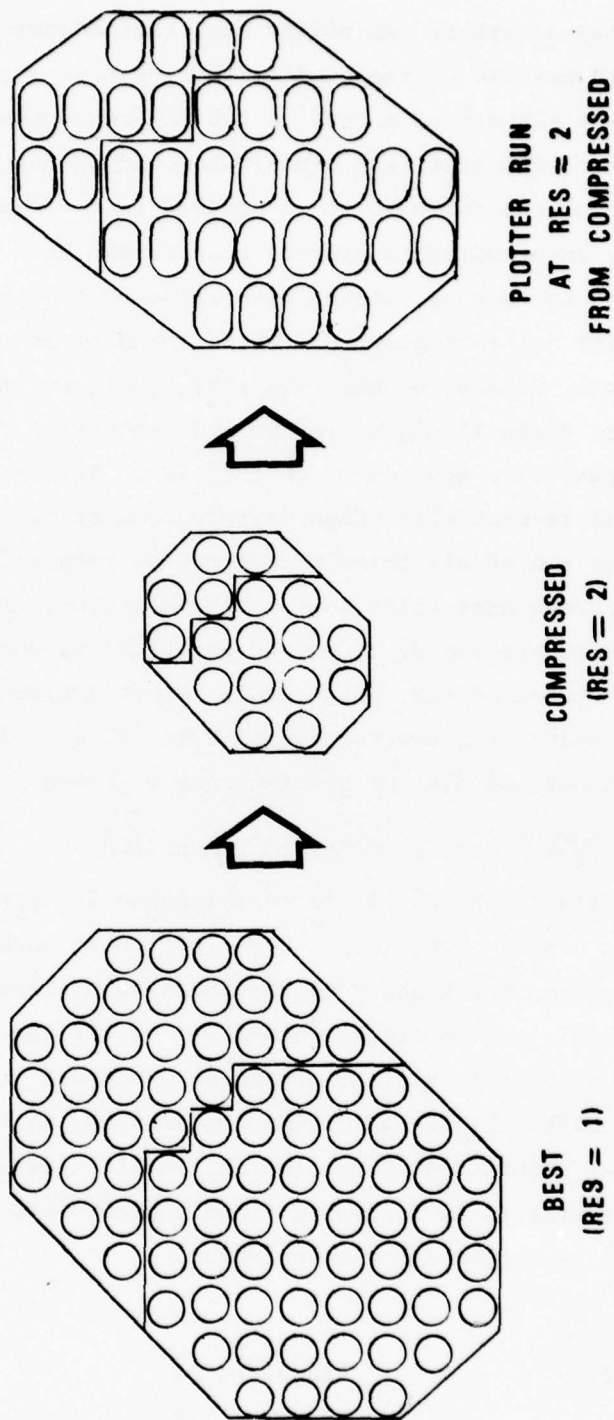


Hardware Expansion
of Section



Same Line Section
at 50 Micron Resolution

The 50 and 100 micron Output Phase resolutions allow for a "rough" quick plot of input data collection (see CTS Phase III time comparisons) for data validation/correction before finalized plot. Owing to processing time requirements at output resolution of 25 microns, this becomes a very useful capability. In addition, capability could also be used, if desired, to obtain "magnified views" of any data fitting within physical plotter limits at two larger scales. Specification of resolution in Phase I is completely independent of Phase III resolution. Also, since Phases I and II (Sort) are fairly rapid processes, it is recommended that the Input Phase (Phase I) be processed at 25 micron resolution (RES=1) for normal processing of normal size data sets for greater line integrity.



DATA COMPRESSION

Figure No. 2-15

Phase III has a restart capability such that a maximum of four minutes of CPU time would have to be repeated in the event of a system crash or an intentional operator-abort of a run. A typical reprocessing time would be about 30 seconds of CPU time, and a minimum would be on the order of 10 seconds. Under normal use of the software it is assumed that a Phase III pass will not be intentionally interrupted to process a different data set in whole or in part. This could be done by dumping disc files to tape and restoring them before the initial job is restarted. Phase I and II do not have a restart capability as such, because of the relatively short run times ordinarily required. However, Phase II can be re-run independent of Phase I provided that the Phase I output files are intact on the disc. Any system crash requiring a full or partial re-boot will often destroy some or all of the disc data files and a complete re-run of all three phases may be required if the restart file or any of the fifteen sort files have to be destroyed. If a particularly large volume job is being processed, it may be practical to dump disc files to a temporary tape between phases, which would permit avoidance of some re-processing even in the event of a destruction of Sort files. If the restart file is destroyed none of the data processing done by Phase III can be recovered.

2.1.1.4.1 Output Processor Functional Design

The input data to the Output Processor phase are fifteen sorted data mass storage files in a seven byte (6 bit byte) per point packed format; the first six of which contains the X and Y center point values respectively, with the seventh byte containing the linewidth value. The files are read in ascending order (file #1 to file #15) or ascending Y address value, in record sizes of 224 words. Due to the packed format implemented, each record when unpacked yields 192 X/Y points and corresponding linewidth values. The mass storage files data format is as follows with only eight words (36 bits) being shown, since the format is cyclic every seven words.

WORD #1
869427654321

Y X

WORD #2
986498654321

LW Y X

WORD #3
765486549876

LW Y

WORD #4
123456654321

X LW Y

WORD #5
109876543210

X LW Y

WORD #6
123456789101

X LW

WORD #7
519194871218

Y X LW

WORD #8
905192151750

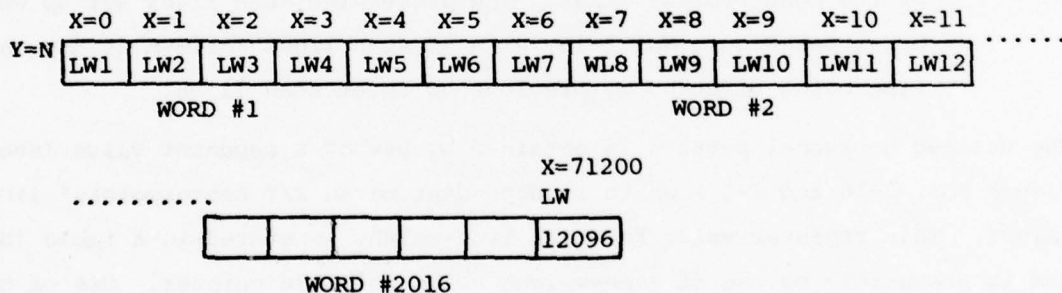
Y X

WORD #9
621195610461

LW

That is, every seven words the Y-values, X-values and lineweights of the center points may be located in high order 18 bits of word W, lower order 18 bits of word W and higher order 6 bits of word W+1 respectively.

As each point and its corresponding lineweight is unpacked, a check is made to determine if the point lies on the present scan line being processed. If so, its word (NWORD) and bit position (BITPOS) within the scan line and similarly within a temporary lineweight storage buffer whose lineweight location is dependent on X coordinate value in the scan line, is calculated. The lineweights are stored within this temporary buffer, NXCNT, in the following manner in consecutive six bit byte (maximum lineweight 0.8 millimeters) six lineweights per word format.



where N is present scan line being processed

This buffer is generated and cleared for each scan line that is processed whenever a point is encountered which does not lie on present scan line.

Each point lying within the same scan line is processed as it is encountered and bits within the appropriate word affected by this X/Y point are set. This is accomplished by a calculation of the maximum and minimum bits affected,

for each X/Y point on scan line, and using the difference of these values as pointers to appropriate word value table (ITAB) to turn-on the correct bits. If the linewidth is even the center mate scan line bits are also set at the same time (see Figure No. 2-16 and 2-17).

Once all points lying within the present scan line being processed have been processed, (i.e. upon encounter of X/Y point not on present, scan line) scan lines affected by this scan line's center points must have their appropriate bits set to generate the correct octagonal spot due to center points' line weights. This octagonal spot generation is accomplished by first calculating the number of scan lines required to exist in core and the number of scan lines required in each of the two mass storage files. No user intervention is required in this step, since automatically the maximum run linewidth, which was determined in the input processing phase, is passed to the output phase for determination of the above mentioned quantities.

EXAMPLE: Suppose the maximum line weight for the run determined by the input phase is 32 (0.8mm) and INCORE=16 which is maximum allowable due to available core storage. Then 16 scan lines in core would be utilized (at 2016 words per scan line) and 8 scan lines on each of the mass storage files. The processing scan lines set up would be as per Figure No. 2-18, with 32 scan lines maximum at any one time being affected by previous or later scan lines.

The desired octagonal pattern is obtained by use of a repeater value (see Figure Nos. 2-16 and 2-17) which is dependent on an X/Y centerpoints' linewidth. This repeater value for each line-weight is stored in a table (NNDUP) and is accessible by use of line-weight value as table pointer. Use of this repeater variable method results in an octagonal pattern with 45° sloped edges. This repeater variable is used to terminate an X/Y center point spot pattern and to generate the octagon pattern for each X/Y center point. Now once all affected scan lines due to present scan line have been processed, a scan line is written to magnetic tape in the chosen Binary or Run-Length-code format (TTYPE Parameter variable) and a new scan line is processed, once all appropriate buffers have been cleared.

