

AD-A098 658

BOEING AEROSPACE CO SEATTLE WA

F/G 13/8

LOW COST HIGH VOLUME RADIOGRAPHIC INSPECTION.(U)

JAN 81 N M LOWRY, H J ABPLANALP, J M TANKE

DAAK40-78-C-0197

UNCLASSIFIED

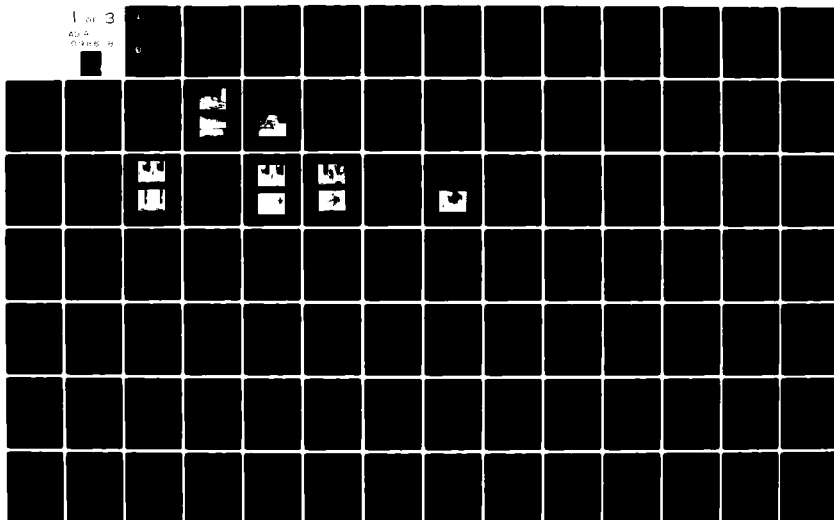
0180-26159-1

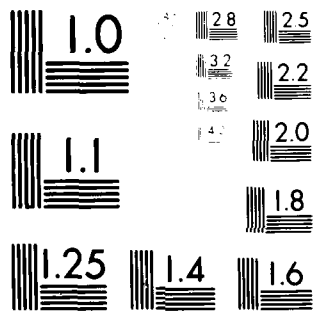
DRSMI/RS-CR-81-1

NL

1 of 3

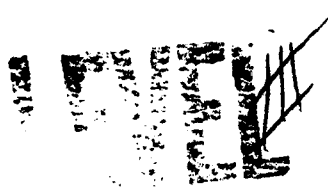
AD-A
0180-26159-1





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A098658



2
BS

AD-E950114

TECHNICAL REPORT TR-RS-CR-81-1

**LOW COST HIGH VOLUME
RADIOGRAPHIC INSPECTION**

**SYSTEMS ENGINEERING DIRECTORATE
MANUFACTURING TECHNOLOGY DIVISION
US ARMY MISSILE LABORATORY**

JANUARY, 1981



U.S. ARMY MISSILE COMMAND

Redstone Arsenal, Alabama 35809

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

DTIC FILE COPY

**DTIC
ELECTE
S MAY 8 1981 D**

A

* 1021, 1 JUL 79 PREVIOUS EDITION IS OBSOLETE

81 3 30 090

DISPOSITION INSTRUCTIONS

**DESTROY THIS REPORT WHEN IT IS NO LONGER NEEDED. DO NOT
RETURN IT TO THE ORIGINATOR**

DISCLAIMER

**THE FINDINGS IN THIS REPORT ARE NOT TO BE CONSTRUED AS AN
OFFICIAL DEPARTMENT OF THE ARMY POSITION UNLESS SO DESIGNATED
BY OTHER AUTHORIZED DOCUMENTS.**

TRADE NAMES

**USE OF TRADE NAMES OR MANUFACTURERS IN THIS REPORT DOES
NOT CONSTITUTE AN OFFICIAL INDORSEMENT OR APPROVAL OF
THE USE OF SUCH COMMERCIAL HARDWARE OR SOFTWARE.**

