# REPORT ON
# CONVERSION OF NCDC MICROFILMS TO ELECTRONIC FILES
# WITH FOSDIC

M. Leighton Greenough

Greenough Data Associates
Rockville, MD 20850

Copy 2

# REPORT ON CONVERSION OF NCDC MICROFILMS TO ELECTRONIC FILES WITH FOSDIC

M. Leighton Greenough
Greenough Data Associates
Rockville, MD 20850

for

National Climatic Data Center
National Oceanographic and Atmospheric Administration
United States Department of Commerce

Prepared under P.O. A5129 with
The Orkand Corporation

October 30,1998

TD-9976 FOSDIC Rescued Data Reels

| Card Deck | Reel# | Missing Reel# | Tape# |
|---|---|---|---|
| 110 | 001-066 | | L50721 |
| 119 | 001-077 | | |
| 116 | 001-160 | | L50723 |
| | 161-320 | | L50852 |
| | 321-481 | 343 | L50855 |
| | 482-596 | | L50798 |
| 117 | 001-122 | | L50879 |
| | 123-244 | (change reel# 326 to 236) | L79798 |
| 118 | 001-145 | 119 | L79795 |
| 128 | 001-125 | | L79797 |
| | 126-250 | 187-207 | L60641 |
| 184 | 001-064 | 012 | L79796 |
| 185 | 001-010 | | |
| 187 | 001-002 | | |
| 189 | 001-023 | | |
| 190 | 001-004 | | |
| 195 | 001-054 | | |
| 196 | 001-016 | | |
| 192 | 001-130 | | L79801 |
| | 131-260 | | L79800 |
| | 261-390 | | L79802 |
| | 391-517 | | L79804 |
| 193 | 001-150 | | L79803 |
| | 151-300 | | L87420 |
| | 301-450 | | L87600 |
| | 451-538 | | L87113 |
| 194 | 001-170 | | L79805 |
| | 171-341 | | L79806 |
| 197 | 001-003 | | L90765 |
| 256 | 748-776 | | |
| 281 | 001-097 | | |
| 541 | 001-021 | | |
| 608 | 001 | | |
| 650 | 001-007 | | |
| 690 | 001 | | |

# Foreword and Reader's Guide

This report describes the project for converting data contained on approximately 3,000 reels of microfilm to a magnetic storage medium. These reels contain approximately 12,000 images each of tabulating cards punched with past weather observations. The equipment utilized was the author's FOSDIC, an instrument for scanning 16 mm roll films.

The report begins with a list of the specific decks and reels involved in the conversion in Section 1. Readers interested primarily in results of the process are referred to Section 4, where performance quality indicators are reviewed. The process itself is outlined in Section 2, a brief description of the equipment, and in Section 3 covering the scanning program in moderate detail. Section 5 on remnant actions lists remaining tasks relative to the scanner and to the sponsoring agency. Section 6 contains notes pertinent to the processing of each deck. More detailed notes relating to processing of individual reels are available from the author.

Appendix A provides a conversion table showing the recorded characters for punched data in the card columns. The remaining Appendices provide a listing of the source program components which make up the scanning program.

## Acknowlergement

While the author was responsible for the scanning program, the subject of the bulk of this report, acknowledgement must be made of the contributions of Nevell L. Greenough, who developed the associated programs for preparing the final data output.

Table of Contents

Table of Contents, cont.

Table of Contents, cont.

## List of Appendices

## 1.0 Introduction

This report covers a project undertaken by the author to convert archival microfilms of punched cards containing climatic data to modern computer form, using proprietary scanning equipment called FOSDIC. Some 3,000 rolls of 16mm film, or approximately 36 million cards, were processed. These films were made under special conditions to facilitate scanning by equuipment of this type.

### 1.0.1 List of films scanned

The films listed below were included in the project work statement. A few films were unavailable at the time of this report. These are marked with comment notes following the list.

| Deck | Description | | No. Reels Specified | No. Reels Received | No. Reels Scanned | Notes |
|---|---|---|---|---|---|---|
| CD-110 | Navy Marine | 1945-June 1951 | 66 | 66 | 66 | |
| CD-116 | Surface Marine | 1949-June 1963 | 596 | 596 | 596 | |
| CD-117 | Navy Marine | July 1951-June 1963 | 244 | 244 | 244 | |
| CD-118 | Japanese Marine | 1937-June 1953 | 145 | 144 | 144 | (1) |
| CD-119 | Japanese Marine | July 1953-1960 | 77 | 77 | 77 | |
| CD-128 | Surface Marine | July 1963-April 1968 | 210 | 229 | 229 | |
| CD-129 | Surface Marine | 1968 | 23 | 122 | 0 | (2) |
| CD-184 | British Marine | April 1953-1961 | 53 | 63 | 63 | (3) |
| CD-185 | USSR Marine | 1957-1958 | 10 | 10 | 10 | |
| CD-187 | Japanese Whaling | | 2 | 2 | 2 | |
| CD-189 | Dutch Marine | | 23 | 23 | 23 | |
| CD-190 | CG Log Book | | 4 | 4 | 4 | |
| CD-192 | German Marine | | 517 | 517 | 517 | |
| CD-193 | Dutch-Netherland | | 538 | 538 | 538 | |
| CD-194 | British Marine | | 341 | 339 | 339 | (4) |
| CD-195 | Navy Ship Logs | 1945-1945 | 54 | 54 | 54 | |
| CD-196 | German Marine | 1949-1953 | 16 | 16 | 16 | |
| CD-197 | Danish Marine | | 3 | 3 | 3 | |
| CD-256 | Surface Marine | 1963- | 72 | 29 | 29 | |
| CD-258 | Surface Marine | | 4 | 170 | 0 | (5) |
| CD-281 | Navy MAR | Beg. -1945 | 17 | 97 | 97 | |
| CD-541 | US RAOBS | 1941-1947 | 21 | 1 | 1 | (6) |
| CD-547 | Payerne RAOBS | 1942-1948 | 1 | 0 | 0 | (7) |
| CD-608 | Hungary RAOBS | 1954-1958 | 1 | 1 | 1 | |
| CD-650 | Formosan RAOBS | 1952-1960 | 7 | 7 | 7 | |
| CD-690 | Israel RAOBS | 1957-1961 | 1 | 1 | 1 | |

Totals as of date 10-30-98 .................................................... 3046 .......... 3061 .......... 3061
Totals after completion of Deck 194 .................................... 3046 .......... 3063 .......... 3063
Totals if missing reels received ............................................ 3046 .......... 3088 .......... 3088

Notes: (applicable 10-30-98)

(1) deck 118, reel 119 not recvd
(2) deck 129, deleted at later NCDC request
(3) deck 184, reel 012 not recvd
(4) deck 194, reels 062, 089 not recvd
(5) deck 258, deleted at later NCDC request
(6) deck 541, 20 reels not recvd
(7) deck 547, 1 reel not recvd

The missing reels wil be scanned when received

## 1.1 Background

The name FOSDIC is an acronym accurately describing its function, Film Optical Sensing Device for Input to Computers. Since the early 1950s ten models of these machines have been produced. Many, including the original , were designed for Census use. The first Fosdic for the Weather Bureau was developed at the Bureau of Standards in the middle 1950s. Subsequently two more machines were introduced. For use with Fosdic, extensive files of weather data contained on punched cards were microfilmed in a style readable by the machine. This greatly compressed the size of the files. The intended goal at the time was to perform data processing operations from the films, it being faster and more convenient than working directly with punched cards. However with advances in computers and magnetic storage techniques, and given the fragility of film, it became more practical to apply the machines to simply converting the data from the microfilms to computer language. By the end of the useful life of these machines, tens of millions of cards had been converted, still leaving many more inaccessible. This led to approaching the Census Bureau, which had remaining scanners from the 1990 operation. Feasibility was demonstrated but production was unavailable because of later machine decommissioning. Then construction of the author's scanner was completed. Feasibilty of Fosdic use for this application was again demonstrated and the present project established.

The project began after demonstration scanning of sample films obtained from the Climatic Data Center. A relatively straight-forward program for scanning these films had indicated overall feasibility. At initiation of the project, the program was expanded to take into account certain anticipated difficulties. Not surprisingly, other problems appeared as the work progressed. Modifications to the program became neccessary from time to time , resulting in a final version of appreciable sophistication. Among its important features is the ease of introducing custom modifications, often needed for successful processing of individual reels. It is the latest version which is the subject of the documentation included in this report.

Measures for monitoring quality evolved during the course of work. In particular, tests for comparison of the data from duplicate runs, required under the contract on only an initial sample, were eventually performed on every deck. Numerous other criteria for evaluating data quality were introduced as the work progressed.

## 1.2 Films of punched cards

The films of interest were made in several modified Recordak rotary microfilm cameras. This type of camera involves continuous synchronized movement of the film and the object, here the direction across the narrow dimension of the punched cards. A slot in the direction of movement, together with the velocity, combine to control the exposure time. With illumination in back of the cards, light reaches the film to produce exposure at the hole locations and between the cards. These areas then appear dense on the (negative) films. Original films so produced are the ones scanned. In a few cases only copy films of the same polarity were available.

## 1.3 Camera modifications

The commercial cameras were modified in several respects for this application. First, the normal lens was supplemented by adding an attachment known as an anamorphic lens. This lens has magnification of close to 2X in one axis and none in the other, perpendicular axis. It is the basic element in the Cinemascope process for motion picture films. It is here used to double the number of cards on a roll of film, allowing over 12,000 cards on a 100-foot film roll. Of additional benefit is the fact that it makes the rectangular punched holes more nearly square, for a better match to the round scanning beam. With the anamorphic lens, the camera reduction ratio is approximately 24X across and 45X along the film.

The second modification was to add a scale made up of 81 bars whose thickness and location match the card stock web space between columns of punched data. These bars are parallel to film movement. The original purpose of the scale was to allow dependable access to each column despite the limitations of then available digital-to-analog converters. Such restrictions are no longer applicable but the outside scale bars remain beneficial for registration purposes. With this scale placed in close contact with the cards in the filming aperture, light can only reach the film in the spaces between the bars. Properly registered cards then show as full-width hole patterns in the overall card image.

Figure 1-1 on the next page is a drawing showing details of typical films. Overall, the effect on aspect ratio due to the anamorphic lens is evident in the lesser than normal short dimension of punched cards and the nearly square holes. The card stock itself, being relatively opaque to the illumination, appears clear or nearly so. Around the card and through the punched holes, the film is dense. The scale bars, as mentioned above, block light to produce clear lines at their positions. Finally there is a clear card bar at beginning of every card location. This bar appears even during card misfeeds.

The Card Bar appears at nominally regular intervals along the film. It is not always distinguishable, particularly in the leader sections of the film, in misfeeds or when the camera stops. Typically along a roll there are many visible card bar images not accompanied by cards. Nevertheless every readable card must have a clearly legible, well positioned Card Bar. Its proper position is with its left side coinciding with the left side of the card image as shown in Figure 1-1.

In the course of scanning, numerous departures from the ideal conditions shown in Figure 1-1 arose. Some expected and unexpected conditions are next covered along with the remedial measures it was found neccessary to include in the scanning program.

### 1.4 Variability in films

This section describes the properties of the films with which a successful scanning routine must cope. Important variables related to films are 1) overall condition, 2) density and 3) variation in image geometry. To these potential problems, the punched cards contribute those of 4) punch misregistration, 5) expansion or contraction of card size, 6) color of card stock and 7) excessive clipping of corners. The discussions which follow cover some anticipated concerns, experiences in the course of scanning and such remedial measures as could be worked out.

#### 1.4.1 Film condition

These are old films, some as much as forty years. There was an initial apprehension

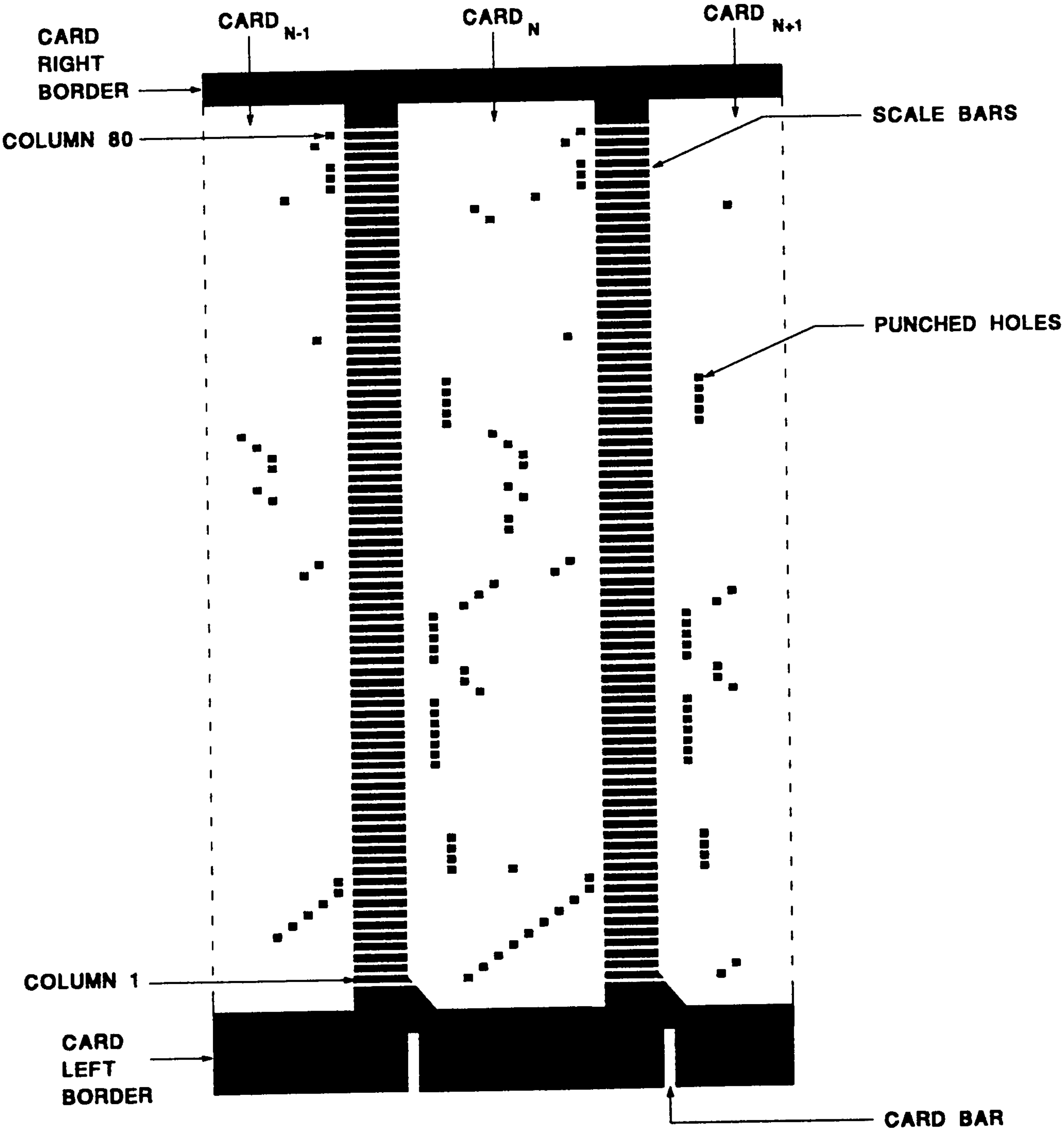


FIGURE 1-1 CARD IMAGES ON NCDC MICROFILMS

about their physical condition, with regard to stickiness and fragility. The former never became evident. Fragility is an issue because high speed film transport is essential for efficiency in throughput. Film movement for instance accounted here for 40 percent of roll processing time. Despite applied cushioning in acceleration and deceleration, the rapid start and stop motion can result in film breakage; this happened in the trailer end of two films.

In microfilms of this age, blemishes in the form of small spots are not uncommon. These original films are negatives, with the result that their opacity simulates that of punched holes. The artificial data so created depends of course upon specific locations. Sometimes the result is an asterisk indicating an illegal punched character, creating data loss rather than substitution. The most effective measure against blemishes is to identify the strongest signal in a numeric or the two strongest in an expected alphabetic, column. Then any blemishes weaker than punching drop out. As is mentioned later in the discussion of quality control, an excessive count of asterisks, when followed up by review scanning, is an important tool for evaluating blemishes. The general conclusion at the end of the project is that, while some were present, the contribution to errors was not major.

Blemishes at certain other locations may produce apparent departures from true rectangularity in the card images. Such distortions are detectable so that erroneous information from this cause can be screened out.

### 1.4.2 Density variation

At project initiation it was anticipated that appreciable variation in overall film density (logarithm of inverse transmittance) might be encountered. This turned out to be an unfounded concern. Some elaborate measures for handling density variation were developed but later discarded as unneccessary complexities. The few occasions of overly dense or light reels were treated with ad hoc customizations.

Some reels were of cards filmed by both transmitted and reflected light. The latter represented an attempt to preserve printed information on the card face. Printing of this nature reduces the difference between the hole signals and the background, complicating the process of thresholding between hole and background signals. For example the double illumination may darken the background area surrounding holes to less than half of that obtained with transmitted light. Considerable experimentation was performed to deal with this factor. The final program contains provisions for automatic recognition of this variable density in the background card stock.

An unexpected density-related problem in the scale zone became noticeable on certain cameras. The common evidence was that card misfeeds caused the camera to decelerate and stop before reducing the illumination. This produced overexposure in the scale bar area immediately following card images. Then the resultant unsuccessful search for scale bars created calibration failures. This was one of the most common causes of such failures.

### 1.4.3 Geometric variation

Microfilm cameras of nominally the same type can be expected to show minor differences in reduction ratio; a typical tolerance is in the order of three percent. Adding the anamorphic attachment increases the possible longitudinal variation (the short card dimension) to possibly five percent. These are steady state conditions fixed in each camera. Automatic size calibration in the Fosdic program finds four corner locations on every card, computing from them the locations of the 960 hole coordinates. This produces 79 equal intervals from columns 1 to 80 and 12 from the top of the card to the "9" row.

Aside from this fixed geometry, variations arise. A possibility is that cards may enter the filming aperture on a slant. Particularly the first few cards filmed from a stack may show a slant of up to one-half degree, approximately the height of a punched hole. The clue to the presence of such slant is the difference in the measured horizontal (in scanner coordinates) locations at columns 1 and 80 on a card. This difference must be scaled accordingly for every column and applied as a correction to its 12 computed hole locations. There is also a vertical (in scanner orientation) shift accompanying any slant; its magnitude is considerably smaller. Correction for this was deemed not worthwhile in view of its complexity and cost in computing time.

A surprisingly common variation in geometric fidelity is evident on the films. In card terms, rows appear out of place, being either above or below their nominal positions. This displacement can reach as much as one-half of the height of a hole. It may possibly be due to fluctuating synchronism between motion of the cards and the film during the time of exposure due to backlash in gearing, or else simply misregistered row punching. The result is the same, that is, dividing the top-to-bottom dimension of a card image by 13 intervals does not always define the row location on an image. The displacement is least in the upper and lower rows and greatest in the central ones.

Another observed phenomenon is that several reels showed a variation equivalent to a photographic stretching of the narrow card dimension, cycling as much as 10 percent with a periodicity of perhaps 30 cards. Watching the viewing screen during processing gave the impression of "breathing" in size.

Finally the most frequent problem observed on the films was due to lateral (columnar) misregistration of the cards with respect to the scale bars during filming. Improper registration, either at the time of punching or of filming, produces partially-occluded, narrowed images of the holes. These create problems for scanning First, the resultant loss in apparent hole size translates into signal reduction. Besides this, the residual hole images are inherently off-center from the normal position because the occlusion occurs from one or the other side of the hole images.

As is described in some detail in regard to the scanning program, the remedial measure for either type of misregistration is to search for the best response in multiple adjacent positions around the nominal hole center. Misregistered holes have only partial coverage by the scanning spot, yielding weakened signals. In the adopted scanning program a fractional response at any nominal hole center hole triggers five rescans of the entire card at modified positions and sensitivity. This greatly reduces the number of missed holes. At the same time, the loss in overall throughput is minimal because the rescan takes place only when neccessary. Typically this adds a few percent to the running time for reels.

1.5 (begins on next page)

1.5 Variability in punched cards

Misregistered punching is of course the principal concern with regard to introduced problems. It can take several forms. Oversized cards, brought about by high humidity in storage before filming, mostly have shown good matching at the column-80 side but with "thin" holes at the opposite side of the cards. Earlier correction techniques, such as shifting the reading positions leftward for the lower-numbered columns, showed considerable benefit, as indicated by fewer dropped punches in designated column groups. It was the method of choice for many initial runs. It has the disadvantage of being time consuming to execute, along with introducing a minor prejudice for properly located punching.

In many instances individual cards showed a combination of zones of excellent and bad hole images. Again the card left side was the location of most occurrences. This suggests that the punching took place in more than one machine.

The color of card stock has some influence on image density. Semi-transparent material should produce denser background, leaving less difference between background and hole signals. The same image condition also appears as a result of filming by reflected light, as was evident on several dozen reels. For this reason the signal thresholding formula in the scanning program includes background density in the computation, performed individually for every column.

Excessively trimmed corners on possibly a hundred cards altogether resulted in calibration failures. On occasion the clipping at top or bottom extended into the cards as much as ten columns. It was possible to observe these incidents by calling for a programmed halt on such failures. For these there was no countermeasure beyond program modification to tolerate clipping for up to three columns.

The final version of the scanning program, with its provision for complete card rescan on any doubtful signal, proved to be the best method for handling the problems brought about by card misregistration in either punching or filming. In the rescan procedure, the examination for holes reads with a small displacement on each side of the nominal center, also above and below, as well as at the center, all with elevated sensitivity. This technique eliminated the need for a great deal of the run-time program customization.

## 2.0 Summary description of Fosdic

All Fosdics have had in common the basic principle of flying spot scanning of film images, using a cathode ray tube which is directed by computer programs. In this arrangement the screen of the Crt is focused by a lens onto the film image and the transmitted light collected by a phototube. Any desired location on the image can then be probed by suitable positioning of the Crt spot by hardware which converts digital instructions to analog deflection. Interactive scanning allows determination of line centers or edge positions. This arrangement is in contrast to conventional systems wherein an entire image is scanned, stored in memory and analyzed. In a Fosdic the photographic image becomes the memory. Increased throughput can therefore result from confining the scanner attention solely to the areas of interest.

One of the useful characteristics of later Fosdics is that in addition to being positionally programmable, the size of the scanning spot is also controllable by the computer. As applied to scanning of images of punched cards, it allows aggregation of film transparency over most of a hole area with a single reading. The distribution of the enlarged disk-like spot is uniform, making multiple readings unneccessary for clustering. Such a characteristic minimizes effects from scratches on the film. Independently of the size of the spot, its incremental movement can be a small fraction of hole dimensions when desired. As is described later these abilities are fully utilized in the scanning program.

Aside from its speed, another detail of the present Fosdic is useful for this service. A reverse illumination system superimposed on the scanning optical path projects a bright 16X enlarged image on a viewing screen showing the film within the scannable aperture. From this enlarged portrayal, it is readily possible to identify the punched holes on the cards by using a simple paper scale. This facilitates comparing optical and electronic records for evaluating the accuracy of data conversion. In addition to the enlarged projection, a smaller image appears on the scanning Crt face. Registration with the scanning pattern is exact because both the projection and scanning beams pass through the same lens. With an 8X close-focus telescope it is possible to evaluate visually the precision of the scanning pattern with respect to the film image.

Scanning takes place while the film is stationary, usually requiring approximately one-twentieth of a second for eight cards. As a result the systems is capable of processing about 5500 cards per minute or slightly over 90 per second.

## 2.1 Components of Fosdic

The latest version of Fosdic is that constructed by the author and shown in Figure 2-1 through 2-4 beginning after the next page. It occupies two cabinets, one containing the computer and power supply, the other comprising the optical and electronic systems. The computer, a 486 DX-66, is at the left in a mini-tower case. Above it is the Iomega Zip disk drive, in which the final output of the project is recorded. Underneath the computer monitor is the scanner control box with power switches and panel counter showing card counts. At the top right in Figure 2-1 is the viewing screen which displays the 16X magnified film image. Film travels from left to right through the film gate where the scanning takes place. Windows on each side of the film gate allow observation of the Crt face with the telescope.

Figure 2-2 is a front-on picture of an image on the viewing screen. The next illustration, Figure 2-3, is a closer photograph of the film-handling panel, showing the windows for Crt observation and the telescope in its adjustable mount. Punched holes are visible as dark spots in the card images. Finally Figure 2-4 shows the Crt face itself with its smaller projected image, about 3.5X the size of the scanned film. As mentioned earlier, this projected image on the Crt and the scanning spot are always in precise registry with each other.

The film transport system presents images to the scanning head in an asynchronous manner, starting and stopping upon command. A motor-driven capstan pulls film through the gate, moving it about three-quarters of an inch at a time. Idler arms in the film path actuate motors on the supply and takeup reels. Small velvet pads at the entrance to the film gate act to remove any extraneous material on the film. Film advance is rapid, typically one-thirtieth of a second for eight card images. The present Fosdic has the ability to examine an area of about one inch of film length, which is equivalent to 10 or 11 cards.

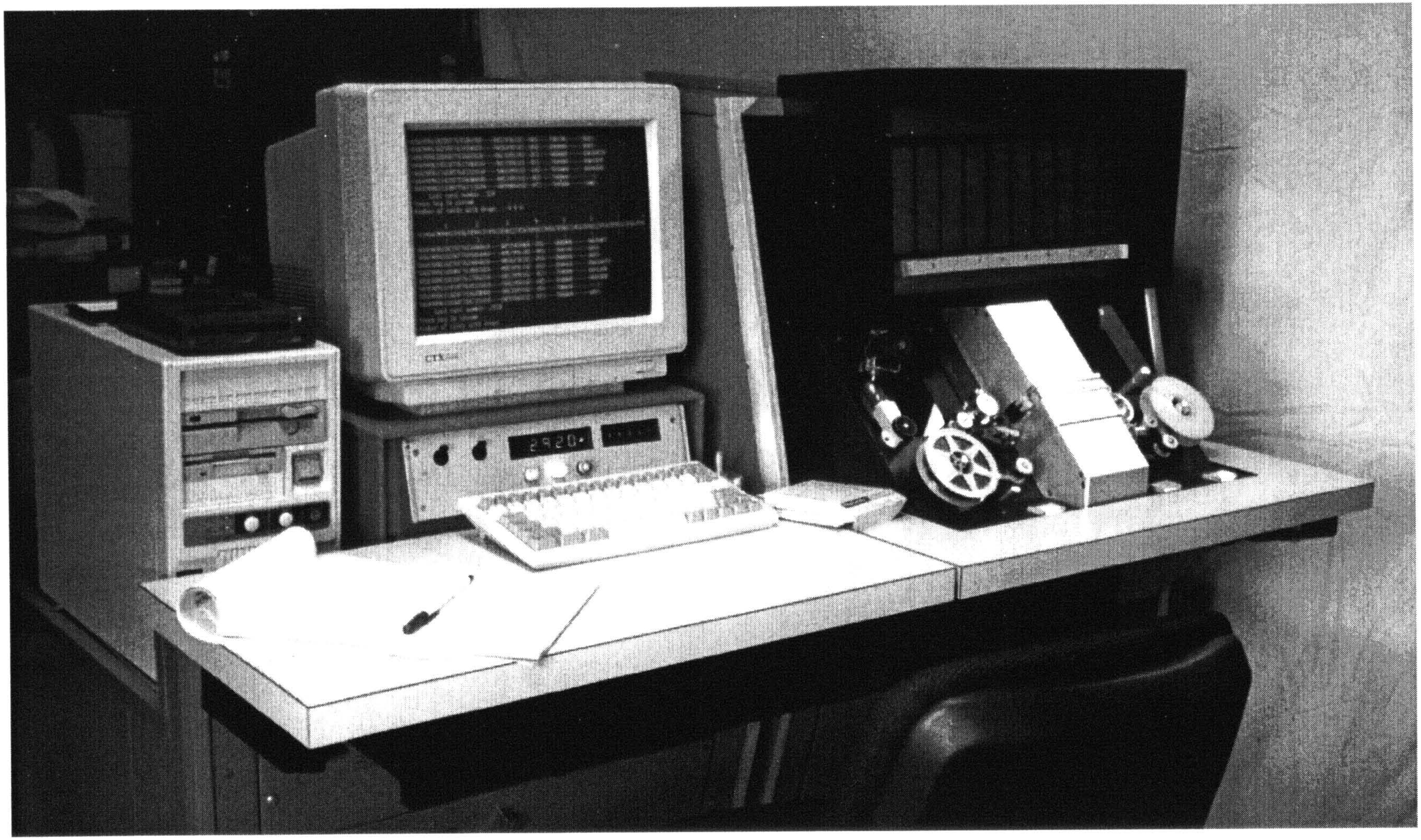

FIGURE 2-1 FOSDIC AT GREENOUGH DATA ASSOCIATES

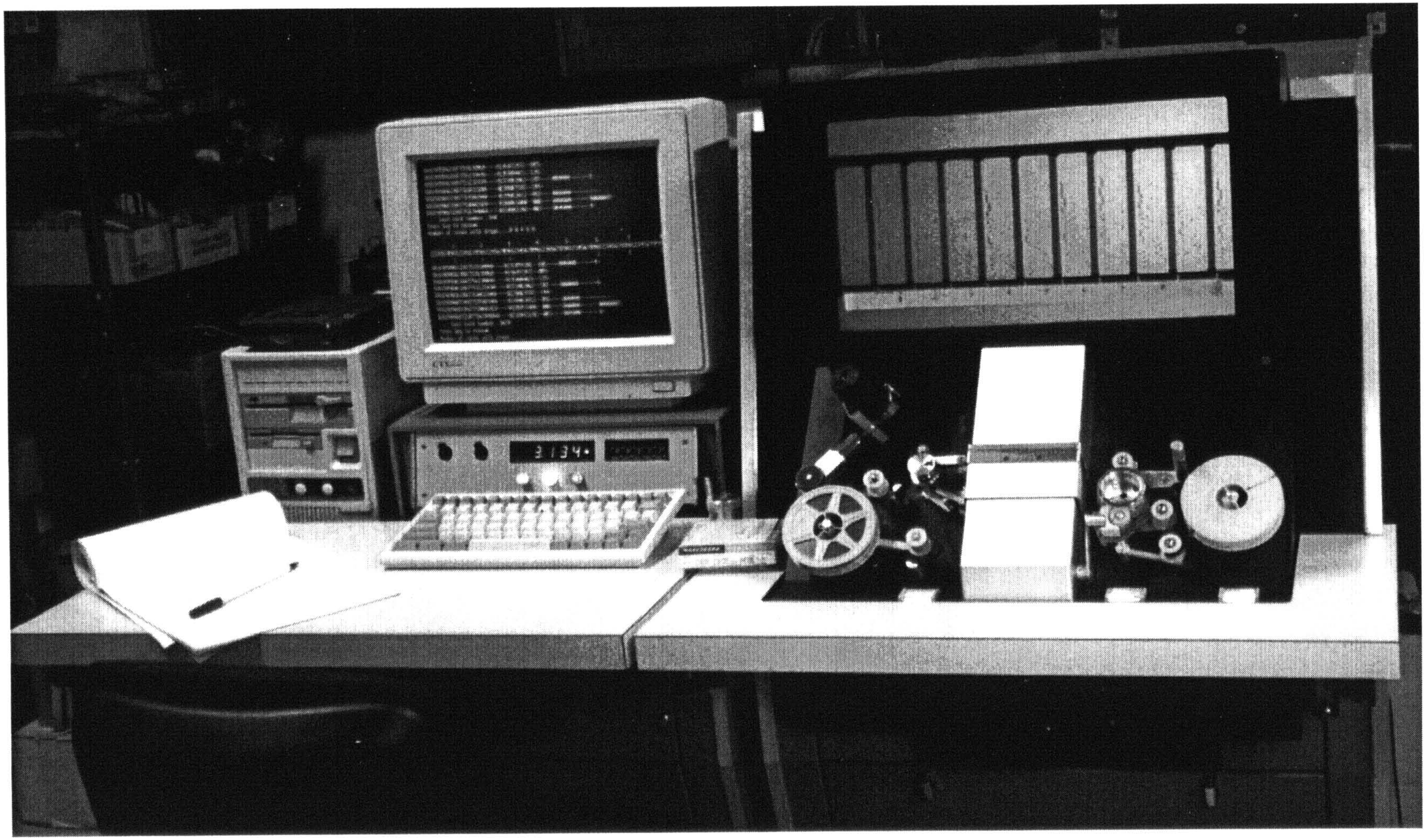

FIGURE 2-2   VIEWING SCREEN SHOWING IMAGES OF NCDC MICROFILMS

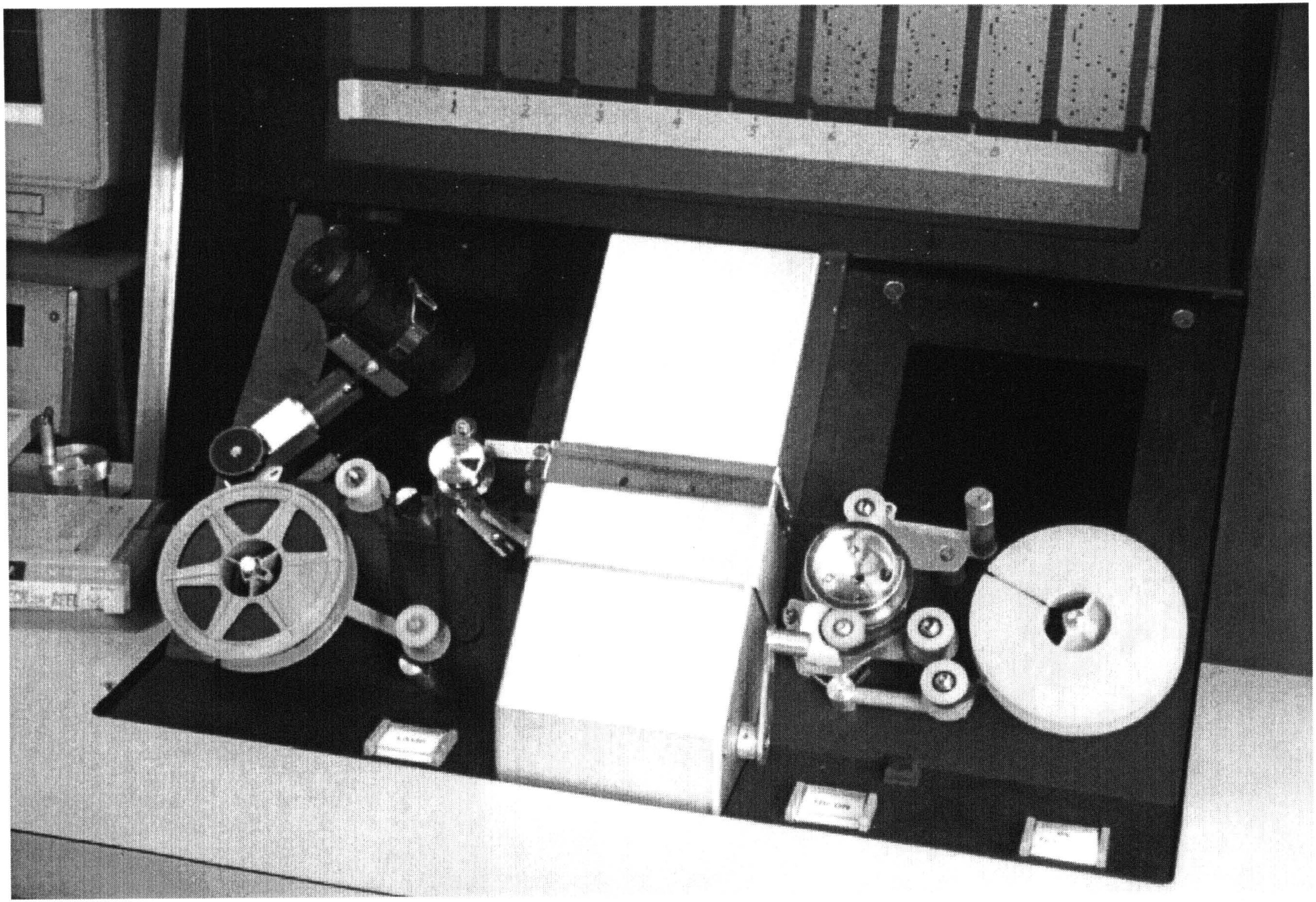

FIGURE 2-3 FOSDIC FILM PANEL

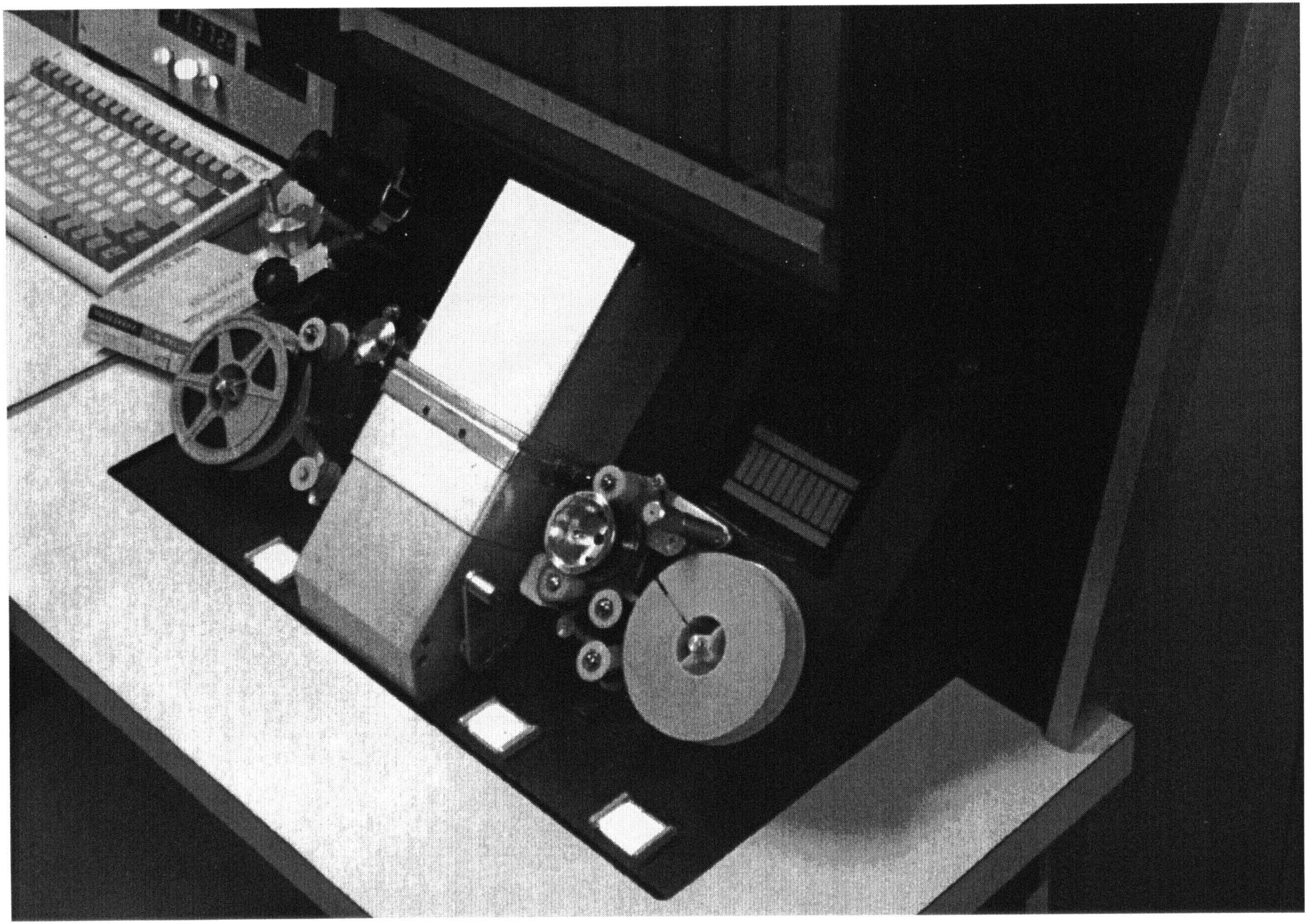

FIGURE 2-4 PROJECTED FILM IMAGE ON FACE OF CATHODE RAY TUBE

## 3.0 Scanning program

All of the operations in connection with Fosdic are conducted in the personal computer environment, where the basic operating system is Microsoft DOS, for Disk Operating System. The scanning program has three major components, whose names and actions are:

LDR.EXE
: Run in the film leader, accept typed identitying information for the reel, measure calibration parameters for the reel and generate a Param file witth the combined information

PCSCAN.EXE
: Perform actual scanning of the approximately 12,000 punched cards on the reel, recording the output data in 80-byte strings in a large file called Datout

TLR.EXE
: Generate a Trailer file containing processing and quality monitoring summaries

These separate programs execute as called for in a short batch file with the name GO.BAT. The Appendices furnish a detailed copy of the source programs in assembler language together with extensive commentary. However an understanding of the language is not required for the present discussion of program actions. The immediate next sections take up the functions performed by the three programs above.