MATES FOR PATTERN

$N+1 \leftrightarrow N-1$

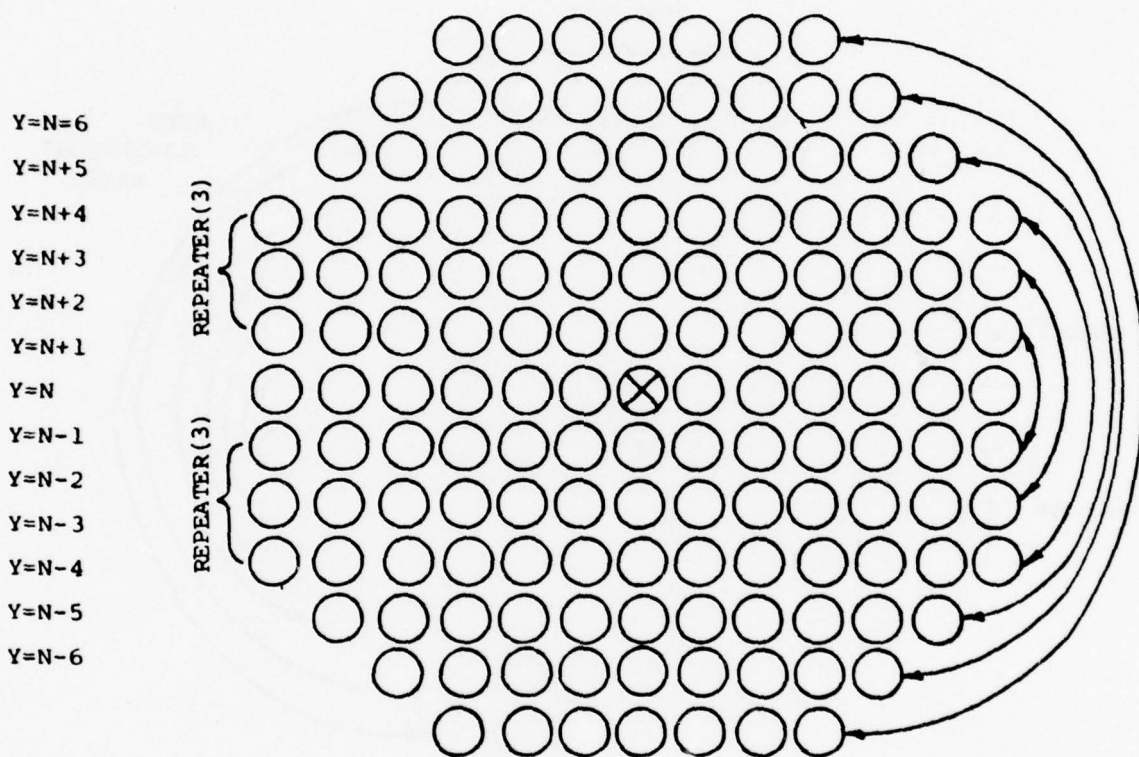
$N+2 \leftrightarrow N-2$

$N+3 \leftrightarrow N-3$

$N+4 \leftrightarrow N-4$

$N+5 \leftrightarrow N-5$

$N+6 \leftrightarrow N-6$

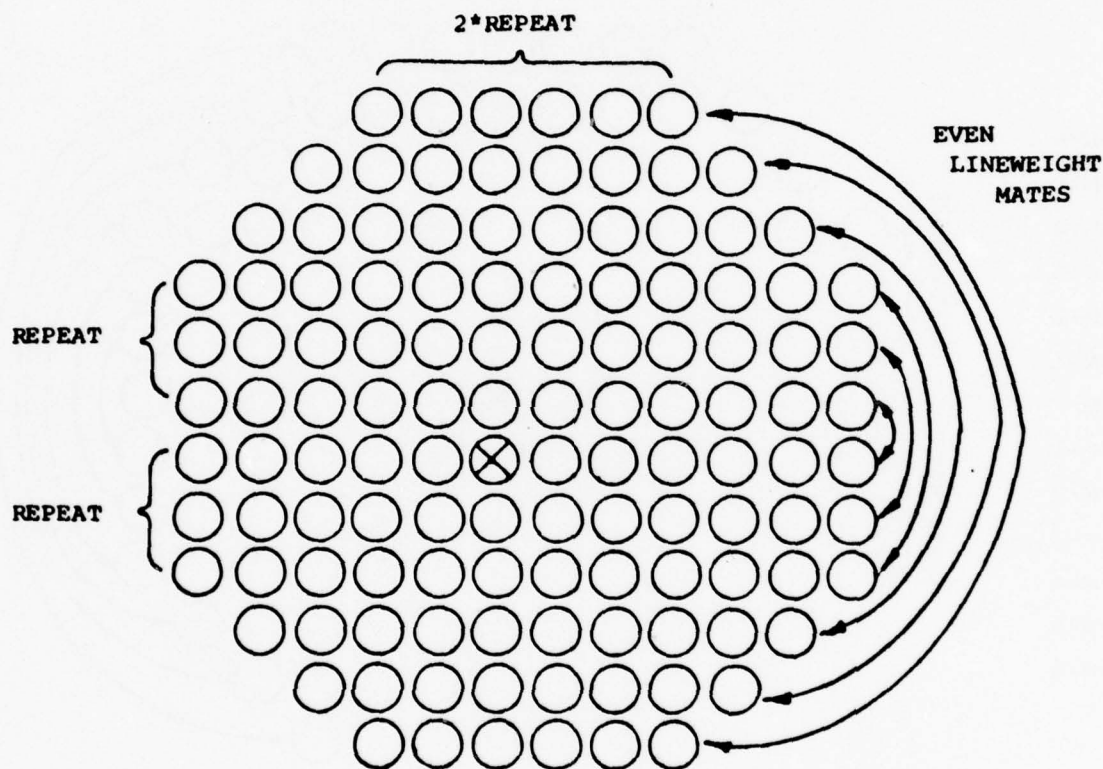


NOTE: The above octagonal pattern illustrates the scan line mates for an odd line-weight along with the repeater scan lines. Arrows designate the mates

OCTAGONAL SPOT PATTERN (ODD)

Figure No. 2-16

LW=12 Y=2000 X=1515
 SCAN LINE IS 2000 NWORD=1515/36+1=43



NOTE: If this X/Y center point were the first to appear in word 43 of scan line 2000, the repeat value would be table NNDUP whose location would be determined by the Line Weight (i.e. NNDUP(12))

OCTAGONAL SPOT PATTERN (EVEN)

Figure No. 2-17

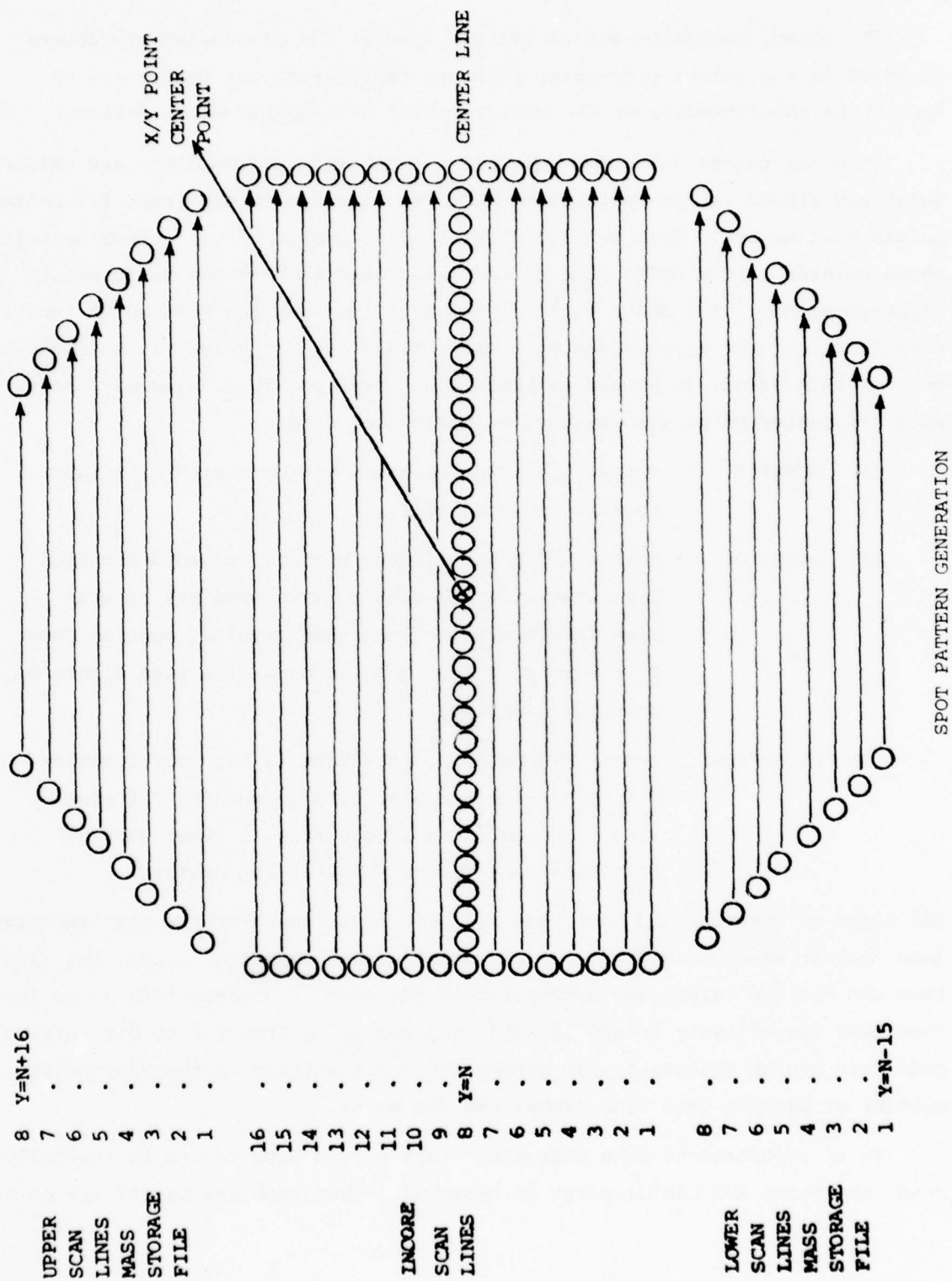


Figure No. 2-18

The above discussion was an overall view of the processing procedures utilized in the output processing phase to familiarize the reader and to facilitate understanding of the more detailed description which follows.