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TECHNICAL REPORT TR-RS-CR-81-1	2. GOVT ACCESSION NO. AD-A098 658	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) LOW COST, HIGH VOLUME RADIOGRAPHIC INSPECTION		5. TYPE OF REPORT & PERIOD COVERED Technical Report
7. AUTHOR(s) Ned M. Lowry Helen J. Abplanalp Jon M. Tanke		6. PERFORMING ORG. REPORT NUMBER D180-26159-1
9. PERFORMING ORGANIZATION NAME AND ADDRESS Quality Assurance Technology Boeing Aerospace Company P.O. Box 3999, M.S. 2P-24 Seattle, WA 98124		8. CONTRACT OR GRANT NUMBER(s) DAAK40-78-C-0197 3783454
11. CONTROLLING OFFICE NAME AND ADDRESS Commander, U.S. Army Missile Command Attn: DRSMI-RST		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS MM&T Project
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE January, 1981
		13. NUMBER OF PAGES 237
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release, distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Non-Film Radiography Remote Part Manipulation Real Time Radiography Digital Image Processing Computer Aided Inspection		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This document describes a prototype non-film radiographic inspection system designed to perform rapid assembly configuration verification of large complex assemblies. The system includes capabilities for: (1) real time X-ray radiography, (2) digital image enhancement, (3) remote part positioning and manipulation and (4) computer aided inspection. The advantages and limitations of each of these capabilities are discussed. A cost/benefit analysis is also included.		

TABLE OF CONTENTS

	Accession For	
	NTIS GRA&I	
	DTIC TAB	
	Unannounced	
	Justification	
FOREWARD		iv
SUMMARY		1
1.0 INTRODUCTION		3
2.0 SCOPE OF WORK		5
3.0 EQUIPMENT DESCRIPTION		7
3.1 MECHANICAL EQUIPMENT		7
3.2 REAL TIME RADIOGRAPHIC EQUIPMENT		11
3.3 IMAGE PROCESSOR		11
3.4 COMPUTER SYSTEM		12
3.5 EQUIPMENT INTERFACING		16
4.0 RADIOGRAPHY		17
4.1 REAL TIME RADIOGRAPHY		17
4.1.1 THE ADVANTAGE OF MOTION		17
4.1.2 METHODS OF REDUCING SCATTERED RADIATION		18
4.1.3 VIEWING OF THE X-RAY IMAGE		18
4.2 IMAGE PROCESSING		20
4.2.1 NOISE REDUCTION		20
4.2.2 EDGE ENHANCEMENT		21
4.2.3 CONTRAST MANIPULATION		26
4.2.4 COLOR REPRESENTATION		26
4.2.5 SOFTWARE CONSIDERATIONS		27
4.3 RADIOGRAPHIC QUALITY		28
4.3.1 PARAMETERS AFFECTING RESOLUTION		29
4.3.2 MEASUREMENT OF RESOLUTION		31
4.4 RADIOGRAPHIC RECORDS		32

Accession For	
NTIS GRA&I	
DTIC TAB	
Unannounced	
Justification	
For	
Distribution	
Availability	
Dist	

A

TABLE OF CONTENTS (Continued)