At the end of scanning, additional processing routines combine the three files from above into the primary output data file for the reel and some secondary log files. These routines are the subject for later description in Section 3.5 and 3.6.

## 3.1 Leader program LDR.EXE

Once a reel of film is installed on the optical panel, action begins when the operator types the characters GO and presses the Enter key. This causes the batch file to start executing through its list of programs which constitute the overall processing system.

The batch file first issues an instruction to disable caching of file recording in order to guarantee later that the temporary RAM memory containing scanned data for a group of 640 cards is fully recorded on the computer hard disk before more scanning occurs.

The leader source program is LDR.ASM, expressed in mnemonics and subsequently assembled to direct machine instructions. It calls for several subroutines in the course of execution; these will be introduced as the description progresses.

Fosdic reads entirely by enabling (turning on) the Crt beam, creating a spot of light at a position on the Crt face determined by the previouly-supplied horizontal and vertical coordinates. The active period of a few microseconds is followed by programmed movement to the next desired position. The time for repositioning is dependent upon the magnitude of positional shift. For the small change between rows in a column, this inactive time is less than the active beam-on time. Thus with exclusively spot reading, this Fosdic differs from previous models where stroke scanning was the operating mode.

3.1.1 FRAME subroutine

To begin, a procedure (subroutine) called FRAME turns on the Crt beam, generating a vertical column of dots near the left center of the scannable aperture. The film drive capstan also begins movement. Because this is at the start of a reel, the film image is of the scale bars, substantially opaque. The scan pattern effectively waits until the first card appears as indicated by light through the relatively clear card image. When this information reaches the computer, it stops the FRAME routine, disabling the Crt beam and telling the capstan to assume its holding mode. After a pause of a few milliseconds for deceleration and settling, the system is ready for the next action.

The first image should be that of an identification card which has punching in the first three columns to designate the deck number and in the last half of the card, a large visual representation of the reel number. No scanning takes place on this card. In the viewing screen the identification card appears near the left side.

3.1.2 WIPE

After FRAME a series of calibration procedures are performed. First there is WIPE, which drives a full raster across the Crt face to activate the phosphorescent screen. This is to counteract a fading phenomenon associated with this type of very fast phosphor.

3.1.3 GAIN and BKGND

Next come two calibration subroutines GAIN and BKGND (for Background). The former measures the film transmittance in the nominally clear, unexposed film outside of the card area. Its purpose is to compensate for the maximum transmittance of the film, which depends upon the film type and development conditions. It then adjusts the overall optical sensitivity of the system for constant signal amplitude. The system thereafter responds in a uniform manner regardless of whether the film stock is very clear or partially opaque. Images from cards then always produce signals less than the reference level target of 980 units resulting from Gain calibration. It may be mentioned here that signals from the punched holes, which are almost opaque, typically range from 80 to 200 on the same scale.

The Background subroutine attempts to determine the general level of signals from the two areas, the card stock and the dense border near the Card Bar and column 1. On account of several factors, including varying illumination in the exposure areas in cameras, the former measurement is of questionable usefulness. It was eventually supplanted by a background reading in every column. The dense border information is obtained from multi-spot readings spanning the border. It shows the lowest signal which can be expected on the film, typically around 60 units. Both GAIN and BKGND utilize weighting schemes to minimize effects from film blemishes. The parameters obtained from these subroutines appear in a file called Param.

3.1.4 CAL DSPL

Results from the aforementioned calibration appear in a display on the computer monitor for operator evaluation with respect to normalcy. The subroutine called CAL_DSPL performs this function.

### 3.1.5 ENTRY

The last subroutine in the leader program is ENTRY. It begins execution with waiting for operator typing of Deck and Reel identification numbers. The system accepts minimal typing in not requiring leading zeroes. In any case, ENTRY places the deck number into the Param file as it is typed, with or without leading zeroes. Any lower case alphabetic characters are automatically capitalized by the subroutine for consistency in display.

The leader program LDR ends by recording the 100-byte Param file, which also contains other information on processing date and start time. All of this is specific to the particular run; the file is written over at the start of the next reel.

## 3.2 Punch card scan program PCSCAN

This and its subroutines form the main part of the scanning system. It has two major components, SCAN which reads the data from eight cards and FRAME which moves the film to the next group of eight. SCAN itself calls other subroutines, all of which return directly to the caller without further branching. FRAME is the same subroutine mentioned before as used by the LDR program. While some of the subroutines execute only once in reel processing, as in the leader program, many are set up for multiple use on every card. Again the subroutines are brought into the discussion here as needed.

Before calling for SCAN to start reading card data, certain preparatory actions are in order. PCSCAN begins with reading the Param file to extract the contained parametric values and typed identification data mentioned above. The first subroutine called is JUSTIFY, which aligns the typed deck and reel numbers into the DOS convention of eight-digit name and three-digit extension.

### 3.2.1 JUSTIFY

This subroutine accepts up to eight characters for reel number, aligning them to the right and supplying leading zeroes to make up a full eight-digit name. It does the same for the reel number, placing the filled reel number in the extension field. It also adds a separating period digit, thereby forming a message which later becomes the name of the file recorded for each reel.

### 3.2.2 Return to PCSCAN, the MSK file

The next action is to read in an auxiliary mask file carrying information specific to the deck regarding the expected type of punched data in each of the 80 columns. This essentially is an interpretation of the column-by-column information in the Reference Manuals. Two types of encoding are accessible in the present scanning program, single digit and alphanumeric with overpunch. Switching between these is the function of the mask file.

Before considering the mask file in detail, it is worthwhile to review the available possibilities. To start, the Fosdic program reads the 12 row positions in every column and sorts the results in order of the three strongest signals above a threshold. The benefits of restricting data coding to the single strongest signal in blemish tolerance have already been touched upon.

Thus encoding for the single strongest response is desirable whenever applicable. The other encoding style accommodates overpunching in the 'Y' (12) row, 'X' (11) row, and a single punch in the numeric '0' through '9' rows. By rendering all of these, it guarantees proper interpretation of valid punched data in the cards. It would be the universal style of choice except that experience has shown increased pickups (false holes) from film blemishes.

The purpose of the .MSK file is to specify to the scanning program those columns requiring alphabetic/overpunch coding. Then the remaining columns can have the benefits of blemish tolerance provided by recording the single strongest digit. To implement this function, a special 80-character mask file is created beforehand for every deck. These files are named with the deck number and .MSK as the extension. Deck identification for selecting the desired mask file is derived from the typed data in the Param file.

Initially in the project activity this approach was applied rigorously until a significant problem became apparent. Many instances were observed where multiple punching occurred in columns other than those designated in the Reference Manuals. Often the extra (usually double) punching appeared in adjacent columns. To take care of these cases as well as possible misunderstanding of instructions in the Manuals, a "learning" approach was adopted. Its introduction is considered an important step in evolution of the scanning program.

Learning applies to columns not already designated as needing encoding for alphanumeric or overpunch data. In double-punched but not so specified columns, an arbitrary number of single reads is permitted, after which the column becomes converted to the full encoding. It remains in that pattern for the remainder of the reel. In effect a potential for substitution is traded for accurate reproduction of multiple punches when shown to be needed. At the same time maximum protection against blemishes is retained as long as feasible. In the adopted imlementation, the second instance of double punching in a column converts its encoding for the rest of the reel to the full alphanumeric/overpunched form.

In setting up the learning approach, the 80-character mask file contains either a '2' for two single read encodings or a '0' for multiple punch interpretation. The file is first loaded with '0' for those columns known to require full encoding. The remaining columns are filled with '2' or other values. Then if unexpected double punching is found in a scanning a column, the appropriate character is decremented by one. When it reaches '0', decrementation is stopped and column interpretation is kept as alphanumeric/overpunch until the end of the reel. A list of mask file contents for all decks is included in Appendix J-2.

At this stage PCSCAN calls for the .MSK file with the appropriate deck name carried by Param and transfers its contents to a memory section for later reference. After some housekeeping actions, PCSCAN calls the subroutine SCAN which controls the actual scanning of up to eight cards. On the subsequent return from SCAN, a loaded data file contains the punched information along with status indicators and memory pointer.

3.3 <u>SCAN program</u>

This is the subroutine responsible for control of other subroutines which perform the actual work of searching and extracting data from the card images. One of its primary actions is to supply parameters to the called functions. Examples are geometric coordinates, directional indicators and signal thresholds. All of the called functions return to SCAN which when completed goes back to PCSCAN.

### 3.3.1 Coordinate scales

The system has three basic axes, horizontal or X, vertical or Y, and tilt or skew. In the first two the units of movement are expressed in mils, one thousandth of an inch, with a range of 0 to 32,767. This scale comes from digital-to-analog converters of 13 bits. In this scale the center is 16,384; deflection of the Crt beam to this value in both axes brings the spot to the center of the scannable aperture. The scannable aperture extends from approximately 5,000 to 27,000 mils horizontally and 9,000 to 23,000 mils vertically. The polarity of movement is that increasing coordinate values cause rightward and downward motion on the Crt and viewing screens.

This scale factor matches a microfilm reduction ratio of 24X very closely, that is, the least programmable unit of one bit produces a movement on the filmed original of one mil. On this basis the dimensions of a card are:

| | |
|---|---|
| Card width: | 7,375 mils nominal<br>Movement in this direction is in the Fosdic vertical (Y) axis |
| Card height: | 1750 mils nominal<br>Horizontal (X) movement |
| Hole width: | 58 mils nominal in Y axis |
| Hole height: | 63 mils nominal in X axis |
| Scale length: | 430 mils approximately in X |

Because the card images are oriented transversely on the film, horizontal movement in Fosdic corresponds to row-to-row excursion in the card images. Increasing horizontal coordinate values therefore proceed down a card. Vertically the opposite is true; higher values are associated with the left side of cards or lower column numbers.

Fosdic has the additional programmable coordinate of Tilt (skew), producing a pattern rotation about the center of the Crt screen. Its range results from ten bits or 0 to 1023, covering the range from approximately 4 degrees counterclockwise to the same clockwise. Here the middle value 512 represents zero tilt.

### 3.3.2 Scan origin

SCAN begins with setting the coordinate origin which is the reference for almost all subsequent action. For the purposes of the present discussion, references to position are with respect to the visual orientation on the Crt or viewing screens unless otherwise noted. The horizontal position of the origin is slightly to the left of the first of a group of the eight central cards of the ten normally visible after film advance. After setting the origin, SCAN calls for the GAIN subroutine.

### 3.3.3 GAIN

This is the same procedure used earlier in the leader program. As before it is performed in the clear unexposed zone near the top of the screen. The effective output of the procedure is to generate the proper input to its associated circuits in the scanner to bring the maximum signal amplitude to the nominal level of 980. It then returns to SCAN to prepare for VTRACK which finds the vertical position of the card images.

### 3.3.4 VTRACK

It is necessary to search for the lateral location of card images to compensate for significant differences in placement among cameras. The specific action of VTRACK is to locate the vertical position of the visual bottom of the first card, its left edge in terms of card orientation. The procedure output is a variable called YCMargin, where the 'C' refers to the coarse coordinate scale whose least programmable bit is 4 mils. The completion of VTRACK marks the end of initialization before undertaking actual card reading.

The procedure uses an array of 12 columns of 16 reading positions about 2 inches (in card dimensions) square. Horizontally the array begins at the origin, thus it covers parts of the scale zone as well as part of the card imge. In each of 12 downward columns it finds the transition from clear card stock to dense band below the first card. The mean location of the four most downward transitions is the product of simple calculations. This algorithm hence rejects contributions from other than the desired positions. With a minor downward shift to approximate the center of the dense band, the resultant becomes YCMargin, the basic reference for proceeding with reading the eight card images in the frame.

In as much as high precision is not a requirement for this measurement, the Crt spot size is large, about 100 mils. The procedure is interactive in that the vertical strokes stop upon detection of the dense band. Its proper action is therefore visible on the face of the Crt. YCMargin typically changes in steps equal to the spot size, allowing easy verification. The relatively coarse resolution is is quite adequate for the next action, which is to start the search process for the first card after return to SCAN.

#### Return to SCAN, preparation for Card Bar Search

The process for finding and reading each of the eight cards is substantially identical from this point. In summary it begins with generating an approximate horizontal reference from the Card Bar visible in Figure 1-1. Then comes a short series of actions to locate the position of the nearby corner of the card. This series repeats with appropriate changes in direction to locate all four corners. From the resultant set of coordinates, hole positions are thereafter calculated. Figure 3-1 on the next page is an enlarged drawing of the scanning actions in the vicinity of the first corner of a card image.

A called subroutine named CENTROID creates a multi-dot stroke horizontally across the Card Bar. This is a universal procedure, requiring prior data on position, dot spacing, number and direction of dots in the stroke, spot size and signal threshold. With these specified in SCAN, calling CENTROID produces the horizontal position of the center of the Card Bar. The vertical coordinate of the stroke, YCMargin, remains unchanged.

### 3.3.5 CENTROID

Centroid as used for finding the horizontal center of the Card Bar is a multi-dot dot stroke

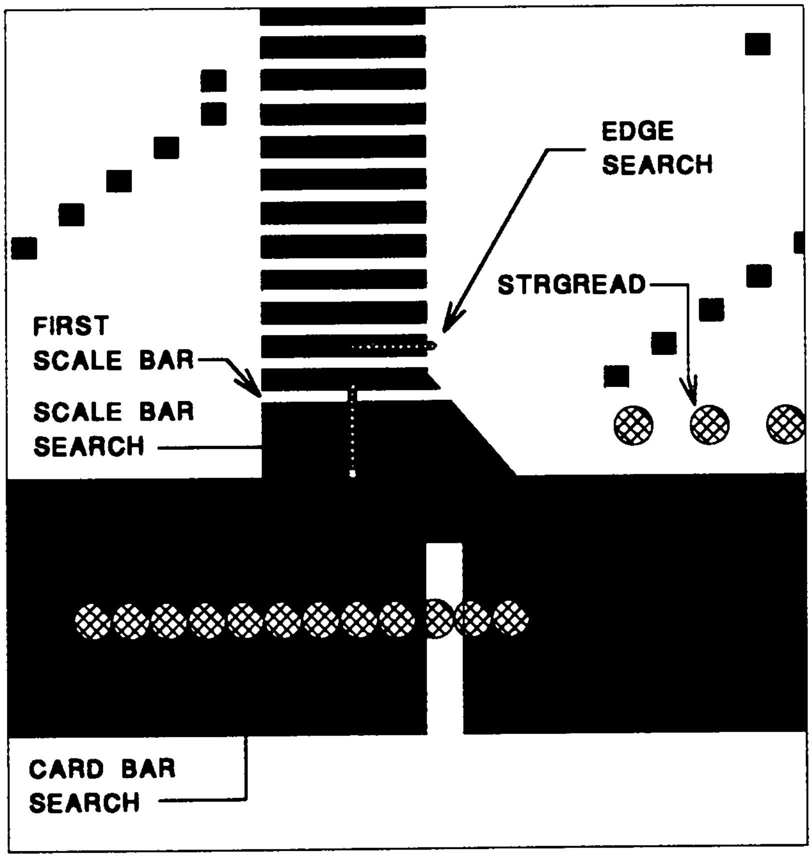


FIGURE 3-1 SCAN SEARCH PATTERNS

of relatively large step and spot size. It also is interactive, ending when it has found the target, offering similar benefits for ready verification by early termination. Its maximum length is approximately three-quarters of the card to card spacing, so that it can assuredly cross the Card Bar when film movement is complete. This is the mechanism for handling variability in stopping position.

CENTROID operates by computing the center of gravity based upon signal amplitudes above a threshold. As the stroke moves across a clear line, signals rise to a peak and decay following the peak. Movement stops and calculations begin when the signals fall below the threshold. The formula for calculation uses moments, the sum of the product of amplitudes and positions, this sum being divided by the sum of amplitudes. The result is derivation of the effective center of the Card Bar. This process is capable of considerable precision, as is necessary for its next usage in finding the centers of Scale Bars.

Return to SCAN, preparation for Scale Bar Search

A few films have shown so much variability in card size in the narrow dimension along the film that use of a fixed interval is not feasible. Accordingly the program calls for CENTROID a second time, directed to the Card Bar for the next card. Subtraction of the coordinates produces the true card interval within the limited resolution of this usage of CENTROID. This card-to-card distance is an element in later determination of corner locations.

The center of the Card Bar defines the stopped horizontal position of card images insufficiently well at this stage to be directly usable for scanning the punch holes. Both horizontal and vertical coordinates need further refinement. This is accomplished by changing to another set of scales in the scanner.

Fosdic has provision for additive Coarse and Fine scale movements, which differ by a factor of four. The Coarse scale covers the range from zero to 32,767, the Fine from zero to 8191. On the coarse scale the smallest programmable increment is four mils, on the fine scale, one mil. So far movements up through recognition of the Card Bar have been on the coarse scale. Once any card has been located approximately using coarse coordinates, further actions such as exact positioning and reading of punched holes involve use of the fine scale exclusively. These actions in the fine scale are replicated for every card, superimposed on the coarse positioning derived from Card Bars.

The next actions form a sequence aimed at finding the precise locations of the corners of a card. These subroutines involved are CENTROID again and EDGES, with a supplemental determination of threshold by STRGREAD (for String Read).

The first object of interest is the vertical location in (fine) Fosdic coordinates of the position of the first column in the scanned card. This corresponds to the center of the dense area between the two bottom-most scale bars. The process involves CENTROID to find the center of the lowest bar, followed by a small upward jump to advance the spot to the column center. Here input parameters to the subroutine facilitate the maximum possible precision.

Preparation for this more critical execution of CENTROID consists of reducing spot size for highest optical resolution, changing to upward vertical motion in fine steps and allowing customizable thresholding. The search vertical origin is in line with the card border at column 1, as is visible in Figure 3-1. Its horizontal location is approximately in the center of the scale between cards, as set by a fixed leftward displacement from the Card Bar.

Aside from the positional preparatory data for locating the scale bar, the threshold for signal amplitudes is most important. Of all of the variables requiring alteration for special film conditions, this is the one most frequently changed. Often the scale area is heavily exposed in filming, resulting in scale bars of reduced thickness or partial transmittance. Both of these conditions lower the useful signal amplitudes. Unless corrected for, higher rates of calibration failure commonly occur. The standard operating procedure has been to allow operator modification of the threshold, expressed as an increment above the signal amplitude at search start location.

3.3.6 Scale Bar Search, CENTROID

As in the previous use of the CENTROID subroutine, computation based on moments determines the center of the first scale bar. The search stops when signals fall below the threshold, again enabling visual evaluation of performance. An advance upward of one-half the nominal column spacing then furnishes a vertical reference for the location of the first column.

Return to SCAN, preparation for STRGREAD

STRGREAD, for StringRead, is a small subroutine added to measure the transmittance of the card stock in preparation for finding the top edge (in card orientation) of cards. Preparation includes coordinate insertion and spot enlargement for blemish rejection.

3.3.7 STRGREAD

The procedure develops the mean amplitude of four spot readings of film transmittance at the border of the card. The basic assumption is that this transmittance is representative of that prevailing at the top of the card (card orientation) near column 1. Typically the output mean amplitude ranges from 900 (on the scale of 0 to 1023) for clear card stock images down to 400 for heavily or reflectively exposed films.

Return to SCAN, preparation for EDGES

Coordinates for the card edge search are indicated in Figure 3-1. After the scale bar search, the vertical position is at the center of the first scale bar, determined by CENTROID. Strgread returns without change to the coordinates at the time of calling, bringing the spot back to the bar center. However as is apparent in the illustration, the top edge of the card may be displaced by trimming of the card corner. Hence a more inboard column is the preferred zone for reliable edge detection. For this reason the track for edge search is shifted to the vertical position of column 2. At the other top corner of cards, even more inset was found necessary to avoid more extreme corner clipping.

Edge search begins its rightward dot-based stroke at the same horizontal coordinate as the search for the scale bar, working from the supplied position, step size (fine), direction and reduced spot size.

### 3.3.8 EDGES

Scanning in edge search follows the path depicted in Figure 3-1. Signals in edge search start from low amplitudes inside the scale area until the spot reaches the card edge. Then a rapid rise takes place on crossing the edge. The true edge position is defined as that at which the signal is midway between the starting value and the transmittance just found in STRGREAD. The mean of these two establishes the edge threshold. A spot at this position would cover a low-signal area with one half while the other half would see the measured transmittance. Such exact positioning cannot be counted upon with the stepped digitized coordinates; instead an interpolation process is employed. Interpolation based on the nearest signals straddling the threshold defines the edge location.

Such a process is capable of good edge resolution with the assumption that the measured transmittance is representative of the card stock. Experience with dense cards bears out the validity of the approach. By taking into account the card stock transmittance, it provides an automatic correction for dense card images; no modification of these parameters was ever found needed.

When EDGES returns to SCAN, the extra one-column advance to column 2 is removed, leaving the half-column shift to the location of column 1. It should be noted that these advances come before any vertical sizing of the card image. At this stage the size is an estimate, feasible because the incremental distance is short. Also, vertical centering of the search track in column 2 is not critical due to the thresholding logic described above. That is, finding the true edge does not require that the edge search track be centered for best (lowest) signal because the threshold automatically adjusts itself to the initial track signal. Thus the parameters for the edge location formula are learned from the prevailing conditions. Good edge location is essential for precise access to punched hole positions.

At this time the precise location of the top of the card (card orientation) at column 1 has been established. The Fosdic coordinates are one-half column space less than the center of the first scale bar for the vertical axis and for the horizontal, the location of the card edge. These two quantities are the X and Y values in the variable known as Column1.

#### Return to SCAN, location of Corners 2 through 4

Using again the procedures CENTROID and EDGES with appropriate changes in the step direction, the next program sequence finds the coordinates of Corner 2 at the bottom (card orientation) edge of column 1. The transmittance measurement for thresholding is not necessary to repeat for Corner 2. After the edge search, the variable Corner2 contains the desired coordinate locations in X and Y.

The action then shifts to Corner 3 at the card top over column 80. It is similar to that for the first corner, using the same three subroutines. It is desirable to repeat Strgread at this end of the card because the illumination in the camera may not be equal transversely across the film.

Finally the program performs the same actions at Corner 4. The result is four sets of X and Y coordinate values expressed in mils. From these it is possible to derive the apparent length and width dimensions of the card as well as the slant. Calculations based upon these values then serve to direct the Crt spot for reading the 960 hole positions on the card.

3.3.9 Check for calibration failure

Before the mathematical process, the values lend themselves to checking for validity. The observed width of a card is available by subtraction of the X dimensions across the card at column 1 and again at column 80. These values should agree within acceptable limits. In this project a limit of 30 mils, about half the dimension of punched holes, was the applied criterion. Differences greater than this are liable to cause unacceptable deviations in the calculation of hole positions.

The same technique for checking the apparent width of a card applies to the Y dimensions. Here also the acceptable limit is 30 mils.

This verification process took place on every card. Those cards which showed excessive difference according to the criteria were declared failures in calibration, with the card data replaced by blank characters. During production scanning, close to one-half of the reels showed zero calibration failures. The failure rate in many decks was less than one card in a thousand; the mean for all reels was less than two in a thousand cards.

Card coordinates which successfully pass the checking process become the input to the next subroutine procedure, CALCSCAL, for calculation based upon the scale coordinates.

3.3.10 CALCSCAL

This subroutine accepts corner coordinates and computes the following:

1) Column to column (columnar) interval in terms of Fosdic vertical fine scale
2) Slant as difference between horizontal fine edge positions at columns 1 and 80
3) Table of row positions from top of card, also in horizontal fine scale

The procedure performs these actions one time on every card image, providing an automatic calibration for size and position, but only on corner data previously declared valid. The values so derived are passed on to the next subroutine, COLSCAN, for Column Scan.

3.3.10.1 Columnar interval

Columnar spacing comes from dividing the span from columns 1 to 80 by 79. Integer division in the computer yields two outputs. These are in mils for the quotient YFQuotient and the remainder YFRemdr. Typically the former is 87 mils while the latter varies according to the exact camera reduction ratio. These values reside in a temporary working memory, available for later use in column scanning.

3.3.10.2 Slant

Calculations for slant use the corner edge locations to derive the mean of the slant at the leading and trailing edges of the card image. Here the result is a bi-polar (can be positive or negative) value named Slant referring to the total between columns 1 and 80. Experience has shown this may range up to plus or minus 80 mils, greater than the size of a punched hole. The calculated slant for each card resides in a temporary memory for later use in COLSCAN.

3.3.10.3 RowTable

Again working with the corner coordinates, calculations extract the positional increments from the top of card images (leading edge of card images) to all 12 rows. These row positions are derived once per card to minimize the computational activity when sequencing from row to row later. The resultant of this part of the subroutine is RowTable, containing 12 distances in the horizontal fine scale from the top of the card to each row. Card geometry dictates that these values start at 1/13th and go to 12/13th of the measured card height.

Internally the calculations for RowTable use the same type of integer arithmetic involving division and summation of remainders to produce the desired proportioning of row positions. This implementation for automatic size adjustment readily accommodates the 10 percent or so variation found in a few reels.

Completion of this subroutine loads the parameters named above and returns control to SCAN to prepare for COLSCAN.

Return to SCAN, preparation for data reading, COLSCAN

This chiefly requires establishing the origin for the grid of 960 reading locations. In card orientation it is at the top of the card and one column to the left of the first, equivalent to column "0." This displacement permits the use of 80 uniform procedures for column reading. Parameters relating to the size of the grid come from the immediately prior CALCSCAL subroutine.

The parameters of threshold for hole detection and size of scanning spot are incorporated in the COLSCAN subroutine rather than being preset. This is due to their dynamic nature, meaning that these parameters are functions of signal amplitudes.

3.3.11 COLSCAN

This subroutine is responsible for executing the grid readings at the potential locations of punched holes. Its form is a sequence of 12 reads per column, starting with the first and repeating 80 times. The 12 reads in a column advance from the top row ("12" or "Y") down to the "9" row. At the end of each column, signal analysis interprets the stored information for inclusion in the 80-character string from the card.

3.3.11.1 Column advance

The initial step in reading in reading a column is an advance from the previous one, which in the Fosdic scale is a vertical retraction from the origin at column 0. This retraction is equal to a sum derived from adding YFQuot (from the CALCSCAL subroutine) for each column advance. YFQuot is in mils, typically 87 or 88. As each quotient adds to its sum, so does a sum of YFRemdr values. This remainder sum is compared with 79 at every column. When it reaches 79, one mil is added to the quotient sum. The remainder sum continues, with 79 subtracted. In this manner, the speed of integer arithmetic is available with no loss of precsion. The mathematical process results in values accurate to one mil, not a limiting factor in scanning performance.

3.3.11.2 Slant correction

Next is correction for slant. Here the input, also from CALCSCAL, is total slant between end columns. It is bipolar, requiring conversion to absolute value before entering the computational process using integer arithmetic. Then with restoration of polarity it yields a correction applied to the horizontal fine scale. The resultant is XFBase, the horizontal coordinate of the top edge of a card at any column after correction for prevailing slant. The remaining task is to add positional information from RowTable to bring the spot to the 12 hole locations in a column.

Reading at each hole position and saving hole signals involves cycling through the RowTable values added to XFBase. This produces 12 amplitudes stored in memory for analysis and conversion to a final recorded character.

Character resolution in every column is complete in itself. The analysis generates a background value, calculates a threshold, compares the signals against the threshold, sorts the row identifications in order of the three strongest signals and encodes the result.

3.3.11.3 Calculation of background

The first step in signal evaluation is to establish a reference background in preparation for calculating the applicable threshold for the column. This background value is intended to represent signals from unpunched holes in the column. The procedure takes advantage of expectation that only one numeric punch occurs in a column. In execution, the five central rows in a column are assumed representative. The sum of these five signals, from which the lowest value is subtracted, then divided by four, is the desired value. It is independent of whether or not one in the group is punched. For interest, a typical value for good films is approximately 900 when gain calibration in clear film produces 980. Dense card images can lower the background to possibly 400.

3.3.11.4 Signal amplitude

Signals from scanning these film negatives reflect the fact that holes are relatively opaque. Signal strength is measured in departure downward from the background. Thus strong signals may be as low as 80. Signals considered weak may be 300. Signals from blemishes can be virtually anything in the overall range. The goal of setting a threshold in these circumstances is to generate a value for best balance between drops and pickups. The ideal algorithm here is one that offers immunity to variations in camera illumination from side to side, film development conditions, color of card stack and, almost above all, card misregistration. The adopted approach provides automatic compensation for some of these factors plus an opportunity for custom modification when indicated.

3.3.11.5 Calculation of threshold

Technically the threshold is the level at which, in the presence of a certain amount of signal variation, half the tries would be declared above and half below. The derivation of a basic threshold level follows a formula which finds the mid point between background and assumed strong signals, the mean of the two. A numeric example would be 480. However experience demonstrates that such a level makes the system too sensitive as evidenced by a large number of pickups. It requires modification for a resonable balance of pickups and drops.

Threshold modification, or the term "fudge" in the program, is a programmed reduction of the background value before entry into the calculation. A typical modification might be 150, which changes the threshold by half this amount. The change enters via a defined parameter, requiring reassembly of the program for alternative values. Thus a specific threshold modification applies to an entire reel of film. In production processing it was not uncommon to rerun reels with program changes in attempts to better the quality indicators.

3.3.11.6 Rescan mode

Related to threshold generation is a procedure added in the course of program evolution. Its intent is to detect signals from partial hole images caused by misregistration in filming. The best conditions for reading partial holes are to reduce the spot size and to place the spot closer to the residual image. This requires multiple readings around hole positions, an unacceptably time-consuming process if universally applied. The final version of the scanning program includes provisions for multiple reading and other desirable temporary modifications but resorts to them only when needed. The switch to five complete rescans of a card takes place whenever the normal high-speed scan shows one or more marginal signals in the entire card. Marginal here refers to hole signals which lie in the range between the threshold and the midpoint from threshold to the (possibly modified) background.

This rescanning technique reduces throughput if frequently invoked in processing the reels. Each occurrence is equivalent to approximately four cards. Commonly with films with good card registration, the additional processing time is insignificant. A count of the number of occurrences is displayed at the end of scanning as an indicator of the difficulty presented by a reel. The observed counts often were as low as 50. It should be noted here that film blemishes can also trigger the multiple scanning., setting up the conditions more favorable to reading marginal hole images.

The switch to multiple reading appears in the scanning program as a change of mode. This allows freedom to incorporate other parametric or procedural changes of potential benefit. Several items have been mentioned above, another is an increase in sensitivity. Positionally the five rescans occur at the nominal hole center, to left and right by 15 mils and above and below by 25 mils. The strongest signal of the five becomes the declared signal amplitude for the row in the column.

3.3.11.7 Mask for alphanumeric/overpunch coding

A special 80-character file for each deck, .MSK described in Section 3.2.2, provides the information as to whether the column data should be encoded for single or multiple punching. The mask file for carrying the style information contains one character per column with the instruction for which encoding style is to be applied. The general approach for the mask file, including its ability to convert automatically from single to multiple punch, was given in Section 3.2.2. The desirability of single-punch recording style has already been mentioned for its superior performance in the presence of blemishes. For these a number such as "2" appears. For full encoding, "0" is the applicable digit. At the end of column analysis, interrogation of the mask file furnishes the indicator for the applicable encoding plan.

3.3.11.8 <u>Character encoding</u>

After column reading of all rows, the 12 amplitudes pass through three stages of logic to find the strongest in each pass. The logic is the same in all three, allowed by replacement of the strongest amplitude found in the first stage with a disqualifying value before executing the next stage. If none of the amplitudes passes the threshold, the sort-by-amplitude process jumps to its end for encoding as a blank column.

The encoding logic for multiple punching allows two overpunches plus one number from 0 to 9. The first step in analysis is to examine the triplet of three strongest row signals to find whether both the top two rows are punched. If so, the remaining row signal is treated as a numeric. If only one of the triplet is punched, it and the accompanying numeric become an alphabetic character. Finally the case of X over 0 carries a separate code.

To encode the described punch data, a number from 0 to 255 is developed from the row numbers 1 through 12. Special logic produces the same number regardless of which one is the stronger in an alphabetic pair. The dual overpunch condition adds an increment to this number to generate unique codes different from the ordinary numeric set. The last step in character interpretation is to pass this number through a translation table TRNSTBL1, listed in Appendix H. Unpunched columns show as the space character. Illegal punching such as two numerics results in an asterisk.

3.3.11.9 <u>Data recording</u>

Each card produces a string of 80 bytes in a file holding the data content of up to 640 cards, or 51,200 bytes. When this file is full, scanning pauses a fraction of a second for adding the data to the single large file called DataOut containing the total content of the reel. During the pause period, a display on the computer monitor shows the accumulated number of rescan occurrences and the number of calibration failures. These provide ongoing indicators of processing difficulty and quality. A full film reel generates approximately 20 such pause periods.

The data generated during rescans overwrite the information recorded from the original scanning by resetting the pointer in the memory section.

3.3.11.10 <u>Return to SCAN</u>

After the last column in a card has been recorded, control returns to SCAN to move to the next card. It is not always possible to advance the search for the next card bar by a fixed amount; the variability in card periodicity can accumulate enough error to cause the search to miss the rightmost cards. The remedy is to shift the search origin based upon the observed position of the previous card bar.

As mentioned in Section 3.3.5 concerning determination of (horizontal) card edge positions, the distance to the next card bar is determined at the beginning of action on any card. This distance is derived using the fine scale, usually around 2180 mils. To set the origin for the next card bar search, this distance is subtracted from the fine coordinates and added to the coarse. Then a nominal card-to-card interval is also added to the coarse coordinates. Thus the origin for card bar search is based upon an assumed interval from the previous card bar rather than a multiple of the assumed interval from the stopped position of the first card. After this the next card scanning repeats the entire sequence in fine coordinates alone, beginning with a new card bar search.

The rightward scanning of card after card continues through the eighth card. Control then passes to a program section called SetXCFrame whose function is to set the horizontal positon at which the FRAME subroutine is to commence. When there are less than eight cards within the scan aperture, as from empty card stacks or misfeeds, the presence or absence of a next adjacent card comes from the output of STRGREAD. This subroutine executes at the card left border. If a card is present the card stock image is clear. Absence of a card makes the film dense, generating a low signal. If comparison against a threshold after the subroutine shows that no card is present, control passes to SetXCFrame from this source as well. SetXCFrame performs the exchange of fine and coarse interval mentioned above. It furthermore adds an increment to bring the horizontal position of the Frame scan to the horizontal center of the assumed next card.

Before calling FRAME it is necessary to determine a valid threshold. In films exposed with supplemented reflected light, the card stock is relatively dense. This condition requires a correction to the threshold in FRAME for reasons which will become evident. The ideal threshold is a value midway between the card stock and averaged scale lines. Another scan procedure, named SCALREAD, determines the latter.

3.3.11.11 SCALREAD

The specific purpose of this procedure is to generate a signal amplitude which is representative of the scale lines when scanned by the defocused spot used in FRAME. Accordingly it reads at eight positions and calculates the mean, naming it ScalTransm for Scale Transmittance. The scan locations are in a vertical line stretching from column numbers 55 to 75. Horizontally the line is in the center of the scale zone immediately to the right of the last card in the group of eight or less.

Return to SCAN, preparation for FRAME

The threshold for use in FRAME is the mean of ScalTransm and Transmtce, the measure of the card stock transmittance signals near column 80, from STRGREAD.

The horizontal position in preparation for FRAME has been described as approximately half-way into the next expected card. Vertically it is around column 70. Although any vertical location inside the card images would be usable, considerations of wear distribution on the Crt screen favor this position. The leader routine LDR described in Section 3-1 also calls the FRAME procedure but places the vertical position near the lower numbered columns.

3.3.11.12 End of SCAN, return to PCSCAN

After the SCAN subroutine has completed its action to extract the punched data from eight cards, or less if fewer are available, it returns to the executive program, PCSCAN. Several indicator flags will have been set to show success or otherwise in the various search procedures. Two of these make it possible to stop the scanning process if any card in the eight has failed calibration or if illegal punching (multiple numerics in a column) has been found in any of the cards. When such a stop occurs, a complete eight-card display of detected punching appears on the computer monitor. This display is also available at any time during a process run, activated by pressing a key. This feature facilitates visual verification of reading accuracy by comparison with the projected film image on the viewing screen.

When PCSCAN has finished its disposal of information from scanning the group of eight cards, it calls for the FRAME subroutine to bring in more cards.