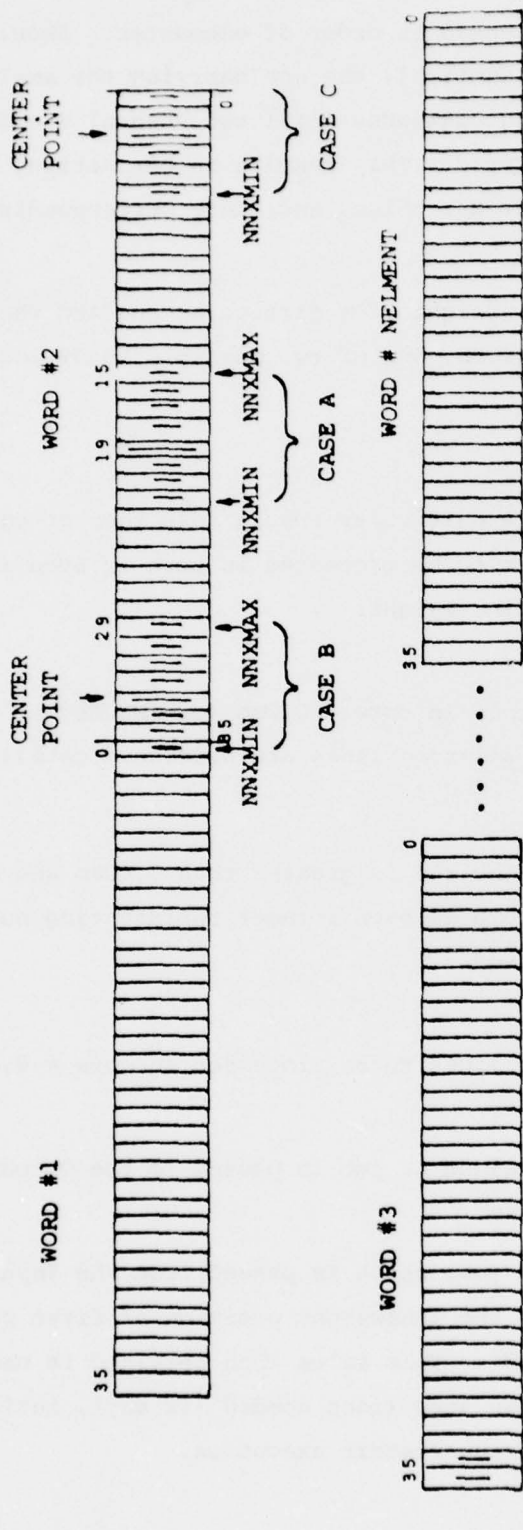
Once the points have been unpacked and the required pointers are calculated and stored in the appropriate buffers, bits due to processed X/Y center points must be set. This setting of bits is accomplished via a look up table whose pointer is the difference of the calculated minimum and maximum bits affected values. The above table, in conjunction with the Fortran OR function, will "turn on" the appropriate bits via a table numeric value, if not presently on. At this point, it should be noted that three possible cases may occur for each X/Y center point with respect to influenced bits.

- (1) GENERAL - center X/Y point affects only bits within its word location in scan line.
- (2) OVERFLOW - center X/Y point affects bits in its word location N in scan line and also bits in word N+1 of scan line (minimum value less than zero*). Special case if occurring in last word of scan line (see Figure No. 2-19 for example).
- (3) UNDERFLOW - center X/Y point affects bits in its word location N in scan line and also bits in word N-1 of scan line. Special case if occurring in first word of scan line (see Figure No. 2-19 for example).

The cases of overflow and underflow will not occur concurrently for lineweights less than or equal to 0.8mm. For the overflow and underflow cases, the minimum and maximum values are appropriately adjusted if greater than 35 or less than zero respectively (since 36 bits in a word). A transfer to bit setting code, via assign statement, now accomplishes bit setting in the appropriate word(s) of present scan line (YNOW) and its mates.

It is important to note that when first sorted data record is initially read, the first X/Y center point is unpacked. Then each new center X/Y point

SCAN LINE OR ROW OF ARRAY BITSET



(A) GENERAL-no underflow or overflow

Y=800 X=53 LW=8 (0.2mm)
 NWORD=53/36+1=2
 BITPOS=MOD(53, 36)=17
 MINIMUM (NNXXMIN)=36-(17-3)=22
 MAXIMUM (NNXXMAX)=36-(17+3+1)=15

(B) UNDERFLOW-previous word affected

Y=800 X=39 LW=8 (0.2mm)
 NWORD=2
 BITPOS=3
 NNXXMIN=36-(3-3)=36
 NNXXMAX=36-(3+3+1)=29

(C) Y=800 X=70 LW=8

BITPOS=34
 NNXXMIN=36-(34-3)=5
 NNXXMAX=36-(34+3+1)=-2

SCAN LINE WORD ARRANGEMENT CENTER X/Y POINT

Figure No. 2-19

is unpacked, the X-Y point is processed in order of encounter. Should two line-center coordinate values be identical, the one carrying the smaller line weight code is dropped. Processing continues until two unequal Y address (different scan lines) are encountered. This results in bit setting for all scan lines influenced by the center X/Y points and their corresponding line-weights.

Depending on the maximum line weight of a particular run and the set value of the Parameter variable INCORE, one of two possible conditions will occur.

(1) FULL WINDOW IN CORE

Maximum line-weight for a particular run is less than or equal to 0.40mm and the number of scan lines to be processed in core is even integer greater than or equal to maximum line-weight.

EXAMPLE: MAXLW=0.175mm

number scan lines in core = 0.2mm (or INCORE=8)

For this case all scan lines are processed totally in core.

(2) HALF WINDOW IN CORE

Maximum line-weight for the run is greater than 0.40mm and number of scan lines to be processed in core is even integer representing half of the actual scan lines to be processed.

EXAMPLE: MAXLW=0.800mm

number of scan lines to be processed in core = 0.4mm (or
INCORE=16)

NOTE: The maximum linewidth of job is passed to the Output Phase thru the Sort Phase.

The maximum linewidth of the job, which is passed from the Input phase to the Output phase, is located in the linewidth position of first point of first sort file (X=0, Y=0, LW=MAXLW). This value when obtained is used to determine the number of mass storage scan lines needed (if any), initialization of parameters and determination of run restart execution.

When processing the first scan line of a run, initialization of window Y addresses starting Y address for run (YSTART) and Y address of present scan line being processed (YNOW) is accomplished. Hence, when a scan line is written to tape, its Y address is the minimum Y address of window presently being referenced. The address value chosen again depends on full windowing or half window case, respectively.

In the half window case, the number of scan lines to be processed in core is specified by the Parameter Statement variable INCORE (16 scan lines in core are utilized). Once the first 16 special case scan lines are processed, each new scan line to be processed will result in writing to tape the minimum window scan line (YMIN) and then processing the new scan line within its buffer location. As each scan line is completed, it is converted to the RLC/BIN Plotter Drive Tape format, either in the normal orientation or in the possible reflected or rotated arrangements, and removed from the system by outputting to tape. The rotating binary window, for mass storage and in core scan lines is then shifted by one line, so that the buffer area represents the next scan line window limits. Scan lines are processed such that the center of the spot pattern is first generated in core and upon completion of these 16 in core scan lines, the minimum scan line from the lower mass storage file is written to magnetic tape in the desired format. Then the least Y coordinate value scan line of the in core lines is written to this lower file, which at the same time the least Y value scan line in the upper mass storage file fills the empty scan line in core slot. The processing develops the mass storage scan lines utilizing a continuously sliding 16 scan line window at maximum (the maximum 0.8mm line weight at the maximum 40 lines/mm resolution), stored on 18 tracks of FH432 drum. At any point in time, the continuously sliding (or rotating window) may appear as in Figure No. 2-20.