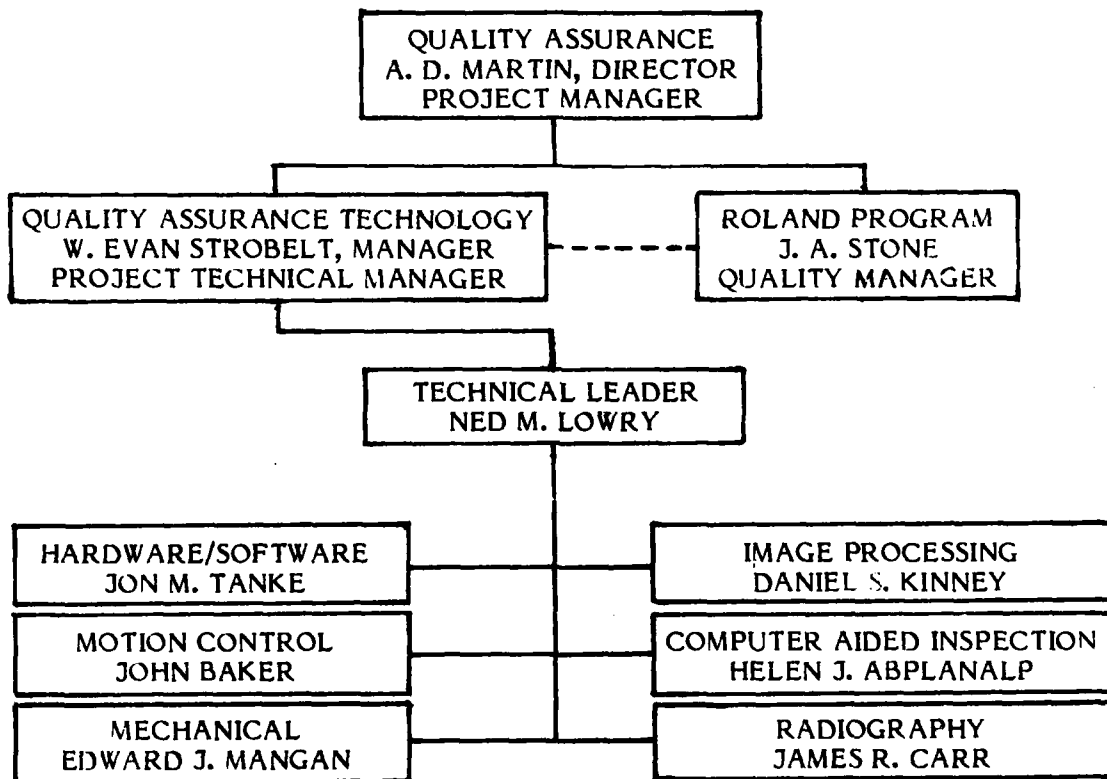
	Page
5.0 COMPUTER AIDED INSPECTION	34
5.1 CONCEPT	34
5.2 SOFTWARE DESIGN	34
5.2.1 INSPECTION	37
5.2.2 PLANNING	37
5.2.3 RETRIEVAL	40
5.3 APPLICATION TO ROLAND FINAL INSPECTION	40
6.0 COST-BENEFIT	42
7.0 CONCLUSIONS AND RECOMMENDATIONS	46
7.1 REAL TIME X-RAY	46
7.1.1 EQUIPMENT	46
7.1.2 APPLICATIONS	46
7.2 IMAGE PROCESSING	46
7.3 COMPUTER-AIDED INSPECTION	47
7.4 ACCOMPLISHMENTS AND RECOMMENDATIONS FOR FUTURE WORK	47
8.0 BIBLIOGRAPHY	48
APPENDICES	
1 STC M-70 I/O SCHEMATICS	A1-1
2 IMAGE PROCESSOR SOFTWARE, LOGIC AND CODE	A2-1
3 MOTOR CONTROL SCHEMATICS	A3-1
4 MOTOR CONTROL SOFTWARE, LOGIC AND CODE	A4-1
5 COMPUTER AIDED INSPECTION SOFTWARE, LOGIC AND CODE	A5-1
6 STC SYSTEM 500 SOFTWARE ADDITIONS, LOGIC AND CODE	A6-1
7 SUPPORTING SOFTWARE DESCRIPTION	A7-1

TABLE OF ILLUSTRATIONS

Figure	Title	Page
1.0-1	Real Time Radiography - Concept	4
2.0-1	Major Program Milestones	6
3.0-1	High Volume Radiographic Inspection System	8
3.1-1	Part Manipulation Table	9
3.1-2	Lead Shielding	9
3.1-3	X-Ray Source and Image Intensifier Configuration	10
3.3-1	Image Processing System Comparisons	13
3.4-1	Computer Comparisons	15
4.2.2-1	Difference Edge Enhancement	22
4.2.2-2	Gradient Edge Enhancement	22
4.2.2-3	Laplacian Edge Enhancement	24
4.2.3-1	Linear Mapping Image Enhancement	24
4.2.3-2	Piecewise Linear Mapping Image Enhancement	25
4.2.3-3	Color Level Slicing	25
4.2.4-1	Pseudocolor Image Enhancement	27
4.3.1-1	Measured Focal Spot Size for Phillips Electronic Instruments, Inc. Model MCN 321	30
4.3.1-2	Resolution Limitations of Prototype System Due to X-Ray Focal Spot, Image Intensifier, and Video Display	31
4.3.2-1	Penetrameter Sensitivity	32
5.2-1	Computer-Aided Inspection Data Base	35
5.2-2	Computer-Aided Inspection System	36
5.2.1-1	Inspection Subsystem	38
5.2.2-1	Planning Screen Formats	39
5.3-1	Roland Image Processing Sequence	41
6.0-1	Real-Time X-Ray System Capital Costs	43
6.0-2	Investment Analysis - Real Time X-Ray	44
6.0-3	Non-film Radiography with Computer-Aided Inspection	45