3.4 FRAME, detailed

This is the same subroutine as called by the leader program. Because some of the fine points in its logic are involved in the card-to-card advance here, the more detailed description is given in this section.

The scannable area to which the Crt spot can have access is equivalent to approximately eleven cards. Inside this zone the central eight cards represent the working space for the scanning program, with allowances for minor variations in positioning. It is the responsibility of the FRAME procedure to advance the film through the this scannable area despite the presence of certain anomalies.

3.4.1 Film irregularities

A brief description of film characteristics is in order here. The majority of cards appear on the films as long series of images separated by occasional breaks. Breaks arise from several causes, due to interruptions at the time of original filming. Card misfeeds, for example, usually result in leader-like sections with card and scale bars present without card images for a space. When the camera feed hopper becomes empty, the camera turns off its illumination, producing a gap of clear film. Card jams generate the least predictable images, sometimes stretching the image of a single card over the normal space of a dozen cards. In summary these are the conditions with which the film movement system must deal. One of the program goals is to prevent these interruptions from appearing in the final recorded information.

3.4.2 Framing logic

On the Crt face and the viewing screen the leftmost of the normal eight card images is the first while the eighth is at the right. The action of the Framing subroutine is to move the film so that there is always an (apparent) card image in the first position. When scanning of eight cards has finished, the procedure finds and holds on to the left edge of the ninth card (its top edge in card orientation). Capstan rotation eventually brings this edge to the first card position, whereupon film movement stops. Scanning then begins on the new card group.

If there are fewer than eight cards, for instance three just before a misfeed, the search for the fourth card recognizes its absence and calls for FRAME. The framing procedure starts film movement and waits at a position one-half card to the right of the last card until another card (or clear film space) comes to the waiting position. It then holds on to the left edge leftward to the stop position. Ater film motion has ceased and a brief settling time allowance, scanning commences on the new first card. It continues with the second card in sequence up through the eighth unless there is another misfeed. Thus the system runs automatically through zones of missing cards. If the apparent next card is instead a clear zone on the film, the program halts for operator action.

The term (apparent) card image referred to above comes about because the Framing logic by itself cannot tell the difference between a true card image and a transitional break from dense film to a clear gap. The FRAME subroutine has therefore performed its function, regardless of whether or not the stopped image is a genuine card.

That discrimination is performed by other logic. If it is unable to verify that the image is a true card, the program pauses, waiting for manual advance to the next good card image. This is the operating procedure initiated by clear gaps in the films. Gaps of this nature produce no recorded data but of course have an adverse effect on throughput. On an average possibly every reel exhibited at least one such halt; a few had up to twenty.

Because of their unpredictable images, the results of card jams are difficult to define. Simple stretchouts ordinarily yield a single calibration failure, recorded as 80 space characters. More complicated distortions can give rise to multiple pseudo-cards with meaningless data.

3.4.3 Edge following

Framing begins with turning on both the capstan motor and a scanning pattern. This pattern is vertical column of 10 dots, placed horizontally in the approximate center of the ninth card at the start. The pattern is edge-seeking by acting according to the following logic. Whenever the column of dots is over the interior of a card image, it moves to the left (visual orientation of Crt screen) by an increment, before the next column actuation. After a few such shifts, the column of dots comes over the scale zone of the card. No longer in the clear, the leftward horizontal drive ceases, leaving the dot column at the horizontal position of the top edge of the card (card orientation). This edge-seeking action is rapid, possibly taking two milliseconds. A few milliseconds later the film starts moving to the left as the capstan responds to the start command. Finding itself in the clear again, the next dot column shifts to the left so as to keep up with the card movement. This is edge-following; with suitable response rates it is capable of staying with the card edge throughout the card advance period.

### 3.4.4 Film stop position

Cessation of film movement occurs when the horizontal location of the followed edge drops below a program-set XCFrameStop. The desired stopping position is with the next group of eight cards nominally centered in the scanning aperture. To accomplish this rquires definition of a fixed scan origin to which is added the offset allowance for film deceleration, XCFrameStop. Altogether these combine to bring the followed card image to a position within the reach of the subsequent search for the first card bar.

Because the capstan drive is a stepper motor, the stopped position is variable over an acceptance band equivalent to about half the card period. This variability is desirable for distributing wear on the Crt screen.

Thus the framing action is to bring the new, ninth card to the first card position. The films include long continuous strings of card images, interrupted occasionally by gaps of clear film, misfeeds where the scale bars continue without card images and sometimes jammed cards. Encountering clear gaps causes the film to stop with the leading edge of the gap in the normal acceptance zone. This is also the usual case with card jams. Misfeeds, where some cards are simply missing but other parts of normal images are present, cause the film to run on until the next valid card appears, stopping it at the first card position. When there are less than eight cards available in a stopped group, a misfeed after the fifth for example, framing begins in an attempt to bring an apparent sixth card to the first card position. By starting from the last recognized card, this logic can cope with a single missing card. If there is no interruption in the scale area, framing automatically brings in the next card.

### 3.4.5 Manual film advance

If the image it stopped on is not recognizable as a true card, scanning pauses with a request for operator intervention. Manual film advance, by turning off power to the capstan, turning it by hand until the next valid card image is in the normal stop band, and restoring power, is the appropriate operator action.

### 3.4.6 FRAME threshold

This edge-following action is obviously dependent upon discrimination between signals derived from the card stock and those from the scale lines. When the cards have been filmed by reflected light, the background stock can be relatively dense, lessening its difference from the scale lines. This is the reason for introducing the SCALREAD subroutine earlier. The threshold for use in the framing procedure is the mean of ScalTransm and Transmtce, as mentioned in Section 3.3.11.11.

In the 10-dot column which constitutes the scanning pattern, not all of the dots must show signals above the threshold for the area to be declared clear. It is sufficient that at least nine meet this requirement. The Crt spot during the subroutine is enlarged so as to prevent punched holes from disqualifying the clear area. Further refinements in the Framing subroutine minimize the duty cycle of beam-on time so as to reduce Crt screen wear as much as possible. Finally the 10-dot column has an interlace of six origin displacements for the same purpose.

3.4.7 End-of-Film detection

An End-of-Film indicator appears during FRAME if the signals during execution reach the saturation value of 1023. To avoid false recognition from film blemishes, five of the ten dots in the vertical scan line must reach saturation. This condition occurs when the end of the film passes beyond the location of the Framing scan. It sets an indicator flag which later terminates the scanning program.

End of FRAME, return to PCSCAN

Upon return to PCSCAN the End of Film flag is checked. When the flag is not set, program control goes on to call SCAN for the next group of cards. If the flag has been set by termination of the reel as above, information from the cards scanned after the last 640-card pause is recorded. This marks the end of PCSCAN and action reverts to the batch file GO. The latter next calls for a wrap-up program named TLR, for Trailer.

3.4.8 End of PCSCAN

The termination of PCSCAN generates a display on the computer monitor which includes the final number of rescan occurrences on the reel. Aside from the displayed data, it loads other accumulated information about the run into the Param file.

At this time the complete card data content is available in a large file called DataOut, usually nearly one megabyte in size.

3.5 TLR program

This program provides a summary display on the computer monitor of results from scanning a reel. Among the items shown are the total number of cards scanned and number which failed calibration , both transferred from the Param file. Also displayed are the times of starting and stopping the run. These items are in addition to identification of the deck and reel.

3.6 FOSCOPY program

This is the final element in the initiating batch file GO. It has several functions, including most importantly that of adding a header to the card data file and naming the result to show deck and reel identification. Other functions are to create a logging file, allow introduction of comments and provide a summary of data on bad cards.

3.6.1 Card data file identification

Data files are labelled with eight digits for the deck, then a period for delimiter, followed by an extension of three digits for the reel number. Both fields are right-justified and zero-filled to the left. An example is 00000110.005 representing reel 5 in deck 110. This information comes from reading the Param file mentioned in Section 3.1.5 Entry.

3.6.2 Header

All card data files include an initial section of 1882 bytes containing general notes and specific information on reel processing. The latter covers deck and reel identification, date scanned, number of cards in file, number of cards failing calibration and provision for two condition codes to be added if desired by the operator at the end of the run.

Appendix K is a sample printout of a header, here for Deck 110, Reel number 5. The first 40 bytes are largely in unprintable binary form, obtained from the Param file for the run. However it is possible to note the Deck "00000110", Reel "005" and the date "05-15-1998". Next comes a description of the content of these 40 bytes. The ending alphanumeric translation table has since been supplemented with characters from overpunching according to Appendix A.

3.6.3 Card data file format

Data from reading the punched information is contained in a file called DataOut, consisting of an end-to-end string of 80 bytes from each card. Each card adds 80 bytes of punched data, or space characters in the event of calibration failure, to this file. Typically DataOut contains nearly a megabyte (from 12,000 multiplied by 80). The Foscopy program attaches this string to the header to form the final file with the identification described above. The card data begins with byte 1882 from the first column in the first card on the film. Thus each reel generates a file whose length approaches one megabytes.

3.6.4 Zip disks

The final form of output data from the project is recordings on 100-megabyte Iomega Zip disks. These disks can hold the data from slightly over one hundred reels of film. In the records from the project, whenever possible the particular group on any disk had a common hundreds digit for ease in handling. For example a disk might contain reels 100 through 199.

The first Zip disks in all decks (except Version 2 of Deck 194, where the last disk was used) contain recordings from duplicate runs to provide samples for the difference tests described in Section 4.5. At least one and usually five samples are available depending upon deck size. The duplicate files carry the identifying extension "A01" for comparison with the normal "001" for instance.

Finally there are four more files on every Zip disk, namely:

FILES.TXT, a file listing the types of files on the Zip disk
NCDCREAD.TXT, a description of the structure of the card data files
REPORT.EXE, an executable .EXE (DOS) program for displaying the data from any card
REPORT.TXT, instructions for use of REPORT.EXE

The output Zip disks are numbered as GDA-01 (for Greenough Data Associates-01) to GDA-37 in order of submission to the sponsor.

3.6.5 Replacement data files

Some of the disks have been or are to be replaced with data from rescanning with the final version of the program. The replacement disks are denoted with with an "A" suffix, as "14A" replaces "14". The relacement disks better reproduce the punched data under marginal filming and punching conditions.

## 4.0 Performance review

Several control indicators allowed monitoring of the apparent quality of data conversion during processing runs. The most important was the count of calibration failures. Other indicators were the number of illegal characters and the number of cards with two or more illegal characters. Finally individual runs displayed the number of cards forcing rescan on account of marginal signals; this was a measure of the deviation from ideal film or card conditions in the reel.

In addition for acceptable processing on a deck, duplicate runs on sample reels must have a low count of cards with differences in recorded data in one or more columns.

In the sections which follow, the available statistics on the above indicators are summarized. Table 4-1 at the end is a listing by individual decks as well as overall figures.

## 4.1 Calibration failure

The overall calibration failure rate was 0.015 percent of the cards processed, equivalent to one card in 6700 or 1.8 cards per 12,000-card reel. This figure results from a sum of 5608 cards indicated as unreadable in the total of 3086 reels scanned. This overall rate includes 65 copy films, whose failure rate was more than four times that of original films.

No original films were rejected as being unsuitable for scanning. A few of the preliminary copy films were so dense as to require reprinting at less exposure. Only one reel remains outstanding on this account at the date of the report.

Many of the observed failures came about from card misfeeds in specific cameras, where misfeeds resulted in overexposure and loss of the scale lines following the last valid image. The likely cause was abrupt cessation of film movement in the camera before lamp turn-off. Another source of failure was larger-than-normal cuts at the corners of cards. Also, artificial contributions to the failure count sometimes occurred from card jams, when irregular images generated non-existent cards. The standard output from calibration failure is s string of 80 "space" characters.

There may be a few instances where the scanning program produced meaningless data. Such data may include long strings of the characters "*" or "9". The program examines output data before recording, looking for more than 20 of these characters in a card. If so found, it declared a calibration failure and replaced the card data with space characters.

Several reels included unpunched cards, producing blank records similar to those from calibration failure but without incrementing the failure count.

To explore the reasons for calibration failures, a program switch allowed repeated scanning after pausing at detected failures. Such investigations were time-consuming and so were conducted principally on reels with high counts. This usually required a second run with the pause option turned on, then more runs with potential corrections if the causes were not simply in the punching.

4.2 Illegal character count

An illegal character, recorded as "*", arises from double punching in the numeric field of a column. The possible sources are punching, pickups from blemishes on the film or Crt screen. By far the most common was the punching. In certain decks it was so consistent that mention was included on the deck processing notes in Section 5.0. To determine whether the causes were again in the punched data, the same pause and exploration procedures as for calibration failure stops were followed. This utilized another option switch in the scanning program.

Pickups, that is, apparent punches where none exists, can be produced from blemishes or dust on either the film or the Crt screen. The source can be isolated to some extent by whether the occurences are random in location (film) or systematic (Crt). While blemishes are usually immovable on films, preventative measures against dust include velvet pads at the entrance to the film gate with newly-cleaned surfaces at the start of reels. Normally the Crt screen was cleaned routinely every few hundred reels. Once or twice, systematic pickups, those occurring in specific columns with a periodicity of eight cards, became evident, requiring additional cleaning and rerunning of affected films.

The illegal character count on one large deck, 193 with 538 reels, is probably typical. It showed 0.11 percent of cards with an illegal character, or one card in 900. Examination of several worst cases led to an estimate that three quarters of the occasions were due to punching. On this assumption perhaps 0.03 percent of the cards recorded an illegal character from blemishes. When an allowance is made for blemishes in locations other than the numeric fields of columns with punches, blemishes and dust may account for as much as 0.05 percent of cards affected by pickups.

The effect of pickups can take several forms. Blank columns appear punched. Single punches convert to either an alphabetic or illegal character. Although it is sometimes possible to extract the true punched data, the importance of minimizing errors from this source gave rise to the monitoring system for illegal characters.

4.3 Cards with two or more illegal characters

Separate monitoring of the number of cards in a reel showing multiple illegal characters resulted in a very low count in the sample above. The figure amounted to 0.0001 percent or one card in 100,000.

4.4 Rescan occurrences

The number of cards triggering the multiple-scan procedure because of marginal signals was another internally monitored performance indicator. In general it shows the "difficulty" or "stress" the particular reel is presenting to the scanning program. The occasions for rescanning are chiefly created by cards with poorly registered holes. Typically films exhibit regions of frequent rescanning mixed with sections where it is rare. If most of a reel requires rescanning the processing throughput drops significantly, even to as low as a quarter of normal.

Results for the example of deck 193 cited above indicate an average 2.1 percent of cards necessitated rescanning, equivalent to 250 times in a reel. Many reels required as few as 30, others up to several thousand. In any case, this provision for automatic rescanning allowed much greater throughput than rerunning with specific customization of the program on individual reels.

4.5 Difference tests

Difference tests are comparisons of data obtained from duplicate runs on the same reel of film. Difference tests on five reels was a requirement included in the project work statement as a measure for establishing confidence in the conversion process. The tests were later found to be of such usefulness that their application was expanded to sampling in every deck.

The overall difference rate, computed as the mean of all reels subject to the comparisons, was 0.17 percent or less than two cards per thousand card images on the films. Difference test results on the small sample available of copy films showed a nearly similar figure.

These tests are particularly sensitive indicators of marginal conditions such as card misregistration or film blemishes. This is of considerable assistance in determining the threshold level for balance between failing to detect "weak" punch images and responding to artifacts like blemishes or dust. Investigations of optimal thresholding techniques accounted for much of the time in development of the scanning program.

Initially in the project all difference tests were conducted at the Climatic Data Center. Later a comparison program was created for the project by an associate of the author which automatically identifies cards with differing data and indicates the non-matching columns. Its use greatly simplifies the otherwise laborious process of visually finding differences in 80-character record pairs. After checks on agreement between tests here and at NCDC, the resulting shortening of turn-around time expedited the exploration of thresholding. Also additions to the scanning program made it possible to stop on specified cards and examine the card images directly on the Fosdic viewing screen. Observations on this basis allowed ready follow-up of proposed thresholding approaches.

4.6 Processing summary

The results of calibration failure and difference tests on individual decks are shown in Table 4-1 on the next page. The Table also includes data on the number of reels scanned in each deck, duplicated from Section 1.0.1 at the beginning of the report.

Table 4-2 lists the records regarding shipments of films. In some cases decks which had been sent back to NCDC were requested again for rescanning. A few, initially unreadable, reels were held back from return to NCDC in anticipation of program modification. These and other details are included in the Table. An entry of "GDA" indicates reels or data files that are presently here, not yet returned. For example in Deck 116, reel 343 awaits return shipment.

Table 4-3 shows the records of output data by deck, useful for determining which Zip disks contain the data for given decks. A description of the format of these Zip disks is included in Section 3.6 on Foscopy, the program which assembles the final data files. In this Table, "GDA" refers to temporarily recorded files which will be combined on a final disk when several more films listed in Section 5 are received.

Table 4-4 rearranges the same information for indexing by output Zip disk to show the contents of deck and reel numbers for each disk.

## Table 4-1 Calibration Failure and Difference Rates by Deck

| Deck | Description | No. Reels Scanned | Calib Fail Rate % of cards | Difference Rate % of cards |
|---|---|---|---|---|
| CD-110 | Navy Marine | 66 | 0.020 | 0.36 |
| CD-116 | Surface Marine | 596 | 0.009 | 0.19 |
| CD-117 | Navy Marine | 244 | 0.027 | 0.07 |
| CD-118 | Japanese Marine | 144 | 0.012 | 0.14 |
| CD-119 | Japanese Marine | 77 | 0.009 | 0.07 |
| CD-128 | Surface Marine | 229 | 0.006 | 0.09 |
| CD-129 | Surface Marine | 0 | | |
| CD-184 | British Marine | 63 | 0.017 | 0.35 |
| CD-185 | USSR Marine | 10 | 0.007 | 0.07 |
| CD-187 | Japanese Whaling | 2 | 0.004 | 0.07 |
| CD-189 | Dutch Marine | 23 | 0.009 | 0.08 |
| CD-190 | CG Log Book | 4 | 0.006 | 0.05 |
| CD-192 | German Marine | 517 | 0.015 | 0.19 |
| CD-193 | Dutch-Netherland | 538 | 0.016 | 0.24 |
| CD-194 | British Marine | 339 | 0.019 | 0.19 |
| CD-195 | Navy Ship Logs | 54 | 0.014 | 0.07 |
| CD-196 | German Marine | 16 | 0.003 | 0.10 |
| CD-197 | Danish Marine | 3 | 0.008 | 0.17 |
| CD-256 | Surface Marine | 29 | 0.016 | 0.12 |
| CD-258 | Surface Marine | 0 | | |
| CD-281 | Navy MAR | 97 | 0.040 | 0.24 |
| CD-541 | US RAOBS | 1 | 0.025 | 0.20 |
| CD-547 | Payerne RAOBS | 0 | | |
| CD-608 | Hungary RAOBS | 1 | 0.008 | 0.09 |
| CD-650 | Formosan RAOBS | 7 | 0.023 | 0.11 |
| CD-690 | Israel RAOBS | 1 | 0.008 | 0.14 |

Total number ............... 3061 reels scanned (10-30-98)

Average calibration failure rate ..................0.015 percent of cards

Average difference rate ............................................... 0.17 percent of cards

## Table 4-2 Film Shipment Records by Deck, page 1 of 2

Status 10-30-98

| Deck | Description | Reels | Date Film Received | Date Film Returned | Notes |
|---|---|---|---|---|---|
| **CD-110** | Navy Marine | 001-066 | 03-18-98 | 09-01-98 | |
| **CD-116** | Surface Marine | 001-596 | 09-18-97 | 11-20-97 | (except reel 343) |
| | | 343 | 03-18-98 | GDA | (original reel 343 recvd) |
| | | 001-005 | 03-04-98 | GDA | |
| **CD-117** | Navy Marine | 001-244 | 11-10-97 | 11-24-97 | |
| | | 001-005 | 03-04-98 | 10-07-98 | Sent again for rescanning |
| | | 006-244 | 07-30-98 | 10-07-98 | Sent again for rescanning |
| **CD-118** | Japanese Marine | 001-145 | 03-18-98 | GDA | (except reels 078,079,098,119) |
| | | 078-079 | 10-26-98 | GDA | (copy reels 078,079 recvd) |
| | | 098 | 10-26-98 | GDA | (copy reel 098 recvd) |
| **CD-119** | Japanese Marine | 001-077 | 03-18-98 | GDA | |
| **CD-128** | Surface Marine | 001-186 | 03-18-98 | GDA | (except reel 112) |
| | | 208-210 | 03-18-98 | GDA | |
| | | 211-250 | 03-06-97 | GDA | |
| | | 112 | 10-26-98 | GDA | (copy reel 112 recvd) |
| **CD-129** | Surface Marine | 187-311 | 03-06-97 | 07-02-98 | Deleted from list |
| **CD-184** | British Marine | 001-064 | 03-18-98 | 09-01-98 | (except reel 012) |
| **CD-185** | USSR Marine | 001-010 | 03-18-98 | 09-01-98 | |
| **CD-187** | Japanese Whaling | 001-002 | 03-18-98 | 09-01-98 | |
| **CD-189** | Dutch Marine | 001-023 | 03-18-98 | 09-01-98 | |
| **CD-190** | CG Log Book | 001-004 | 03-18-98 | 09-01-98 | |
| **CD-192** | German Marine | 001-517 | 11-19-97 | 01-28-98 | (1) |
| | | 001-517 | 07-03-98 | 10-07-98 | Sent again for rescanning |
| | | 013 | 10-26-98 | GDA | (copy reel 013 recvd) |
| **CD-193** | Dutch-Netherland | 082-086 | 11-19-97 | 08-11-98 | |
| | | 093 | 11-19-97 | 08-11-98 | |
| | | 099-104 | 11-19-97 | 08-11-98 | |
| | | 106 | 11-19-97 | 08-11-98 | |

## Table 4-2 Film Shipment Records by Deck, page 2 of 2

Status 10-30-98

| Deck | Description | Reels | Date Film Received | Date Film Returned | Notes |
|---|---|---|---|---|---|
| **CD-193** (cont.) | Dutch-Netherland | 001-081 | 12-18-97 | 08-11-98 | |
| | | 087-092 | 12-18-97 | 08-11-98 | |
| | | 094-098 | 12-18-97 | 08-11-98 | |
| | | 105 | 12-18-97 | 08-11-98 | |
| | | 107-538 | 12-18-97 | 08-11-98 | |
| **CD-194** | British Marine | 001-046 | 06-28-97 | 11-02-97 | |
| | | 103-341 | 06-28-97 | 11-02-97 | |
| | | 001-046 | 09-18-98 | GDA | Sent again for rescanning |
| | | 103-341 | 09-18-98 | GDA | Sent again for rescanning |
| | | 047-102 | 08&09-98 | GDA | (copy reels in this range sent in several batches) |
| **CD-195** | Navy Ship Logs | 001-054 | 03-18-98 | GDA | |
| **CD-196** | German Marine | 001-016 | 03-18-98 | 09-01-98 | |
| **CD-197** | Danish Marine | 001-003 | 03-18-98 | 09-01-98 | (except reel 002) |
| | | 002 | 10-26-98 | GDA | (copy reel 002 recvd) |
| **CD-256** | Surface Marine | 748-746 | 03-18-98 | 07-02-98 | |
| **CD-258** | Surface Marine | 748-916 | 02-13-98 | 07-02-98 | Deleted from list |
| | | 917 | 03-18-98 | " | " |
| **CD-281** | Navy MAR | 001-097 | 03-18-98 | GDA | |
| **CD-541** | US RAOBS | 001-021 expected | | | To be scanned when recvd |
| | | 021 | 10-26-98 | GDA | (copy reel 021 recvd) |
| **CD-547** | Payerne RAOBS | 001 expected | | | To be scanned when recvd |
| **CD-608** | Hungary RAOBS | 001 | 10-26-98 | GDA | (copy reel 001 recevd) |
| **CD-650** | Formosan RAOBS | 001-007 | 03-18-98 | 09-01-98 | |
| **CD-690** | Israel RAOBS | 001 | 03-18-98 | 09-01-98 | |

**Notes:**

(1) 18 reels were held back for scanning with modified program, then returned with deck

(2) See Section 5.0 Remnant actions, for reels not yet received.

## Table 4-3 Data Return Records by Deck, page 1 of 2

Status 10-30-98

| Deck | Description | Reels | Date Data Returned | Recorded on GDA-No. | Notes |
|---|---|---|---|---|---|
| **CD-110** | Navy Marine | 001-066 | 08-04-98 | **31** | |
| **CD-116** | Surface Marine | 001-596 | 11-13-97 | **05-10** | (except reel 343) |
| | | 343 | 09-10-98 | **19A** | (reel 343 recvd later) |
| | | 001-005 | 05-18-98 | **26A** | Updated by 6 floppy disks |
| | | | 06-04-98 | | Duplicate floppy disks |
| **CD-117** | Navy Marine | 001-244 | 11-20-97 | **11-13** | Replaced by new data |
| | | 001-005 | 05-18-98 | **26A** | Updated by 6 floppy disks |
| | | | 06-04-98 | | Duplicate floppy disks |
| | | 001-244 | 10-08-98 | **11A-13A** | New data from rescanning |
| **CD-118** | Japanese Marine | 001-145 | 07-22-98 | **29-30** | Updated by 6 floppy disks Note 1 |
| **CD-119** | Japanese Marine | 001-077 | 07-22-98 | **28** | |
| **CD-128** | Surface Marine | 001-186 | 08-05-98 | **33-35** | (except reel 112) |
| | | 208-210 | 08-05-98 | " | |
| | | 211-250 | 08-05-98 | " | |
| | | 112 | GDA | **37** | (reel 112) |
| **CD-129** | Surface Marine | 187-311 | | | Deleted from list |
| **CD-184** | British Marine | 001-064 | 07-22-98 | **27** | (except reel 012) |
| **CD-185** | USSR Marine | 001-010 | 07-22-98 | **27** | |
| **CD-187** | Japanese Whaling | 001-002 | 07-22-98 | **30** | |
| **CD-189** | Dutch Marine | 001-023 | 07-22-98 | **27** | |
| **CD-190** | CG Log Book | 001-004 | 07-22-98 | **30** | |
| **CD-192** | German Marine | 001-517 | 01-27-98 | **14-19** | Replaced by new data Note 2 |
| | | 001-517 | 09-12-98 | **14A-19A** | New data from rescanning |
| **CD-193** | Dutch-Netherland | 001-538 | 03-24-98 | **20-25** | Replaced by new data |
| | | 001-538 | 07-02-98 | **20A-25A** | New data from rescanning |

## Table 4-3 Data Return Records by Deck, page 2 of 2

Status 10-30-98

| Deck | Description | Reels | Date Data Returned | Recorded on GDA-No. | Notes |
|---|---|---|---|---|---|
| **CD-194** | British Marine | 001-005 | 08-20-97 | **01** | Replaced by new data |
| | | 006-046 | 10-08-97 | **02-04** | Replaced by new data |
| | | 103-341 | 10-08-97 | " | Replaced by new data |
| | | 001-341 | GDA | **01A-04A** | New data from rescanning |
| **CD-195** | Navy Ship Logs | 001-054 | 08-05-98 | **32** | |
| **CD-196** | German Marine | 001-016 | 07-22-98 | **28** | |
| **CD-197** | Danish Marine | 001-003 | 07-22-98 | **28** | (except reel 002) |
| | | 002 | GDA | **37** | |
| **CD-256** | Surface Marine | 748-746 | 07-22-98 | **30** | |
| **CD-258** | Surface Marine | 748-917 | | | Deleted from list |
| **CD-281** | Navy MAR | 001-097 | 08-05-98 | **36** | |
| **CD-541** | US RAOBS | 001-020 expected | | | To be scanned when recvd |
| | | 021 | GDA | **37** | |
| **CD-547** | Payerne RAOBS | 001 expected | | | To be scanned when recvd |
| **CD-608** | Hungary RAOBS | 001 | GDA | **37** | |
| **CD-650** | Formosan RAOBS | 001-007 | 07-22-98 | **30** | |
| **CD-690** | Israel RAOBS | 001 | 07-22-98 | **30** | |

Notes:

(1) reels 078, 079, 098 missing

(2) 18 reels held back for delayed scanning, data not included on GDA-14 to 19
New files on GDA-14A to 19A are complete.

GDA indicates material at Greenough Data Associates, 10-30-98, to be included on GDA-37

## Table 4-4 Data Return Records by GDA Disk page 1 of 3

| GDA Disk Number | Date Last Submitted | Deck | Reels | Notes |
|---|---|---|---|---|
| **01A** | GDA | **194** | **001-099** | \ |
| | | | **A01-A05** | These replace GDA 01-04 |
| **02A** | GDA | **194** | **100-199** | submitted 10-08-97 |
| **03A** | GDA | **194** | **200-299** | Note: sample A11-A15 & 011-015 |
| **04A** | GDA | **194** | **300-341** | / for Difference Test is on GDA-04A |
| **05** | 11-13-97 | **116** | **001-099** | \ GDA-26A replaces GDA-05 |
| | | | **A01-A05** | / for reels 001-005 and A01-A05 |
| **06** | 11-13-97 | **116** | **100-199** | |
| **07** | 11-13-97 | **116** | **200-299** | |
| **08** | 11-13-97 | **116** | **300-399** | --- (reel 343 contained on GDA-19A) |
| **09** | 11-13-97 | **116** | **400-499** | |
| **10** | 11-13-97 | **116** | **500-596** | |
| **11A** | 10-08-98 | **117** | **001-099** | \ |
| | | | **A01-A05** | These replace GDA 11-13 |
| **12A** | 10-08-98 | **117** | **100-199** | submitted 11-20-97 |
| **13A** | 10-08-98 | **117** | **200-244** | / |
| **14A** | 09-10-98 | **192** | **001-099** | \ |
| | | | **A01-A05** | (except unrecorded reel A02) |
| **15A** | 09-10-98 | **192** | **100-199** | |
| **16A** | 09-10-98 | **192** | **200-299** | These replace GDA 14-19 |
| **17A** | 09-10-98 | **192** | **300-399** | submitted 01-27-98 |
| **18A** | 09-10-98 | **192** | **400-499** | |
| **19A** | 09-10-98 | **192** | **500-517** | / |
| **20A** | 07-02-98 | **193** | **001-099** | \ |
| | | | **A01-A05** | |
| **21A** | 07-02-98 | **193** | **100-199** | |
| **22A** | 07-02-98 | **193** | **200-299** | These replace GDA-20-25 |
| **23A** | 07-02-98 | **193** | **300-399** | submitted 03-24-98 |
| **24A** | 07-02-98 | **193** | **400-499** | |
| **25A** | 07-02-98 | **193** | **500-538** | / |
| **26A** | 03-31-98 | **110** | **001-005** | \ |
| | | | **A01-A05** | |
| | | **116** | **001-005** | |
| | | | **A01-A05** | |
| | | **117** | **001-005** | This replaces GDA-26 |
| | | | **A01-A05** | submitted 03-31-98 |

**(26A continued on next page)**

## Table 4-4 Data Return Records by GDA Disk page 2 of 3

| GDA Disk Number | Date Submitted | Deck | Reels | Notes |
|---|---|---|---|---|
| (**26A** cont from page 4-9) | | **118** | **001-005** | |
| | | | **A01-A05** | |
| | | **119** | **001-005** | |
| | | | **A01-A05** | |
| | | **184** | **001-005** | |
| | | | **A01-A05** | |
| | | **192** | **001-005** | This replaces GDA-26 |
| | | | **A01-A05** | submitted 03-31-98 |
| | | **193** | **001-005** | |
| | | | **A01-A05** | |
| | | **195** | **001-005** | |
| | | | **A01-A05** | |
| **27** | 07-22-98 | **184** | **001-064** | -- (except reel 012, to be recorded when recvd) |
| | | | **A01-A05** | |
| **28** | 07-22-98 | **119** | **001-077** | |
| | | | **A01-A05** | |
| | | **196** | **001-016** | |
| | | | **A01** | |
| | | **197** | **001-003** | -- (except reel 002, to be recorded on GDA-37) |
| | | | **A01** | |
| **29** | 07-02-98 | **118** | **001-099** | -- (except reels 078,079,098, to be recorded on GDA-37) |
| | | | **A01-A05** | |
| **30** | 07-22-98 | **118** | **100-145** | -- (except reel 119, to be recorded when recvd) |
| | | **187** | **001-002** | |
| | | | **A01** | |
| | | **190** | **001-004** | |
| | | | **A01** | |
| | | **256** | **748-776** | |
| | | | **A48** | |
| | | **650** | **001-007** | |
| | | | **A01** | |
| | | **690** | **001** | |
| | | | **A01** | |
| **31** | 08-05-98 | **110** | **001-066** | |
| | | | **A01-A05** | |

## Table 4-4 Data Return Records by GDA Disk page 3 of 3

| GDA Disk Number | Date Submitted | Deck | Reels | Notes |
|---|---|---|---|---|
| **32** | 08-05-98 | **195** | **001-054**<br>**A01-A05** | |
| **33** | 08-05-98 | **128** | **001-099**<br>**A01-A04** | |
| **34** | 08-05-98 | **128** | **100-186** | (except reel 112, to be recorded on GDA-37 |
| **35** | 08-05-98 | **128** | **208-250** | |
| **36** | 08-05-98 | **281** | **001-097**<br>**A01-A05** | |
| **37** | GDA | **118** | **078,079** | To be recorded |
| | | **118** | **098** | To be recorded |
| | | **128** | **112** | To be recorded |
| | | **197** | **002** | To be recorded |

Notes:

GDA indicates material remaining at Greenough Data Associates, 10-30-98

5.0 <u>Remnant actions</u>

At the date of report preparation, certain items in the project are incomplete. Most evident is that a few reels have not yet been received, or at least, not in a form suitable for scanning.

| | | |
|---|---|---|
| | Deck 118 | Reel 119 not yet received |
| | Deck 184 | Reel 012 not yet received |
| (1) | Deck 194 | Reels not yet received in scannable form: 062, 089, |
| (2) | Deck 541 | Reels 001- 020 not yet received |
| | Deck 547 | Reel 001 not yet received |

The total of the above yet to be scanned appears to be 25 reels. It is regarded as an obligation of the project to process these reels upon receipt. However substitutions are acceptable, even if the total number may somewhat exceed the originally specified number of 3046.

Several decks were re-scanned with a newer version of the program to enhance recovery of punched data under marginal conditions. To assure replacement of the earlier data with the new information, the following decks should be recopied:

| | |
|---|---|
| Deck 117 | Replace data files with new version on GDA-11A through -13A |
| Deck 192 | Replace data files with new version on GDA-14A through -19A |
| Deck 193 | Replace data files with new version on GDA-20A through -25A |
| Deck 194 | Replace data files with new version on GDA-01A through -04A |

The original files have identical Zip disk nomenclature without the suffix A.

<u>Suplemental Notes, December 3, 1998</u>

Notes:

1) In Deck 194, reel 062 needs to be copied with lower density
   Reel 089 has been processed and included in the data files on GDA-1A
2) In Deck 541, all 21 reels have been processed and included on GDA-37

DECKNOTES

## 6.0 Processing notes by Deck

The following pages of Section 6 contain information on these topics below for every deck that was scanned in the project:

**Scanned:** Date the of the final submitted record. For Decks 117, 192, 193 and 194, the dates are those of the second runs, whose data files replace the first versions.

**Recorded on:** Zip disk identification on which the files are recorded. For example, GDA-31 contains the listed files for Deck 110.

**Calibration failure rates:** "Mean for deck," the total number of counted calibration failures, divided by the number of reels scanned in the deck. Failures resulted in 80 "space" characters instead of erroneous data. In some cases extra failures were created by film anomalies, leading to an apparent higher count. When causes for excessive failure counts were determined, these are noted.

"Failure rate," the total number of calibration failures in the deck, divided by the number of cards scanned. The overall rate for all scanned decks was 0.015 percent of cards failing calibration.

**Difference test data:** In the reel pairs shown, the number of cards with one or more differences in apparent punched data, divided by the number of card pairs compared, i.e. the number of cards on one reel.
"Mean for sample," the total number of cards with differences in the sample group, divided by the number of card pairs. The overall rate for cards with differences was 0.17 percent.

**Notes:** Comments on the processing

Individual data sheets follow for decks in numeric order.

Processing Notes by Deck

## Deck 110, Navy Marine, 1945-June 1951

Scanned:

| | | |
|---|---|---|
| 001-005 | 04-26 & 05-15-98 | |
| A01-A05 | 04-26 & 05-15-98 | |
| 006-066 | 05-26 to 05-27-98 | |

Recorded on:

| | |
|---|---|
| 001-066 | GDA-31 |
| A01-A05 | GDA-31 |

Calibration failure rate:

Mean for deck = 2.5 — Unreadable(1)cards per reel

Failure rate = 0.020 percent — Cards found unreadable

Difference test data:

| | | | |
|---|---|---|---|
| 001 vs A01 | 0.31 percent | |
| 002 A02 | 0.40 | |
| 003 A03 | 0.17 | |
| 004 A04 | 0.61 | |
| 005 A05 | 0.31 | Data from 05-15-98, submitted on \floppy disks of 05-18-98 |

Mean for sample 0.36 percent — Cards with differences in duplicate runs

Notes:

1) Many of these cards were severely clipped at top of right side, causing calibration failures, a specific example is reel 064.

## Processing Notes by Deck

### Deck 116, Surface Marine, 1949-June 1963

Scanned:

| | | |
|---|---|---|
| 001-596 | 10-24 to 11-08-97 | Except reel 343, not received \at time of first scanning |
| 343 | 07-22-98 | Received later |
| A01-A05 | 07-22 to 07-22-98 | |

Recorded on:

| | | |
|---|---|---|
| 001-099 | GDA-5 | Reels 001-005 to be replaced |
| 001-005 | GDA-26A | Replacement for reels 001-005 |
| A01-A05 | GDA-26A | Replacement for reels A01-A05 |
| 100-199 | GDA-6 | |
| 200-299 | GDA-7 | |
| 300-399 | GDA-8 | Reel 343 missing |
| 343 | GDA-19A | Includes reel 343 |
| 400-499 | GDA-9 | |
| 500-596 | GDA-10 | |

Calibration failure rate:

| | |
|---|---|
| Mean for deck = 1.1 | Unreadable cards found per reel |
| Failure rate = 0.009 percent | Cards found unreadable |

Difference test data:

| | | | |
|---|---|---|---|
| 001 vs | A01 | 0.20 percent | |
| 002 | A02 | 0.22 | |
| 003 | A03 | 0.11 | |
| 004 | A04 | 0.25 | |
| 005 | A05 | 0.16 | |
| Mean for sample | | 0.19 percent | Cards with differences in duplicate runs |

Notes:

1) Acquire reel 343 data file from GDA-19A
2) Use GDA-26A for Difference tests, contains 001-005, A01-A05
3) Reel 051: many punches of rows 2 and 9 (illegal) in col 27
4) Reel 111: many illegal characters, light streak near col. 50
5) Reel 282: column 79 often punched in X and 9 rows
6) Reels 545,546: many blank cards

## Processing Notes by Deck

### Deck 117, Navy Marine

Scanned:

| | |
|---|---|
| 001-244 | 04-21 to 09-25-98 |
| A01-A05 | 04-21 to 04-22-98 |

Recorded on:

| | |
|---|---|
| 001-099 | GDA-11A |
| A01-A05 | " |
| 100-199 | GDA-12A |
| 200-244 | GDA-13A |

Calibration failure rate:

| | |
|---|---|
| Mean for deck = 3.2 | Unreadable cards found per reel |
| Failure rate = 0.027 percent | Cards found unreadable |

Difference test data:

| | | |
|---|---|---|
| 001 vs | A01 | 0.12 percent |
| 002 | A02 | 0.07 |
| 003 | A03 | 0.01 |
| 004 | A04 | 0.09 |
| 005 | A05 | 0.03 |

Mean for sample 0.07 percent — Cards with differences in duplicate runs

Notes:

1) Card jams in filming this deck were more common then usual.
2) Data shown are from second run of entire Deck 117. GDA disks with "A" suffix replace original submissions without suffix. GDA-11A through 13A directly replace GDA-11 through 13.