In the case of full window, all scan lines affected by the current scan line being processed (YNOW) are processed totally in core by implementation of the same continuously sliding window. Scan lines are written to tape for the special case of the initial INCORE scan lines beginning with the first scan line which is $\text{INCORE}/2$ scan lines away from the present scan line (YNOW). For example, if the number of scan lines to be processed in core is 14

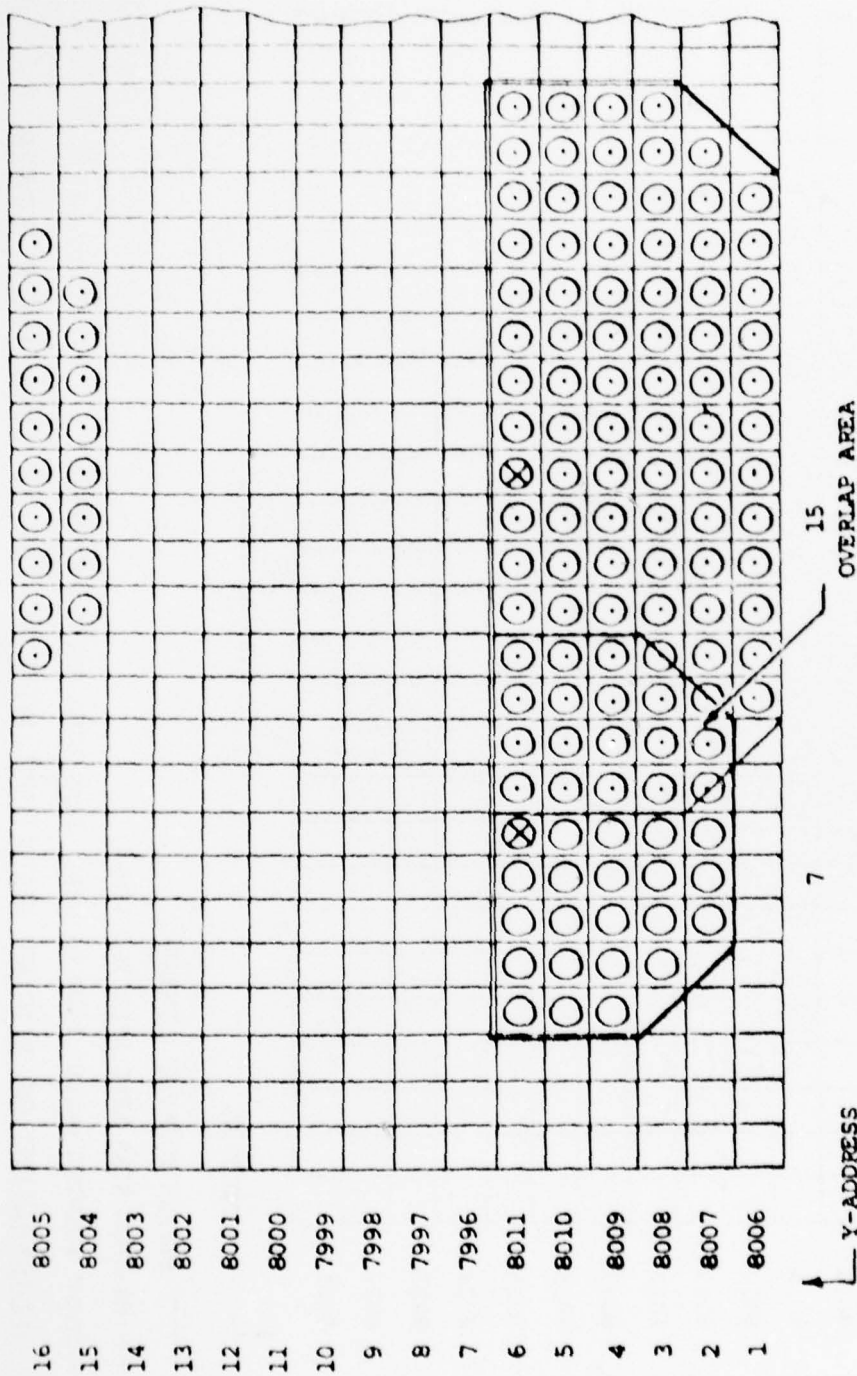
(INCORE=14), scan lines will commence being written to tape when present scan lines being processed minus 7 is greater than the starting Y address of the run (YSTART). Upon completion of the initial scan lines processing, each new scan line will result in a scan line written to tape whose address is YNOW-7.

Processing for this case is identical to the half window case except that all scan lines are processed in core and no mass storage files are required. The only other remaining difference is the Y address utilized upon writing a scan line to tape.

The overall general method for spot pattern generation is identical for either case of full or half window, in that all output line weights are built up by arrays of a single 0.025mm spot size into a "binary image" (one bit per spot, zero if white, and one if black). The array shapes will be octagonal for line weights of 150, 200, 250, 300,....450, 500, 650, 750, 800 microns, and will degenerate into a "square without corners" at 125 or 100 microns. Except in the case of 100 or 125 micron line weights being plotted at output resolutions of 20 lines/mm or 10 lines/mm this approach will eliminate all "edge ripple" by providing at least 50% overlap of adjacent array patterns for the full range of valid and 97% overlap for the heavier line weights. The difficulties of variation in line density are eliminated in this approach because there is no overlap of the elemental 25 micron spots regardless of the overlap of the spot pattern laid down, and each elemental spot is exposed only once regardless of the number of array patterns which happen to cover the element. Figure No. 2-22b illustrates the case of generating a 175 micron line weight by overlapping of arrays. Also some array spot patterns are illustrated in Figure No. 2-22a.

The last in core scan lines of a run are a special case in that once scan line address Y=50,800 has been processed, no new scan lines are loaded into the processing buffer (BITSET) and the appropriate scan lines are written to tape in RLC or Binary mode.

LOWER SPOT PATTERN



PRESENT POINTER VALUE (YMIN OR YMAX) OF SCAN LINE BEING PROCESSED

(YMIN) SCAN LINE WRITTEN TO TAPE

ILLUSTRATION OF SLIDING (ROTATING)

WINDOW FOR 16 SCAN LINES

NOTE: The line-weights for the two centers X/Y points are 0.225mm and 0.4mm respectively. Also note the overlapping of spots area.

Figure No. 2-20

UPPER SPOC PATTERN

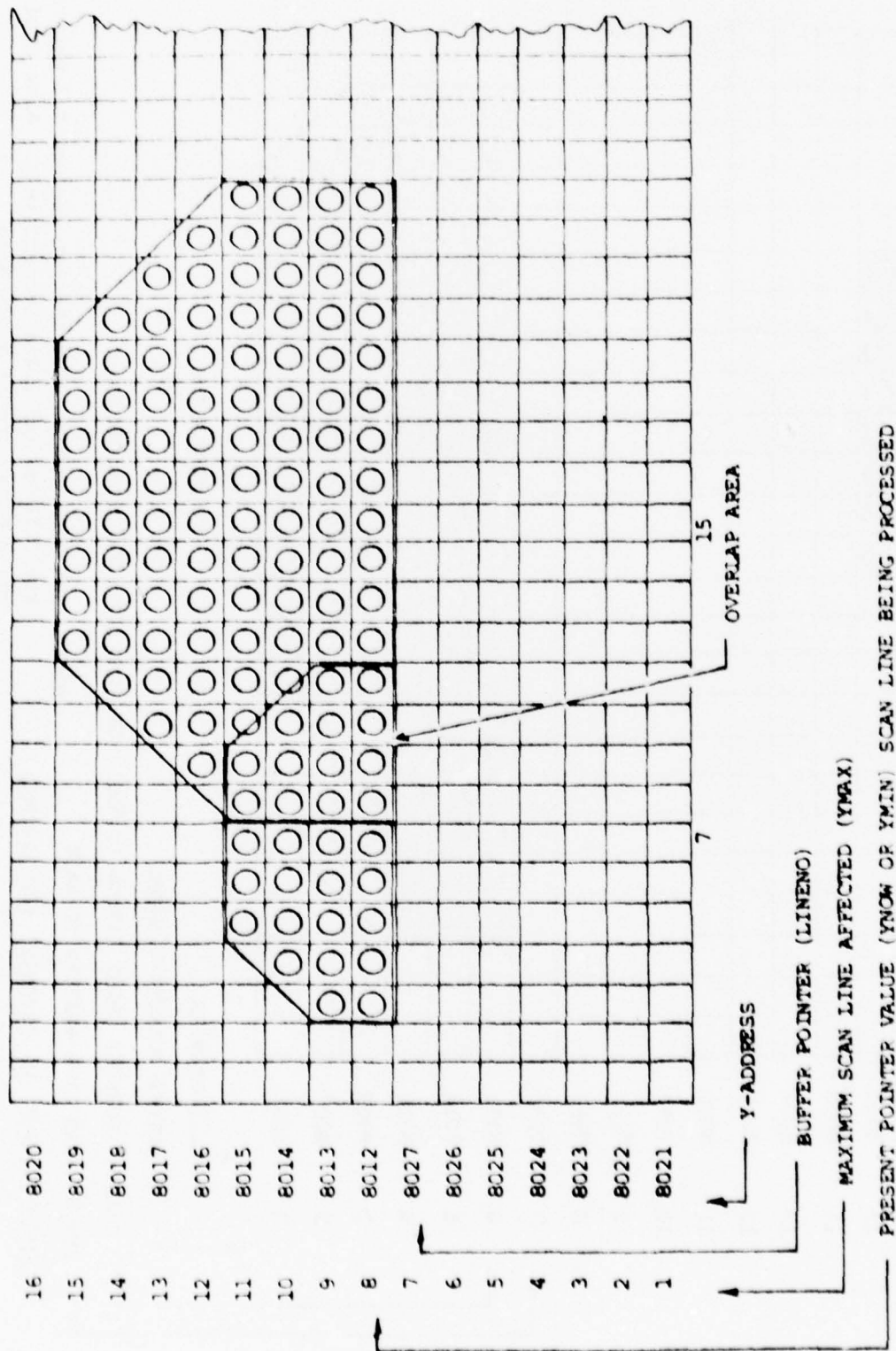
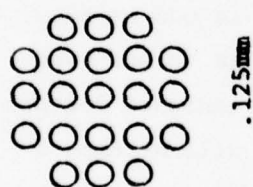
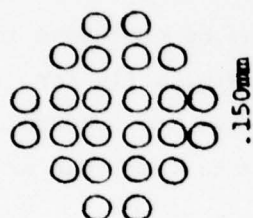
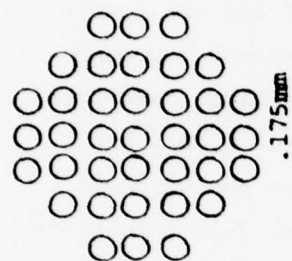
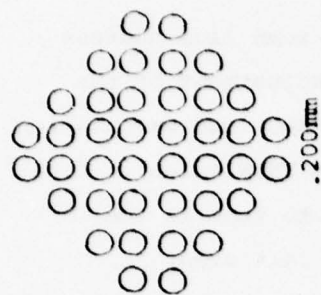
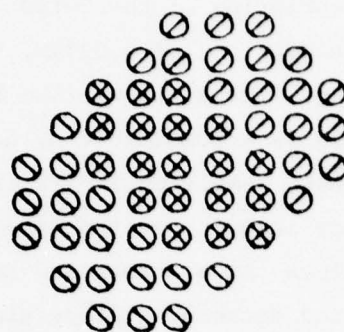