FOREWARD

This project was made possible through initial and continuing support of Boeing Aerospace Co. management and through the dedicated efforts of its engineers and technicians. Those persons who were directly involved in the project are shown below. There are also many others who also contributed invaluable to the success of the project.



PROJECT ORGANIZATION

SUMMARY

A prototype system has been assembled to establish the feasibility of performing assembly verification using advanced non-film radiographic techniques. The equipment has been configured to provide real time X-ray radiography, digital image enhancement, remote part positioning, and computer-aided inspection.

Real time X-ray techniques have been applied to an inert ROLAND missile which was assembled for this purpose. When viewing parts of relatively uniform thickness, an X-ray image intensifier coupled to a vidicon camera provides adequate sensitivity for configuration inspection applications. However, when a part has very thick areas next to very thin areas, the high energies required to inspect the thick areas allow sufficient energy levels in the thin areas to blind or damage some vidicon cameras. Applications of this nature require use of a different camera system, such as a fluorescent screen and an isocon camera.

Provisions for remote control of part motion are necessary to take full advantage of real time radiography. A prototype part manipulation table and control unit has been assembled to perform remote part positioning either manually or by a computer.

When higher resolution than provided by the real time system is required, the use of image processing is indicated to increase the image contrast and detail sharpness. A broad range of digital image processing techniques has been examined. Techniques found advantageous were video frame averaging, contrast modification, gray level slicing, Laplacian and gradient edge enhancement, and pseudocolor. Useful evaluation aids included split screen for comparison of two images and profile and histogram graphics. The evaluation has produced an optimized image processing sequence for each critical inspection area on the ROLAND missile.

A computer-aided inspection system has been developed to perform parts positioning, image enhancement sequences, and records storage and retrieval. The final acceptance decisions are made by the operator. Cost/benefit studies indicate that high volume manufacturing applications could benefit from computer-aided inspection procedures as well as image processing techniques.

Image storage techniques have been evaluated with respect to cost, quality, and ease of retrieval. High density digital magnetic tape is the preferred choice. However, for applications not requiring a computer for image processing or records management, video tape provides an adequate image storage medium.

As a result of this project it has been concluded that real time X-ray should be implemented into all inspection facilities performing radiographic assembly verification (e.g., ROLAND). In addition, it has been determined that development of systems with higher resolution and sensitivities would make possible wider application of this technique, resulting in the replacement of most film-based radiography.

1.0 INTRODUCTION

Many products, such as the ROLAND missile, must pass an X-ray inspection prior to release from the manufacturer. This is to assure that all internal parts are present and assembled correctly. Without verification of internal structure by radiographic inspection, a defective missile could be deployed. Such a missile could malfunction and potentially cause catastrophic failure.

Radiographic examination of critical low-rate-production components and assemblies traditionally has made use of film processing methodology. The use of film-based methods is prohibitive for high rate production due to the high labor and film costs and excessive flow time. The development of a non-film radiographic inspection method was clearly necessary to reduce the costs of assembly verification of high volume products.

Fluoroscopy offers the best potential to replace conventional film radiography. Applications of real time systems have been limited because the X-ray image is generally of lower quality than film. Low sensitivity has been a major drawback in applications requiring detection of very small flaws, but when examination requirements are primarily related to hardware configuration, spatial resolution is of lesser importance.

The purpose of the project has been to develop and demonstrate a prototype non-film radiographic method for verifying the configuration of large, complex parts. The ROLAND missile final assembly inspection was chosen as the target application for this project, although the techniques developed could be used for inspecting the internal configuration of any complex missile or part.