# Processing Notes by Deck

## Deck 118, Japanese Marine, 1937-June 1953

Scanned:

| | | |
|---|---|---|
| 006-145 | 05-18 to 05-20-98 | Reels 078,079,098,119 \not received |
| 001-005 | 04-23-98 | |
| A01-A05 | 04-23-98 | |
| 078-079 | 10-27-98 | Reels 078,079 |
| 098 | 10-27-98 | Reel 098 |

Recorded on:

| | | |
|---|---|---|
| 001-099 | GDA-29 | Except reels 078, 078, 098 |
| 100-145 | GDA-30 | Except reel 119, not received |
| 078,079 | GDA-37 | |
| 098 | GDA-37 | |

Calibration failure rate:

| | |
|---|---|
| Mean for deck = 1.4 | Unreadable cards found per reel |
| Failure rate = 0.012 percent | Cards found unreadable |

Difference test data:

| | | |
|---|---|---|
| 001 vs A01 | | 0.24 percent |
| 002 | A02 | 0.14 |
| 003 | A03 | 0.06 |
| 004 | A04 | 0.12 |
| 005 | A05 | 0.13 |

Mean for sample 0.14 percent — Cards with differences in duplicate runs

Notes:

1) Column 51 often punched 0/1, illegal character
2) Reel 114, cards with heavily cut corners, causing calibration failures
3) Reel 119 not yet recvd (12-03-98)

Processing Notes by Deck

## Deck 119 Japanese Marine, July 1953-1960

Scanned:

| | | |
|---|---|---|
| 006-0077 | 05-20 to | 05-21-98 |
| 001-005 | | 04-28-98 |
| A01-A05 | | 04-28-98 |

Recorded on:

| | |
|---|---|
| 001-077 | GDA-28 |
| A01-A05 | GDA-28 |

Calibration failure rate:

| | |
|---|---|
| Mean for deck = 1.1 | Unreadable cards found per reel |
| Failure rate = 0.009 percent | Cards found unreadable |

Difference test data:

| | | |
|---|---|---|
| 001 vs | A01 | 0.12 percent |
| 002 | A02 | 0.04 |
| 003 | A03 | 0.05 |
| 004 | A04 | 0.08 |
| 005 | A05 | 0.04 |

Mean for sample 0.07 percent — Cards with differences in duplicate runs

Notes:

1) Reel 068, cards with 0/1 punching, illegal character

## Processing Notes by Deck

### Deck 128 Surface Marine, July 1963-April 1968

Scanned:

| | | | |
|---|---|---|---|
| 001-250 | 06-03 to | 06-08-98 | Reel 112 not received |
| A01-A05 | 06-03 | 06-07-98 | |
| 112 | | 10-27-98 | Reel 112 recvd |

Recorded on:

| | | |
|---|---|---|
| 001-099 | GDA-33 | |
| 100-186 | GDA-34 | except reel 112 |
| 208-250 | GDA-35 | |
| 112 | GDA-37 | Reel 112 |

Calibration failure rate:

| | |
|---|---|
| Mean for deck = 0.8 | Unreadable cards found per reel |
| Failure rate = 0.007 percent | Cards found unreadable |

Difference test data:

| | | | |
|---|---|---|---|
| 001 vs | A01 | 0.05 percent | |
| 002 | A02 | 0.25 | |
| 003 | A03 | 0.06 | |
| 004 | A04 | 0.06 | |
| 005 | A05 | 0.04 | |
| Mean for sample | | 0.09 percent | Cards with differences in duplicate runs |

Notes:

## Processing Notes by Deck

### Deck 184 British Marine, April 1953-1961

Scanned:

| | | |
|---|---|---|
| 006-064 | 04-27 to 04-30-98 | Reel 012 not received |
| 001-005 | 04-20 05-13-98 | |
| A01-A05 | 04-20 05-13-98 | |

Recorded on:

| | |
|---|---|
| 001-064 | GDA-27 |
| A01-A05 | GDA-27 |

Calibration failure rate:

Mean for deck = 2.0 — Unreadable cards found per reel

Failure rate = 0.017 percent — Cards found unreadable

Difference test data:

| | | |
|---|---|---|
| 001 vs | A01 | 0.41 percent |
| 002 | A02 | 0.61 |
| 003 | A03 | 0.15 |
| 004 | A04 | 0.34 |
| 005 | A05 | 0.26 |

Mean for sample 0.35 percent — Cards with differences in duplicate runs

Notes:

1) Although the workflow statement listed 53 reels, 64 were received. All were scanned. Images in Reels 054 through 064 were much denser than normal, requiring program customization.
2) Reel 012 to be scanned when received.

# Processing Notes by Deck

## Deck 185 USSR Marine, 1957-1958

Scanned:

| | |
|---|---|
| 001-010 | 05-23-98 |
| A01 | 05-23-98 |

Recorded on:

| | |
|---|---|
| 001-010 | GDA-27 |
| A01 | GDA-27 |

Calibration failure rate:

| | |
|---|---|
| Mean for deck = 0.9 | Unreadable cards found per reel |
| Failure rate = 0.008 percent | Cards found unreadable |

Difference test data:

| | | |
|---|---|---|
| 001 vs A01 | 0.07 percent | |
| Mean for sample | 0.07 percent | Cards with differences in duplicate runs |

Notes:

1) Deck 185 small, difference sample limited to one reel

Processing Notes by Deck

## Deck 187 Japanese Whaling

Scanned:

| | |
|---|---|
| 001-002 | 05-28-98 |
| A01 | 05-28-98 |

Recorded on:

| | |
|---|---|
| 001-002 | GDA-30 |
| A01 | GDA-30 |

Calibration failure rate:

| | |
|---|---|
| Mean for deck = 0.5 | Unreadable cards found per reel |
| Failure rate = 0.004 percent | Cards found unreadable |

Difference test data:

| | | |
|---|---|---|
| 001 vs A01 | 0.07 percent | |
| Mean for sample | 0.07 percent | Cards with differences in duplicate runs |

Notes:

1) Deck 187 small, difference sample limited to one reel

# Processing Notes by Deck

## Deck 189 Dutch Marine

Scanned:

| | |
|---|---|
| 001-023 | 05-29 to 06-01-98 |
| A01 | 05-29-98 |

Recorded on:

| | |
|---|---|
| 001-023 | GDA-27 |
| A01 | GDA-27 |

Calibration failure rate:

| | |
|---|---|
| Mean for deck = 1.0 | Unreadable cards found per reel |
| Failure rate = 0.009 percent | Cards found unreadable |

Difference test data:

| | | |
|---|---|---|
| 001 vs A01 | 0.14 percent | |
| Mean for sample | 0.14 percent | Cards with differences in duplicate runs |

Notes:

1) Column 73 sometimes punched 0/1, illegal character
2) Deck 189 small, difference sample limited to one reel

# Processing Notes by Deck

## Deck 190 CG Log Book

Scanned:

| | |
|---|---|
| 001-004 | 05-28-98 |
| A01 | 05-28-98 |

Recorded on:

| | |
|---|---|
| 001-004 | GDA-30 |
| A01 | GDA-30 |

Calibration failure rate:

| | |
|---|---|
| Mean for deck = 0.8 | Unreadable cards found per reel |
| Failure rate = 0.006 percent | Cards found unreadable |

Difference test data:

| | | |
|---|---|---|
| 001 vs A01 | 0.05 percent | |
| Mean for sample | 0.05 percent | Cards with differences in duplicate runs |

Notes:

1) Deck 190 small, difference sample limited to one reel

# Processing Notes by Deck

## Deck 192 German Marine

Scanned:

| | | | |
|---|---|---|---|
| 007-517 | 07-06 to | 07-20-98 | |
| 001-006 | 04-24 | 04-25-98 | |
| A01-A05 | 04-24 | 04-25-98 | A02 omitted, card jam makes card count variable |

Recorded on:

| | | |
|---|---|---|
| 001-099 | GDA-14A | |
| A01-A05 | GDA-14A | A02 omitted(Note 1) |
| 100-199 | GDA-15A | |
| 200-299 | GDA-16A | |
| 300-399 | GDA-17A | |
| 400-499 | GDA-18A | |
| 500-517 | GDA-19A | |

Calibration failure rate:

Mean for deck = 1.8 — Unreadable cards found per reel

Failure rate = 0.015 percent — Cards found unreadable

Difference test data:

| | | |
|---|---|---|
| 001 vs | A01 | 0.24 percent |
| 003 | A03 | 0.16 |
| 004 | A04 | 0.24 |
| 005 | A05 | 0.14 |

Mean for sample 0.19 percent — Cards with differences in duplicate runs

Notes:

(1) A card jam caused non-reproducible card counts in multiple trials, disqualifying difference matching using A02. Meaningless or blank data may show on up to 20 cards on reel 002.
2) Many calibration failures due to image overexposure following card misfeeds
3) Reel 147 has long scratch near column 70
4) Reel 274 has long scratch near column 22
5) Reel 313 has long scratch near column 65
6) Reel 336 has high calibration failure count from cut edges of cards
7) Reels 341 and 351 may have artificially high calibration failure rate without data loss
8) Reel 378 has long streak near card 7476
9) Reel 414 has scratches near start
10) Data shown are from second run of entire Deck 192. GDA disks with "A" suffix replace original submissions without suffix. GDA-14A through 19A directly replace GDA-14 through 19.

# Processing Notes by Deck

## Deck 193 Dutch-Netherland

Scanned:

| | | |
|---|---|---|
| 001-538 | 06-10 to | 06-24-98 |
| A01-A05 | 06-10 | 06-12-98 |

Recorded on:

| | |
|---|---|
| 001-099 | GDA-20A |
| 100-199 | GDA-21A |
| 200-299 | GDA-22A |
| 300-399 | GDA-23A |
| 400-499 | GDA-24A |
| 500-538 | GDA-25A |

Calibration failure rate:

| | |
|---|---|
| Mean for deck = 2.0 | Unreadable cards found per reel* |
| Failure rate = 0.016 percent | Cards found unreadable* |

Difference test data:

| | | |
|---|---|---|
| 001 vs | A01 | 0.26 percent |
| 002 | A02 | 0.21 |
| 003 | A03 | 0.20 |
| 004 | A04 | 0.25 |
| 005 | A05 | 0.26 |

Mean for sample 0.24 percent — Cards with differences in duplicate runs

Notes:

* As a result of card jams during filming, possibly one-half of the total shown for this deck may be artificially generated, without actual loss of data from cards.

1) Column 13 with illegal character frequently punched 0/1
2) Column 19 with illegal character (sampled) punched ¼
3) Several reels showed narrow cards left border, required program customization.
4) Many calibration failures due to image overexposure following card misfeeds
10) Data shown are from second run of entire Deck 193. New GDA disks labeled 20A through 25A directly replace earlier numbers 20 through 25.

# Processing Notes by Deck

## Deck 194 British Marine, revised 12-03-98

Scanned:

| | | |
|---|---|---|
| 001-341 | 08-06 to 11-30-98 | Less reel 062, copy not yet \received in scannable form |
| A11-A15 | 10-08-98 | Sample for difference test, \see Note 1 |

Recorded on:

| | | |
|---|---|---|
| 001-099 | GDA-1A | Except reel 062 |
| 100-199 | GDA-2A | |
| 200-299 | GDA-3A | |
| 300-341 | GDA-4A | |

Calibration failure rate:

| | |
|---|---|
| Mean for deck = 2.3 | Unreadable cards found per reel |
| Failure rate = 0.019 percent | Cards found unreadable |

Difference test data:

| | | |
|---|---|---|
| 011 vs A11 | 0.15 percent | |
| 012 A12 | 0.32 | |
| 013 A13 | 0.23 | |
| 014 A14 | 0.06 | |
| 015 A15 | 0.18 | |
| Mean of sample | 0.19 percent | Cards with differences in duplicate runs |

Note:

1) After multiple runs of original sample A01-A05, reels A11-A15 were adopted as a more representative sample of deck 194.
2) Copy film of reel 062 will be scanned when received
Outstanding number of reels at current date (12-03-98): 1

# Processing Notes by Deck

## Deck 195 Navy Ship Logs, 1945-1945

Scanned:

| | |
|---|---|
| 006-054 | 05-22-98 |
| 001-005 | 04-28-98 |
| A01-A05 | 04-28-98 |

Recorded:

| | |
|---|---|
| 001-054 | GDA-32 |
| A01-A05 | GDA-32 |

Calibration failure rate:

| | | |
|---|---|---|
| Mean for deck | = 1.7 | Unreadable cards found per reel |
| Failure rate | = 0.014 percent | Cards found unreadable |

Difference test data:

| | | |
|---|---|---|
| 001 vs | A01 | 0.04 percent |
| 002 | A02 | 0.07 |
| 003 | A03 | 0.06 |
| 004 | A04 | 0.05 |
| 005 | A05 | 0.11 |

Mean of sample 0.07 percent — Cards with differences in duplicate runs

Notes:

1) Reel 024 has high calibration failure rate (0.09%) due to number of corner cuts at lower right edge of some cards

Processing Notes by Deck

## Deck 196 German Marine, 1949-1953

Scanned:

| | |
|---|---|
| 001-016 | 05-27 to 05-28-98 |
| A01 | 05-27-98 |

Recorded:

| | |
|---|---|
| 001-016 | GDA-28 |
| A01 | GDA-28 |

Calibration failure rate:

| | |
|---|---|
| Mean for deck = 0.4 | Unreadable cards found per reel |
| Failure rate = 0.003 percent | Cards found unreadable |

Difference test data:

| | | |
|---|---|---|
| 001 vs A01 | 0.10 | |
| Mean of sample | 0.10 percent | Cards with differences in duplicate runs |

Notes:

1) Deck 196 small, difference sample limited to one reel

# Processing Notes by Deck

## Deck 197 Danish Marine

Scanned:

| | | |
|---|---|---|
| 001-003 | 05-28, 1988 | Reel 002 omitted, not recvd |
| A01 | 05-28, 1998 | |
| 002 | 10-26, 1998 | Reel 002 recvd |

Recorded on:

| | | |
|---|---|---|
| 001-003 | GDA-28 | Reel 002 absent |
| A01 | GDA-28 | |
| 002 | GDA-37 | Reel 002 recvd and scanned |

Calibration failure rate:

| | | |
|---|---|---|
| Mean for deck = 0.4 | | Unreadable cards found per reel |
| Failure rate | = 0.003 percent | Cards found unreadable |

Difference test data:

| | | |
|---|---|---|
| 001 vs A01 | 0.10 percent | |
| Mean of sample | 0.10 percent | Cards with differences in duplicate runs |

Notes:

1) Reel box labelled 002 contained film of Deck 159, Reel 088
2) Deck 197 small, difference sample limited to one reel

## Processing Notes by Deck

### Deck 256 Surface Marine, 1963-

Scanned:

| | | |
|---|---|---|
| 748-776 | 06-02-98 | See note (1) below |
| A48 | 06-02-98 | |

Recorded on:

| | |
|---|---|
| 748-776 | GDA-30 |
| A48 | GDA-30 |

Calibration failure rate:

| | |
|---|---|
| Mean for deck = 1.9 | Unreadable cards found per reel |
| Failure rate = 0.016 percent | Cards found unreadable |

Difference test data:

| | | |
|---|---|---|
| 748 vs A48 | 0.12 percent | |
| Mean of sample | 0.12 percent | Cards with differences in duplicate runs |

Notes:

1) Original work statement specified 72 reels; 29 were received. All were scanned. Later information indicated that these may be of limited interest.
2) Deck 256 small, difference sample limited to one reel

# Processing Notes by Deck

## Deck 281 Navy MAR, Beg -1945

Scanned:

| | | |
|---|---|---|
| 001-097 | 06-25 to 06-30-98 | |
| A01-A05 | 06-25 | 06-26-98 |

Recorded on:

| | |
|---|---|
| 001-097 | GDA-36 |
| A01-A05 | GDA-36 |

Calibration failure rate:

| | |
|---|---|
| Mean for deck = 4.8 | Unreadable cards found per reel* |
| Failure rate = 0.40 percent | Cards found unreadable* |

Difference test data:

| | | | |
|---|---|---|---|
| 001 vs | A01 | 0.12 percent | |
| 002 | A02 | 0.17 | |
| 003 | A03 | 0.49 | |
| 004 | A04 | 0.20 | |
| 005 | A05 | 0.22 | |
| Mean of sample | | 0.24 percent | Cards with differences in duplicate runs |

Notes:

* More numerous defects in films of Deck 281 than normal, overexposed at misfeeds, many card jams

## Processing Notes by Deck

### Deck 541 USRAOBS, 1941-1947

Scanned:

| | |
|---|---|
| 021 | 10-27-98 |
| A21 | 10-27-98 |
| 001-020 | 11-30-98 |

Recorded on:

| | |
|---|---|
| 001-021 | GDA-37 |
| A21 | GDA-37 |

Calibration failure rate:

| | | |
|---|---|---|
| Mean for deck | = 5.8 | Unreadable cards found per reel |
| Failure rate | = 0.048 percent | Cards found unreadable |

Difference test data:

| | | |
|---|---|---|
| 021 vs A21 | 0.20 percent | |
| Mean of sample | 0.20 percent | Cards with differences in duplicate runs |

Notes:

1) Deck 541 small, difference sample limited to one reel

# Processing Notes by Deck

## Deck 547 Payerne RAOBS, 1942-1948

This reel will be scanned when received

# Processing Notes by Deck

## Deck 608 Hungary RAOBS, 1954-1958

Scanned:

| | |
|---|---|
| 001 | 10-28-98 |
| A01 | 10-28-98 |

Recorded on:

| | |
|---|---|
| 001 | GDA-37 |
| A01 | GDA-37 |

Calibration failure rate:

| | |
|---|---|
| Mean for deck = 1.0 | Unreadable cards found per reel |
| Failure rate = 0.008 percent | Cards found unreadable |

Difference test data:

| | | |
|---|---|---|
| 001 vs A01 | 0.09 percent | |
| Mean of sample | 0.09 percent | Cards with differences in duplicate runs |

Notes:

1) Deck 608 small, difference sample limited to one reel

## Processing Notes by Deck

### Deck 650 Formosan RAOBS

Scanned:

| | |
|---|---|
| 001-007 | 06-03-98 |
| A01 | 06-03-98 |

Recorded:

| | |
|---|---|
| 001-007 | GDA-30 |
| A01 | GDA-30 |

Calibration failure rate:

| | | |
|---|---|---|
| Mean for deck | = 2.7 | Unreadable cards found per reel |
| Failure rate | = 0.023 percent | Cards found unreadable |

Difference test data:

| | | |
|---|---|---|
| 001 vs A01 | 0.11 percent | |
| Mean of sample | 0.11 percent | Cards with differences in duplicate runs |

Notes:

1) Deck 650 small, difference sample limited to one reel

# Processing Notes by Deck

## Deck 690 Israel RAOBS

Scanned:

| | |
|---|---|
| 001 | 05-28-98 |
| A01 | 05-28-98 |

Recorded on:

| | |
|---|---|
| 001 | GDA-30 |
| A01 | GDA-30 |

Calibration failure rate:

| | | |
|---|---|---|
| Mean for deck | = 1.0 | Unreadable cards found per reel |
| Failure rate | = 0.009 percent | Cards found unreadable |

Difference test data:

| | | |
|---|---|---|
| 001 vs A01 | 0.14 percent | |
| Mean of sample | 0.14 percent | Cards with differences in duplicate runs |

Notes:

1) Deck 690 small, difference sample limited to one reel

# APPENDIX

# Apendix A-1 Coding Plan for Fosdic scanning of NCDC films

CODES FOR NUMERIC AND ALPHABETIC PUNCHED DATA

| Hollerith Rows | Fosdic Rows | Fosdic hex | Fosdic decimal | ASCII Nmbr | ASCII Char |
|---|---|---|---|---|---|
| 12 only | 1 only | 10h | 1 | 38 | & |
| 12 + 11 | 1 + 2 | 12h | 18 | 42 | ! |
| 12 + 0 | 1 + 3 | 13h | 19 | 42 | ) |
| 12 + 1 | 1 + 4 | 14h | 20 | 65 | A |
| 12 + 2 | 1 + 5 | 15h | 21 | 66 | B |
| 12 + 3 | 1 + 6 | 16h | 22 | 67 | C |
| 12 + 4 | 1 + 7 | 17h | 23 | 68 | D |
| 12 + 5 | 1 + 8 | 18h | 24 | 69 | E |
| 12 + 6 | 1 + 9 | 19h | 25 | 70 | F |
| 12 + 7 | 1 + 10 | 1Ah | 26 | 71 | G |
| 12 + 8 | 1 + 11 | 1Bh | 27 | 72 | H |
| 12 + 9 | 1 + 12 | 1Ch | 28 | 73 | I |
| 11 only | 2 + none | 20h | 2 | 45 | - |
| 11 + 0 | 2 + 3 | 23h | 35 | 42 | ( |
| 11 + 1 | 2 + 4 | 24h | 36 | 74 | J |
| 11 + 2 | 2 + 5 | 25h | 37 | 75 | K |
| 11 + 3 | 2 + 6 | 26h | 38 | 76 | L |
| 11 + 4 | 2 + 7 | 27h | 39 | 77 | M |
| 11 + 5 | 2 + 8 | 28h | 40 | 78 | N |
| 11 + 6 | 2 + 9 | 29h | 41 | 79 | O |
| 11 + 7 | 2 + 10 | 2Ah | 42 | 80 | P |
| 11 + 8 | 2 + 11 | 2Bh | 43 | 81 | Q |
| 11 + 9 | 2 + 12 | 2Ch | 44 | 82 | R |
| 0 only | 3 only | 30h | 3 | 48 | 0 |
| 0 + 1 | 3 + 4 | 34h | 52 | 45 | * |
| 0 + 2 | 3 + 5 | 35h | 53 | 83 | S |
| 0 + 3 | 3 + 6 | 36h | 54 | 84 | T |
| 0 + 4 | 3 + 7 | 37h | 55 | 85 | U |
| 0 + 5 | 3 + 8 | 38h | 56 | 86 | V |
| 0 + 6 | 3 + 9 | 39h | 57 | 87 | W |
| 0 + 7 | 3 + 10 | 3Ah | 58 | 88 | X |
| 0 + 8 | 3 + 11 | 3Bh | 59 | 89 | Y |
| 0 + 9 | 3 + 12 | 3Ch | 60 | 90 | Z |
| 1 only | 4 only | 40h | 4 | 49 | 1 |
| 1 + 2 | 4 + 5 | 45h | 69 | 42 | * |
| 1 + 3 | 4 + 6 | 46h | 70 | 42 | * |
| 1 + 4 | 4 + 7 | 47h | 71 | 42 | * |
| 1 + 5 | 4 + 8 | 48h | 72 | 42 | * |
| 1 + 6 | 4 + 9 | 49h | 73 | 42 | * |
| 1 + 7 | 4 + 10 | 4Ah | 74 | 42 | * |
| 1 + 8 | 4 + 11 | 4Bh | 75 | 42 | * |
| 1 + 9 | 4 + 12 | 4Ch | 76 | 42 | * |
| 2 only | 5 only | 50h | 5 | 50 | 2 |
| 2 + 3 | 5 + 6 | 56h | 86 | 42 | * |
| 2 + 4 | 5 + 7 | 57h | 87 | 42 | * |
| 2 + 5 | 5 + 8 | 58h | 88 | 42 | * |
| 2 + 6 | 5 + 9 | 59h | 89 | 42 | * |
| 2 + 7 | 5 + 10 | 5Ah | 90 | 42 | * |
| 2 + 8 | 5 + 11 | 5Bh | 91 | 42 | * |
| 2 + 9 | 5 + 12 | 5Ch | 92 | 42 | * |

## CODES FOR NUMERIC AND ALPHABETIC PUNCHED DATA (cont)

| Hollerith Rows | Fosdic Rows | Fosdic hex | Fosdic decimal | ASCII Nmbr | ASCII Char |
|---|---|---|---|---|---|
| 3 only | 6 only | 60h | 6 | 51 | 3 |
| 3 + 4 | 6 + 7 | 67h | 103 | 42 | * |
| 3 + 5 | 6 + 8 | 68h | 104 | 42 | * |
| 3 + 6 | 6 + 9 | 69h | 105 | 42 | * |
| 3 + 7 | 6 + 10 | 6Ah | 106 | 42 | * |
| 3 + 8 | 6 + 11 | 6Bh | 107 | 42 | * |
| 3 + 9 | 6 + 12 | 6Ch | 108 | 42 | * |
| 4 only | 7 only | 70 | 7 | 52 | 4 |
| 4 + 5 | 7 + 8 | 78 | 120 | 42 | * |
| 4 + 6 | 7 + 9 | 79 | 121 | 42 | * |
| 4 + 7 | 7 + 10 | 7A | 122 | 42 | * |
| 4 + 8 | 7 + 11 | 7B | 123 | 42 | * |
| 4 + 9 | 7 + 12 | 7C | 124 | 42 | * |
| 5 only | 8 only | 80h | 8 | 53 | 5 |
| 5 + 6 | 8 + 9 | 89h | 137 | 42 | * |
| 5 + 7 | 8 + 10 | 8Ah | 138 | 42 | * |
| 5 + 8 | 8 + 11 | 8Bh | 139 | 42 | * |
| 5 + 9 | 8 + 12 | 8Ch | 140 | 42 | * |
| 6 only | 9 only | 90h | 9 | 54 | 6 |
| 6 + 7 | 9 + 10 | 9Ah | 154 | 42 | * |
| 6 + 8 | 9 + 11 | 9Bh | 155 | 42 | * |
| 6 + 9 | 9 + 12 | 9Ch | 156 | 42 | * |
| 7 only | 10 only | A0h | 10 | 55 | 7 |
| 7 + 8 | 10 + 11 | ABh | 171 | 42 | * |
| 7 + 9 | 10 + 12 | ACh | 172 | 42 | * |
| 8 only | 11 only | B0h | 11 | 56 | 8 |
| 8 + 9 | 11 + 12 | BCh | 188 | 42 | * |
| 9 only | 12 only | C0h | 12 | 57 | 9 |

## CODES FOR DOUBLE OVERPUNCH WITH SINGLE NUMERIC PUNCH

| Hollerith Rows | Fosdic Rows | Fosdic hex | Fosdic decimal | ASCII Nmbr | ASCII Char |
|---|---|---|---|---|---|
| Y+X only | 1+2 only | EDh | 237 | 33 | ! |
| Y+X + 0 | 1+2 + 3 | F0h | 240 | 43 | + |
| " 1 | " + 4 | F1h | 241 | 94 | ^ |
| " 2 | " + 5 | F2h | 242 | 60 | < |
| " 3 | " + 6 | F3h | 243 | 62 | > |
| " 4 | " + 7 | F4h | 244 | 36 | $ |
| " 5 | " + 8 | F5h | 245 | 37 | % |
| " 6 | " + 9 | F6h | 246 | 61 | = |
| " 7 | " +10 | F7h | 247 | 91 | [ |
| " 8 | " +11 | F8h | 248 | 93 | ] |
| " 9 | " +12 | F9h | 249 | 92 | \ |

Notes: Sequence of rows in Hollerith code is:
12-11-0-1-2-3-4-5-6-7-8-9 or:
Y -X -0-1-2-3-4-5-6-7-8-9

Sequence in Fosdic scanning program is:
1-2-3-4-5-6-7-8-9-1-11-12

Remaining codes not shown above are all (42) "*"

# Appendix B-1 LDR.ASM Program

The following is a printout of the source program LDR.ASM described in Sections 3.0 to 3.1.5 of the text. Written in assembler language, it requires assembly using TASM LDR to create an LDR.OBJ file. Called subroutines are similarly assembled. The group of OBJ files is then linked with the instruction TLINK @LDR0. This yields the operating program LDR.EXE. The file LDR0 is a response file containing the names of all subroutines called on in execution of LDR.EXE. Appendix E lists the contents of LDR0. This approach makes LDR0 the equivalent of a library file but one that is more easily modified for program customization than the standard type.

There are two other files referred to in the program listing below. These are called INCLUDE files, here named SCAN_VAR and SCAN_PCD, for Scan Variables and Scan Procedures. The two files are definitions of the Global variables and procedures available in common to all elements of the overall program.

---

```
;LDR.ASM Program to run leader in on NCDC films
;Calibrates Gain with called sr GAIN, displays and records Gain DAC and MaxSig
;Measures CardSig and MinSig with called sr BKGND, displays and records same
;Stores output data in <Parameters> and creates file PARAM, as listed below
;Linked via LDR0

.386
IDEAL
MODEL SMALL
STACK 128

DATASEG

XC0      dw      7000              ;XC origin
YC0      dw      19100             ;YC origin
XF0      dw      4096              ;XF origin
YF0      dw      8000              ;YF origin
Thrld0    dw     400               ;signal threshold for frame scan stroke
XC       dw      ?                 ;working XC
YC       dw      ?                 ;working YC
XF       dw      ?                 ;working XF
YF       dw      ?                 ;working YF
XCFrame dw       ?                 ;XC when calling for Frame
;YCFrame dw       ?
XCFrameStop dw   ?                 ;XC left threshold for turning film drive off
ParamData dw     10 dup   (?)       ;ParamData +0 = Gain DAC
                                   ;           +2 = MaxSig
                                   ;           +4 = CardSig
                                   ;           +6 = MinSig
                                   ;+8 through 18 = Spare
ParamText db     80  dup (32),0;       ',20 dup (32),0 ;this must follow
Parameters
xy        dw     100 dup (20)
;(Parameter values are superimposed on above message for display purposes)
signon    db     0Ah,0Dh,'Film started',0        ;sign on message
stallmsg db      13,10,'Film Drive stalled, please restart program',0
filename db      'Param',0
handle   dw      5
Junk     dw      100 dup (?)       ;JIC
prtbuf   db      20  dup (?)       ;JIC
Data5    dw      ?
nmbrcdsscnd dw   ?
INCLUDE "Scan_var"
ParamHandle dw   ?
;FlagEoF dw       ?
Transmtce dw     800
```

```
CODESEG
INCLUDE "Scan_Pcd"

start:
        mov ax,@data
        mov ds,ax
        mov es,ax

        mov dx,316h             ;Reset
        mov ax,01
        out dx,ax
        mov ax,03               ;set deflection settling delay to minimum
        out dx,ax

        mov dx,308h             ;set initial Gain DAC for advancing film
        mov ax,512
        out dx,ax
        mov dx,30Ch             ;set Tilt to null
        mov ax,500
        out dx,ax
;--------------------------------------------------------------------------
                        ;Preparation for Frame sr
        mov ax,[XC0]            ;set XCFrame position XC0+2000
        add ax,2000
        mov [XCFrame],ax
                                ;randomize XCFrame to distribute wear on CRT
        mov ah,2Ch              ;use seconds and hundredths in time data, dx
        int 21h
        shr dx,5                ;divide by 32 to limit excursion to apx 500
                                ;mils
        add [XCFrame],dx        ;add to XCFrame to shift origin

        mov ax,[XC0]            ;set XCFrameStop position XC0+1500
        add ax,1500
        mov [XCFrameStop],ax

        mov [YCFrame],12000


        mov dx,304h             ;set and issue XF to initial value (center)
        mov ax,4096
        out dx,ax
        mov dx,306h             ;set and issue YF to initial value
        mov ax,8000
        out dx,ax

;        mov [CardSig],650       ;dummy values to create threshold for Frame
                                ;  sr
        mov [Transmtce],800
        mov [MinSig],50
;--------------------------------------------------------------------------
        call Frame              ;Runs film leader to first card and stops
                                ;  for entry of deck and reel numbers
;--------------------------------------------------------------------------
StallCheck:                     ;Want to go directly to Stop if Frame stalled
        mov dx,31Ah
        in ax,dx
        test ax,4h              ;bit 02 high indicates TakeUp Idler is slack
        jne StallStop
;--------------------------------------------------------------------------
        call WIPE              ;Activate CRT screen
;--------------------------------------------------------------------------
        call GAIN               ;Calibrate Gain, find Gain DAC and MaxSig
;--------------------------------------------------------------------------
       call BKGND               ;Calibrate CardSig and MinSig, add Spare
;--------------------------------------------------------------------------
```

```
        call CAL_DSPL               ;Display calibration parameters
;---------------------------------------------------------------------
        call ENTRY                  ;Add kbd Deck, etc data
;---------------------------------------------------------------------
        mov [ParamData+8], 0
        mov [ParamData+10],0
        mov [ParamData+12],0
        mov [ParamData+14],0
        mov [ParamData+16],0
        mov [ParamData+18],0

        mov [ParamData+43],49
        mov [ParamData+44],57
        mov [ParamData+45],57
        mov [ParamData+46],56

CreateParamFile:
        mov dx,offset filename          ;create and open PARAM file
        mov ah,03Ch
        mov cx,0
        int 21h
        mov [ParamHandle],ax

        mov bx,[ParamHandle]            ;write PARAM file
        mov cx,100
        mov ah,40h
        mov dx,offset paramdata
        int 21h
stop:
        mov ah,04Ch                     ;close file and exit LDR program
        int 21h
;---------------------------------------------------------------------
StallStop:
        mov cx,1
        mov di, offset stallmsg         ;Restart msg
        call strwrite
        mov cx,2
@wtl:   mov bx,65000
@wt2:    dec bx
        jnz @wt2
        loop @wtl
        jmp start

End start
```

## Appendix B-2 FRAME.ASM Subroutine procedure

This procedure is assembled with the command TASM FRAME, automatically including it in LDR0 and PCSCAN0. A description of its actions is covered in Section 3.1.1 for LDR0 and in more detail for PCSCAN0, in Sections 3.4 to 3.4.7.

```
;--------------------------------------------------------------------------------
;FRAME.ASM Procedure to frame on NCDC films

;Scan pattern makes strokes of 10 Reads, interlaced 6X, checked for
;  Clear/Dense at end of strokes
;Scan follows (top) edge of card designated by XCFrame from calling program
;Film drive stops when left side threshold XCFrameStop is reached
;Vertical origin is YCFrame from calling program; XF and YF are jammed to 4096
;  and 8000 in procedure and not referred to afterwards
;Horizontal origin is XCFrame loaded by calling program
;Threshold supplied by calling program

.386
IDEAL
MODEL SMALL

DATASEG
msg1    db       'Turn on Film Drive',0
ClearCount dw    0
EoFCount   dw    ?
Interlace  dw    50;100
StepF      dw    300
FlagEOF    dw    ?
XCStopped  dw    ?
FrmSig     dw    800                    ;default CardSig for setting threshold
YCFrame    dw    ?
Thrld      dw    ?
ScalTransm dw    ?

INCLUDE "Scan_Var"

CODESEG
INCLUDE "Scan_Pcd"
proc Frame
        pusha

        mov [FlagEoF],0              ;reset End of Film flag
        mov dx,310h             ;start capstan
        mov ax,01h
        out dx,ax

        mov dx,30Ah             ;set Focus for large spot
        mov ax,1000
        out dx,ax

        mov dx,304h             ;set XF to origin
        mov ax,4096
        out dx,ax
        mov dx,306h             ;set YF to origin
        mov ax,8000
        out dx,ax

        mov ax,[XCFrame]        ;put floor under XCFrame for at least 2-card
                                ;advance
        cmp ax,11000            ;make XCFrame at least 12000
        jg @13
        mov ax,12000
```

```
@13:    add ax,1000             ;advance XCFrame, the initial horiz location
        cmp ax,27000            ;limit deflection to 27,000
        jle @13a
        mov ax,27000
@13a:
        mov [XCFrame],ax        ;  of Frame Scan, 1000 above XCFrame
        mov [XC],ax
;---------------------------------------------------------------------------
                                ;Threshold calcs
        mov ax,[ScalTransm]     ;
        add ax,[Transmtce]
        shr ax,1
        mov [Thrld],ax
        cmp [Thrld],250;300         ;put floor of 250;300 in Thrld
        jg @14
        mov [Thrld],250;300
@14:
        cmp [Thrld],500         ;and ceiling of 500
        jl @15
        mov [Thrld],500
@15:
;----------------------------------------------------------------------------
NewRaster:
        mov [Interlace],0       ;reset Interlace

        mov cx,6                ;set for 6 vertically interlaced strokes
NewStroke:                      ;here at start of every stroke
        mov [ClearCount],0

        mov dx,302h             ;send out first YC
mov ax,[YCFrame];               ;vertical retraction from YC0 for sweep origin
        add ax,[Interlace]
        mov [YC],ax
        out dx,ax

        mov dx,300h             ;;set current XC.  Limits are XCFrame, max
        mov ax,[XC]             ;;                    "    "  XCFrameStop, min
        out dx,ax               ;issue XC
                                ;coords all loaded for first read in stroke
        mov bx,1200             ;reduce duty cycle of beam with 70µs delay
dly1:   dec bx                  ;  between strokes
        jne dly1
        mov [EoFCount],0

        mov dx,318h             ;start Read at first position in stroke
        in ax,dx

        push cx                 ;pushed from Interlace to Read counting
        mov cx,10               ;No. of Reads in column
Read:
        cmp cx,1                ;Skip increment below if last read in stroke
        je @@1

        mov dx,302h
        mov ax,[YC]
        add ax,[StepF]          ;add YC step
        mov [YC],ax             ;save new YC
        out dx,ax

@@1:    mov dx,318h             ;Reads in loop
@10:    in ax,dx
        test ax,8000h           ;wait for end of Read
        jne @10
        and ax,1023             ;leave only useful bits
```

```
        cmp ax,1023               ;check for End of Film as indicated by
        jne @18                   ;saturated photosignal
        inc [EoFCount]
        cmp [EoFCount],5
        jb @18
        mov [FlagEoF],1
@18:    cmp ax,[Thrld]            ;looking for signals above threshold
        jl contin1                ;discard low signals and skip to continue
        inc [ClearCount]          ;increment ClearCount if high signal
contin1:
        loop Read
        pop cx                    ;popped for return from Read to Interlace
                                  ;  counting
                         ;Through here at end of every stroke of 10 Reads
;-----------------------------------------------------------------------------
StallCheck:
        mov dx,31Ah               ;;Want to terminate Frame Scan when
        in ax,dx                  ;;  TakeUp Idler is slack (318h bit 02 high)
        test ax,004h              ;;  return immediately to calling program
        jne Stop                  ;;  to prevent sustained Frame Scan
;-----------------------------------------------------------------------------
StopCheck:                        ;Check edge-following XC with threshold below
        mov ax,[XCFrameStop]      ; which film drive is to be turned off
        cmp [XC],ax               ;want to stop when followed edge reaches
        jg contin5                ;  left side at XC=XCFrameStop, then jump out
                                  ;through here if left side reached
        mov dx,310h               ;tell film drive to stop
        mov ax,0
        out dx,ax

contin5:                          ;want scan to continue until stopped indicated
        mov dx,31Ah               ;checking for Stopped Indicator (00h low)
        in ax,dx                  ;
        test ax,1h
        jne contin6
        jmp Stop                  ;  to Stop
;-----------------------------------------------------------------------------

DataAnalysis:                     ;Was this stroke in clear (card) or
                                  ;dense (scale) area of image?
contin6:
        cmp [ClearCount],8      ;check Clear count at end of every stroke of 10
        jng DriveContin           ;requires 9 or more high signals in stroke
                                  ;  to be called Clear
                                  ;executes next instruction if Clear
        sub [XC],200              ;shove XC down when in clear zone
;-----------------------------------------------------------------------------
DriveContin:             ;through here until followed edge reaches left side
        add [Interlace],100       ;move stroke YC down 100 for interlace
        loop toNewStroke
        jmp toNewRaster
toNewStroke:
        jmp NewStroke
                          ;through here at end of every raster of 6 strokes
toNewRaster:
        jmp NewRaster             ;repeat raster of interlaced strokes
;-----------------------------------------------------------------------------
Stop:   mov dx,302h               ;set YC to origin
        mov ax,[YC0]
        mov [YC],ax
        out dx,ax                 ;issue YC0
        popa
        ret

ENDp  Frame                       ;Normal exits - 31Ah bit 02=0
END                               ;Emergency       "     "  =1   TakeUp Idler
                                  ;  is slack, need to turn off Beam
```