Figure No. 2-21

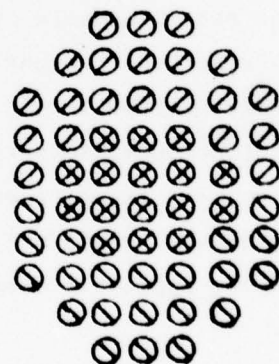


ARRAYS OF ELEMENTAL SPOTS

Figure No. 2-22a



TWO .175mm SPOT ARRAYS,
.106mm OVERLAP ALONG
45° ANGLE



TWO .75mm SPOT ARRAYS,
.100mm OVERLAP ALONG AXIS

OVERLAP OF ELEMENTAL SPOT ARRAYS

Figure No. 2-22b

Scan line may be such that successive corresponding scan line address difference is greater than one. This will result in an adjustment of the windowing Y scan line addresses, line number counter (LINENO) and calculation of the next scan line to be processed (i.e. adjustment in YNOW value). Likewise, an appropriate number of scan line will be written to tape in the designated format before processing of scan line YNOW is initiated.

The above description of the Output Processor pertains to all three possible output resolutions of 0.025mm, 0.05mm and 0.1mm. For the latter two resolutions, the core sizes will be reduced to one half and one fourth respectively. This is accomplished by having each bit represent a 0.05mm or 0.1mm binary spot, resulting in 1/2 or 1/4 of the buffer sizes needed to process the "binary image". In the case of the resolution of 0.05mm and 0.1mm, all scan lines can be processed totally in core, thus avoiding all mass storage I/O and decreasing processing time. Figure No. 2-23 illustrates how each line weight is treated at these higher resolutions. As noted in above figure, the values in parenthesis are used to represent the respective line-weight. Also, for 0.1mm resolution, all scan lines may be processed in core since the highest relative line-weight is 0.2mm (8). Similarly for 0.05mm resolution, all relative line weights less than or equal to 0.400mm(16) may be processed totally in core (i.e. all actual line weights less than or equal to 0.8mm).

In conclusion, the scan lines are output to an odd or even scan line tapes dependent upon whether the present scan line address is odd or even. Also, the output tapes are called for in pairs by the program as the need arises. Thus, a sufficient multiple sets of tapes must be assigned for program output tapes. Upon read or write errors, the program will attempt a recovery by attempting several reads/writes. If unsuccessful, processing will be terminated and the appropriate error condition encountered will be output to the printer.

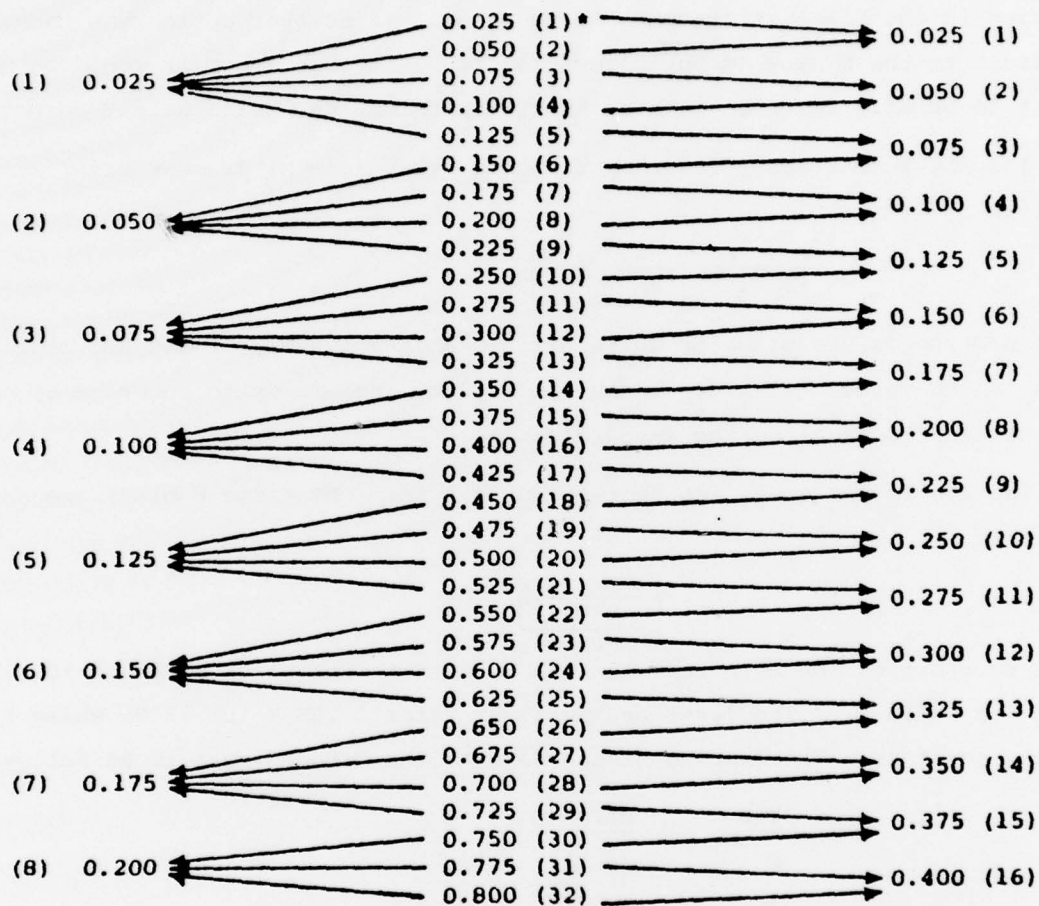
2.1.1.4.2 Output Processor RLCBIN Subroutine

When a scan line in core is no longer affected by successive scan lines, its buffer record pointer (LINENO) along with the scan line is prepared by this subroutine for output to one of two Plotter Drive tapes.

LINE-WEIGHTS
0.1mm RESOLUTION
(UNITS IN MILLIMETERS)

LINE-WEIGHTS
0.025mm RESOLUTION
(UNITS IN MILLIMETERS)

LINE-WEIGHTS
0.05mm RESOLUTION
(UNITS IN MILLIMETERS)



*NOTE: () Value designates program representation of line-weights.

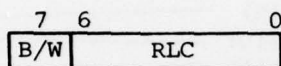
LINE-WEIGHT CONVERSION FOR 0.05MM AND 0.1MM FROM 0.025MM

Figure No. 2-23

Should the first scan line to be output be an odd Y address, a "false", even Y addressed scan line is supplied in Binary or Run-length code. This is done so that the Plotter Drive Tapes reading always begins with the even scan line or same tape unit.

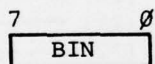
The scan line may now be output in RLC or BIN, depending upon user specification in the Namelist statement data cards. If neither option was chosen, a default to the Binary output format results. At this stage it would be pertinent to briefly describe each of the two possible scan line output formats.

The RLC format specifies that the data bytes (one MT frame-8 bits) are in the form



where B/W represents Black or White bits (0=White; 1 = black) and RLC represents a Run Length value of $1-127_{10}$ (number of .025mm increments to power-on or power-off the laser beam along the X-coordinate).