Recent improvements in fluoroscopic screens, electro-optical imaging devices and image processing technology have substantially reduced the gap between film and direct-viewing systems. The improved technology has made it feasible to develop a non-film system for examination of ROLAND hardware which will meet inspection requirements at a significantly lower cost, ultimately resulting in greater reliability due to the capability for more complete inspection coverage.

A real time fluoroscopic X-ray inspection system has been developed to meet the goals of reduced cost and flow time. The concept of real time X-ray is shown in Figure 1.0-1. The X-rays which pass through the test part fall on a fluoroscopic screen instead of a photosensitive X-ray film. The fluoroscopic screen converts the X-rays to visible light, and a camera transmits the image to a remote monitor. Remote control of part motion allows the operator to inspect the part at every angle and position. The real time image may be digitized so that image enhancement techniques can be performed upon it. The digitized image may be saved on a magnetic tape or disc associated with a computer system. Automatic control of parts positioning and image processing allows a computer-aided inspection scheme to be used. With this system, techniques have been developed that could be applied to final assembly verification of the ROLAND missile.

It is estimated that a cost reduction of 8:1 and flow time reduction of 5:1 could be achieved through the implementation of non-film methods. For the ROLAND missile this would result in a savings of over \$1.1 million based on a production of 10,000 units.

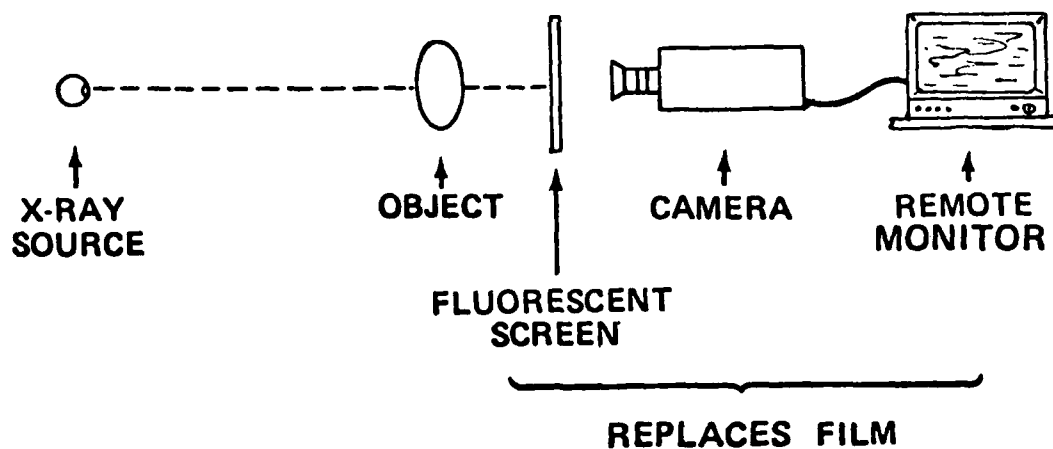


FIGURE 1.0.1 REAL TIME RADIOGRAPHY - CONCEPT

2.0 SCOPE OF WORK

The scope of this project covers the evaluation of three areas: real time X-ray, image processing, and computer-aided inspection. The real time X-ray system includes a fluoroscopic image intensifier tube, an antimony trisulfide vidicon camera, a TV monitor, and a remotely-controlled part manipulation table. Digital image processing was executed under control of a computer system, and was performed on digitized images. The computer-aided inspection system controls routine procedures, such as part positioning, image processing sequences, and storage and retrieval of records and images. Computerized decision-making was not investigated and did not fall within the scope of this project.

The prototype inspection system was applied to assembly verification of the ROLAND missile propulsion unit. Since this inspection was the target application for the project, care was taken to assure that all ROLAND radiographic inspection requirements could be met.

The major program milestones and their interrelationships are diagrammed in Figure 2.0-1. After the real time imaging system was assembled, part manipulation and image processing systems were added. A prototype computer-aided evaluating inspection system was then developed. Optimization of the real-time images of image processing techniques, and determination of the best testing procedure for the computer-aided inspection were accomplished. It was then possible to perform a cost/benefit analysis for these systems. An industry demonstration was given on September 17 and 18, 1980.