## Appendix B-3 WIPE.ASM Subroutine procedure

This procedure is assembled with the command TASM WIPE, automatically including it in the linking file LDR0. A description of its action is given in Section 3.1.2.

```
;-----------------------------------------------------------------------
;WIPE.ASM Procedure to activate CRT phosphor

.386
IDEAL
MODEL SMALL

DATASEG
signon  db      0Ah,0Dh,'Activate CRT screen',0
space   dw      ' ',0
XC0Wipe dw      (5000)
YC0Wipe dw      (10000)
INCLUDE "Scan_Var"

CODESEG
INCLUDE "Scan_Pcd"
proc wipe
        mov dx,308h     ;set Gain
        mov ax,512
        out dx,ax
        mov dx,30Ah     ;set Focus to max
        mov ax,1000
        out dx,ax
        mov dx,30Ch     ;set Tilt to null
        mov ax,512
        out dx,ax

        mov dx,304h     ;set XF to center
        mov ax,4096
        out dx,ax
        mov dx,306h     ;set YF to center
        mov ax,4096
        out dx,ax

        mov dx,316h
        mov ax,01       ;reset
        out dx,ax
        mov ax,12h      ;enable counter
        out dx,ax
        mov ax,03       ;Settling delay=0
        out dx,ax

        mov dx,300h     ;set XC at start of raster
        mov ax,[XC0Wipe]
        out dx,ax
        mov [XC],ax
;-----------------------------------------------------------------------

NewCol:                         ; here at start of column
        mov dx,302h             ;set YC origin of column
        mov ax,[YC0Wipe]
        out dx,ax
        mov [YC],ax

        mov dx,300h
        add [XC],25
        mov ax,[XC]             ;advance XC for next column
        out dx,ax
        mov [XC],ax
```

```
;-----------------------------------------------------------------------------
        mov dx,302h             ;send out first YC
        mov ax,[YC]
        out dx,ax
;--------------------------------------------------------------------------
        mov bx,100              ;settling delay
dly1:   dec bx
        jne dly1
;--------------------------------------------------------------------------

        mov dx,318h             ;start Read at first position in column
        in ax,dx

Read:   mov dx,302h
        add [YC],50             ;YC step
        mov ax,[YC]
        out dx,ax
        mov [YC],ax

        mov dx,318h             ;Read in loop
@10:    in ax,dx
        test ax,8000h           ;wait for end of Read
        jne @10

        cmp [YC],23000          ;check for YC at end of column
        jle Read                ;go back for more Reads until YC>23000

        cmp [XC],28000          ;check for end of raster at end of each column
        jle NewCol              ;repeat columns until XC>28000

        ret
Endp   wipe
End
```

# Appendix B-4 GAIN.ASM Subroutine procedure

This procedure is assembled with the command TASM GAIN, automatically including it in LDR0 and PCSCAN0. A description of its action is given in Section 3.3.3.

```
;------------------------------------------------------------------------------
;GAIN.ASM Procedure to calibrate Gain
;Generates slanted stroke in clear area of film above card image on screen
;Aims for max photosignal of 980 on scale of 1023
;Finds mean of 4 highest signals in stroke of 10 Reads
;Varies Gain DAC input in successive approximations to achieve 980
;Makes 3 trials with open-end corrections
;Output is Gain (DAC) at Parameters+0, MaxSig at Parameters+2

.386
IDEAL
MODEL SMALL

DATASEG
;-----------------------------------------------------------------------------
;Fudges
GainFudge  dw   0;-200;0       ;Fudge to reduce Gain from calculated value,
                                ;  may be required on dense copy films
;-----------------------------------------------------------------------------
Data0      dw   10 dup (0)     ;memory area for 10 Reads in slanted Gain
stroke
XC0Gain    dw   15000          ;XC origin for Gain scan
YC0Gain    dw   6500           ;YC origin for Gain scan, also YF=8000 (-4096)

Thrld0     dw   300            ;initial threshold
Thrld1     dw   300            ;working threshold
ThrldSum   dw   0              ;accumulated sum of 4 peak signals in Gain
                                ;  scan
Gain0      dw   512            ;initial Gain DAC setting
Gain1      dw   ?              ;working and final Gain DAC value
NmbrTries  dw   ?              ;iterations in Gain calibration settling
MaxSig     dw   ?              ;mean of 4 highest signals in Gain scan stroke
PeakPos    dw   ?              ;location of intermediate peaks in 4 passes
;ParamData  dw   ?              ;Storage location for parameter data as
generated

INCLUDE "Scan_Var"

CODESEG
INCLUDE "Scan_Pcd"
proc Gain
        mov dx,308h            ;set initial Gain value
        mov ax,[Gain0]
        mov [Gain1],ax
        out dx,ax

        mov dx,30Ah            ;set Focus to max
        mov ax,1000
        out dx,ax

        mov dx,316h
        mov ax,33              ;Settling delay=3
        out dx,ax

        mov dx,304h            ;set initial XF
        mov ax,[XF0]
        mov [XF],ax
        out dx,ax
```

```
        mov dx,306h             ;set initial YF
        mov ax,[YF0]
        mov [YF],ax
        out dx,ax

        mov [NmbrTries],0       ;initialize iteration counter
;-----------------------------------------------------------------------------
NewStroke:
        mov dx,302h             ;set YC at start of stroke
        mov ax,[YC0Gain]
        out dx,ax
        mov [YC],ax
        mov di,0

        mov dx,300h             ;send out first XC
        mov ax,[XC0Gain]
        mov [XC],ax
        out dx,ax               ;coords all loaded for first read in stroke

        mov bx,30               ;settling delay
dly1:   dec bx
        jne dly1

        mov dx,318h             ;start Read at first position in stroke
        in ax,dx

        mov cx,10               ;No. of Reads in stroke
Read:
        cmp cx,1                ;Skip increment below if last read in stroke
        je @@1

        mov dx,300h             ;create slanted stroke of 10 Reads
        add [XC],600            ;XC step
        mov ax,[XC]
        out dx,ax
        mov dx,302h             ;YC step
        add [YC],100
        mov ax,[YC]
        out dx,ax

@@1:    mov dx,318h             ;Read in loop
@10:    in ax,dx
        test ax,8000h           ;wait for end of Read
        jne @10
        and ax,1023
        mov [Data0+di],ax
        inc di
        inc di
        loop Read
                                ;Here at end of stroke
        Jmp DataAnalysis
DataAnalysisRtn:
        inc [NmbrTries]
        cmp [NmbrTries],3
        jge Stop                ;Go out after 3 tries
        jmp NewStroke           ;Path to keep trying 3 times
;-----------------------------------------------------------------------------
DataAnalysis:           ;Want mean of 4 highest signals in stroke of 10
        mov [ThrldSum],0        ;reset sum of 4 peaks
        mov cx,4                ;set cx for repeating to find the 4 highest
                                ;  peaks
@@7:                            ;recycle to here for 4-peak search
        mov di,0                ;reset data pointer
        mov bx,[Thrld0]
        mov [Thrld1],bx         ;Thrld1 now has initial value for pass
                                ;  through 10 data values
```

```
        push cx                 ;push from peak counting to cycling through
                                ;  Data0
        mov cx,10               ;Examine 10 data values
@@5:    mov ax,[Data0+di]
        cmp ax,[Thrld1]
        jl @@6                  ;jump to @@6 if signals below initial
threshold
        mov [Thrld1],ax         ;save amlitude and position if signals above "
        mov [PeakPos],di
@@6:
        inc di
        inc di
        loop @@5
        pop cx                  ;pop cx from cycling data to counting peaks
                                ;Thrld1 has the highest in each execution of
10X
                                ;PeakPos has its position in terms of di
        mov bx,[PeakPos]
        mov [Data0+bx],0        ;Zero out the interim peak value so next
        mov ax,[Thrld1]         ;  peak can be found.  Then go through 10
again
        add [ThrldSum],ax       ;ax has Thrld1 resulting from 10X looping
        loop @@7                ;go back to peak-seeking for total of 4 times
        mov ax,[ThrldSum]
        shr ax,2                ;divide sum by 4 to derive mean
        mov [MaxSig],ax         ;End of Data Analysis
                                ;MaxSig has mean of 4 highest peaks,
                                ;  which is the effective gain indicator
                                ;continue below with Set Gain
;-----------------------------------------------------------------------------
SetGain:                ;Next, want to change gain so that mean of peaks
                                ;  is approximately 980 after 3 tries
        mov dx,308h             ;apply correction to Gain, Port 308h

        mov ax,980
        sub ax,[MaxSig]         ;ax has correction, the signed departure
                                ;  from 980. Result is positive for low gain
        add ax,[Gain1]          ;Sum correction and prior value of Gain
        mov [Gain1],ax
@1:
        mov [ParamData],ax     ;for transfer and display of Gain DAC
        out dx,ax
                          ;Gain is now calibrated
;-----------------------------------------------------------------------------
        mov cx,1
        cmp [NmbrTries],2
        jne @@11
        mov ax,[MaxSig]         ;for save and display of max signals
        mov [ParamData+2],ax
@@11:
        jmp DataAnalysisRtn
;----------------------------------------------------------------------------
stop:
        mov dx,308h             ;apply Gain Fudge here
        mov ax,[Gain1]
        add ax,[GainFudge]
        out dx,ax

        ret
ENDp Gain
END
```

## Appendix B-5 BKGND.ASM Subroutine procedure

This procedure is assembled with the command TASM BKGND, automatically including it in the linking file LDR0. A description of its action is given in Section 3.1.3.

```
;---------------------------------------------------------------------------
;BKGND.ASM Procedure to scan widely across film card and left edge band,
;  finding typical CardSig and MinSig
;Input is MaxSig at Parameters+2
;Output is CardSig at Parameters+4, MinSig at Parameters+6
;Makes a single vertical stroke of 40 Reads, beginning over card stock and
;  ending below dense band (YCMargin)
;Finds CardSig as mean of 4 highest signals in first 10 Reads in stroke
;Finds MinSig as mean of 4 lowest signals in last 20 Reads

.386
IDEAL
MODEL SMALL

DATASEG
XC0Bkgnd dw     9000            ;fixed XC origin
YC0Bkgnd dw     13000           ;fixed YC origin
Data0    dw     40 dup (0)      ;Stored amplitudes for 40 reading positions
                                ;  used for finding minimum signal mean
Data1    dw     40 dup (0)      ;Duplicate of above used in moment calculations
Data2    dw     40 dup (0)      ;Storage of modified amplitudes in calculations
Thrld0   dw     400             ;fixed initial value
Thrld1   dw     400             ;working threshold
ThrldSum dw     0               ;working sum of peak or valley signal values
MinSig   dw     (?)             ;mean of 4 lowest signals in Bkgnd stroke
CardSig dw      (?)             ;mean of 4 highest signals over card image
Junk1    dw     (?)             ;JIC
ValleyPos dw    (?)             ;working YCs needed for clearing in 4-pass
                                ;  routines
PeakPos dw      (?)             ;same for developing mean CardSig
INCLUDE "Scan_Var"

CODESEG
INCLUDE "Scan_Pcd"
proc Bkgnd

        mov dx,40Ah             ;set Focus to max
        mov ax,1000
        out dx,ax

        mov dx,316h
        mov ax,33               ;Settling delay=3
        out dx,ax
;---------------------------------------------------------------------------

NewStroke:
        mov di,0                ;di points to YC positions in stroke

        mov dx,300h             ;set XC at start of stroke
        mov ax,[XC0Bkgnd]
        out dx,ax
        mov [XC],ax

        mov dx,302h             ;send out first YC
        mov ax,[YC0Bkgnd]
        mov [YC],ax
        out dx,ax               ;coords all loaded for first read in stroke

        mov bx,30               ;settling delay
dly1:   dec bx
        jne dly1
```

```
        mov dx,318h              ;start Read at first position in stroke
        in ax,dx

        mov cx,40                ;No. of Reads in stroke
Read:
        cmp cx,1                 ;Skip increment below if last read in stroke
        je @@1

        mov dx,302h
        add [YC],100             ;YC step
        mov ax,[YC]
        out dx,ax
        mov dx,300h              ;XC step
        mov ax,[XC]
        add ax,16
        out dx,ax
        mov [XC],ax

@@1:    mov dx,318h              ;Read in loop
@10:    in ax,dx
        test ax,8000h            ;wait for end of Read
        jne @10
        and ax,1023
        mov [Data0+di],ax        ;at termination Data0+ has 40 signal values

        mov [Data1+di],ax        ;  Data1+ has same values
        inc di
        inc di
        loop Read
                                 ;Here at end of stroke
        jmp DataAnalysis
DataAnalysisRtn:                 ;Return from Data Analysis
        jmp Out1
;---------------------------------------------------------------------------

DataAnalysis:   ;Want mean of 4 highest values in first 10 reading positions
        mov [ThrldSum],0         ;clear threshold sum
        mov cx,4                 ;set cx for repeating to find 4 highest peaks
@@70:                            ;recycle to here for 4-peak search
        mov di,0                 ;clear data pointer to first of 40 Reads
        mov bx,[Thrld0]
        mov [Thrld1],bx          ;Thrld1 now has initial value for passes
                                 ;  through first 10 data points
        push cx                  ;push cx from peak counting to cycling through
                                 ;  Data0
        mov cx,10                ;Examine 10 amplitudes
@@50:
        mov ax,[Data0+di]
        cmp ax,[Thrld1]
        jl @@60                  ;jump to @@60 if signals below threshold
        mov [Thrld1],ax          ;save new Thrld1
        mov [PeakPos],di         ;save di as pointer to YC where peak occurred

@@60:
        inc di
        inc di
        loop @@50
        pop cx                   ;pop cx from cycling data to counting peaks
                                 ;Thrld1 has highest in each execution of 10X
                                 ;PeakPos has its position in terms of di
        mov bx,[PeakPos]
        mov [Data0+bx],0         ;clear last peak amplitude to find next highest
        mov ax,[Thrld1]

        add [ThrldSum],ax        ;ax has threshold sum accumulated during 10X
                                 ;  looping
        loop @@70                ;go back to peak-seeking for total of 10 times
```

```
        mov ax,[ThrldSum]
        shr ax,2                ;divide sum by 4 for mean
        mov [CardSig],ax        ;save as Card Signal
        mov [ParamData+4],ax   ;save for transfer and display
;------------------------------------------------------------

@2:                             ;Next, want mean of 4 lowest signals in
                                ;  last 20 of 40 strokes
        mov [ThrldSum],0        ;reset sum of 4 valleys

        mov cx,4                ;set cx for repeating to find the 4 lowest
                                ;  valleys
@@7:                            ;recycle to here for 4-valley search
        mov di,40               ;reset data pointer
        mov bx,[Thrld0]
        mov [Thrld1],bx         ;Thrld1 now has initial value for pass
                                ;  through last 20 data values
        push cx                  ;push from valley counting to cycling through
                                ;  Data0
        mov cx,20               ;Examine last 20 data values
@@5:    mov ax,[Data0+di]

        cmp ax,[Thrld1]         ;compare signal amplitude against working
                                ;  threshold
        jg @@6
        mov [Thrld1],ax
        mov [ValleyPos],di      ;save valley position in terms of di
@@6:    inc di
        inc di
        loop @@5
        pop cx                  ;pop cx from cycling data to counting valleys
                                ;Thrld1 has the lowest in each execution of 20X
                                ;ValleyPos has its position in terms of di
        mov bx,[ValleyPos]
        mov [Data0+bx],999      ;max out the interim valley value so next
        mov ax,[Thrld1]         ;  valley can be found.  Then go through 20 again
        add [ThrldSum],ax       ;ax has Thrld1 resulting from 20X looping
        loop @@7                ;go back to valley-seeking 4X

        mov ax,[ThrldSum]
        shr ax,2                ;divide sum by 4 to derive mean (MinSig)
        mov [MinSig],ax         ;End of Data Analysis
                                ;MinSig has mean of 4 valleys
        mov [ParamData+6],ax   ;save MinSig

Out1:
        ret
ENDp Bkgnd
END
```

## Appendix B-6 CAL_DSPL.ASM Subroutine procedure

This procedure is assembled with the command TASM CAL_DSPL, automatically including it in the linking file LDR0. A description of its action is given in Section 3.1.4.

```
;-------------------------------------------------------------------------------
;CAL_DSPL.ASM Procedure to display calibration values from Parameters
;Displays Title and column headings for Gain DAC, MaxSig, CardSig,
;MinSig, Thrld and YC Margin
;Input is pre-loaded <Parameters> file
;Sole output is display

.386
IDEAL
Model small
Stack 128

DATASEG

Title2  db       '                       CALIBRATION DISPLAY',0
downline db      13,10,0
dash2line db     80 dup (61),0
Label1  db       13,10,'     GAIN      MAX        CARD       MIN',0
Label2  db       13,10,'     DAC       SIG        SIG        SIG',0
INCLUDE "Scan_Var"

CODESEG
INCLUDE "Scan_Pcd"
 proc Cal_Dspl
                                        ;Display Headings
        mov di,offset dash2line         ;top bar
        call strwrite
        mov di,offset downline          ;skip line
        call strwrite
        mov di,offset Title2            ;Title
        call strwrite
        mov di,offset downline          ;skip line
        call strwrite
        mov di,offset Label1            ;upper line of column headings
        call strwrite
        mov di,offset Label2            ;lower line of headings
        call strwrite
        mov di,offset downline          ;skip line
        call strwrite
        mov di,offset downline          ;skip line
        call strwrite
;-------------------------------------------------------------------------------
                                ;Display parameters
@20:    mov ah,3                        ;;find cursor location
        int 10h                         ;;leave vert position (in dh) alone
                                        ;;use horiz position (dh) to move
                                        ;;cursor
                                        ;;  along columns

        mov cx,4                        ;display  4 values from calibration
        mov si,0
        mov dl,5
@10:
        mov ah,2h                       ;enable cursor control
        int 10h
        mov ax,[paramdata+si]           ;step through values
        call prtax
        add dl,10
        add si,2
        loop @10
```

```
        mov di,offset downline          ;add blank line
        call strwrite
        mov di,offset downline          ;add blank line
        call strwrite

        ret

ENDp Cal_Dspl
END
```

# Appendix B-7 ENTRY.ASM Subroutine procedure

This procedure is assembled with the command TASM ENTRY, automatically including it in the linking file LDR0. A description of its action is given in Section 3.1.5.

```
;---------------------------------------------------------------------------
;ENTRYX.ASM  Procedure to generate header record
;Can type in up to 11 numbers and letters for deck and reel numbers
;Format is Deck (8 char max), Period, Reel (3 char max)
;Echoes input directly but changes lower case to upper for storage
;Picks up Date and Start Time
;Displays heading information
;Displays Deck, Reel, Date and Start Time as a byte string <Parameters1>
;Output is above info saved in global string msg <Parameters1>
;SR is called by LDR program after film stop and calibration for Gain, MaxSig,
;  CardSig, MinSig

.386
IDEAL
Model small


DATASEG
dashline  db
'---------------------------------------------------------------------------
---',0
dash2line db
'===========================================================================
===',0
downline1 db    13,10,0

HdrTitle  db 13,10,'                    FOSDIC SCAN HEADER DISPLAY
               ',13,10,0
Label1    db 13,10,'|    FILENAME,    |     DATE     |   START   |    END    |  TOTAL
| FAIL  | CNDTN |',0
Label2    db 13,10,'|    DECK■REEL    |   SCANNED    |   TIME    |   TIME    |  CARDS
| CARDS | CODE  |',0
Label3    db 13,10,'|                 |              |           |           |
|       |       |',13,10,0
DataLine  db      '|                 |   -  -1998   |   :  :    |   :  :    |
|       |       |',13,10,0

filename db  13,   'PARAM',0
Prompt1   db    'Type <Deck Number> (8 char max) <Period> <Reel Number> (3
max) ... ',0

buffer  db      5 dup (32)        ;;;was 6
xxx     dw      0
INCLUDE "Scan_var"

CODESEG
INCLUDE "Scan_Pcd"
proc Entry
                                  ;display heading messages
        mov cx,1
        mov di,offset dashline          ;dash line
        call strwrite

        mov di,offset HdrTitle          ;Title
        call strwrite

        mov di,offset downline1         ;blank line
        call strwrite                   ;displays message "Type Deck...etc"
        mov di,offset prompt1
        call strwrite
```

```
;-----------------------------------------------------------------------------
        mov si,2                        ;si places data along displayed line
        mov di,2;0                      ;di    "      "  in ParamText mem
        call Get_Key_Msg

        mov ah,2Ah                      ;Get Date (Month and Day)
        int 21h

        mov si,17                       ;insert Month
        mov di,17;14
        mov bl,dh
        call Convert1

        mov si,20                       ;insert Day
        mov di,20;17
        mov bl,dl
        call Convert1
;-----------------------------------------------------------------------------
                                        ;Get Start Time
        mov ah,2Ch
        int 21h

        mov si,30                       ;insert hours
        mov di,30;27
        mov bl,ch
        call Convert1

        mov si,33                       ;insert minutes
        mov di,33;30
        mov bl,cl
        call Convert1

        mov si,36                       ;insert seconds
        mov di,36;33
        mov bl,dh
        call Convert1
;-----------------------------------------------------------------------------
display1:

        mov cx,1
        mov di, offset Label1           ;upper line of column headings
        call strwrite

        mov di, offset Label2           ;lower line of headings
        call strwrite

        mov di, offset Label3           ;vert lines
        call strwrite

        mov cx,1                        ;display DataLine
        mov di, offset DataLine         ;contains Deck, Reel, Date, Start Time
        call strwrite

        mov di,offset downline1         ;blank line
        call strwrite

        mov di,offset dash2line          ;dash line
        call strwrite

        ret
;=============================================================================
Get_Key_Msg:                ;Routine for processing keyed data

        mov ah,0Ah              ;;Load keyed data
        mov dx,offset buffer    ;;Buffer specifies size of buffer, max char+1
        int 21h                 ;;Buffer+1 carries number of char typed
```

```
                                ;;Buffer+2 to Buffer+13 is typed data field,
                                ;;  left-justified
        cmp [Buffer+2],13       ;jump around G-K-M if no typed char
        jne @3
        jmp @32                 ;;this is shortcut if no typed char
@3:
        mov bx,0                        ;transfer typed characters
        mov ch,0                        ;  to output file
        mov cl,[Buffer+1]               ;cx now has number of typed chars
        cmp cl,12                       ;limit to 12 char including Period
        jng @11
        mov cl,12
@11:
        mov al,[Buffer+2+bx]            ;convert to upper case
        cmp al,90
        jng @15
        sub al,32                       ;change lower to upper case
@15:
        mov [DataLine+bx+si],al         ;load typed chars into DataLine msg

        mov [ParamText+bx+di],al        ;load typed chars into file string
        inc bx
        loop @11
@32:
        mov di,offset downline1         ;blank line
        call strwrite

        ret                             ;data field is left-justified
;=====================================================================
Convert1:               ;format single byte into 2 dec digits for time fields
        mov bh,0
@10:    sub bl,10
        jl EndTens
        inc bh
        jmp @10
EndTens:
        add bh,48
Units:
        add bl,58
        mov [DataLine+si],bh            ;display Tens digit via DataLine msg
        mov [ParamText+di],bh           ;save Tens in <Parameters1> for later
                                        ;  transfer to PARAM file
        mov [DataLine+si+1],bl          ;display Tens digit via DataLine msg
        mov [ParamText+1+di],bl         ;save Tens in <Parameters1> for later
ee:                                     ;  transfer to PARAM file
        ret
;=====================================================================
ENDp entry
END
```

## Appendix B-8 UTIL.ASM Subroutine procedure

This procedure is assembled with the command TASM UTIL, automatically including it in LDR0 and PCSCAN0. It is called by various subroutine procedures

```
;-----------------------------------------------------------------------------
;UTIL.ASM Utility routines for displaying a space from "spacex", the contents
of the ax register from "Prtax" and line feed from "Line"

.386
IDEAL
MODEL small

Global       Spacex:proc            ;Horizontal space character
Global       Line:proc              ;CR and LF
Global       Prtax:proc             ;Print contents of ax register

DATASEG

spacexx db 32,0
lineyy  db 13,10,0
prtbuf  db 20 dup (?)
CODESEG
extrn strwrite:proc,bintoascdec:proc
include "Scan_Pcd"
proc Spacex
        push ax
        push cx
        push di
        mov cx,1
        mov di,offset spacexx
        call strwrite
        pop di
        pop cx
        pop ax
        ret
ENDp spacex
proc line
        push ax
        push cx
        push di
        mov cx,1
        mov di,offset lineyy
        call strwrite
        pop di
        pop cx
        pop ax
        ret
ENDp line

proc Prtax
        push ax
        push cx
        push di
        mov cx,1
        mov di,offset prtbuf
        call bintoascdec
        call strwrite
        pop di
        pop cx
        pop ax
        ret
ENDp Prtax
END
```

## Appendix C-1 PCSCAN.ASM Program

The following is a printout of the source program PCSCAN.ASM described in Sections 3.2. to 3.3.11.12 and 3.4.8 of the text. Written in assembler language, it requires assembly using TASM PCSCAN to create a PCSCAN.OBJ file. Called subroutines are similarly assembled. The group of OBJ files is then linked with the instruction TLINK @PCSCAN0. This yields the operating program PCSCAN.EXE. The file PCSCAN0 is a response file containing the names of all subroutines called on in execution of PCSCAN.EXE. Appendix G-1 lists the contents of PCSCAN0. This approach makes PCSCAN0 the equivalent of a library file but one that is more easily modified for program customization than the standard type.

There are two other files referred to in the program listing below. These are called INCLUDE files, here named SCAN_VAR and SCAN_PCD, for Scan Variables and Scan Procedures. The two files are definitions of the Global variables and procedures available in common to all elements of the overall program.