The Bin format specifies that the data bytes (1MT frame 8 bits) are in the form



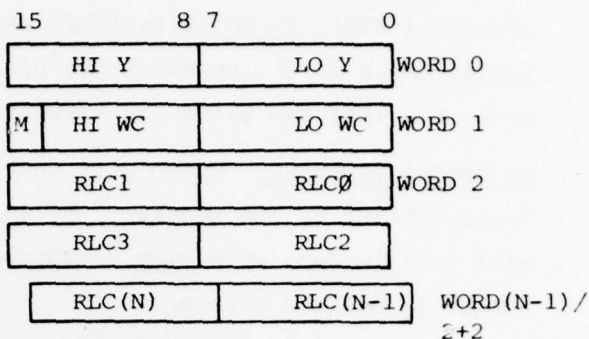
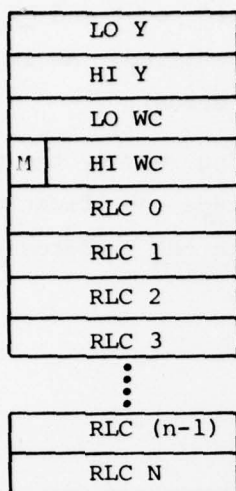
where each bit of the byte represents a .025 increment Black or White (0 = White - 1 = Black). The laser beam will traverse 0 .LE X .LE 71199 while Y remains constant. The mapping of the data to the X-coordinate is as follows:

<u>BYTE</u>	<u>BITS</u>	<u>X</u>
0	0-7	0-7
1	0-7	8-15
⋮	⋮	⋮
8899	0-7	71192-71199

(Note that the mapping is low to high order bits).

RLC MT FORMAT: Each MIT RLC record will contain a Y-address, a Mode bit, the Word Count and the RLC data. The records are variable length. The following is the full specification of the RLC format.

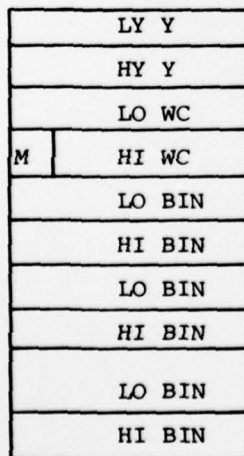
MT FRAMES



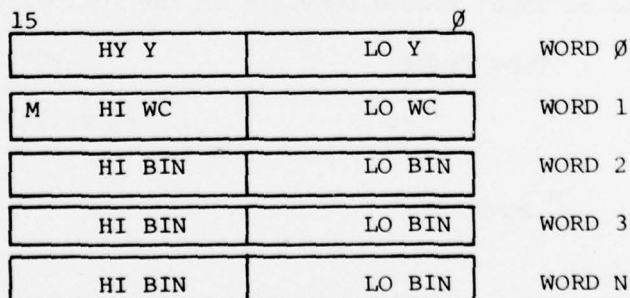
where HI Y and LO Y (bits 15-0) represent the Y-address, M represents Mode (0=RLC - 1=BIN), HI WC and LO WC (Bits 14-0) represent the word count (actual data words which follow the word count) and RLC represents a Run Length Code as specified. Note that although the records are variable in length, they are always an even number of bytes (8 bits). Also, the last two RLCs will be repeated along the X-coordinate from the X position that was produced by the previous RLC to the X position 71199 (end of active area). (Usually the last two RLC should each have a value of 127_{10} ; 127_{10} increments of white).

BIN MT FORMAT: Each MT BIN record will contain a Y-address, a Mode bit, the Word Count and the Binary data. The record are variable length. The following is a full specification of the BIN format:

MT FRAMES



WORD (PDP 11/40)



where HI Y and LO Y (Bits 15-0) represent the Y-address, M represents Mode ($\emptyset = \text{RLC} - 1 = \text{BIN}$), HI WC (Bits 14-0) represents the word count and HI BIN and LO BIN (bits 15-0) represent the binary image as specified. As in the RLC mode, the last two bytes are "repeater" bytes (all zeros).

MT RECORD ORDERING: There will be two magtapes being read by the plotter or being written by the scanner. Because of the two tape specification, one tape will contain data with even Y-addresses and one with odd Y-addresses. An even number is defined here as:

$$N = K * 2$$

where $\emptyset \leq K \leq 25399$.

An odd number is defined here as:

$$N = (K * 2) + 1$$

where $\emptyset \leq K \leq 25399$.

Note that $\emptyset \leq N \leq 50799$ since there are 50800 Y-addressable locations along the Y-coordinate.

Both the RLC Mode and BIN Mode records with Y-addresses of the form $N = K * 2$ as defined above will be contained on one set of tapes and those records with Y-addresses of the form $N = (K * 2) + 1$ as defined above will be contained on another set of tapes. Figures 2-24 and 2-25 illustrate the MT record structure.

The last data record K of each tape must be followed by either an "end of tape" or an "end of plot" record. The former occurs on those tapes which are not the last tapes to be plotted and the latter occurs on two tapes which are the last to be plotted.

The "end of tape" record consists of the following short record:

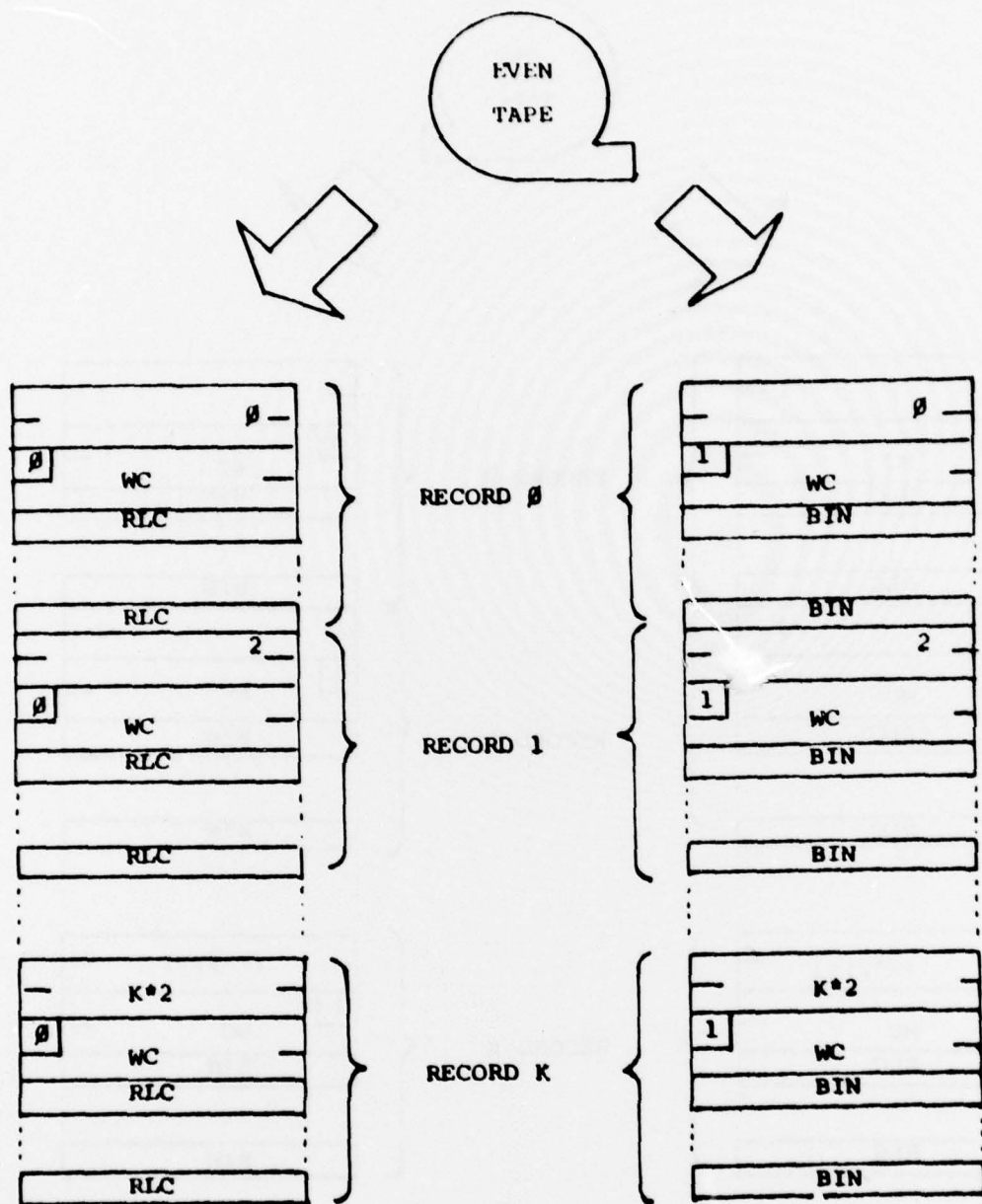
Tape Frames

PDP 11/40

-1

-1 = $(177777)_8$

^M WC= $\emptyset$

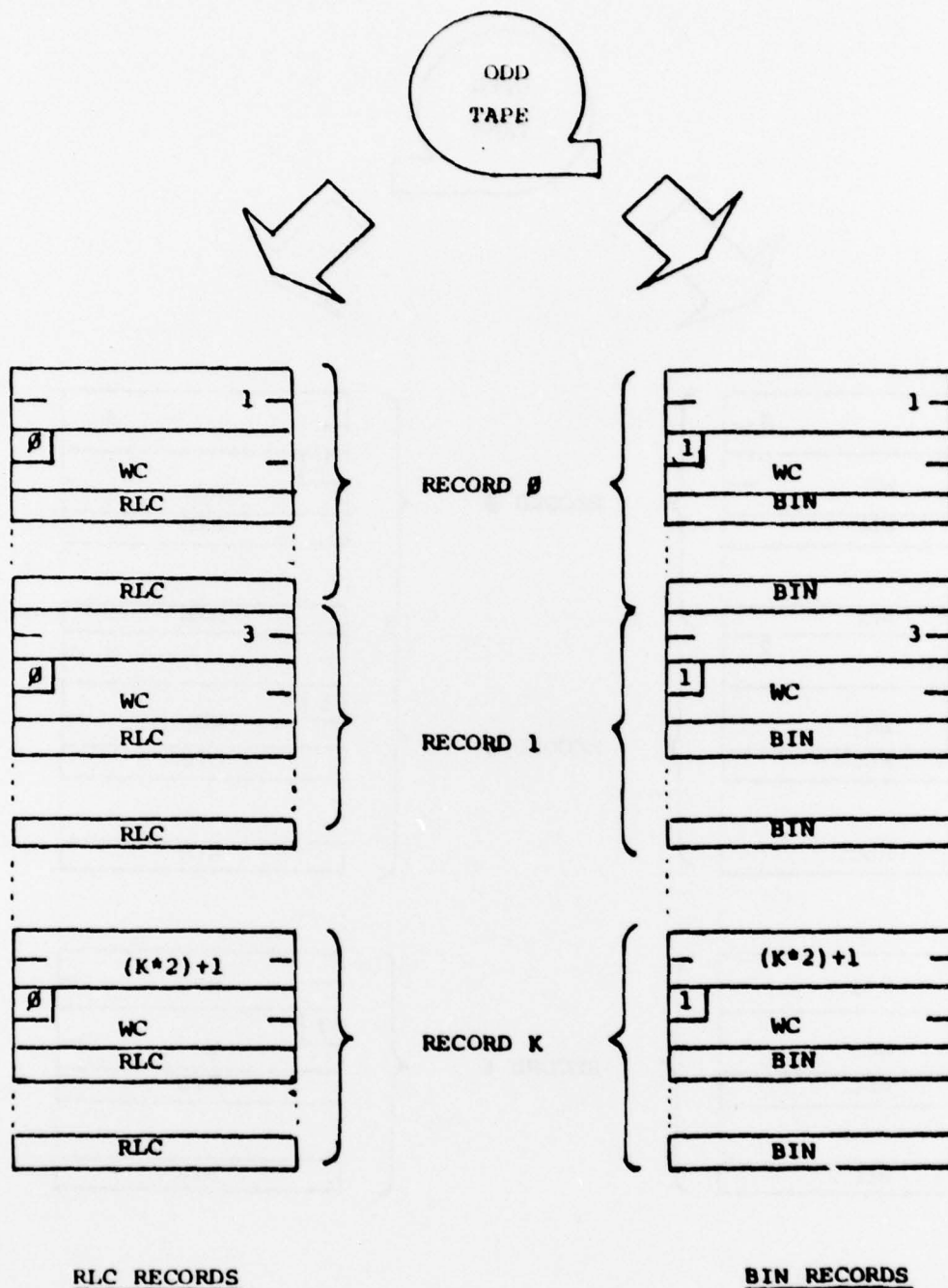


RLC RECORDS

BIN RECORDS

"EVEN" TAPE FORMAT

Figure No. 2-24



"ODD" TAPE FORMAT

Figure No. 2-25

The "end of plot" record consists of the following short record:

Tape Frames	PFP 11/40
-2	-2 = (1777776 _b)
M WC=0	

In both cases, M represents whether binary or RLC mode.

Scan line processing in this subroutine is described as follows:

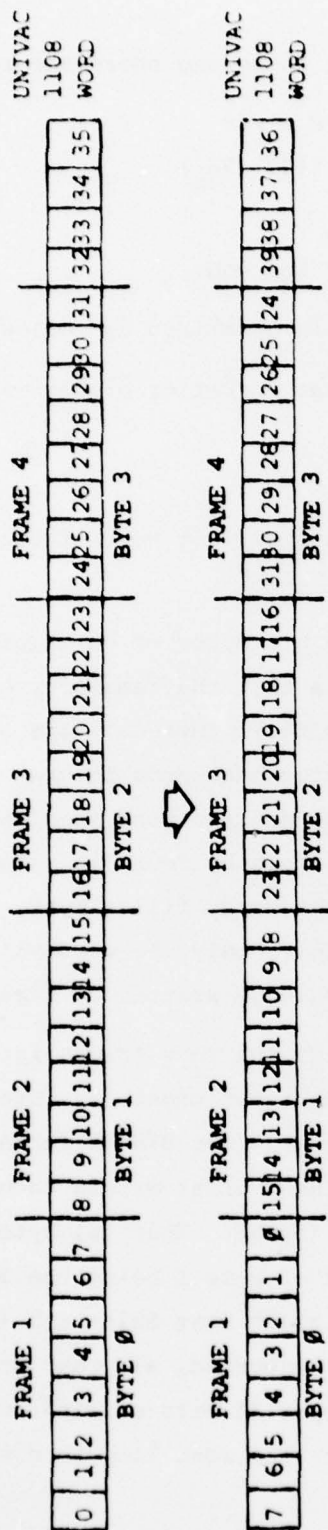
In the Binary mode of processing, no great amount of processing is required. The only two processing tasks are:

- (1) Repeater Byte processing
- (2) Y-address and Word Count setting in the first word of each scan line.
- (3) Inversion of all scan line bytes

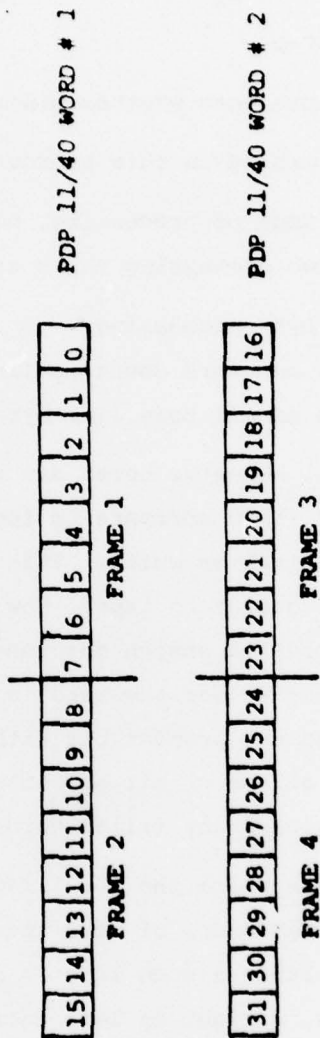
For the Binary case, repeater bytes are two 8 bit bytes of on or off bits utilized by the PDP 11/40 software to indicate that the remainder of the scan line is either all black or white. This results in physical tape saving since smaller records are output to tape. Now working backwards in scan line buffer (higher to lower words) a search for the first nonzero word results. At this point, depending upon whether the word is divisible by four (so as not to split PDP 11/40 words) and the present bit within the word, filler bytes (8 bits) are generated with all on or all off bits. Thus, only the word value of the on encountered bit, plus any filler words, would be written to tape.

Due to method in which the PDP 11/40 loads the tape frames into core, frames loaded in lower order of word and then higher order, an interchange of bits of each byte within a scan line is necessary (for BINARY format only). Via a look-up table, a right to left increasing X order within each byte (frame) is accomplished by the subroutine for Binary format. That is, byte 0 (frame 1) will contain bits 8-15 with right-most bit of byte 1 being the 8th bit, etc. Byte N will contain bits 8*N thru 8*N+7 with right-most being 8*N (see Figure No. 2-26). Once the input buffer is properly inverted, all that now remains is to place the scan line Y-address in the upper 16 bits of first word of scan line and the word count in following 16 bits. The scan line is now written to the even or odd Plotter Drive tape.

WORD #1



2-56



**BINARY FORMAT UNIVAC-1108 TO PDP 11/40
WORD CONVERSION**

Figure No. 2-26

In the case of Run length code format, a scan line must be placed in the RLC format by utilization of a Run-length code buffer. Counter variables are implemented in this phase of processing to count strings of on or off bit counts which are broken up if necessary, in maximum counts of 127 and then placed in the appropriate byte of words in Run length code buffer. The repeater bytes for this case are generated in a similar manner to the Binary format, except that in this case, the repeater bytes contain the lower 7 bits on and the upper bit off if the remainder of scan line is white, or all 8 bits (of two or more bytes) on if remainder of scan line is black. Since the 16 bit words of the PDP 11/40 and the 36 bit words of the UNIVAC 1108 are not evenly divisible, and 8 bit bytes Run length codes are utilized, every two words of the UNIVAC 1108 word scan lines contains one-half of a byte or frame. This byte is treated distinctly in that a bit count cannot simply be loaded, via the Field function, into an eight bit field. In this case, the appropriate bits are set by splitting the byte into half and placing these four bit pairs in the appropriate word locations. Word padding will result if Univac 1108 word count is not multiple of four; so that partial PDP 11/40 words are not obtained.

The selection of which of the two alternative output tape record formats is to be used for a given scan line is based solely on the spatial data density present. Figure No. 2-27 illustrates the relative efficiency of three types of data compaction schemes: binary image, 7-bit magnitude Run-length-Coding (RLC) and 15-bit magnitude RLC. (Note that the illustrated minimum 7-bit RLC record size of 561 actually only holds true for a scan line which is entirely blank except for a short run of black at a high X-coordinate value.) On the basis of a data density analysis, it was determined that the 15-bit RLC would be used relatively infrequently, since it is more efficient than the 7-bit RLC only for very low spatial densities, less than approximately 3 features per inch of scan line. For this reason, it was not proposed for use in the Raster Finishing Plotter. The efficiency breakpoint between 7-bit RLC and binary image falls at a data density of 63.5 features per inch of scan line, or a total of 4,450 feature crossings per scan line. Since it is conceivable that the total may, on occasion, be more than double that amount (theoretically, almost triple is possible), the binary image form will be used for maximum storage efficiency of all such very high density scan lines.