---

```
;PCSCAN.ASM  Executive program for production scanning of NCDC films

;Reads in file Param to transfer parameters determined during LDR program
;Starts program in normal manner.
;Calls SCAN which in turn uses subroutines:
;       VTRACK          Finds lateral position of card images
;       CENTROID        Finds centers of lines
;       STRGREAD        Measures transmittance of card borders
;       EDGES           Finds edges of card images
;       CALCSCAL        Calculates scaling factors for 960 reading positions
;       COLSCAN         Reads 80 columns at locations calculated above
;       SPOTREAD        Measures transmittance at specified locations
;       SCALREAD        Measures average transmittance of scale zone
;SCAN returns to PCSCAN
;FRAME is second major routine; brings in new frame of 8 cards
;End of FRAME repeats SCAN.  Cycle continues until:

;       Keypress any key except "Enter" or "r" intercepts call to FRAME,
;         goes to Display Data for 8 cards in view, waits for next keypress
;       Next keypress, any key except "Enter","r","q","a", or "z"
;         resumes normal sequence of SCAN and FRAME
;       Keypress "r" repeats scan of 8 cards in view
;       Keypress "q" creates End of Film
;       Keypress "z" causes stop on Fail Calibrate until toggled again
;       Keypress "a" causes stop on Artifact until toggled again
;       Keypress "Enter" terminates program
;Scanning pauses every 640 cards to record results of batch
;At same time, displays accumulated number of drops in Must-Have cols
;At end, the program adds Number of Cards Scanned and other data
;  to PARAM file for display by Trailer file TLR
;Reads in DDDDDDDD.msk to Maskbuf, 160 bytes, for designation of blank (No-Punch)
;  required punched data (Must-Have) or single-punch columns

.386
IDEAL
MODEL small
stack 64

DATASEG

Title1      db  13,10,25 dup (32),'SCANNING INITIATED',0
sign_on1    db  13,10,10,'Press any key except Enter to display contents of 8
cards visible on screen',0
sign_on2    db  13,10,'Press Enter to terminate program',0
```

```
msg1        db  'Unable to find Mask file',0
msg2        db        '     Last card number..',0
msg3       db  13,10,'Press key to resume',13,10,'Number of cards rescanned....
            ;',0
msg4        db  13,10,10,'Program Terminated',0
msg5        db  13,10,'Program Terminated...END OF FILM',0
msg6        db  13,10,'Gain DAC..',0
msg7        db        '     MaxSig..',0
msg8        db        '          Number of cards scanned..',0
Tens        db        '          1         2         3         4         5
  6        7         8',0
U n i t s                                d b
'12345678901234567890123456789012345678901234567890123456789012345678901234567
890',0
Ticks       db        '          |         |         |         |         |
  |        |         |',0
filename    db      'PARAM',0
ParamData   dw      10 dup (?)
ParamData1  dw      10 dup (32)  ;
                                 ; +0 = Gain1   (Gain DAC), left in place
                                 ; +2 = MaxSig  Near 980 by calibration
                                 ; +4 = CardSig
                                 ; +6 = MinSig
                                 ; +8 = NmbrCdsScnd
                                 ;+10 = NmbrCdsFld
                                 ;+12 = DirtSpot ??
                                 ;+16 = Spare
                                 ;+18 = Spare
ParamText1 db   80 dup (32)     ;Must follow ParamData1
OutMsg      db   8 dup (48),46,3 dup (48)      ;filename for Deck.Reel
space       db   32,0           ;print space
spaces      db   10 dup (32),0
dash2line   db   13,10,78 dup (61),0
prtbuf      db   20 dup (?)     ;JIC
outfile8    db   640 dup (45)   ;Output data memory for 8 cards
xx          db   0              ;delimiter for ending display
FlagRep     dw   (?)           ;Flag for selecting keyboard-controlled actions
                                ;  FlagRep=0 for normal SCAN/FRAME cycling
                                ;  FlagRep=1 or more for repeating SCAN on same
                                ;  frame

filename1     db    'DataOut',0
Maskfilename  db    'xxxxxxxx.msk',0
Maskbuf      db    160 dup (45)
Buffer        db    2,3 dup (32)
CndtnTitle    db    28 dup (32),'SCAN CONDITION CODES',0
CndtnCode1    db    'First.... ',0
CndtnCode2    db    13,10,'Second... ',0

DataHandle    dw    ?
ParamHandle   dw    ?
MaskHandle    dw    ?
BlockPtr      dw    ?
DataBlock     db    52000 dup  (?)       ;the processed block of 51,200 may be
                                           ;  exceeded by 639, hence the extra
BlankCols     dw    ?                     ;nmbr of designated blank columns,
"No-Punch"
PunchCols     dw    ?                     ;nmbr of designated punched cols,
"Must-Have"
TotalBlankCols dd   ?
NmbrCdsScnd    dw   ?                ;total number of cards scanned
NmbrCdsRescnd  dw   ?
NmbrCdsFld     dw   ?                ;total number of cards with failed
calibration
NmbrColDrops   dw   ?                ;Total nmbr of columns with drops in
Must-Have cols
NmbrColPickups dw   ?                ;Total nmbr of columns with pickups in
```

```
No-Punch cols

PrevCdsFld      dw  ?                ;working location
PrevCdsPickups dw  ?
PrevColPickups dw  ?             ;working location
PrevCdsDrops   dw  ?             ;working location
NmbrCdsDrops   dw  ?        ;Total nmbr of cards with drop(s) in Must-Have cols
NmbrCdsPickups dw   ?            ;Total nmbr of cards with one or more illegal
characters
NmbrCdsArtif   dw   ?            ;total number of cards with artifacts
ResumeScan     dw   ?            ;flag for restarting after film failure stop
AllowFailCalStop dw ?            ;flag for enabling stop on Fail Calibrate
AllowArtiStop dw    ?             ;flag for enabling stop on Artifacts
AftrFirstFrm   dw   ?
Tilt           dw  500               ;start Tilt at 500
TiltLeft       dw  ?

msg101         db  13,8 dup (10),30 dup (42),' Stop for DROP ',32 dup (42),0
msg102         db  13,8 dup (10),24 dup (42),' Stop for ILLEGAL CHARACTER ',26
dup (42),0
msg103        db  13,8 dup (10),25 dup (42),' Stop for FAIL CALIBRATE ',28 dup
(42),0
msg104         db  13,10,'XFCornerDiff.... ',0
msg105         db  '     YFCornerDiff.... ',0
msg106         db  13,10,'Thrld.... ',0
INCLUDE            "Scan_Var"

CODESEG
INCLUDE "Scan_Pcd"

start:
        mov ax,@data
        mov ds,ax
        mov es,ax

        mov dx,30Ch                     ;start Tilt at 500
        mov ax,[Tilt]
        out dx,ax

        mov dx,316h                     ;reset counter
        mov ax,01h
        out dx,ax


        mov ah,3Dh                      ;open file PARAM
        mov dx,offset filename
        mov al,2
        int 21h
        mov [ParamHandle],ax

        mov ah,3Fh                      ;read file PARAM of100 bytes
        mov bx,[ParamHandle]            ;into ParamData1+
        mov dx,offset ParamData1
        mov cx,100
        int 21h

        mov ax,[ParamData1+2]           ;load MaxSig from file
        mov [MaxSig],ax
        mov ax,[ParamData1+4]           ; load CardSig
        mov [CardSig],ax
        mov ax,[Paramdata1+6]           ;load MinSig
        mov [MinSig],ax
;-----------------------------------------------------------------------
        call justify            ;right-justify Deck.Reel msg
                                ;OutMsg has filename for specified Deck.Reel
                                ;ParamText1+2 has Deck.msk
;-----------------------------------------------------------------------
```

```
        mov ah,3Dh              ;open file DDDDDDDD.MSK for reading
        mov dx,offset Maskfilename
        mov al,0
        int 21h
        mov [MaskHandle],ax
        jc NotFind
        mov [MaskBuf],'X'       ;this should be overwritten by mask
        mov ah,3Fh              ;read from file to MaskBuf
        mov dx,offset MaskBuf
        mov cx,80
        mov bx,[MaskHandle]
        int 21h

        mov ah,3Eh              ;close file
        int 21h
;------------------------------------------------------------------------------

        mov cx,1
        mov di,offset dash2line
        call strwrite
        call line
        mov di,offset Title1            ;display Title
        call strwrite
        mov di,offset sign_on1          ;msg "Press any key....."
        call strwrite
```



```
        mov di,offset sign_on2          ;msg "Press Enter..."
        call strwrite
;--------------------------------------------------------------------------
        mov ah,03Ch                     ;create file DataOut
        mov dx,offset filename1
        mov cx,0
        int 21h
        mov [DataHandle],ax

        mov [NmbrSigs],         0
        mov [SigSum],           0
        mov [junk],             0
        mov [junk+2],           0
        mov [NmbrCdsScnd],      0       ;reset number of cards scanned to 0
        mov [NmbrCdsRescnd],    0       ;reset number of cards rescanned
        mov [NmbrCdsFld],       0       ;reset number of cards failed
        mov [NmbrCdsArtif],    0       ;reset number of cards with artifact(s)
        mov [BlockPtr],         0       ;reset block pointer
        mov [TotalBlankCols],   0       ;reset blank columns
        mov [FlagRep],          0       ;reset repeat flag to normal cycling
        mov [NmbrCdsDrops],     0       ;reset number of cards with drops
        mov [NmbrCdsPickups],   0       ;reset number of cards with pickups
        mov [AllowArtiStop],    0       ;reset Stop on Artifacts
        mov [AllowFailCalStop],0        ;reset Stop on Fail Calibrate
        mov [NmbrColDrops],    0       ;reset total nmbr of drops in Must-Haves
        mov [NmbrColPickups],    0            ;reset total nmbr of pickups in
No-Punches
        mov [AftrFirstFrm],     0       ;reset AftrFirstFrm

        mov dx,316h                     ;reset counter
        mov ax,01h
        out dx,ax
;------------------------------------------------------------------------------
Scan1:
JMP NODISPLAY
        call line
        mov ax,[Card]                   ;derive pointer for DataBlock from Card
        dec ax                          ;Card starts with 1, want 0
```

```
        mov bx,80
        mul bx                          ;ax has incremental nmbr of cols for frame
        mov bx,[BlockPtr]               ;show data for up to 8 cards, 640 chars
        cmp bx,0
        jne @@1
        add bx,51200
@@1:
        sub bx,ax
                                        ;bx has offset from DataBlock base
        mov cx,640                      ;display 640 char in DataBlock
        mov si,0
        mov ah,2
@30a:    mov dl,[DataBlock+bx+si]
        int 21h
        inc si
        loop @30a

NoDisplay:
        cmp [FlagRep],1                 ;Retract Block Pointer in Repeat mode
        jne @33                         ;  to allow data replacement
        sub [BlockPtr],640
@33:
     ;  mov ax,[NmbrCdsDrops]            ;save entering drop (card) count for
     ;  mov [PrevCdsDrops],ax            ;  comparison with exiting count

;call prtax
        mov ax,[NmbrCdsPickups]         ;save entering pickup (card) count for
        mov [PrevCdsPickups],ax         ;  comparison with exiting count
        mov ax,[NmbrCdsFld]             ;save entering total fail calib count
        mov [PrevCdsFld],ax             ;  for comoparison with entering count
;---------------------------------------------------------------------------
mov cx,3;2;50;0                         ;pre-scan cycling delay if wanted
wt9: mov bx,32000;65000
wt6: dec bx
jnz wt6
loop wt9
;---------------------------------------------------------------------------
        call SCAN
;---------------------------------------------------------------------------
;        mov cx,100;0;0                 ;post scan cycling delay if wanted
;@5c:    mov bx,65000                   ;approx 4 ms per cx
;@6a:    dec bx
;        jnz @6a
;        loop @5c
;ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
                                    ;Optional stops
        mov bx,0                    ;prepare for Stop Test

                                    ;An Artifact is either a pickup in a No-Punch
                                    ;  column or a drop in Must-Have col
                                    ;Stop on any pickups in frame of 8
        mov ax,[NmbrCdsPickups]
        sub ax,[PrevCdsPickups]
;mov ax,0
        test [AllowArtiStop],1  ;is stopping on artifact allowed? (from "a")
        je @@37                 ;no, continue at @@37
        cmp ax,0                    ;cmp on 0 to stop for artifacts (illegal char)
        je @@37
        add bx,1                    ;skip this if no '*'
;pusha
;mov cx,1
;mov di,offset msg101               ;msg "Stop for DROP(S)"
;call strwrite
;popa

@@37:
```

```
;@@37 cont.
;        mov ax,[NmbrColPickups]
;        sub ax,[PrevColPickups]
;push ax
;;mov ax,[AllowArtiStop]
;call line
;call prtax
;pop ax
        test [AllowArtiStop],1
        je @@37a
        cmp ax,0                  ;cmp on 0 to stop for pickup(s)
        je @@37a
        add bx,1                  ;skip this if no pickups
pusha
mov cx,1
mov di,offset msg102
call strwrite
popa

@@37a:                            ;Stop on frame with a fail calibrate card
                                  ;FailCalStop is 0 for no failure in frame of 8
                                  ;     "      " 1 for any fail calibration in 8

        mov ax,[NmbrCdsFld]
        sub ax,[PrevCdsFld]
        test [AllowFailCalStop],1  ;is stopping on fail calib allowed (from "z")
        je Stops
        cmp ax,0
        je Stops
        add bx,1                  ;do only if fail calibrate cards in frame of 8
pusha
mov cx,1
mov di,offset msg103              ;announce Fail Calibrate
call strwrite
mov di,offset msg104              ;msg "XFCornerDiff...."
call strwrite
mov ax,[XFCornerDiff]
call prtax                        ;show value
mov di,offset msg105              ;msg "YFCornerDiff...."
call strwrite
mov ax,[YFCornerDiff]             ;show value
call prtax
popa
;ssss------------------------------------------------------------------

Stops:                            ;Unconditional stops
        mov dx,31Ah               ;stop on designated card count
        in ax,dx                  ;  100h means preset count reached
        cmp ax,100h
        jg continul
        jmp display1
continul:
        add bx,[Pause]            ;stops for film break or card jam
                                  ;film break or jam raises bx by 8

        cmp bx,0
        je NoFilmStop             ;jump if Pause not set before return from SCAN
        jmp Display1             ;through here if film problem, stop with display

NoFilmStop:                       ;through here if no film stop
        mov [ResumeScan],0        ;next keystroke resumes in SCAN with Frame
;---------------------------------------------------------------------
                                          ;Set up data block
        cmp [BlockPtr],51200              ;sets data block size before transfer
        jb @16                            ;  to DataOut (640 cards)
        mov ax,[NmbrCdsRescnd]            ;display current total number of cards
```

```
call prtax                    ;  with one or more unexpected drops
call spacex
mov ah,2
mov dl,'('
int 21h
mov ax,[NmbrCdsFld]
call prtax
mov ah,2
mov dl,')'
int 21h
call spacex
;-----------------------------------------------------------------------------
        call FillFile                   ;Record a block of 640 cards
;-----------------------------------------------------------------------------
        mov [BlockPtr],0
@16:
        cmp [FlagRep],0                 ;check repeat flag for cycling control

        je Kpp                          ;bypass display if FlagRep=0
        jmp Display1
;-----------------------------------------------------------------------------
                        ;Test for "Display and Pause" keypress
Kpp:    mov ah,1                ;first, test for any keypress
        int 16h
       je Frame1                ;no kp, continue with next frame
                                ;if keypress, check character for Enter
        mov ah,0                ;read kb char
        int 16h                 ;al may be Enter or any other char
        cmp al,13               ;compare for Enter
        je Stop                 ;if Enter go to Stop
                                ;through here if not Enter
;-----------------------------------------------------------------------------
Display1:
        call line               ;2 blank lines
        call line
        mov cx,1
        mov di,offset Tens      ;set up scale for labeling columns
        call strwrite
        mov di,offset Units
        call strwrite
        mov di, offset Ticks
        call strwrite

        mov ax,[Card]              ;derive pointer for DataBlock from Card
        dec ax                        ;Card starts with 1, want 0
        mov bx,80
        mul bx                       ;ax has incremental nmbr of cols for frame
        mov bx,[BlockPtr]          ;show data for up to 8 cards, 640 chars
        sub bx,ax
                                        ;bx has offset from DataBlock base
        mov cx,640                   ;display 640 char in DataBlock
        mov si,0
        mov ah,2
@30:    mov dl,[DataBlock+bx+si]
        int 21h
        inc si
        loop @30
        mov cx,80
        mov si,0
        mov ah,2
@31:    mov dl,[MaskBuf+si]
        int 21h
        inc si
        loop @31

        mov di,offset msg2    ;"Last card is number..."
        call strwrite
```

```
        mov ax,[NmbrCdsScnd]  ;show card count
        call prtax
        mov di,offset msg3        ;msg "Press any key to Resume"
        call strwrite
;---------------------------------------------------------------------------
@@37:
                              ;End of display, wait for next keypress
                              ;Test for keypress to resume cycling
Kpwait:
        mov ah,1              ;Wait for any keypress
        int 16h
        jz kpwait             ;no kp yet, wait

        mov ah,0              ;read keypress
        int 16h
        cmp al,13             ;compare for Enter
        je Stop               ;go out if Enter

        cmp al,122            ;looking for "z" to set flag allowing stop
        jne @@38              ;  on fail calibrate
        inc [AllowFailCalStop] ;toggle for disabling stop when even
        jmp KPwait
@@38:
        cmp al,97             ;looking for "a" to set flag allowing stop
        jne @@38a             ;  on artifact
        inc [AllowArtiStop]   ;toggle for disabling stop when even
        jmp KPwait
@@38a:
        cmp al,113            ;looking for "q"
        jne continll
        mov [FlagEoF],1
        jmp EndofFilm
continll:
        cmp al,114            ;If not Enter or "a" or "q" or "z",
        jne @001              ;, resume cycling with msgl below
        mov [FlagRep],1
       ;inc [FlagRep]         ;if kp="r" move flag away from 0, repeat scan
        jmp @002a
@001:   mov [FlagRep],0       ;if not "r" restore flag to 0
@002a:  cmp [FlagRep],0       ;branch according to Flagtest for Repeat mode
        je Framel             ;continue normal scan by going to next frame
        jmp scanl             ;repeat again
Framel:

@001b:
        cmp [ResumeScan],1    ;switch for resuming after film stop
        je SkipFrame          ;skip next Frame after manual film advance
StartDelay:
        mov dx,30Eh                    ;start a delay period of 40ms
        mov ax,870
        out dx,ax
;===========================================================================
      call FRAME              ;moves film to next group of 8 cards
;===========================================================================
CheckDelay:                            ;if Frame took longer than 40ms,
                                       ;  want to introduce delay of 100ms
        mov dx,318Ah
        test ax,8000h
        je SkipFrame
        mov dx,30Eh                    ;start another delay, 100ms, if Frame
        mov ax,1000                    ;  time is longer than 40ms
        out dx,ax
        mov dx,31Ah                    ;wait for end of 100ms
wt8:    in ax,dx
        test ax,8000h
        jne wt8
```

```
SkipFrame:
        mov [FailTrack],    0               ;reset FailTrack
     ;   mov [FailCardBar1],0                ;reset FailCardBar
        mov [FailEdge1],    0               ;reset FailEdge
        ;mov [ResumeScan],  0
;        mov dx,31Ah                        ;;test for TakeUp Idler slack
;        test ax,004h                       ;;  if so, need to turn off Beam
;        jne Stop

EndofFilm:                          ;Test for End of Film via FlagEOF
        cmp [FlagEOF],1                     ;FlagEOF=1, End of Film
        jne Scan1                           ;if not EOF, continue cycling with
                                            ;  Scan and Frame
                                            ;if EOF, turn off CRT beam and capstan
                                            ;FlagEOF is set by high signals in Frame
                                            ;through here only if End of Film

        call line
        mov di,offset msg5                  ;msg "Program Terminated, End of Film"
        call strwrite
        jmp stop1
stop:
        mov cx,1
        mov di,offset msg4                  ;msg "Program Terminated"
        call strwrite
stop1:                                     ;through here on termination of program

        mov cx,1
        mov di,offset msg8                  ;"Number of cards scanned"
        call strwrite
        mov ax,[NmbrCdsScnd]                ;the total number of cards
        call prtax
;----------------------------------------------------------------------------
                                    ;Omit Stop Time if not End-of-Film
        cmp [FlagEOF],1
        jne CardsScnd
                                            ;Get Stop Time, insert in ParamText1+
        mov ah,2Ch                          ;get current Time
        int 21h

        mov si,41                           ;place hours
        mov bl,ch
        call Convert1

        mov si,44                           ;place minutes
        mov bl,cl
        call Convert1

        mov si,47                           ;place seconds
        mov bl,dh
        call Convert1
;----------------------------------------------------------------------------
CardsScnd:                          ;Add data on number of cards scanned, etc.
        mov cx,1                    ;  to ParamData1 for later recording in PARAM
        mov ax,[NmbrCdsScnd]
        mov [ParamData1+8],ax
        mov di,offset paramtext1+53
        call bintoascdec

        mov ax,[NmbrCdsFld]                 ;add number of cards failed
        mov di,offset ParamText1+62
        mov [ParamData1+10],ax
        call bintoascdec

        cmp ax,0                                 ;ensure printing of 0 failed
        jne CalcArtif                            ;  if applicable
        mov [ParamText1+62],48
```

```
;-----------------------------------------------------------------------------
CalcArtif:                                   ;Add keyed Condition Codes
        mov di,offset dash2line
        call strwrite
        call line
        call line
        mov di,offset CndtnTitle
        call strwrite                    ;display title
        call line
        mov di,offset CndtnCode1
        call strwrite                    ;display Cndtn1 msg
        mov di,70
        call Get_Key_Msg                 ;add first keyed code to output file
        mov di,offset CndtnCode2
        call strwrite                    ;display Cndtn2 msg
        mov di,71
        call Get_Key_Msg                 ;add second keyed code to output file
;-----------------------------------------------------------------------------
                                 ;Write modified PARAM file
        mov bx,[paramhandle]     ;move pointer to 8 bytes from start of PARAM
        mov ah,042h
        mov cx,0
        mov dx,8
        mov al,0
        int 21h
        mov ah,40h                ;Record ParamText1 in file PARAM, 92 bytes

        mov cx,92
        mov dx,offset ParamData1+8
        int 21h
;-----------------------------------------------------------------------------
        call FillFile
;-----------------------------------------------------------------------------
        mov ax,[NmbrCdsRescnd]           ;display current total number of cards
call line
call prtax                       ;  with one or more unexpected drops
        jmp exit1
NotFind:                                 ;Display NotFind msg
        mov cx,1
        mov di,offset msg1
        call strwrite
Exit1:
        mov ah,04Ch
        int 21h                  ;END OF PROGRAM
;=============================================================================
;                          Called internal subroutines follow
Convert1:                                ;makes 2 digits from 1 byte
        mov bh,0
@10:    sub bl,10
        jl EndTens
        inc bh
        jmp @10
EndTens:
        add bh,48
;and Units:
        add bl,58
        mov [ParamText1+si],bh
        mov [ParamText1+si+1],bl
        ret                      ;End of Convert1 routine
;-----------------------------------------------------------------------------
FillFile:                                ;writes block of 60000 bytes or
        pusha                            ;residual at End of Film
        mov ah,40h
        mov dx,offset DataBlock
        mov cx,[BlockPtr]
        mov bx,[DataHandle]
        int 21h
```

```
        mov ah,045h
        mov bx,[datahandle]
        int 21h
        mov bx,ax
        mov ah,03Eh
        int 21h
        popa

        ret                 ;End of FillFile routine
;---------------------------------------------------------------------------------
Get_Key_Msg:                ;Routine for processing keyed data
        mov ah,0Ah                  ;;Load keyed data
        mov dx,offset buffer        ;;Buffer specifies size of buffer, max char+1
        int 21h                         ;;Buffer+1 carries number of char typed


        mov al,[Buffer+2]           ;;Buffer+2 is typed data field, left-justified
        cmp al,13                     ;jump around G-K-M if no typed char
        jne @3
        jmp @Out                    ;;this is shortcut if no typed char
@3:
        mov [ParamText1+di],al            ;load typed char into file string
@Out:                                   ;data field is left-justified
        ret                 ;End of keyed Condition Codes routine
;---------------------------------------------------------------------------------
Justify:                    ;Routine to right-justify Deck and Reel message
        mov cx,8                              ;Looking for "." in input msg for
justification
        mov si,0
@1:     cmp [ParamText1+2+si],46    ;cmp for period
        je @2
        inc si
        loop @1
@2:                                 ;si has location of period in input message
        mov bx,8
        sub bx,si                   ;bx has offset between period positions
                                      ;  in input and output messages, 0 to 7
        mov cx,9
@3a:     mov al,[ParamText1+2+si]
        cmp al,32                   ;ASCII digits must be greater than space (32)
        jng @5
        mov [Outmsg+si+bx],al       ;transfer real digit to output
        dec si
        loop @3a                    ;OutMsg has been loaded with Deck
                                    ;bx still has difference in period location  @5:
                                    ;  in original and final msgs

        mov si,11                   ;Find si for rightmost (Reel) digit
        sub si,bx
        mov cx,3
@6:     mov al,[ParamText1+2+si]
        cmp al,32                   ;Reel nmbr can include letters
        jg @5a                      ;jump out of loop when rightmost digit of Reel
        dec si                      ;   is found
        loop @6
                                    ;si has position of units of Reel nmbr for input
@5a:                                ; Deck, 3 char to right of period in input data
        mov bx,11
        mov cx,3                    ;transfer up to 3 digits to output
@7:     mov al,[ParamText1+2+si]
        cmp al,46
        je @9
        mov [OutMsg+bx],al
        dec si
        dec bx
        loop @7
@9:
```

```
;@9 cont.
        mov cx,12                   ;transfer 12-byte Deck.Reel msg to ParamText1
        mov si,0                    ;  for inclusion in modified PARAM
@8:     mov al,[OutMsg+si]
        mov [ParamText1+2+si],al
        inc si
        loop @8

        mov cx,8                    ;transfer first 8 bytes for Deck name in Mask
        mov si,0                    ;   file
@10a:   mov al,[OutMsg+si]
        mov [Maskfilename+si],al
        inc si
        loop @10a
        ret

        mov dx,310h
        mov ax,00h
        out dx,ax
        ret
End start;
```

## Appendix C-2 SCAN.ASM Subroutine procedure

This procedure is assembled with the command TASM SCAN, automatically including it in the linking file PCSCAN0. A description of its actions is given in Section 3.3 to 3.3.11.12.

```
;---------------------------------------------------------------------------
;SCAN.ASM  Production procedure for scanning punch card images

;Scans up to 8 cards, returning with inset for starting Frame
;Inset is Card Bar search origin for last card
;Called by PCSCAN.
;Calls procedures VTrack, Centroid, Edges, CalcScal, ColScan, Util, MTA
;Each procedure returns to Scan
;YCMargin set by VTrack
;To customize for variable card size, changes are at lines

.386
IDEAL
MODEL small

DATASEG
;--------------------------------------------------------------------------
;Fudges
ThrldMod12 dw   100;150         ;Working thrld is this increment above first
ThrldMod34 dw   100;150         ;  signal; approx. scale factor: 200 = 1 volt
                                ;Use 100 for thin or dense scale lines,
                                ;  150 for normal
ThrldCardBar dw 300;100         ;Fixed value
;--------------------------------------------------------------------------
SigSum   dd     0
NmbrSigs dd     0
MeanCdSig dw    0

Card     dw     1               ;Cards 1 through 8
XC0      dw     7000
YC0      dw     19100
XF0      dw     4096            ;XF origin, fixed
YF0      dw     8000            ;YF    "      "

Port     dw     304h            ;Default = XF
Step     dw     40              ;Default value
MaxCnt   dw     32              ;Default value
Focus    dw     1000            ;default value
;Thrld    dw     300             ;default threshold, replaced by ParamData1+8
CardBar dw  16000               ;dummy value
CardPeriod  dw  2000            ;dummy value
SaveXF      dw  ?
FrmSpan     dw  2000            ;dummy value
LeftEdge    dw  16000           ;dummy value
CardSig dw      600             ;dummy
MinSig  dw      50              ;dummy
Corner1 dw      2 dup (0)       ;corner at top of card, col 1,  |
Corner2 dw      2 dup (0)       ;    "   bottom           "     |first word XF
Corner3 dw      2 dup (0)       ;    "     top          col 80  |second  "  YF
Corner4 dw      2 dup (0)       ;    "   bottom           "     |
NoCard  dw      ?
;DirtSpot dw     ?               ;Count of signals in blank cols
XFCornerDiff dw 0               ;Non-rectangularity, size diff, cols 1 and 80
YFCornerDiff dw 0               ;Non-rectangularity, size diff, rows 12 and 9
prtbuf  db      30 dup (?)      ;for displaying info
junk    dw      100 dup (?)     ;JIC
msg2    db      32,0            ;horiz space
Flag    dw      0               ;Flag for special treatments in subroutines
```

```
XC      dw      ?               ;working XC
YC      dw      5000;?              ;  "    YC
XF      dw      ?               ;  "    XF
YF      dw      ?               ;  "    YF
XCFrame dw      ?               ;XC origin for use by Frame procedure
;YCFrame dw      ?               ;YC origin for use by Frame procedure
XCFrameStop dw  ?               ;default XC for stopping film movement
;NmbrCdsScnd dw  0               ;Total number of cards scanned
;NmbrCdsFld  dw  0               ;Number of cards failing squareness check
Error   dw      10  dup (0)     ;JIC flag set identifying cards with errors

FailTrack        dw ?   ;<1>  fail flag in VTrack,   for runout
FailBar          dw ?   ;<8>  fail flag in Centroid, for runout
FailEdge         dw ?   ;<16> fail flag in Edges,    for runout
FailBar1         dw ?   ;<2>  fail flag in Centroid, for high first read
FailEdge1        dw ?   ;<4>  fail flag in Edges,    for high first read
FailCalGeom      dw ?   ;<32> fail flag in GeomCheck, for non-rectangularity
FailArtifact     dw ?   ;<64> fail flag for stopping on artifacts

;FailCardBar1     dw ?  ;stored fail frame flag for card bar search, hi 1st read
;FailCalCardBar    dw ?   ;stored fail calib flag for card bar search, runout
;FailCalEdge       dw ?   ;stored fail calib flag for edge search, runout
;FailCalScaleBar   dw ?   ;stored fail calib flag for scale bar search, runout
FailCard         dw ?
FailCardinFrame  dw ?
;FailCorner       dw ?
CardJam          dw ?
;BarFailCount     dw ?
ThrldMod    dw    ?
Want        dw    ?
;SpotTransm dw    ?
;ResumeScan  dw    ?
Pause   dw   ?
TiltLeft dw  ?
TiltOut  dw ?
INCLUDE "Scan_Var"

CODESEG
INCLUDE "Scan_Pcd"
proc Scan                       ;Through here only at start of routine
       ; mov dx,316h             ;reset
       ; mov ax,01h
       ; out dx,ax

      ;  mov ax,12h              ;start counter if desired
      ;  out dx,ax

        mov ax,[XC0]            ;load XC position for stopping film movement
 add ax,3300
        mov [XCFrameStop],ax
      ;  mov [junk+20],        0
      ;  mov [junk+22],        0
      ;  mov [FailCardBar1],   0
        mov [FailTrack],      0
        mov [FailCardinFrame],0
;                               Set calibration parameters one time per frame
        mov dx,308h             ;set Gain, may not need this
        mov ax,[ParamData1]
        out dx,ax
    ;   mov  dx,30Ch            ;set to Tilt null
    ;   mov ax,500
    ;   out dx,ax

        mov dx,30Ah             ;set initial Focus
        mov ax,1000
        out dx,ax
```

```
;-----------------------------------------------------------------------------
        mov dx,316h             ;settling delay, min.
        mov ax,03h
        out dx,ax
                        ;End of initialization, done once at start of SCAN
;-------------------------------------------------------------------------------

        call Gain       ;Calibrates Gain using Gain DAC, range 0 to 1023
;-------------------------------------------------------------------------------
                        ;Preparation for VTrack; XC and YC set in VTrack
        mov dx,304h
        mov ax,[XF0]
        mov [XF],ax
        out dx,ax               ;issue XF=XF0
        mov dx,306h
        mov ax,[YF0]
        mov [YF],ax
        out dx,ax               ;issue YF=YF0
;------------------------------------------------------------------------------
        call VTrack             ;finds YCMargin for vertical registration
;-------------------------------------------------------------------------------
;mov ax,[YCMargin]
;call prtax
;call spacex

ErrorCheck1:                    ;VTrack returns with YC=YCMargin, XC=XC0
                                ;XF and YF not used
                                ;FailTrack=0/1
        cmp [FailTrack],0       ;check for FailTrack
        je FirstCard            ;
        inc [Card]              ;need to increment card nmbr in this shortcut
        jmp FilmPause           ;go back to PCSCAN for Film stop
FirstCard:                      ;search for Card Bar #1 uses this origin
        mov [Card],1            ;Initialize Card to 1
;------------------------------------------------------------------------------
NewCard:                        ;through here every card
        mov [CardJam],          0
        mov [Pause],            0
        mov [FailCard],         0

        mov dx,300h             ;XC advances approx 2180 every card
        mov ax,[XC]
        out dx,ax               ;issue updated XC

        mov dx,302h
        mov ax,[YCMargin]
;add ax,100
        mov [YC],ax
        out dx,ax               ;issue YC=YCMargin



                                ;no delay needed here
        mov dx,304h
        mov ax,[XF0]
        mov [XF],ax
        out dx,ax               ;issue XF

        mov dx,306h
        mov ax,[YF0]
        mov [YF],ax
        out dx,ax               ;issue YF
;-----------------------------------------------------------------------------
                                ;Initialize for each card
                                ;XC advanced for next card at end of previous
                                ;  card
```

```
                                ;Set parameters for Centroid SR
PrepCardBar:
        mov [Port],304h         ;set for horiz movement
        mov [Step],80           ;set step
        mov [MaxCnt],30         ;allow 30 steps
        mov [Focus],1000        ;set large spot
        mov ax,[ThrldCardBar]
        mov [Thrld],ax          ;End of coordinate initialization for
                                ;  Centroid on Card Bar
        cmp [Card],2
        jne @7
@7:
;-------------------------------------------------------------------------
        Call Centroid           ;Horizontal position of Card Bar
;-------------------------------------------------------------------------
ErrorCheck2:                    ;Returns with Center of Card Bar and
                                ;  XF=last Read, XC=XC0, YC=YCMargin, YF=8000
                                ;  FailBar1 and FailBar, 0/1
       mov ax,[FailBar1]        ;FailBar1=0/1, failure causes film break stop

       add [FailCard],ax
       mov ax,[FailBar]         ;check status of FailBar=0/1
       add [FailCard],ax        ;store failure of Card Bar search
                                ;  for check for FailCalibrate

        mov ax,[Center]         ;Save horizontal center of Card Bar
        mov [CardBar],ax
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        jmp CardAdv
CardAdvRtn:
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        cmp [Card],9            ;check for 9th card, skip to SetXCFrame
        jg SetXCFrame
                        ;Preparation for finding (vert) center of first scale bar
        mov dx,304h             ;move to scale zone horizontally
        mov ax,[CardBar]
        sub ax,200
        out dx,ax
        mov [XF],ax

        mov dx,306h             ;YF vert origin for search for first scale bar,
        mov ax,[YF0]            ;  Corner 1
sub ax,350              ;advance 400 toward scale bar to shorten scan
        mov [YF],ax
        out dx,ax

        mov [Port],306h         ;load parameters for Centroid, Corner 1
        mov [Step],-8
        mov [MaxCnt],64
        mov [Focus],200
        mov [Thrld],0
mov ax,[ThrldMod12]
mov [ThrldMod],ax;;;100;150      ;Thrld becomes first amplitude+150
                                ;  with limits 200 to 400
                                ;End of prep for first scale bar
;-------------------------------------------------------------------------
        Call CENTROID           ;Find center of first scale bar, Corner1,Col= ½
;-------------------------------------------------------------------------
ErrorCheck3:                    ;on return, XF is in scale zone, -200 from Bar
                                ;YF is in Center at centroid of first scale bar
        mov ax,[FailBar]        ;FailBar=0/1 for runout
        add [FailCard],ax       ;enable FailCalibrate from FailBar
        mov ax,[FailBar1]       ;FailBar1=0/1 for high first signal
        add [FailCard],ax       ;enable FailCalibrate from FailBar1
```

```
        mov dx,306h             ;jump to center of first inter-bar space, Col -½
        mov ax,[Center]          ;this is center of first bar as found
        sub ax,43                ;move to center of column 1
        mov [YF],ax              ;
        mov [Corner1+2],ax
        out dx,ax                ;this is YF for Col 1, Corner1

        mov ax,[CardBar]         ;prep for StrgRead
        add ax,600
        mov [XF],ax              ;load XF for origin of StrgRead
        add [YF],150
        mov [Focus],1000

;----------------------------------------------------------------------------
        call StrgRead            ;measure transmittance at card left border
;----------------------------------------------------------------------------
        mov ax,[Transmtce]       ;low transmittance indicates No-Card next
cmp ax,120;200              ;use absolute thrld of 200
        jb SetXCFrame            ;where XF for last CardBar is converted to
                                 ;  XC position for framing

                            ;Preparation for Edges SR to find top edge of card
        mov dx,304h              ;issue XF for origin of top edge search, col 1
        mov ax,[CardBar]
        sub ax,200
        mov [XF],ax
        out dx,ax

        mov dx,306h              ;use col 2 for edge search to avoid cut corner
        mov ax,[Corner1+2]
        sub ax,87
        mov [YF],ax
        out dx,ax

        mov [Port],304h          ;parameters for search for edge, Corner 1
        mov [Step],16            ;direction is increasing XF
        mov [MaxCnt],50
        mov [Focus],200
        mov [ThrldMod],0         ;Thrld calculated in proc using first signal
                                 ;  and Transmtce at card left edge
;----------------------------------------------------------------------------
        Call EDGES               ;Top edge of card, Col 1, Corner 1
;----------------------------------------------------------------------------
                                 ;End of prep for top edge col 1ErrorCheck4:
                                 ;returns with FailEdge1=0/1, FailEdge=0/1
        mov ax,[FailEdge]        ;check for search runout
        add [FailCard],ax        ;enable FailCalibrate

        mov ax,[Edge]            ;store XF for Corner1
        mov [Corner1],ax
                                 ;prep for Corner2 Centroid
        mov dx,304h              ;move to bottom of card
        mov ax,[CardBar]
;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
add ax,[CardPeriod];2000
sub ax,200
;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        mov [XF],ax
        out dx,ax                ;issue XF for scale zone

        mov dx,306h
        mov ax,[YF0]             ;set YF origin for finding first scale bar
sub ax,350                  ;YF retraction=400 to shorten scan
        mov [YF],ax
        out dx,ax                ;XF and YF now in for Centroid
```

```
;-------------------------------------------------------------------------
        call SPOTREAD              ;reads for jam at origin of scale bar search
;-------------------------------------------------------------------------
                                   ;returns with CardJam 0/1 every card
                                   ;check CardJam after calibration check
        mov [Port],306h            ;load parameters for Centroid, Corner2
        mov [Step],-8
        mov [MaxCnt],64
        mov [Focus],200
        mov [Thrld],0
mov ax,[ThrldMod12]
mov [ThrldMod],ax                  ;Thrld becomes first amplitude+150
                                   ;  with limits 200 to 400
                                   ;End of prep for first scale bar, Corner2
;-------------------------------------------------------------------------
        Call CENTROID          ;Find center of first scale bar, Corner2, Col= ½
;-------------------------------------------------------------------------
ErrorCheck5:                       ;on return, XF is in scale zone between cards
                                   ;YF is in Center at centroid of first scale bar
        mov ax,[FailBar]           ;FailBar=0/1 for runout
        add [FailCard],ax          ;enable FailCalibrate

        mov dx,306h               ;jump to center of first inter-bar space, Col -½
        mov ax,[Center]            ;this is center of first bar as found
        sub ax,43                  ;
        mov [YF],ax                ;
        mov [Corner2+2],ax         ;save YF, Corner2
                                   ;XF unchanged from Centroid
        sub ax,87
        out dx,ax                  ;YF for Col 2, bottom of card, avoid cut

                        ;Preparation for Edges SR to find bottom edge of card
        mov [Port],304h            ;parameters for search for edge, Corner 1
        mov [Step],-16
        mov [MaxCnt],50
        mov [Focus],200
        mov [FailEdge],0
        mov [FailEdge1],0
        mov [ThrldMod],0           ;Thrld calculated in proc using first signal
                                   ;  and Transmtce at card left edge
                                   ;End of prep for bottom edge col 1
;-------------------------------------------------------------------------
        Call EDGES                 ;Bottom edge of card, Col 1, Corner2
;-------------------------------------------------------------------------
ErrorCheck6:                       ;returns with FailEdge1=0/1, FailEdge=0/1
        mov ax,[FailEdge]          ;check for runout
        add [FailCard],ax          ;enable FailCalibrate

        mov dx,316h                ;set for long settling delay
        mov ax,0A3h
        out dx,ax
                                   ;Save positions for Corner2
        mov ax,[Edge]
        mov [Corner2],ax           ;save XF in Corner2+0
                                   ;Prep for Centroid, last scale bar Corner3
        mov dx,304h                ;load XF with 200 retraction from Card Bar
        mov ax,[CardBar]           ;  for start of scale bar search
        sub ax,200
        mov [XF],ax
        out dx,ax                  ;issue XF
                                   ;move (up) to near col 80 at top of card
        mov dx,306h                ;YF origin for search for scale bar at Col 80½
        mov ax,[YF0]
        sub ax,7700
        mov [YF],ax
        out dx,ax                  ;issue YF
```

```
        mov [Port],306h          ;parameters for Centroid, Corner3
        mov [Step],8
        mov [MaxCnt],64
        mov [Focus],200
        mov [Thrld],0
mov ax,[ThrldMod34]
mov [ThrldMod],ax;;;;100;150      ;Thrld becomes first amplitude+150       TTT
                                 ;  with limits 200 to 400
;----------------------------------------------------------------------
       Call CENTROID             ;Corner3, Col 80½
;----------------------------------------------------------------------
ErrorCheck7:                     ;returns with FailBar1=0/1, FailBar=0/1
        mov ax,[FailBar]         ;FailBar=0/1 for runout
        add [FailCard],ax

        mov dx,306h              ;save returned center of scale bar

        mov ax,[Center]
        add ax,43                ;mov to center of Col 80
        mov [YF],ax
        mov [Corner3+2],ax
        out dx,ax                ;Col 80
                                 ;want film transmittance at right edge of card
        mov ax,[CardBar]
        add ax,600
        mov [XF],ax
        sub [YF],150
        mov [Focus],1000
;----------------------------------------------------------------------
        call StrgRead            ;measure transmittance at card right border
;----------------------------------------------------------------------
                                 ;returns with Transmtce, mean signal at
                                 ;  4 positions at right side border of card
                                 ;used for thrld in Centroid next

                                 ;Prep for edge search for Corner 3
        mov dx,304h
        mov ax,[CardBar]
        sub ax,200
        mov [XF],ax
        out dx,ax                ;issue XF for edge search origin

        mov dx,306h
        mov ax,[Corner3+2]
        ;;;;;;add ax,87                ;searches at col 79 to avoid cut corner
add ax,261                            ;search at ccol 78, avoid supercut corner

        mov [YF],ax
        out dx,ax                ;issue YF for edge search origin

        mov [Port],304h          ;parameters for Corner 3 edge search
        mov [Step],16
        mov [MaxCnt],50
        mov [Focus],200
        mov [ThrldMod],0         ;Thrld calculated in proc using first signal

                                 ;  and Transmtce at card right border
;----------------------------------------------------------------------
       Call EDGES                ;Corner3, Col 80 nominal, col 79 actual
;----------------------------------------------------------------------
ErrorCheck8:                     ;returns with FailBar1=0/1, FailBar=0/1
        mov ax,[FailEdge]        ;check for runout
        add [FailCard],ax        ;enable FailCalibrate

        mov ax,[Edge]            ;save position of edge, Corner3
        mov [Corner3],ax
        mov [XF],ax
```

```
                                  ;Preparation for Corner4
        mov dx,304h               ;move XF to bottom of card, column 80
        mov ax,[CardBar]
add ax,[CardPeriod];2000
sub ax,200
        mov [XF],ax
        out dx,ax

        mov dx,306h               ;set YF to origin for Centroid, Corner 4
        mov ax,[YF0];[LeftEdge]
        sub ax,7700               ;retract (up on crt) YF origin for Centroid
        mov [YF],ax
        out dx,ax

        mov [Port],306h           ;parameters for Centroid, Corner4
        mov [Step],8
        mov [MaxCnt],64
        mov [Focus],200
        mov [Thrld],0
        mov ax,[ThrldMod34]
        mov [ThrldMod],ax         ;Thrld becomes first amplitude+ThrldMod34
                                  ;  with limits 200 to 400
;------------------------------------------------------------------------------
       Call CENTROID              ;Corner4, Col 80½
;------------------------------------------------------------------------------
ErrorCheck9:                      ;returns with FailBar1=0/1, FailBar=0/1
        mov ax,[FailBar]          ;FailBar=0/1 for runout
        add [FailCard],ax

        mov dx,306h               ;save YF at center of scale bar at Col 80½
        mov ax,[Center]
        add ax,43                 ;mov to center of Col 80
        mov [YF],ax
        mov [Corner4+2],ax        ;Col 80
        add ax,87
        out dx,ax                 ;Col 79

                                  ;Prep for edge search for Corner4
        mov [Port],304h           ;parameters for Corner4 edge search
        mov [Step],-16
        mov [MaxCnt],50
        mov [Focus],200
        mov [ThrldMod],0          ;Thrld calculated in proc using first signal
                                  ;  and Transmtce at card right edge
;------------------------------------------------------------------------------
        Call EDGES                ;Bottom edge of card, Col 80, Corner4
;------------------------------------------------------------------------------
ErrorCheck10:                     ;returns with FailEdge1=0/1, FailEdge=0/1
        mov ax,[FailEdge]         ;check for runout
        add [FailCard],ax         ;enable FailCalibrate

        mov ax,[Edge]             ;Save positions for Corner4
        mov [Corner4],ax
        mov [XF],ax
        mov dx,304h
        mov ax,[YF]
        out dx,ax
                                  ;All 4 corners have been located for this card
;------------------------------------------------------------------------------
FindTilt:                                 ;Calibrate Tilt on first frame only
        cmp [AftrFirstFrm],0              ;
        jne contin11
        cmp [card],1                      ;select Card1 only
        jne contin10
        mov ax,[Corner3+2]                ;save YF at Corner3 on Card1
        mov [TiltLeft],ax                 ;name it TiltLeft
```

```
contin10:
        cmp [Card],8                    ;select Card8
        jne contin11
        mov ax,[Corner4+2]              ;save YF at Corner4 on Card8 as

        sub ax,[TiltLeft]               ;ax has YF difference, (Card8-Card1)
        mov dx,0
        mov bx,2                        ;want to divide difference by 2.5
        imul bx                         ;start by mul by 2
        mov bx,5
        idiv bx                         ;then div by 5
        add ax,[Tilt]                   ;add correction to original tilt
        mov [TiltOut],ax
;sub ax,15                               ;TiltFudge
        mov dx,30Ch
        out dx,ax
        mov [AftrFirstFrm],1            ;this limits action to first frame
contin11:
        mov dx,306h                     ;move spot position to Corner 1
        mov ax,[Corner1+2]              ;  for settling
        out dx,ax

        mov dx,304h
        mov ax,[Corner1+0]
        out dx,ax
        mov [XF],ax
;-----------------------------------------------------------------------------
CalibrationCheck:
        jmp CheckGeom                   ;Check for rectangularity, every card
CheckGeomRet:                           ;return here every card with
                                        ;  FailGeom=0/32
        cmp [FlagRep],0                 ;increment card scanned count
        jne @16                         ;  if not in repeating mode
        inc [NmbrCdsScnd]
        mov dx,316h                     ;increment Panel Counter
        mov ax,22h
        out dx,ax
@16:
        cmp [FailCard],0                ;check for any failure
        jne FailCal                     ;paths diverge:
                                        ;if failure, go off to FailCal
                                        ;if ok, continue to CalcScal
                                ;End of prep for CalScal
;-----------------------------------------------------------------------------
Fudge:
;        sub [Corner1],15
;        sub [Corner3],15
;        sub [Corner4],15
;        sub [Corner2],15
;         add [Corner1+2],20
;         add [Corner2+2],20
;         add [Corner3],10
;         add [Corner4],10
;-----------------------------------------------------------------------------
Call CalcScal      ;for calculating:
                            ;   column advance
                            ;   base line slant correction
                            ;   row spacing and generation of RowTable
;-----------------------------------------------------------------------------
                                ;Prep for ColScan
        mov dx,306h             ;move YF to column= -1 as origin for
        mov ax,[Corner1+2]      ;  column incrementation
        add ax,[Corner2+2]      ;use mean of Corners 1 and 2
        shr ax,1
        add ax,[YFQuot]
        mov [YF],ax
        out dx,ax
```

```
        mov dx,304h              ;go to top edge of card, is XF origin for
        mov ax,[Corner1+0]       ;  column scanning
        mov [XF],ax              ;later supplemented by correction for slant
        out dx,ax
                                 ;Focus set earlier just after Corner4???
                                 ;Threshold set in ColScan
;-----------------------------------------------------------------------
       Call ColScan              ;Reads 960 hole locations
;-----------------------------------------------------------------------
AfterColScan:            ;ColScan returns with XC for Card Bar search origin for
                         ;  last card scanned, for use by Frame procedure
                         ;YC = YCMargin,    XF = 4096,   YF = 8000
                         ;Error=1 if fail, 0 otherwise
        inc [Card]
        cmp [Card],8             ;check for 9th card, skip to SetXCFrame
        jg SetXCFrame
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
mov ax,[XC]                        ;advance XC for next card
add ax,[CardPeriod];2180
mov [XC],ax
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        mov dx,302h
        mov ax,[YCMargin]
        out dx,ax

        jmp NewCard              ;go back to NewCard after cards 1 through 7
;-------------------------------------------------------------------------
        mov dx,302h
        mov ax,[YC0]
        mov [YC],ax
        out dx,ax                ;issue YC
;-------------------------------------------------------------------------
SetXCFrame:                      ;Prep for ScalRead near Corner4 of last card
        mov ax,[Corner4];[CardBar]
        add ax,200
        mov [XF],ax
        mov [YF],9500
        mov [Focus],1000
;-------------------------------------------------------------------------
        call ScalRead            ;Measure transmittance in Scale Zone
;--------------------------------------------------------------------------
                                 ;returns with ScalTransm for use in threshold
                                 ;  calcs for Frame proc
                                 ;Want XF for last CardBar converted to XC
                                 ;  supplement
        mov ax,[CardBar]         ;Need to convert (XF-4096) to eqivalent
        sub ax,4096              ;  XC increment, add it to XC, remove from XF
        add ax,[XC]
        add ax,2180
        mov [XCFrame],ax
        mov [XF],4096
mov ax,[YC0]
sub ax,10000

mov [YCFrame],ax
        jmp exit1
;-------------------------------------------------------------------------
FailCal:                                 ;Here after calib failure
                                         ;Record 80 "space" chars

        cmp [FlagRep],0
        jne Calcscal
        mov cx,80
        mov bx,[BlockPtr]
IncPtr: mov [DataBlock+bx],32
        inc bx
        loop IncPtr
        add [BlockPtr],80                ;advance Block Pointer
```

```
        push cx                         ;time delay after dummy fill
        mov cx,2
wt1:    mov bx,65000
wt3:    dec bx
        jnz wt3
        loop wt1
        pop cx
                                   ;could be:
                                        ;simple Fail Calib
                                        ;Card Jam
        cmp [FlagRep],1
        je @@16
        inc [NmbrCdsFld]                ;increment failed cards count if
@@16:                                   ;  not in Repeat mode
                                   ;if film failure, go back to PCSCAN via FilmPause
                                   ;if not card jam, continue with next card
        cmp [CardJam],0
        jne FilmPause
        mov [FailCardinFrame],1 ;here for calib failure but not card jam
        jmp AfterColScan          ;End of FailCal wrap-up, continue with next card
FilmPause:
        mov [ResumeScan],1
        mov [Pause],8
        inc [Card]                 ;increment Card number for unscanned cards
;------------------------------------------------------------------------
Out1:
exit1:
        ret                        ;END OF PROCEDURE
;========================================================================
CheckGeom: ;a patch for checking XF and YF squareness, through here every card

        mov ax,[Corner2]                ;find XF size at col 1 and
        add ax,[Corner3]                ;  sub XF size at col 80
        sub ax,[Corner1]             ;;difference indicates non-rectangularity
        sub ax,[Corner4]
        cmp ax,0                        ;calc absolute XF Corner difference
        jnl @@10X
        neg ax
@@10X:                                  ;ax has absolute XF corner difference
        cmp ax,30                       ;max XF diff=29 mils
        jl @@2X
        mov [FailCardinFrame],1
        mov [FailCard],1
        mov [XFCornerDiff],ax           ;save failed XFCD for display
@@2X:
        mov ax,[Corner1+2]              ;find YF size at top of card
        add ax,[Corner4+2]              ;  sub YF size at bottom of card
        sub ax,[Corner2+2]            ;difference indicates non-rectangularity
        sub ax,[Corner3+2]
        cmp ax,0                        ;calc absolute YF Corner difference
        jnl @@10Y
        neg ax
@@10Y:              ;ax has absolute YF corner difference
        cmp ax,15                       ;max YF diff=20 mils
        jl @@2Y
        mov [FailCardinFrame],1
        mov [FailCard],1
        mov [YFCornerDiff],ax           ;save failed YFCD for display
@@2Y:
        jmp CheckGeomRet                ;End of GeomCheck patch
;------------------------------------------------------------------------
CardAdv:                           ;Patch to determine CardPeriod, the XF advance
                                   ;  for cards of variable size
                                   ;Set parameters for Centroid SR
PrepNextCardBar:                   ;here from line 200
        mov ax,[XF]                ;save entering XF
        mov [SaveXF],ax
```

```
        mov ax,[CardBar]          ;set XF origin for second stroke to find
        add ax,1500               ;  CardPeriod
        mov [XF],ax

;mov dx,306h                      ;Use this to see second stroke
;mov ax,7950
;out dx,ax

        mov [Port],304h           ;set for horiz movement
        mov [Step],80             ;set step
        mov [MaxCnt],30           ;allow 30 steps
        mov [Focus],1000          ;set large spot
        mov ax,[ThrldCardBar]
        mov [Thrld],ax            ;End of coordinate initialization for
                                  ;  Centroid on Card Bar
;---------------------------------------------------------------------------
        Call Centroid             ;Horizontal position of next Card Bar
;-----------------------------------------------------------------------------
        mov ax,[Center]
        sub ax,[CardBar]
        mov [CardPeriod],ax
;call prtax                       ;Use this to display CardPeriods
;call line
        mov ax,[SaveXF]
        mov [XF],ax
        jmp CardAdvRtn            ;near line 200
;-----------------------------------------------------------------------------

ENDp scan
End
```

## Appendix B VTRACK.ASM Subroutine procedure

This procedure is assembled with the command TASM VTRACK, automatically including it in PCSCAN0. A description of its action is given in Section 3.3.4.

```
;------------------------------------------------------------------------
;VTRACK.ASM Subroutine to find vertical location of YCMargin
;Makes 12 YC strokes searching for card left edge (below card on screen)
;Strokes stop on reaching dense area at card left edge
;Mean of 4 highest YC values is declared card edge, recorded in Parameters+8
;Initial XC, XF, YF fixed positions during routine
;Final XC is at horizontal origin for Card Bar search
;Final YC is at vertical origin (YCMargin) for Card Bar search

.386
IDEAL
MODEL small

DATASEG
LeftBorder dw   300               ;Use 300 normally or 250 for thin card left
                                  ;   border
Datal   dw      20 dup (0)        ;storage area for peak signal from each col
ThrldSum dd     0                 ;working sum for accumulating 4 peak YCs
YCThrld  dw     (?)               ;working peak YC threshold
ThrldSig dw     200;300               ;working signal threshold, absolute
YCMargin dw     19000             ;default value
RunoutCount dw  ?                 ;count of runout scans, 3 req'd for TrackFail
;FailTrack dw    0
INCLUDE "Scan_Var"

CODESEG
INCLUDE "Scan_Pcd"
proc vtrack
        pusha
        mov ax,[XC0]              ;set XC for origin of scan raster
add ax,1000
        mov [XC],ax
      ;  mov bx,[MaxSig]           ;;calculate signal threshold for edge search
     ;   add bx,[MinSig]           ;;formula is (MaxSig+2*MinSig)/4
     ;   add bx,[MinSig]           ;;
     ;   shr bx,2                  ;;
      ;  mov [ThrldSig],bx         ;;store signal threshold

        mov [RunoutCount],0
        mov [FailTrack],0         ;reset FailTrack
        mov di,0                  ;di carries Column count
NewCol:                           ;start of column
        mov dx,300h               ;starts at XC0, advances 11 steps of 200
        mov ax,[XC]
        out dx,ax

        mov dx,302h               ;set YC origin - YC0-4000
        mov ax,[YC0]
        sub ax,4000
        mov [YC],ax
        out dx,ax                 ;issue YC origin

        mov bx,30                 ;settling delay
@11:    dec bx
        jnz @11

        mov dx,318h               ;start first read
        in ax,dx
```

```
        mov cx,16                 ;16 Reads in column
Read:
        cmp cx,1                  ;skip next if last Read in col
        je @@1

        mov dx,302h               ;advance YC by 100
        add [YC],100
        mov ax,[YC]
        out dx,ax

@@1:    mov dx,318h               ;take in signal amplitude
@10:    in ax,dx
        test ax,8000h
        jne @10
        and ax,1023               ;leave only useful bits

cmp ax,[ThrldSig]       ;compare signal with threshold
        jl EdgeFound
                                  ;finish the col of 16 Reads if Edge not found
        loop Read                 ;otherwise continue to finish 16 Reads in col

;------------------------------
        inc [RunoutCount]         ;through here on runout of any stroke
        cmp [RunoutCount],3       ;need 3 strokes for FailTrack
        jl @11a
        mov [FailTrack],1         ;show Track failure, may be clear area,
@11a:                             ;  caused by runout
        jmp EndCol
;-------------------------------------------------------------------------
EdgeFound:                        ;saves YC and starts next col for sigs<300
        mov ax,[YC]               ;save YC of first signal below threshold
        sub ax,100                ;correction
        mov [Data1+di],ax         ;save YC for left edge as the data for each
col

EndCol:                           ;through here at end of col, likely <16 max
        cmp di,22                 ;look for last column if no Edge found
        jge EdgeCalc
        inc di                    ;advance XC by 200 and col cntr di by 2
        inc di
        add [XC],200              ;
        jmp NewCol
                        ;Through here at end of every column of up to 16 Reads

                                  ;[Data1+di] has YCs for 12 card edge scans
;-------------------------------------------------------------------------

EdgeCalc:               ;Analysis for mean of 4 highest YC values in 12 cols
        mov eax,0
        mov cx,4
        mov [ThrldSum],0          ;clear ThrldSum
@80:    mov si,0
        mov ax,[YC0]              ;move original to working threshold
        sub ax,4700               ;displacement, nominal 15300
        mov [YCThrld],ax          ;YCThrld is working YC threshold

@70:    mov ax,[Data1+si]         ;load ax with next data (YC for edge) value
        cmp ax,[YCThrld]          ;compare it with YCThrld
        jl @60                    ;skip next if edge<YCThrld

        inc ax                    ;a fudge which chooses first 4 of equql peaks
        mov [YCThrld],ax          ;temporarily save any peak value,
        mov bx,si                 ;save si when the peak was found
@60:                              ;  for next pass
        inc si
        inc si
```

```
        cmp si,24                   ;check for going through 12 iterations
        jl @70                      ;repeat for total of 12 times
        mov [Data1+bx],0            ;zero out this peak in preparation for next
                                    ;  pass
        mov ax,[YCThrld]
        add [ThrldSum],eax
        loop @80

        sub [ThrldSum],4            ;minor  correction to remove fudge
        shr [ThrldSum],2            ;divide by 4
        mov eax,[ThrldSum]          ;save card left edge YC
add ax,[LeftBorder]                 ;declare YCMargin from left card edge
        mov [YCMargin],ax
        mov dx,302h
        out dx,ax                   ;return with YC=YCMargin,issued
@55:
        mov dx,300h                 ;return with XC=XC0,issued
        mov ax,[XC0]
        mov [XC],ax
        out dx,ax
        popa
        ret

ENDp    vtrack
END
```

## Appendix C-4 CENTROID.ASM Subroutine procedure

This procedure is assembled with the command TASM CENTROID, automatically including it in the linking file PCSCAN0. A description of its action is given in Sections 3.3.5 and 3.3.6.

```
;-----------------------------------------------------------------------------
;CENTROID.ASM  Universal procedure to locate center of target line

.386
IDEAL
MODEL small

DATASEG
DataAmpl dw     10
ProdSum dd      0
AmplSum dd      0
XYF      dw     10000           ;Default value for working position along stroke
Center   dw     5000
prtbuf   db     20 dup (?)
DataAmpl1 dw    ?               ;first signal amplitude, used in thrld calc
FirstRead dw    ?               ;flag, =0 before first read, =1 after
ThrldCentroid dw ?              ;calculated thrld
INCLUDE "Scan_Var"

CODESEG
INCLUDE "Scan_Pcd"
proc Centroid
        pusha

        mov [FirstRead],0
        mov [ProdSum],  0
        mov [AmplSum],  0
        mov [DataAmpl], 0
        mov [DataAmpl1],0
        mov [FailBar],  0
        mov [FailBar1], 0
;-----------------------------------------------------------------------------
;                       Initialize parameters
        mov dx,30Ah
        mov ax,[Focus]
        out dx,ax
;mov dx,316h            ;for cro sync in focus adjustment
;mov ax,022h
;out dx,ax
;-----------------------------------------------------------------------------
;                       Begin stroke
;               Identify axis of stroke for loading XYF with initial value
;                 in desired axis
        mov dx,[Port]
        cmp [Port],304h
        je @1
        mov ax,[YF]
        jmp @2
@1:     mov ax,[XF]             ;load XF into XYF if Port=304h
@2:     mov [XYF],ax
        out dx,ax               ;issue XYF (XF or YF from loaded Port)

        mov dx,318h             ;start Read at first position in stroke
        in ax,dx

        mov dx,316h             ;shorten settling delay to minimum
        mov ax,03h
        out dx,ax
```

```
        mov cx,[MaxCnt]           ;maximum number of reads in column
Read:
        cmp cx,1
        je @@1
        mov dx,[Port]              ;is 304h for XF or 306h for YF
        mov ax,[XYF]
        add ax,[Step]              ;advance in stroke
        mov [XYF],ax
        out dx,ax

@@1:    mov dx, 318h              ;Reads
wtl:    in ax,dx
        test ax,8000h
        jne wtl
        and eax,1023              ;eax has signals
;---------------------------------------------------------------------------
                                  ;Threshold determination
        cmp [FirstRead],0         ;save first signal amplitude
        jne @@21
        mov [DataAmpl1],ax        ;will use first amplitude in thrld calc
@@21:
        cmp [Thrld],0             ;select preset or calculate threshold
        jne DataAnalysis          ;Thrld=0 means calculate; any other, use it
;---------------------------------------------------------------------------
                                  ;here for calculated thrld
                                  ;calculate thrld as sum of FirstRead and
        mov bx,[DataAmpl1]        ;  ThrldMod
        add bx,[ThrldMod]         ;add increment to first signal when needed
                                  ;  for thrld calc
        cmp bx,200                ;apply range limits of 200 to 300
        jg @001
        mov [Thrld],200
@001:   cmp bx,300
        jl @002
        mov bx,300
@002:   mov [Thrld],bx            ;save calculated thrld
;===========================================================================
DataAnalysis:             ;Data analysis with calculation of centroid
                                  ;eax has amplitude
        mov [FirstRead],1         ;set flag for bypassing thrld calcs
                                  ;  on subsequent reads
        cmp ax,[Thrld]             ;Thrld brought in either from calling SCAN or
        jge @@3                   ;  calculated from from first signal & ThrldMod
        mov eax,0                  ;load eax with 0 for signals below threshold
        jmp @@4
@@3:                              ;ax has signals above threshold, or 0 for below
        sub ax,[Thrld]
@@4:
        mov [DataAmpl],ax         ;save amplitude above threshold
        add [AmplSum],eax         ;AmplSum has cumulative sum of above
        mov ebx,0                 ;clear high order bits
        mov bx,[XYF]              ;Note: bx has (next) position

        mul ebx                   ;dx has high order 16 bits of product
        add [ProdSum],eax         ;accumulate product of amplitude*distance
;---------------------------------------------------------------------------
;                         Read cycling and termination check
@@5:                              ;Want to stop stroke if DataAmpl is 0
                                  ;  and AmplSum is not 0,
                                  ;Means that signals are past the peak and
                                  ;  lower than threshold
        cmp [DataAmpl],0
        jne @@7
        cmp [AmplSum],0           ;through here if DataAmpl=0
        je @@7
        jmp CalcMom               ;terminate reading upon finding low signal
                                  ;  after accumulating values above threshold
```

```
@@7:                                    ;  otherwise continue reading
        loop toRead                     ;repeat to continue stroke
        jmp OutFail
toRead: jmp Read
;                           End of reading in stroke
;-------------------------------------------------------------------------

CalcMom:                                ;Calculate moment
        mov eax,[ProdSum]               ;Formula:  Center=ProdSum/AmplSum
        mov ebx,[AmplSum]
        cmp ebx,0
        je continl

        div ebx                        ;eax has high order 16 bits of quotient (moment)
        sub ax,[Step]                   ;correct for current position
Outl:
        mov [Center],ax                 ;Centroid position along stroke, XF or YF

                                        ;Look for high first signal showing
                                        ;  start in clear
        mov ax,[DataAmpl1]
        cmp ax,[Thrld]
        jl continl
        mov [FailBar1],1
Continl:
        mov bx,170                      ;10 us delay
@@10:   dec bx
        jnz @@10
        popa

        ret                             ;END OF PROCEDURE
;=========================================================================
OutFail:
        mov [FailBar],1                     ;Fail=1 indicates Card or Scale Bar
        jmp Outl                            ;  not seen by end of stroke

ENDp centroid
End
```

## Appendix C-5 EDGES.ASM Subroutine procedure

This procedure is assembled with the command TASM EDGES, automatically including it in the linking file PCSCAN0. A description of its action is given in Section 3.3.8.

```
;----------------------------------------------------------------------------
;EDGES.ASM  Universal scan procedure to locate edges, opaque to clear
;               Interpolates to find location of position whose signal
;                 equals threshold
;               Reports "1" in FailEdge on scan runout

.386
IDEAL
MODEL small

DATASEG

Edge     dw      10000              ;default
DataAmpl dw      100 dup (0)        ;signal amplitudes
TempXYF dw       (?)                ;temporary XYF mem
XYF      dw      (?)                ;Fine coordinate value, XF or YF
XYF0     dw      ?                  ;initial coordinate origin for Edges stroke
FirstRead dw     ?                  ;identifies first signal,=0/1
zzz      dw      100 dup (0)

INCLUDE "Scan_Var"

CODESEG
INCLUDE "Scan_Pcd"
proc Edges

        pusha
;                       Initialize parameters
        mov dx,30Ah
        mov ax,[Focus]
        out dx,ax
;---------------------------------------------------------------------------
start1:
        mov [FirstRead],  0             ;reset FirstRead
        mov [FailEdge],   0             ;reset runout failure flag

        mov [FailEdge1],  0             ;reset high first read flag

                        ;Clear data memory
        push di
        push si
        mov si,0
        mov ax,SEG DataAmpl
        mov es,ax
        mov di,offset DataAmpl
        mov ax,0
        mov cx,200
        cld
        rep stosw
        pop si
        pop di
;                        End of memory clear
;--------------------------------------------------------------------------
;                       Identify axis of stroke
        mov dx,[Port]
        cmp [Port],304h
        je @1
        mov ax,[YF]
        jmp @2
@1:     mov ax,[XF]
```

```
;------------------------------------------------------------------------
;                          Begin stroke
@2:     mov [XYF],ax                    ;ax has either XF or YF as appropriate

        out dx,ax
mov [XYF0],ax
        mov dx,318h             ;first Read
        in ax,dx

        mov cx,[MaxCnt]         ;maximum number of reads in column
Read:
        mov bx,[XYF]            ;Transfer position of current Read
        mov [TempXYF],bx        ;  to TempXYF leaving ax alone

        cmp cx,1                ;skip advance if last Read
        je @@1
        mov dx,[Port]           ;advance XF
        mov ax,[XYF]
        add ax,[Step]           ;add Step
        mov [XYF],ax
        out dx,ax               ;issue advance

@@1:    mov dx, 318h            ;Reads
wt1:    in ax,dx
        test ax,8000h
        jne wt1
        and ax,1023
;----------------------------------------------------------------------------

        mov [DataAmpl+di],ax    ;save signals in DataAmpl+
        cmp [DataAmpl],300
        jle contin5
        mov [FailEdge1],1
        jmp Out1
;---------------------------------------------------------------------------

contin5:                                ;pick out first signal amplitude for
Thrld calcs
        cmp [FirstRead],0       ;Thrld is (DataAmpl + Transmtce)/2
        jne Analysis
        add ax,[Transmtce]      ;ax has sum of first signal and Transmtce at
        shr ax,1                ;  left or right card edge
        add ax,[ThrldMod]       ;add threshold modification if needed
cmp ax,100;200               ;apply range limits 100 to 400
        jg @001
mov ax,100;200
@001:   mov [Thrld],ax
        cmp ax,400
        jl @002
        mov ax,400
@002:   mov [Thrld],ax
        mov [FirstRead],1
;---------------------------------------------------------------------------
@003:
        jmp Analysis            ;Looking for position whose signal equals
threshold
AnalysisRtn:                    ;Come back here afterwards

        inc di
        inc di
        loop Read
        mov [FailEdge],1       ;"1" indicates failure to find edge, runout
        jmp Out1
;                               End of stroke generator
```

```
;--------------------------------------------------------------------------
Analysis:                           ;Data analysis
                                    ;ax has amplitude for position past threshold
;                      Search for signals immediately before and after
threshold
        cmp ax,[Thrld]
        jng AnalysisRtn             ;return to scanning if signal below threshold
;--------------------------------------------------------------------------

;                                    Interpolation, here when signal exceeds
threshold
        mov bx,[DataAmpl+di]
        sub bx,[DataAmpl+di-2]      ;bx has amplitude difference
        mov dx,[Thrld]              ;
        sub dx,[DataAmpl+di-2]      ;dx:ax has Thr50 minus prior amplitude

        mov ax,0                    ;  and bx has amplitude difference
        cmp dx,bx                   ;avoid divide overflow
        jae Out1
                                    ;Multiply by Step
        div bx                      ;ax now has interpolated increment relative to
        mov bx,[Step]               ;  65536
        cmp bx,0                    ;looking for neg Step
        jge @01                     ;jmp if pos Step
        neg bx                      ;if Step is neg, reverse polarity tempo
        mul bx                      ;multiply Step by fractional increment
        neg dx                      ;reverse polarity back to negative
        jmp @02                     ;go to summation with last position, the one
@01:                                ;just past threshold
        mul bx
@02:                                ;dx has interpolated increment
                                    ;XYF has position past threshold
        add dx,[TempXYF]
        sub dx,[Step]
        mov [Edge],dx               ;This is Edge, interpolated to threshold
;--------------------------------------------------------------------------
Out1:
        mov bx,200                  ;minor time delay
@@3:    dec bx
        jnz @@3

;mov ax,[DataAmpl]                   ;declares FailEdge1=1 if first amplitude
;cmp ax,400                          ;  is clear (>400), is used in Stopping Logic
;jle contin5                         ;  in PCSCAN
;mov [FailEdge1],1
;contin5:

        popa
        ret                         ;END OF PROCEDURE
;==========================================================================


ENDp Edges
END
```

# Appendix C-6 CALCSCAL.ASM Subroutine procedure

This procedure is assembled with the command TASM CALCSCAL, automatically including it in the linking file PCSCAN0. A description of its action is given in Sections 3.3.10 through 3.3.10.3.

```
;----------------------------------------------------------------------------
;CALCSCAL.asm  Procedure to calculate scaling constants
;Interprets corner coordinates into scale factors for reading punched holes
;Generates scaled constants for column spacing in YF
;    "        "       "         slant correction in XF
;    "        "       "         row spacing in XF, creates RowTable

.386
Ideal
Model small

DATASEG
RowCorrIn dw    0                  ;'0' disables RowCorr, '1' enables

Height  dw      ?
CardWidth dw    ?
Slant   dw      ?
SlantNeg dw     ?
YFQuot  dw      ?                ;Used for column positioning
YFRemdr dw      ?
SlantRemdr dw   ?                ;Used for slant correction
XFQuot  dw      ?                ;This group used in making up RowTable
XFRemdr dw      ?
RowTable dw     26 dup (?)
SlantSum dw     ?
SlantRemdrSum dw ?
XFRemdrSum dw   ?
XFBase  dw      ?
RowTableSum dw  ?
RowCorr dw        0,1,5,10,15,20,20,15,10,5,1,0
INCLUDE "Scan_Var"

CODESEG
INCLUDE "Scan_Pcd"

proc CalcScal
;----------------------------------------------------------------------------
;                        YF Section - 80 Column locations in YF
;                         Calculate quotient and remainder for Column Advance
        mov ax,[Corner1+2]      ;derive mean span between cols 1 and 80
        add ax,[Corner2+2]
        sub ax,[Corner3+2]
        sub ax,[Corner4+2]
        shr ax,1
        mov [CardWidth],ax      ;save span between cols 1 and 80

        mov dx,0
        mov bx,79
        div bx                  ;ax=87 nominal
        mov [YFQuot],ax         ;Save quotient as base for col. no.
                         ;  incrementation
        mov [YFRemdr],dx        ;Save remainder in YFRemdr

;                         YFQuot has integer quotient, YFRemdr has remainder
;                         End of YF section
;----------------------------------------------------------------------------
```

```
;                          Slant - Calculation of XF Base correction
        mov ax,[Corner3+0]       ;Derive mean of card slant using cols 1 and 80
        add ax,[Corner4+0]
        sub ax,[Corner1+0]
        sub ax,[Corner2+0]
        sar ax,1                  ;ax has Slant in mils between cols 1 and 80
        mov [Slant],ax            ;save Slant, clockwise slant is positive
;                          End of Slant section
;------------------------------------------------------------------------------
;                          Calculate quotient and remainder for row intervals
        mov ax,[Corner2]         ;derive mean of card height from corner XFs
        add ax,[Corner4]
        sub ax,[Corner1]
        sub ax,[Corner3]
        shr ax,1
        mov [Height],ax          ;save card height
        mov dx,0
        mov bx,13                ;card height is 13 row spaces
        div bx                   ;ax=134 nominal
        mov [XFquot],ax          ;Save quotient as base for col. no.
incrementation
        mov [XFRemdr],dx         ;Save remainder for fractional incrementation
;                           End of Row Intervals section
;------------------------------------------------------------------------------
;                                Create RowTable of 12 row positions measured
;                                   from top of card, using above parameters
;                                Inputs:  XFQuot, XFRemdr, scaled to height
        mov di,0
        mov [XFRemdrSum],0
        mov [RowTable],0
        mov [RowTableSum],0

        mov cx,12                ;set for 12 iterations beyond row 0 (card
edge)
NewRow:                          ;start with processing the remainder
        mov ax,[XFRemdrSum]
        add ax,[XFRemdr]         ;ax has accumulated remainders

        cmp ax,13;7              ;use half interval as breakpoint for rounding
off
        jl @@9                   ;through below when sum exceeds breakpoint

        sub ax,13                ;retract sum of remainders by whole interval
        add [RowTableSum],1      ;apply 1-mil correction to position

@@9:    mov [XFRemdrSum],ax      ;save new remainder sum
        mov ax,[RowTableSum]
        add ax,[XFQuot]          ;add the integer quotient
        mov [RowTableSum],ax     ;RowTableSum accumulates 25 half-row coords
;------------------------------------------------------------------------------
RowTableCorrections:             ;possibly add a correction to RowTable values
        cmp [RowCorrIn],0        ;raises reading positions toward top of card
        je @12
        sub ax,[RowCorr+di]
@12:    mov [RowTable+di],ax     ;Then load RowTable+di as values are generated
;------------------------------------------------------------------------------
        inc di
        inc di
        loop NewRow             ;start new row
                        ;End of RowTable calculations
;------------------------------------------------------------------------------
Out1:
        ret

ENDp CalcScal
End
```