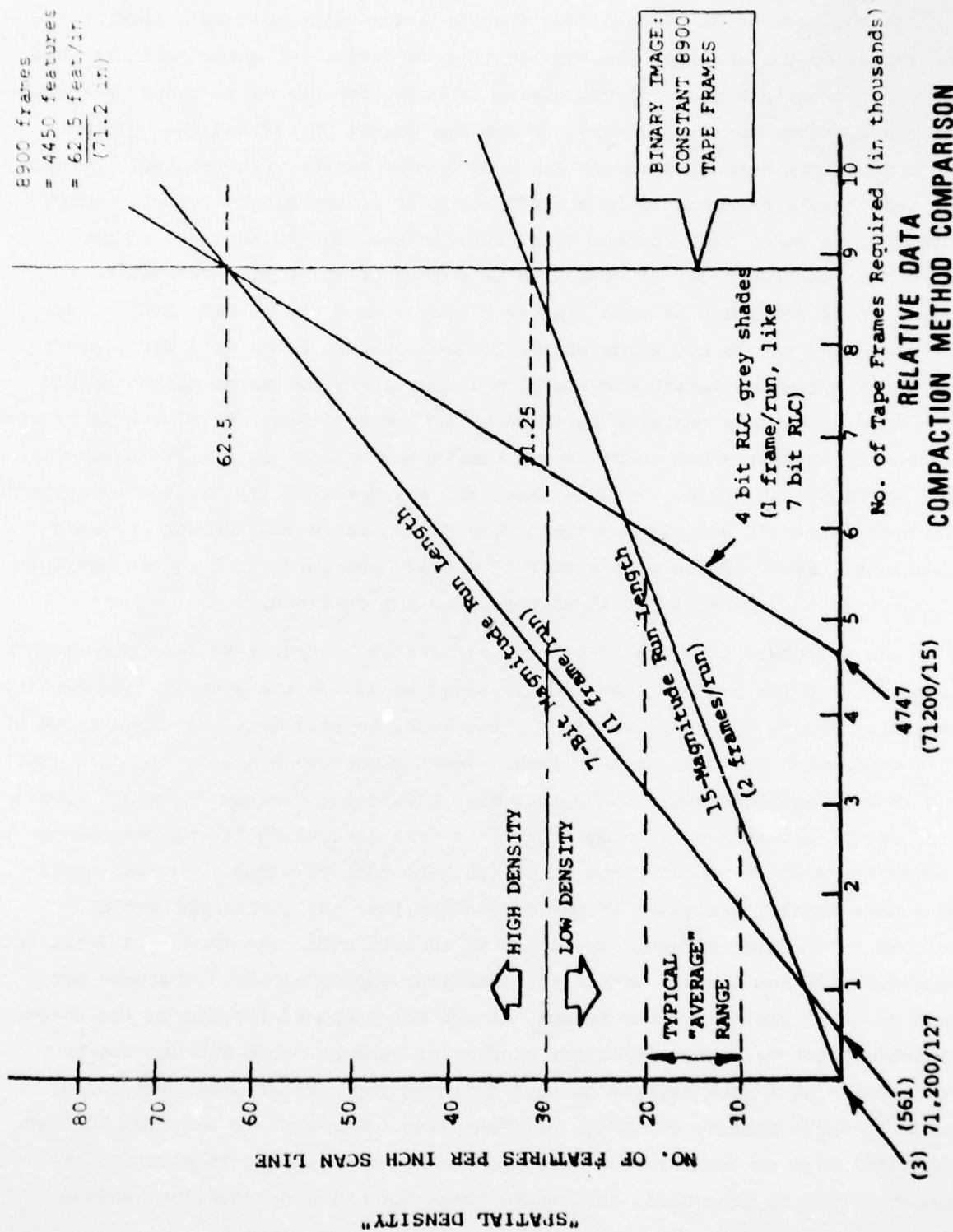


Figure No. 2-27

Taking the final binary image of each scan line completed, the Output Processor will proceed to develop the sequence of RLC words needed to represent that line, and pack them appropriately in a buffer area such that a subsequent tape write would result in each 8-bit value (7-bit RLC plus one bit on/off code) falling into separate frame ("tape characters") on a 9-channel tape. If a limit of approximately 4800 RLC codes (4800 B/W or feature transitions per scan line) will be accumulated before the end of the scan line has been reached, the efficiency breakpoint will be exceeded, and the binary option should be chosen. However, since the receiving buffer area planned for the plotter minicomputer controller is only 2,400 words (16 bits each), the binary image line will be split into two physical tape records. The plotter hardware will recognize the bit in the second control word which indicates a binary image plot line, and after completing the first record (one-half line), will automatically cycle one full revolution before proceeding to plot the second-half line. (Calculations show that the second-half record will not be completely into the minicomputer core at the time when the first-half plot is done. The extra revolution is needed to allow for completion of the reading of the second-half tape record.) Note that the spatial data density at which this "double cycle" plot of a given scan line is invoked is approximately double the level set in paragraph 4.1.3.1 of the statement of work.

Since a given plot run will commonly require more than one reel of tape (approximately 7 will be needed for an entire plot at a density of 30 features per inch of scan line or 4 for 15 features/inch), the output process will stop momentarily when physical end-of-tape is sensed during a tape write. The tape will be backspaced over the record which was in progress, and the record will be overwritten by a minimum length record containing a -1 as the Y address word. Tape writing will recommence with the aborted record whenever a new output tape is ready. Final end-of-plot (code -2 as the Y address) will also appear in a separate, short tape record following all valid data.

In conclusion, in order to meet the specification of plotting a full-format (127cm X 178cm) area within a one hour time limit, it will be necessary that all Plotter Drive Tapes produced will be at 1,600 bpi density (Phase Encoded).

SECTION 3
CONCLUSIONS AND SUGGESTIONS

3.0 Incorporated Time Reduction Methods

Three main methods were used to speed up processing time after program code isolation of excessive time consumption areas. These areas of program code were located by use of DMAAC supplied time function called ZTIME. As a result, the following steps were taken to reduce processing time.

- o Replacement of approximately 135 Fortran statements by a call to an assembly language subroutine for the conversion of internal binary-image buffers to Run-Length-Code form output. The assembly language subroutine, CBR, requires only 87 words of memory in contrast to approximately 450 words required by the Fortran generated machine code, and executes about 8 times faster on "typical" data collections (the differential in operating speed would widen as the spatial density of the imagery increases).
- o Many time-consuming FLD (field) function usages were replaced by other methods of performing the desired operations. The only remaining FLD uses are in sections of code which are executed relatively infrequently, such that the impact on operating speed is negligible.
- o It was found that the remaindering MOD (M,N) function generated much slower executing code when N is a constant than the equivalent construction,

$$M-N*(M/N)$$

or MOD function by definition. The MOD function always uses a 10.25 microsecond integer divide regardless of the value of N, whereas the latter construction uses a combination of multiply and shift instructions which execute about 3.5 times faster than the integer divide. Because of this discovery, all frequently executed MOD's where the second argument (N) is a constant, were replaced by the faster form.

A number of minor modifications of technique were also implemented whenever a time savings could be proven. All of these changes were implemented in both the full-width (178cm in X maximum) and half-width (89cm or 35.04 inches in X maximum) Output phase versions described previously in this text. The half-width version (which was not a contractual obligation and was not conceived until late in the effort) can be used with an appreciable time savings (70%) on small-format data collections regardless of the maximum lineweight present, and the full-width version will run 50% faster when no lineweights exceed 0.4mm (approximately 0.016 inches).

None of the above speed-up changes were implemented in the "all Fortran" versions of Phase III (FP3, FP3/HW, FP3/R2, FP3/R4) since the time savings gained by the last two improvements and miscellaneous changes are minor in comparison to the significant time improvement gained by the first change, the substitution of an assembly language subroutine for Run-Length-Code generation.

3.1 Time Reduction Recommendations

Due to time restraint of contract deadline and the focus of reduction of processing time for the entire Cartographic Test Standard plot as a whole, the following suggestions were not attempted. In addition, for some reason, the window utilized for Synectics final testing (window, rotation, etc. options) did not surface a time problem as was noticed at acceptance testing on a window of cartographic data of the CTS. Thus, this problem was not analyzed in the processing time reduction stage.

- o More time tests by use of ZTIME routine should be done to determine what remaining program code areas are consuming a great deal of time. Then once these code areas have been isolated, a determination of whether or not the program code may be time optimized.
- o Consideration of windowing of scan line words with respect to X-components to avoid unnecessary lineweight buffer word checking (NXCNT).

- o Maintain scan line buffer word pointer for last words of scan line set to reduce DO loop range.
- o Possibly try whole byte checking for all bits on condition within nonzero words in Run-Length-Code generation.
- o Mass storage I/O may be reduced to 1/10 the time used on the acceptance testing if the mass storage files required for lineweights greater than 0.016 inches are assigned to (or file space obtained) the FH-432 drum instead of the 8440 disc. The access and word transfer rates are considerably faster for the former versus the latter.