## Appendix C-7 COLSCAN.ASM Subroutine procedure

This procedure is assembled with the command TASM COLSCAN, automatically including it in the linking file PCSCAN0. A description of its action is given in Sections 3.3.11 through 3.3.11.10.

```
;---------------------------------------------------------------------------
;COLSCAN.asm  Procedure to read 80 columns
;               Uses parameters from CalcScal SR
;               Requires Includes "TrnsTbl1" and "Scan_Var" in DATASEG
;               Requires Include "Scan_Pcd" in CODESEG
;               XC = horizontal origin of search for Card Bar, fixed
;               YC = vertical      "           "          "      YCMargin
;               Initial XF = mean of Corners 1 and 2, top edge of card
;               Initial YF = Col (-1) issued in proc, needed for rescan
;               Final XF = 4096, origin for Card Bar search
;               Final YF = 8000
;               Final XF = 4096, origin for Card Bar search
;               Checks for weak signals, one or more causes 5-position rescan
;                 of entire card, with offsets
;               Weak signals are those falling between 50% and 75% of span ;
;                 between ColBkgnd and Minsig
;               Weak signal check is conducted at every column
;               Recognized punches have signals between 50% of span and MinSig
;               Offsets from central position are 15 mils to left side of
;                 punched hole, at nominal center, at 15 mils to right (card ;
;                 orient) also 25 mils above and below center
;               Compares signal amplitudes at the 5 positions in each row,
;                 picks best as signal for this row and column hole location
;               Displays rescan count every 640 cards
;               Spot diameter apx 30 mils

.386
IDEAL
MODEL SMALL

DATASEG
;-------------------------------------------------------------------------
;Fudges

SpotSize0      dw    500       ;spot size for Mode0
SpotSize1      dw    400       ;spot size for Mode1
BkgndFudge     dw    ?         ;working BkgndFudge
BkgndFudge0    dw    150       ;used to lower thresholds for dense films, Mode0
BkgndFudge1    dw    100;50        ;used to set thresholds for thin holes, Mode1
;-------------------------------------------------------------------------
Read5          dw    ?
ColResult      dw    0                   ;Identification of row with strongest
signals
CountIllegals dw     ?
ColSig         dw    26 dup (1001)       ;temp storage for signals in column
WeakSigs       dw    ?
ColBkgnd       dw    ?
Pickup         dw    ?
YFRemdrSum     dw    (?)                 ;working remainder in division for
col-col advance
SlantSum       dw    (?)                 ;slant from col 1 to col 80
SlantRemdrSum dw     (?)                 ;working remainder in division for
Slant
SlantCorr      dw     ?
SlantNeg       dw     ?
SlantQuot      dw     ?
XFBase         dw    (?)                 ;XF at top edge of card at specified
                                         ;  col
```

```
;DATASEG continued
Min5       dw    ?
Sum5       dw    ?
InpA       dw    ?
InpB       dw    ?
InpC       dw    ?
ColThrld   dw    ?
ColThrldWeak dw ?
ColThrld1 dw     ?                  ;working thrld in search for strongest signal
ColThrld2 dw     ?                  ;working thrld in search for second strongest
ColThrld3     dw ?                  ;working thrld in search for third strongest
BlockPtr0 dw     ?                  ;temp storage of blockpointer
OffsetXF  dw     0,0,0,-25,25       ;XF positional offsets for reading in search
OffsetYF  dw     0,15,-15,0,0       ;YF same
deltaXF   dw     ?
deltaYF   dw     ?
Mode      dw     ?

INCLUDE "TrnsTbl1"
INCLUDE "scan_var"

CODESEG
INCLUDE "Scan_Pcd"

proc colscan                ;through here at beginning of every card
                            ;start in Mode=0
                            ;scan all 80 columns in single-read in nominal center
                            ;at end of data, review number of weak signals
                            ;if any weak ssignal in single-read pass, call for
                            ;  rescanning all 80 columns in multi-read pass,
                            ;order of hole rescan is 1)center 2)left 3)right,
                            ;  4)above center 5)below center, card orientation
                            ;final data is best seen in any of the 5 positions
                            ;record the data after these passes as best in row
                            ;report drops only after multi-read pass scan
                            ;resume scan of next card in single read mode

        mov [Mode],0                ;start card in single-read mode
        mov ax,[SpotSize0]          ;set initial spot size for Mode0 in ax
        mov bx,[BkgndFudge0]        ;set initial background fudge for Mode0 in bx
;---------------------------------------------------------------------------

RepeatCard:                         ;return here for rescanning card in Mode=1
        cmp [Mode],0                ;here to switch fudges according to Mode
        je @@1
        mov ax,[SpotSize1]          ;change spot size for Mode1
        mov bx,[BkgndFudge1]        ;change bkgnd fudge for Mode1
@@1:
        mov dx,30Ah                 ;issue spot size for reading punched holes

        out dx,ax                   ;typically 30 mils in Mode0, 22 mils in Mode1
        mov [BkgndFudge],bx         ;issue bkgnd fudge

        mov [Read5],   0
        mov [WeakSigs],0
        mov [CountIllegals], 0
        mov [NmbrColPickups],0
        mov dx,306h                 ;use mean of YFs for Corners1 and 2 for
        mov ax,[Corner1+2]          ;  reference
        add ax,[Corner2+2]
        shr ax,1                    ;derive mean
;sub ax,20                          ;place for YF fudge
        add ax,[YFQuot]             ;mov to (card) left by one column
        mov [YF],ax
        out dx,ax                   ;issue YF origin for card, (Column=-1)
```

```
        mov dx,304h             ;derive XF origin for card, (Corner1)
        mov ax,[Corner1+0]
        mov [XF],ax
        out dx,ax

        mov bx,170              ;settling time delay apx 10 us
wt2:    dec bx
        jne wt2

        mov ax,[BlockPtr]       ;save entering BlockPtr for this card
        mov [BlockPtr0],ax
;--------------------------------------------------------------------------
;                       YF application for column position
;                       Inputs: YFQuot, YFRemdr
        mov dx,316h             ;shorten settling time to minimum
        mov ax,03h
        out dx,ax

        mov [YFRemdrSum],   0   ;initialize some variables
        mov [SlantRemdrSum],0
        mov [SlantCorr],    0
        mov [SlantNeg],     0
        mov si,             0   ;si addresses columns, si=0 is for column 1
;--------------------------------------------------------------------------
        mov cx,80               ;set for 80 iterations starting from -01
NewCol:                         ;iteration starts with one-column advance
        mov [ColThrld],998
        cmp [Mode],1
        jne RepeatCol
        cmp [Read5],0
        jne RepeatCol
        push cx
        push di
        mov di,0
        mov cx,12
ClearCol:
        mov [ColSig+di],1002
        add di,2
        loop ClearCol
        pop di
        pop cx
RepeatCol:
        mov bx,[Read5]          ;prep for deriving offsets.  do this at start
        mov ax,[OffsetXF+bx]    ;  of every column instead of at every read
        mov [deltaXF],ax        ;load deltaXF with next XF offset
        mov ax,[OffsetYF+bx]
        mov [deltaYF],ax        ;load deltaYF with next YF offset

        cmp [Read5],0           ;use Read5=0 for sole execution of
                                ;  YFCalculation
        jne ScanStart           ;do column advance only once per column
;--------------------------------------------------------------------------
YFCalculation:                  ;start with processing remainder
        mov ax,[YFRemdrSum]
        add ax,[YFRemdr]        ;ax now has sum of remainders
        cmp ax,79               ;when sum of remainders exceeds one mil,
                                ;  remove one mil, keep on with remainder sum
        jl @@2
        sub ax,79               ;retract sum of remainders by whole interval
        sub [YF],1              ;apply 1-mil upward correction to YF position
@@2:
        mov [YFRemdrSum],ax     ;remainder sum cannot exceed 158
        mov dx,306h
        mov ax,[YF]             ;apply integer part of positional increment
        sub ax,[YFQuot]         ;subtract toward higher numbered cols
        mov [YF],ax
        out dx,ax               ;issue YF for column location
;                       End of column YF application
```

```
;----------------------------------------------------------------------------
;                        Slant application   Through here every column
;                                Input:  Slant, Corner1
        mov ax,[Slant]
;ADD AX,20                       ;a FUDGE for biasing Slant
        cmp ax,0                ;reverse to positive polarity if Slant is
negative
        jge @5                  ;processing is done in positive polarity only
        neg ax
        mov [SlantNeg],1        ;flag for negative slant

@5:     mov dx,0                ;prep for division
        mov bx,79               ;want increment for each column
        div bx
        mov [SlantQuot],ax      ;dx has remainder, ax and SlantQuot the
                                ;  quotient
        mov bx,[SlantRemdrSum]  ;bx has remainder Sum

        add bx,dx               ;add new remainder to Sum
        cmp bx,79               ;when sum of remainders exceeds 79, want to
        jl @@3                  ;  subtract 79 and add 1 to correction
        sub bx,79
        inc [SlantCorr]
@@3:
        mov ax,[SlantCorr]      ;want total of correction terms
        add ax,[SlantQuot]
        mov [SlantCorr],ax      ;sum of terms from quotient and remainders
        mov [SlantRemdrSum],bx  ;save updated remainder sum

        mov ax,[Corner1]        ;add or subtract correction
        cmp [SlantNeg],0        ;  slant direction
        je @@4
        sub ax,[SlantCorr]      ;through here for negative slant
        jmp @@5
@@4:    add ax,[SlantCorr]      ;through here for positive slant
@@5:    mov [XFBase],ax         ;XFBase is at top edge of (slanted) card,
;                        End of Slant application
;----------------------------------------------------------------------------
ScanStart:                      ;Start of punched data reading in column

                        ;RowTable has 12 scaled individual offsets from XFBase
        mov di,    0            ;  at top of card.   ax has XFBase
        mov [Sum5],0
        mov bx,    0            ;bx will identify row with strongest signals
        mov [InpA],0
        mov [InpB],0
        mov [InpC],0
        mov [Min5],999          ;arbitrary high initial thrld
;----------------------------------------------------------------------------

XFCalculation:
        mov dx,304h             ;do while scanning and loading ColSig
        mov ax,[XFBase]
        add ax,[RowTable+di]
;sub ax,95                      ;XF fudge for offsetting top row
        add ax,[deltaXF]        ;add offset to XF in single & multi-read scans
        mov [XF],ax             ;  offset deltaXF is zero in single-read mode
        out dx,ax               ;issue first read coord in column

        cmp [Mode],0            ;bypass deltaYF in single-read mode to save
time
        je @20                  ;  no bypass in multi-read mode
        mov dx,306h             ;
        mov ax,[YF]
        add ax,[deltaYF]        ;add offset to YF in multi-read scan, Mode=1
        out dx,ax
@20:
```

```
;@20:
        mov dx,318h             ;call for first read
        in ax,dx
        add di,2                ;offset di by 2    (di carries rows)
                                ;  (di=0 at top of card)
        push cx
        mov cx,12
NewRow:                         ; here at start of column
        cmp cx,1
        je @7                   ;skip the advance for last Read

        mov dx,304h
        mov ax,[XFBase]
        add ax,[RowTable+di]
        add ax,[deltaXF]        ;add offset in mult-read scans
        out dx,ax               ;
        mov [XF],ax             ;YF and XF-RowTable have current values
@7:
        mov dx,318h             ;Reads
wtl:    in ax,dx
;============================================================================
                           ;Switch for shift between single and multi-read
        cmp [Mode],0            ;skip switch in single read, Mode=0
        je go_on
        test ax,8000h           ;here to try again for better signal in
        jne wtl                 ;  another offset position, want lowest value
        and ax,1023             ;new value is in ax
        cmp ax,[ColSig+di-2]    ;use new ax if lower than last best signal
        jb go_on1
        mov ax,[ColSig+di-2]    ;load ax with last best signal otherwise
        jmp BkgndCalculate

go_on:  test ax,8000h           ;through here in Mode 0
        jne wtl
        and ax,1023
go_on1: mov [ColSig+di-2],ax    ;saves 12 amplitudes in ColSig+

;----------------------------------------------------------------------------
BkgndCalculate:                 ;here both Modes
                                ;compute background level as mean of 4 mid-row
        cmp [read5],0           ;  unpunched locations, at hole center only
        jne cont1               ;actually add up 5 and subtract the lowest
        cmp di,10               ;Want min. sig in group of 4 central rows

        jb cont1                ;Take in readings on 5 rows, throw out the
        cmp di,18               ;  lowest as a possible hole signal
        ja cont1
                                ;through here for rows 4-8
        cmp ax,[Min5]           ;is done while scanning on middle 5 rows
        ja cont2
        mov [Min5],ax           ;save temp min in group of 5
cont2:
        add [Sum5],ax           ;add all 5 sigs to Sum5
        cmp di,18
        jne cont1

        mov bx,[Sum5]           ;is calculated for every column
        sub bx,[Min5]           ;sub contribution from lowest sig in middle 5
        shr bx,2                ;divide by 4 for mean ColBkgnd
        mov [ColBkgnd],bx       ;
cont1:
        add di,2
        loop toNewRow           ;go back to cover 12 rows
        jmp @@65
toNewRow: jmp NewRow

@@65:           pop cx          ;ColSig+ is now loaded with row signals
;call show                      ;Scanning completed in column
```

```
;-----------------------------------------------------------------------
OffsetSwitch:
        cmp [Mode],0            ;switching for offset shifts
        je Threshold            ;bypass if in single-read mode (Mode=0)
        add [Read5],2           ;Read5 can be 0 = no read offset (card orient)
        cmp [Read5],10          ;             2 = offset left
        jae Threshold           ;             4 = offset up
        jmp repeatcol           ;             6 = offset right
                                ;             8 = offset down
;-----------------------------------------------------------------------
Threshold:                      ;Threshold determination
        mov ax,[ColBkgnd]
        sub ax,[BkgndFudge]     ;a fudge for dense films, 100=normal
        mov bx,ax
        sub bx,60               ;span is in bx
        shr bx,2                ;bx has 1/4 span
        sub ax,bx               ;ax has ColBkgnd-1/4 span
        mov [ColThrldWeak],ax   ;save it
        sub ax,bx               ;ax has ColBkgnd-1/2 span
        shr bx,1                ;bx now has 1/8 span
                                ;ColThrld is 1/2 span below
(ColBkgnd-ColFudge)
        mov [ColThrld],ax       ;is constant for column; save it
;call show
        mov [ColThrld1],ax      ;initialize all three working thresholds
        mov [ColThrld2],ax
        mov [ColThrld3],ax
;-----------------------------------------------------------------------
        mov dx,304h             ;go to top of card for next column;
        mov ax,[XFBase]         ;can get XF increment under way early
        out dx,ax
;-----------------------------------------------------------------------
DataAnalysis:                   ;performed after all data loaded

;call show
        mov bx,0                ;through here at end of every column
        mov di,2
        push cx
        mov cx,12
Strongl:                        ;Find row with lowest signal
        mov ax,[ColSig+di-2]    ;ax has 12 signal amplitudes

        cmp ax,[ColThrldWeak]   ;skip further examination if signal is too
                                ;  weak
        jg @14al                ;  (value too high)
        cmp ax,[ColThrld]
        jl @14a                 ;jump if signal value below standard ColThrld
                                ;through here for signal values between
                                ;  ColThrldWeak and ColThrld
        inc [WeakSigs]          ;save incidence of weak signal, to trigger
@14a:                           ;  rescan of card
        cmp ax,[ColThrld1]      ;try signal against standard, stiffer Thrld
        jg @14al                ;skip next 2 instructions for signals above
                                ;  thrld
                                ;through here for (strong) signal values less
                                ;  than standard Thrld, ColThrld
        mov [ColThrld1],ax      ;save the strongest (lowest amplitude) signal
                                ;  so far
        mov bx,di               ;bx has di word for last row in which valley
                                ;  seen
@14al:
;call show
                                ;bx will point to row with min signal unless
                                ;  all signals are above 1000, then bx=0
        add di,2                ;advance di for next row
                                ;   (di carries rows*2, di=2 for top row)
        loop Strongl            ;take on next row
```

```
        pop cx                  ;ColSig+bx in first pass is best signal in col
        mov [InpA],bx           ;save (word) location of strongest signal
        mov [ColSig+bx-2],1001  ;disqualify strongest signal for next pass
                                ;(bx-2) has 2X location of least signal in
                                ;  column
                                ;ColThrld1 has strongest signal in column
        shr [InpA],1            ;convert (word) location to byte

                                ;InpA byte has position of strongest or
                                ;  0 if blank
;call show                      ;End of Strong1.
;-------------------------------------------------------------------------------
BlankTest:                      ;Skip search for second and third strongest if
        cmp [InpA],0            ;  first is blank
        je Encode
;-------------------------------------------------------------------------------
Strong2In:                      ;find location of second strongest signal
        mov di,2                ;  also disqualify its amplitude
        mov bx,0
        push cx
        mov cx,12
Strong2:
        mov ax,[ColSig+di-2]            ;load signals into ax for comparison
        cmp ax,[ColThrld2]              ;  with working thrld
        jg @14b
        mov [ColThrld2],ax
        mov bx,di
@14b:   add di,2
        loop strong2
        pop cx
        mov [InpB],bx
        mov [ColSig+bx-2],1002
        shr [InpB],1            ;End of Strong2, continue with Strong3
;-------------------------------------------------------------------------------
Strong3In:                      ;find location of third strongest signal
        mov di,2
        mov bx,0
        push cx
        mov cx,12
Strong3:
        mov ax,[ColSig+di-2]
        cmp ax,[ColThrld3]
        jg @14c
        mov [ColThrld3],ax
        mov bx,di
@14c:   add di,2
        loop strong3
        pop cx
        mov [InpC],bx
        mov [ColSig+bx-2],1003   ;don't really need this except for testing

        shr [InpC],1
;-------------------------------------------------------------------------------
MaskCheck:                      ;Check mask file for encoding instructions
;call show
        cmp [MaskBuf+si],'0'
        jg on6
        jmp DblOvrPunch         ;if MaskBuf+si is 0, allow full encoding
on6:                            ;through here if MaskBuf greater than 0,
        cmp [InpB],0            ;  indicating single read
        jng DblOvrPunch         ;if 2 or more punches, dec MaskBuf
        mov [InpB],0            ;  and jam InpB & InpC=0
        mov [InpC],0            ;means that after 2 double punches in a
        dec [MaskBuf+si]        ;  column, it changes to full alpha encoding
;-------------------------------------------------------------------------------
DblOvrPunch:    ;At this point, InpA has location of row with strongest signal
                ;               InpB        :        "       second strongest
                ;               InpC        "        "       third strongest
```

```
;call show
                ;For blank col, all three = 0 but blank cols have bypassed
                ;  here
                ;Next step is to look for combination of two overpunches
                ;  (YX) and a numeral (0-9)
        mov [ColResult],0       ;initialize ColResult
        cmp [InpA],1            ;is InpA=1?
        je @01
        cmp [InpB],1            ;is InpB=1?
        je @02
        cmp [InpC],1            ;is InpC=1?
        je @03
        jmp Encode              ;go on to encode alpha char according to data
                                ;can't be YX overpunch without a '1"
;--------------------------------------------------------------------------

@01:    cmp [InpB],2            ;here if InpA=1, check for YX=AB
        je OutC
        cmp [InpC],2            ;check for YX=AC
        je OutB

@02:    cmp [InpA],2            ;here if InpB=1, check for YX=BA
        je OutC
        cmp [InpC],2            ;check for YX=BC
        je OutA

@03:    cmp [InpA],2            ;here if InpC=1, check for YX=CA
        je OutB
        cmp [InpB],2            ;check for YX=CB
        je OutA
        jmp Encode              ;means Y and X rows are not both punched
;-----------------------------------------------------------------------

OutA:   mov ax,[InpA]           ;come to 1 of these 3 when both Y and X
        jmp OutYX               ;  rows punched
OutB:   mov ax,[InpB]           ;the one used is the one which is not 1 or 2
        jmp OutYX
OutC:   mov ax,[InpC]
        jmp OutYX
OutYX:
        add ax,237              ;add offset=237 for access to codes for
numerals
        mov [ColResult],ax      ;  accompanying YX overpunching. YX only = 237
        jmp Translat
;-------------------------------------------------------------------------
Encode:                         ;Want to encode InpA and InpB (the two
strongest)
                                ;  with lower value in high nibble of
ColResult
                                ;InpA has row position of strongest signal
                                ;InpB has next strongest
                                ;Simplifies encoding to have single entry for
                                ;any alpha character

                                ;swap nibbles if needed for lower ColResult
        mov ax,[InpA]
        mov bx,[InpB]
        cmp ax,bx
        ja @003
        shl ax,4
        jmp @004
@003:   shl bx,4
@004:   add ax,bx
        mov [ColResult],ax      ;al has entry for Translate table
```

```
                                ;ColResult low byte has two nibbles:
                                ;high nibble has lower of InpA or InpB
                                ;low nibble has higher of InpA or InpB
                                ;InpA or InpB=1 for top (Y) row of card
                                ;ColResult low byte enters the translate table
                                ;End of processing of ColResult
                                ;through here at end of DataAnalysis
                                ;here every column including blanks
;-----------------------------------------------------------------------------
Translat:
;call show
        mov bx,offset TrnsTbl1  ;name of look-up table, 256 bytes "TrnsTbl1"
        xlatb                   ;al has converted value from look-up table
                                ;through here every col

        mov bx,[BlockPtr]       ;Block Ptr is column pointer from start of
        mov [DataBlock+bx],al   ;  DataBlock (51200 cols, 640 cards)
        inc [BlockPtr]
;---------------------------------------------------------------------------
        cmp al,'*'              ;want to declare FailCal if more than 8 '*'
        jne @@8;next1           ;  on a card
        inc [NmbrColPickups]
@@8:    cmp al,'9'              ;same for '9' or  combined total of 8
        jne next1
        inc [NmbrColPickups]

next1:  inc si
        loop toNewCol           ;start the next column, or go to ModeSwitch
        jmp ModeSwitch          ;  at end of card
toNewCol:
        mov [Read5],0
;call show
        jmp NewCol
;--------------------------------------------------------------------------
ModeSwitch:                     ;do this if in Mode=0
;;;;mov [WeakSigs],0             ;use this to disable rescanning
        cmp [WeakSigs],0        ;check for indication of weak signal
        je Exit1                ;go out to scan next card if strong signals
        cmp [Mode],0
        jne Mode1
                                ;through here if WeakSigs not equal 0 in Mode0
        mov ax,[BlockPtr0]      ;here if rescanning required
        mov [BlockPtr],ax       ;retract BlockPtr
        mov [Mode],1

        cmp [FlagRep],1
        je @@21
        inc [NmbrCdsRescnd]     ;increment nmbr of cards rescanned if not
                                ;  repeat
;-----------------------------------------------------------------------------
@@21:
        jmp RepeatCard          ;go back to initiate card Mode=1 for repeat 5X
Mode1:                          ;here with results for Mode=1
        mov [Mode],0
Exit1:                          ;go back to Scan for next card
        cmp [NmbrColPickups],0
        je on3
        cmp [FlagRep],1
        je on3
        inc [NmbrCdsPickups]
on3:
        cmp [NmbrColPickups],20
        jle RestoreCoords
                                ;through here when 5 or more '*' in card,
        mov ax,[BlockPtr0]      ;  make it record 80 spaces instead
        mov [BlockPtr],ax
```

```
        mov cx,80               ;record 80 spaces
        mov bx,[BlockPtr]
IncPtr: mov [DataBlock+bx],32
        inc bx
        loop IncPtr
        add [BlockPtr],80
        cmp [FlagRep],1
        je RestoreCoords
        inc [NmbrCdsFld]
;-----------------------------------------------------------------------
RestoreCoords:
        mov dx,304h             ;move XF and YF back to origin of
        mov ax,4096             ;  Card Bar search for this card
        mov [XF],ax
        out dx,ax
        mov dx,306h
        mov ax,8000
        out dx,ax

;call show                               ;through here after every card
        ret                     ;END OF PROCEDURE
;======================================================================
show:
        cmp [Card],8
        jne out2
        cmp si,1
        jne out2
        call line
        push si
        mov si,0
        push cx
        mov cx,12
con:    mov ax,[colsig+si]
      ;  mov ax,[junk+10+si];[ColThrldWeak]
        call prtax
        call spacex
        inc si
        inc si
        loop con
        pop cx
        pop si
        call line
out2:
        ret

ENDp colscan
End
call show at line 298
```

Appendix C-8 STRGREAD.ASM Subroutine procedure

This procedure is assembled with the command TASM STRGREAD, automatically including it in the linking file PCSCAN0. A description of its action is given in Section 3.3.7.

```
;----------------------------------------------------------------------
;STRGREAD.ASM Procedure to measure transmittance in specified location
;Output is mean transmittance signal from four reads with defocused spot
;Reads four positions spaced 200 mils horizontally
;Returns with entrance coordinates and variable named Transmtce for
;  Transmittance

.386
IDEAL
MODEL small
stack 128

DATASEG

Transmtce  dw   ?
XF1      dw     ?
YF1      dw     ?
Include "Scan_var"


CODESEG
Include "Scan_Pcd"
proc StrgRead
        mov dx,30Ah             ;set Focus
        mov ax,[Focus]
        out dx,ax

;        mov bx,65000                ;settling time for focus
;wtla:    dec bx
;        jnz wtla

        mov ax,[XF]             ;save entrance coords
        mov [XF1],ax
        mov ax,[YF]
        mov [YF1],ax

        mov dx,304h             ;set first read position
        mov ax,[XF]
        out dx,ax
        mov dx,306h
        mov ax,[YF]
        out dx,ax

        mov dx,318h             ;first read
        in ax,dx

        mov [Transmtce],0       ;Transmtce begins as a sum, then divided by 4
                                ;  for output value
        mov cx,4
Read:   cmp cx,1
        je @@1

        mov dx,304h
        mov ax,[XF]
        add ax,200
        mov [XF],ax
        out dx,ax
```

```
@@1:    mov dx,318h
wtl:    in ax,dx
        test ax,8000h
        jne wtl
        and ax,1023
;------------------------------------------------------------------------
        add [Transmtce],ax      ;when loop is left, sum is 4x typical signal
        loop Read               ;  at specified location
        shr [Transmtce],2       ;final transmittance signal
;------------------------------------------------------------------------

        mov ax,[XF1]            ;restore but not issue entrance coords
        mov [XF],ax
        mov ax,[YF1]
        mov [YF],ax
ret
Endp strgread
End
```

Appendix C-9 SPOTREAD.ASM Subroutine procedure

This procedure is assembled with the command TASM SPOTREAD, automatically including it in the linking file PCSCAN0. This procedure serves as part of the logic to detect card jams.

```
;---------------------------------------------------------------------------
;SPOTREAD.ASM Procedure to measure transmittance in specified location
;Output is transmittance signal from one read with defocused spot
;Returns with entrance coordinates and variable SpotTransm
;A high signal amplitude at its location below column 1 sets CardJam=1

.386
IDEAL
MODEL small
stack 128

DATASEG

ThrldJam dw     200
SpotTransm dw   ?
XF1     dw      ?
YF1     dw      ?
INCLUDE "Scan_var"

CODESEG
Include "Scan_Pcd"
proc SpotRead
        mov dx,30Ah             ;set Focus
        mov ax,[Focus]
        out dx,ax
        mov [CardJam],0         ;reset CardJam

       ; mov ax,[XF]             ;save entrance coords
       ; mov [XF1],ax
       ; mov ax,[YF]
       ; mov [YF1],ax

        mov dx,304h             ;set read position
        mov ax,[XF]
        out dx,ax
        mov dx,306h
        mov ax,[YF]
        out dx,ax

        mov dx,318h             ;read
        in ax,dx

@@1:    mov dx,318h
wt1:    in ax,dx
        test ax,8000h
        jne wt1
        and ax,1023

        mov [SpotTransm],ax
        cmp ax,[ThrldJam]       ;
        jl @@2
        mov [CardJam],1         ;Card jam sets CJ=1
@@2:
       ; mov ax,[XF1]            ;restore but not issue entrance coords
       ; mov [XF],ax
       ; mov ax,[YF1]
       ; mov [YF],ax
ret
Endp SpotRead
End
```

## Appendix C-10 SCALREAD.ASM Subroutine procedure

This procedure is assembled with the command TASM SCALREAD, automatically including it in PCSCAN0. A description of its action is given in Section 3.3.11.11 in the text.

```
;-----------------------------------------------------------------------------
;SCALREAD.ASM Procedure to measure transmittance in Scale Zone
;Output is mean transmittance signal from four reads with defocused spot
;Reads 8 positions spaced 500 mils vertically
;Returns with entrance coordinates and the variable named ScaleTransm

.386
IDEAL
MODEL small

DATASEG
;ScalTransm  dw   ?
XF1      dw      ?
YF1      dw      ?
Include "Scan_var"

CODESEG
Include "Scan_Pcd"
proc ScalRead
        mov dx,30Ah               ;set Focus
        mov ax,[Focus]
        out dx,ax

        mov ax,[XF]               ;save entrance coords
        mov [XF1],ax
        mov ax,[YF]
        mov [YF1],ax

        mov dx,304h               ;set first read position
        mov ax,[XF]
        out dx,ax
        mov dx,306h
        mov ax,[YF]
        out dx,ax

        mov dx,318h               ;first read
        in ax,dx

        mov [ScalTransm],0        ;ScalTransm begins as a sum then divided by 8
                                  ;  for output value
        mov cx,8
Read:   cmp cx,1
        je @@1

        mov dx,306h
        mov ax,[YF]
        add ax,250
        mov [YF],ax
        out dx,ax

@@1:    mov dx,318h
wt1:    in ax,dx
        test ax,8000h
        jne wt1
        and ax,1023
;---------------------------------------------------------------------------
        add [ScalTransm],ax       ;when loop is exited, sum is 8x typical signal
        loop Read                 ;  at specified location
        shr [ScalTransm],3        ;final Scale transmittance signal, mean of 8
                                  ;  reads
```

```
;         cmp [ScalTransm],300      ;apply range limits 300 to 900
;         jg @@10
;         mov [ScalTransm],300
;@@10:     cmp [ScalTransm],900
;         jl @@11
;         mov [ScalTransm],900
@@11:
;-----------------------------------------------------------------------------
        mov ax,[XF1]              ;restore but not issue entrance coords
        mov [XF],ax
        mov ax,[YF1]
        mov [YF],ax

        ret

Endp scalread
End
```

# Appendix D-1 TLR.ASM Subroutine procedure

This procedure is assembled with the command ASM TLR. A description of its actions is given in Section 3.5.

```
;-----------------------------------------------------------------------------
;TLR.ASM, Program to read the file PARAM after end of PCSCAN

.386
IDEAL
MODEL small
Stack 128

DATASEG
filename db     'PARAM',0              ;;file size is 100 bytes
ParamData dw   10 dup (32)             ;;first 10 words are decimal values
ParamText db   80 dup (32)             ;;remainder is text from Entry
downline1 db    13,10,0
dash2line db    13,10,78 dup (61),0

HdrTitle  db 13,10,   '                        FOSDIC SCAN TRAILER DISPLAY
                 ',0
Label1    db 13,10,10,'|   FILENAME,  |    DATE     |  START  |   END   |
TOTAL  | FAIL  | CNDTN |'
Label2    db 13,10,   '|   DECK■REEL  |  SCANNED    |  TIME   |  TIME   |
CARDS  | CARDS | CODES |'
Label3    db 13,10,   '|              |             |         |         |
   |       |       |'

DataLine  db 13,10,   '|              |   - -       |  :  :   |  :  :   |
   |       |       |',13
room      db    20 dup (32)

Buffer  db      2,3 dup (32)
CndtnCode1 db   'First Condition Code... ',0
CndtnCode2 db   13,10,'Second Condition Code... ',0
space   db      0,0,0,0
prtbuf  db      20 dup (?)

CODESEG
INCLUDE "Scan_Pcd"

start:
        mov ax,@data
        mov ds,ax
        mov es,ax

        mov ah,03Dh                     ;open file PARAM
        mov al,0
        mov dx,offset filename
        int 21h

        mov ah,3Fh                      ;read contents of PARAM
        mov bx,5                        ;  to ParamData1, passing through
        mov cx,100                      ;  ParamText1
        mov dx,offset ParamData         ;current file size is 100 bytes
        int 21h
;---------------------------------------------------------------------------
        mov cx,1
        mov di,offset CndtnCode1
        call strwrite
        mov si,73
        call Get_Key_Msg
        mov di,offset CndtnCode2
        call strwrite
```

```
        mov si,75
        call Get_Key_Msg
;----------------------------------------------------------------------
        mov cx,1
        mov di,offset dash2line
        call strwrite
        mov di,offset downline1
        call strwrite

        mov cx,1                            ;display top line of boundaries
        mov di,offset Hdrtitle
        call strwrite

        mov cx,80                           ;display upper line of headings
        mov si,0
        mov ah,2
@30:    mov dl,[Label1+si]
        int 21h
        inc si
        loop @30

        mov cx,80                           ;display lower line of headings
        mov si,0
        mov ah,2
@30a:   mov dl,[Label2+si]
        int 21h
        inc si
        loop @30a

        mov cx,80                           ;display row with boundary extenders
        mov si,0
        mov ah,2
@30b:   mov dl,[Label3+si]
        int 21h
        inc si
        loop @30b

        mov cx,80                           ;display row with boundary extenders
        mov si,0
        mov ah,2
@30c:   mov dl,[Label3+si]
        int 21h
        inc si
        loop @30c

        mov cx,80                           ;OR ParamText1 and DataLine+2
        mov si,0                            ;(DataLine begins with cr,lf)
@32:    cmp [ParamText+si],32
        jne @31
        cmp [DataLine+si+2],32              ;here if ParamText1 is 32
        je @31                              ;if DataLine is also 32, jump to out
32
        mov al,[DataLine+si+2]              ;here if DataLine has something other
        mov [ParamText+si],al              ;
@31:    inc si
        loop @32

        mov cx,80                           ;display DataLine combined with
        mov si,0                            ;  ParamText
        mov ah,2
@30d:   mov dl,[Paramtext+si]
        int 21h
        inc si
        loop @30d
```

```
        mov ah,2                          ;insert lower left boundary
        mov dl,13
        int 21h
        mov dl,179
        int 21h
        mov cx,1
        mov di,offset downline1
        call strwrite
        mov di,offset dash2line
        call strwrite
        mov di,offset downline1
        call strwrite

        mov ah,4Ch
        int 21h

;==========================================================================
Get_Key_Msg:            ;Routine for processing keyed data

        mov ah,0Ah                ;;Load keyed data
        mov dx,offset buffer      ;;Buffer specifies size of buffer, max char+1
        int 21h                   ;;Buffer+1 carries number of char typed
        mov al,[Buffer+2]         ;;Buffer+2 is typed data field,
                                  ;;  left-justified
        cmp al,13                 ;jump around G-K-M if no typed char
        jne @3
        jmp @Out                  ;;this is shortcut if no typed char
@3:
        mov [DataLine+si],al           ;load typed char into DataLine msg
        ;mov [ParamText+bx+di],al      ;load typed char into file string
@Out:
        ret                            ;data field is left-justified

END start
```

Appendix E-1 SCAN_VAR List of global variables included in all programs and subroutines of LDR and PCSCAN

```
;SCAN_VAR  A list of variables used with LDR and PCSCAN

global  XC:word, YC:word         ;Working Coarse coordinate values
global  XF:word, YF:word         ;Same for Fine
global  XC0:word, YC0:word       ;Scan Origin in scan aperture, Coarse coordinates
global  XF0:word, YF0:word       ;Same for Fine coords
global  YCMargin:word            ;Vertical position for finding Card Bar
global  Port:word                ;Axis indicator - 304h=XF, 306h=YF
global  Step:word                ;Increment in search strokes
global  MaxCnt:word              ;Number of reading positions allowed for strokes
global  Focus:word               ;Focus count value
global  Card:word                ;Card number of 8 in frame
global  CardExam:word            ;Card no. selected for printout
global  Gain0:word               ;Default initial value
global  Gain1:word               ;Final Gain DAC input for MaxSig=980
global  MaxSig:word              ;Determined highest signal
global  MinSig:word              ;Determined lowest signal
global  CardSig:word             ;Determined card stock signal
global  Thrld:word               ;Threshold value
global  BandCenter:word          ;Center of left edge band
global  CardBar:word             ;Determined center of target line in Centroid, SR
                                 ;  expressed in XF
global  Cal_Data:word:5          ;part of data conveyed by file PARAM
global  Center:word              ;Output data from Centroid SR in XF or YF
                                 ;  in XF or YF according to axis of stroke
global  Edge:word                ;Output data from Edges SR in XF or YF
global  LeftEdge:word            ;Determined position of left edge in Centroid SR in YF
global  Corner1:word:2           ;XF and YF at lower left of card image
global  Corner2:word:2           ;XF and YF at lower right of card image
global  Corner3:word:2           ;XF and YF at upper left of card image
global  Corner4:word:2           ;XF and YF at lower right of card image
;global  DirtSpot:word           ;Count of corner-restores and sigs in blank cols
global  NextCard:word
global  XFCornerDiff:word        ;Non-rectangularity, XF size difference, cols 1 & 80
global  YFCornerDiff:word        ;Non-rectangularity, YF size difference,
global  NoCard:word              ;Card absence indicator

global  YFQuot:word
global  YFRemdr:word
global  YFRemdrSum:word
global  Slant:word
global  XFQuot:word
global  XFRemdr:word
global  XFRemdrSum:word
```

```
global   RowTable:word:26
global   xx:byte
global   ParamData:word:10
global   ParamText:byte:86

global   Blob:word                 ;Failure indicator for blob removal, fail=1
;global  CardinError:word          ;flag showing card failed geometry check
global   Report:word:4             ;vehicle for displaying parameters
global   Status:word               ;Working status word
global   Tilt0:word                ;Initial Tilt, scale 0 to 1023
global   Thrld:word                ;Threshold
global   Flag:word
global   FlagEOF:word              ;End of Film=1
global   FlagRep:word              ;Flag for repeating mode,=1 for repeat
global   junk:word:100             ;Notepad
global   PrtTbl:word:100           ;Vehicle for printout

global   NmbrCdsScnd:word          ;Total number of cards scanned
global   NmbrCdsFld:word           ;Total number of cards that failed
global   NmbrCdsRescnd:word        ;Total number of cards rescanned
global   NmbrColDrops:word         ;Total number of column with drops in Must-Haves
global   NmbrColPickups:word       ;Total number of columns with pickups in No-Punches
global   NmbrCdsDrops:word         ;Total number of cards with drops in Must-Have cols
global   NmbrCdsPickups:word       ;Total number of cards with pickups in No-Punch cols
global   PrevColDrops:word         ;tempo storage for NmbrColDrops
global   PrevCdsDrops:word         ;tempo storage for NmbrCdsDrops
global   PrevColPickups:word       ;tempo storage for NmbrColPickups

global   XCFrameStop:word          ;Leftside location for stopping film drive
global   XCFrame:word              ;XC for start of Frame routine
global   YCFrame:word              ;YC for start of Frame routine
global   ParamData1:word:10
global   ParamText1:byte:80
global   BlockPtr:word
global   ParamHandle:word
global   DataHandle:word
global   DataBlock:byte:52000
global   Maskbuf:byte:160
global   CardError:word
global   XCStopped:word
global   FailTrack:word            ;FT=1 indicates clear area after Frame, not card
global   FailEdge:word             ;runout Failure indicator for Edges proc
global   FailEdge1:word            ;hi-sig Failure indicator for Edges proc
global   FailBar:word              ;Failure indicator for Centroid proc
global   FailBar1:word             ;high first sig failure in Centroid proc
global   FailCalCardBar:word       ;stored calib failure indicator for Card Bar
global   FailCardBar1:word         ;stored frame failure indicator for Card Bar
```

```
global  FailCalEdge:word          ;stored calib failure indicator for Edges
global  FailCalScaleBar:word      ;stored calib failure indicator for Scale Bar
global  FailCalGeom:word

global  Cease:word                ;for emergency stopping and display
global  SigSum:dword
global  NmbrSigs:dword
global  ThrldMod:word
global  Want:word
global  Transmtce:word
global  FrmSig:word
global  PunchCols:word
global  StopOnArti:word
global  ScalTransm:word
global  MeanBlank:word
global  ThrldFrame:word
global  CardJam:word
global  SpotTransm:word
global  ThrldJam1:word
global  ThrldJam:word
global  ResumeScan:word
global  Pause:word
global  FailCardinFrame:word
global  AftrFirstFrm:word
global  Tilt:word
global  Pickup:word
```

## Appendix F-1 SCAN_PCD List of global procedures included in all programs and subroutines of LDR, PCSCAN and TLR

```
;SCAN_PCD  A list of procedures used with LDR and PCSCAN

global  Scan:proc          ;central program procedure for handling scan subroutines
global  Centroid:proc      ;Finds center of target line
global  Edges:proc         ;Finds position of edge, opaque to clear transition
global  CalcScal:proc      ;Generates scaling factors for Width, Slant and Height
global  ColScan:proc       ;Scans all 960 punch positions in 8 card images
global  Gainpc1:proc       ;Calibrates Gain, sets MaxSig near 980
global  VTrack:proc        ;Finds YCMargin
global  Spacex:proc        ;Horizontal space    Located in Util.asm
global  Line:proc          ;LF and CR                     "
global  Prtax:proc         ;Prints contents of ax         "
global  Print0:proc        ;Prints messages at end of scanning
global  Frame:proc         ;Film drive
global  Strwrite:proc      ;in MTA
global  Bintoascdec:proc   ;in MTA
global  FrmScan:proc       ;Scan pattern and film drive for leader
global  Gain:proc          ;calibrates Gain DAC and MaxSig
global  Bkgnd:proc         ;calibrates CardSig and MinSig
global  VTrack:proc        ;calibrates YCMargin
global  Cal_Dspl:proc      ;displays calibration parameters
global  Entry:proc         ;enters and displays keyboard data
global  Framex:proc
global  BlobRemv:proc      ;attempts to remove blobs
global  Wipe:proc
global  Movon:proc
global  colscanr:proc
global  scanr:proc
global  strgread:proc
global  scalread:proc
global  spotread:proc
```

## Appendix G-1 Linking files LDR0, PCSCAN0, MTA

LDR0 and PCSCAN0 are files listing the subroutines used in the programs LDR.EXE and PCSCAN.EXE respectively. Linking is accomplished with these instructions:

```
tlink @LDR0  generates LDR.EXE
tlink @PCSCAN0    generates PCSCAN.EXE
```

Listings:

---

**LDR0:** LDR FRAME WIPE GAIN BKGND CAL_DSPL ENTRY UTIL ,,, MTA

---

where:

| | |
|---|---|
| LDR | Calls for leader film movement |
| FRAME | Runs film in to Reel ID (first) card |
| WIPE | Activates Crt phosphor screen |
| GAIN | Adjusts sensitivity of optical system |
| BKGND | Measures density of card images |
| CAL_DSPL | Displays calibration parameters |
| ENTRY | Calls for deck and reel identification via keyboard |
| UTIL | Facilitates displays |
| MTA | Library routines for displays, described below |

---

**PCSCAN0:** PCSCAN SCAN GAIN VTRACK CENTROID EDGES STRGREAD SPOTREAD CALCSCAL COLSCAN SCALREAD FRAME UTIL ,,, MTA

---

| | |
|---|---|
| PCSCAN | Executive program for scanning |
| SCAN | Calls subroutines for scanning |
| GAIN | Adjusts optical sensitivity for card images |
| VTRACK | Finds transverse location of card images on film |
| CENTROID | Finds centers of Card and Scale bars |
| EDGES | Finds edges of card images |
| STRGREAD | Measures average film transmittance at designated zones in card images |
| SPOTREAD | Measures transmittance at designated positions in card images |
| CALCSCAL | Calculates movement scale factors from card corner locations |
| COLSCAN | Controls scanned positions at 960 hole locations on card images |
| SCALREAD | Finds average transmittance in scale zone |
| FRAME | Controls film movement |
| UTIL | Facilitates displays |
| MTA | Library routines for displays, described below |

**MTA:**

This is a library of routines to facilitate the display of numeric values in a register and strings of text from sections of memory. From the collection in the library, BINTOASCDEC and STRWRITE have been utilized extensively. The source for MTA-LIB is "Mastering Turbo Assembler" by Tom Swann, pages 208 and 182 respectively, published 1989 by Hayden Books.

## Appendix H-1 TRNSTBL1 Translation table

This table is referred to in COLSCAN.ASM to translate Fosdic row data identification to final recorded characters.

---

```
;TRNSTBL1 An INCLUDE file for the output data translation table
;
;              0         10        20        30        40        50
;              |         |         |         |         |         |
TrnsTbl1 db   ' &-0123456789*****!)ABCDEFGHI******(JKLMNOPQR*****'
Ext1     db   '***STUVWXYZ***************************************'
Ext2     db   '**************************************************'
Ext3     db   '**************************************************'
Ext4     db   '*************************************!**+^<>$%=[]\'
Ext5     db   '*****!'
```

## Appendix J-1 GO.BAT Executive batch file

This small batch program enables the sequence of programs comprising the overall scanning system.

```
--------------------------------------------------------------------------------
                    ;Contents of GO.BAT

smartdrv c-/q             ;disables SMARTDRV delays scanning of new data until
                          ;  old is recorded on hard disk
ldr                       ;starts LDR.EXE runs in film leader and acquires
                          ;  keyed reel identification
pcscan                    ;starts PCSCAN.EXE scans entire reel
tlr                       ;starts TLR.EXE adds timing and quality control data
foscopy c:                ;starts FOSCOPY combines card data and header to
                          ;  create final file for reel
```

# Appendix J-2 List of .MSK Mask files

Mask files are called by PCSCAN.ASM to convey information regarding selection of numeric or alphanumeric/overpunching in specific columns, as obtained from Reference Manuals for individual decks. The action of these files is described in Section 3.3.11.7.

---

```
        The contents of sample file 00000110.MSK are as listed below:

22222222222222222202222222222222222022000222022022222222222222220222222222222222
```

Notes:

Columns with "2" indicate expected single punch i.e. numeric, Y or X
  After 2 double punches these convert to "0" for full alphanumeric/overpunch encoding.

Columns with "0" indicate expected possible double or triple punches, alphanumeric/overpunch.

The above example (Deck 110) shows expected possible multiple punches in columns 19 36 39 40 41 45 48 64.

---

The contents of all DDDDDDDD.MSK files are as listed below. Except as noted, columns not listed are filled with "2".

| Deck | Columns with "0" for alphanumeric/ overpunch data |
|---|---|
| 110 | 19 36 39-41 45 48 64 |
| 116 | 1 25 27 37 50 57 60 65 |
| 117 | 36 37 39-41 45 47 51 56 62 68 |
| 118 | 29 32 34 41 44 58 |
| 119 | 59 64 69 |
| 128 | 2 12 18 20 22 23 32 39 43 45 62-65 70-74 77 |
| 184 | 1 33 34 36 37 46 52 63 |
| 185 | 57 |
| 189 | 20 32 35 43 46 50-52 59-61 70 71 |
| 190 | 30 |
| 192 | 3 5 10 14-18 27 28 53 58 60 61 |
| 193 | 3 10 13 30 56 57 |
| 194 | 10 |
| 195 | 3 6 8 18 39 42 45 48 |
| 196 | 3 6 8 18 39 42 45 48 |
| 256 | 18 46 |
| 281 | 1 39 50 56 58 65 |
| 541 | none |
| 547 | reel not recvd |
| 608 | 24 40 56 72 |
| 650 | 38 69 |
| 690 | 15 18 32 35 49 52 66 69 |

Appendix K-1 HEADER.TXT Header file

This is the Header record attached to all card data files by FOSCOPY after the end of scanning on a reel. It contains identification information and data on the results of the processing for the specific run. The encoding table given in this record has since been augmented with additional codes to handle overpunching with numerics; the most complete table is located in Appendix A.

```
--------------------------------------------------------------------------------
Byte offset
0-1     Offset from start of file to start of data   2-byte unsigned integer
2-10    Deck number <nos.,letters,()>     8 ASCII chars plus NULL terminator
11-14   Reel number                       3 ASCII digits plus NULL terminator
15-25   Date scanned <mm-dd-yyyy>        10 ASCII digits plus NULL terminator
26-27   Number of card images in file                2-byte unsigned integer
28-29   Number of cards failing scan validity check  2-byte unsigned integer
30      Condition Code 1                             1-byte unsigned character
31      Condition Code 2                             1-byte unsigned character
32-33   Integer constant '42', byte-order indicator  2-byte unsigned integer
34-38   Spares,initially space characters
39      NULL, end of parameter section of header

All integers are in low-byte, high-byte format.
80 characters per card, fixed-length records, no delimiters

Character translation:

Row       ASCII
Punched   Char
-------   -----
none      space
 12         &
 11         -
  0         0
  1         1
  2         2
  3         3
  4         4
  5         5
  6         6
  7         7
  8         8
  9         9
 12,1       A
 12,2       B
 12,3       C
 12,4       D
 12,5       E
 12,6       F
 12,7       G
 12,8       H
 12,9       I
 11,1       J
 11,2       K
 11,3       L
 11,4       M
 11,5       N
 11,6       O
 11,7       P
 11,8       Q
 11,9       R
```

| | |
|---|---|
| 0,2 | S |
| 0,3 | T |
| 0,4 | U |
| 0,5 | V |
| 0,6 | W |
| 0,7 | X |
| 0,8 | Y |
| 0,9 | Z |

Note: Any combination not listed above translates to an "*" character.

Prepared by Greenough Data Associates, Rockville, MD under P.O. A4352 from Orkand Coporation, Asheville, NC under contract to National Climatic Data Center, Department of Commerce, Asheville, NC.

Data follows after CR-LF

------------------------------------------------------------------------------