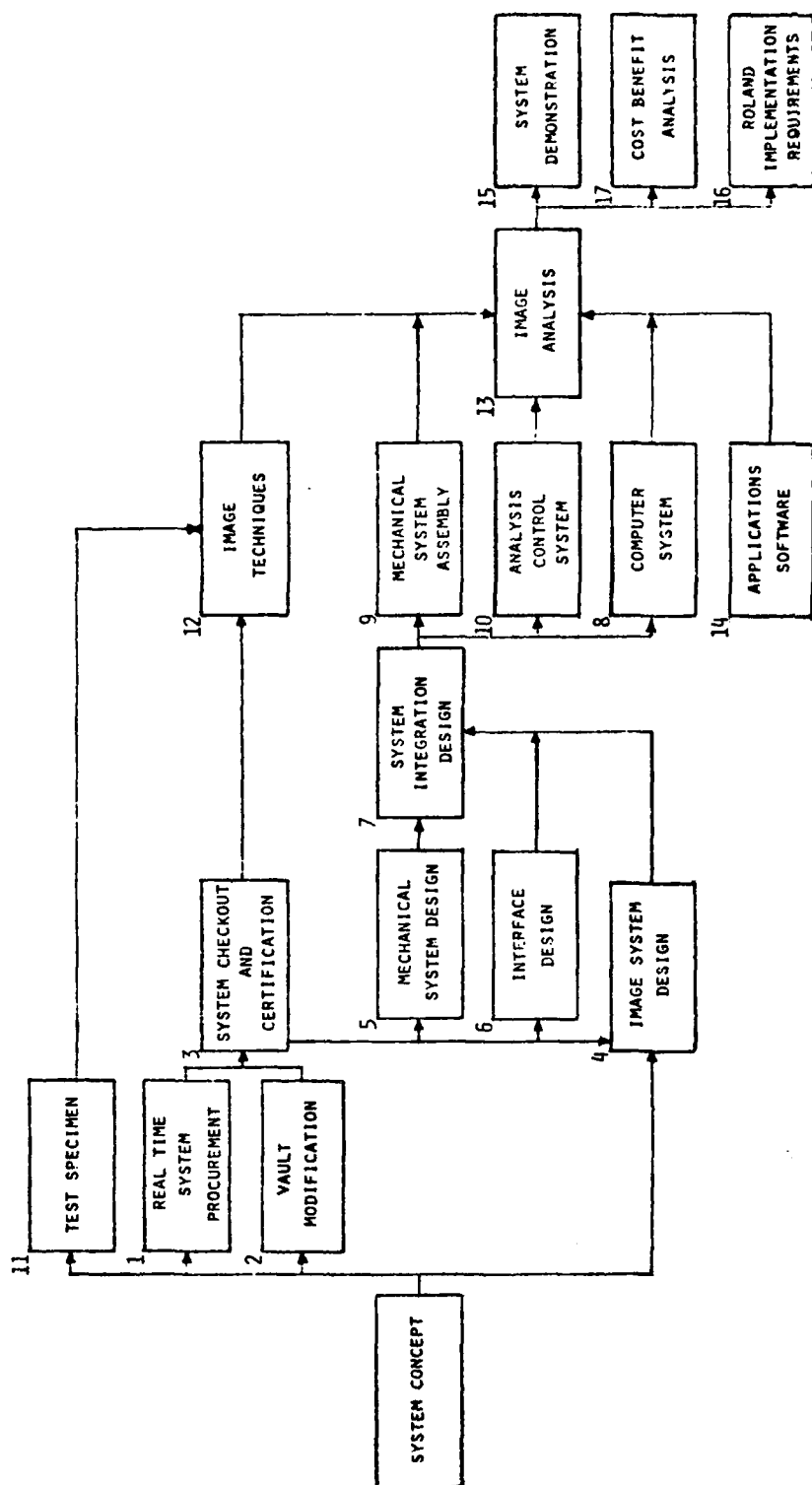


FIGURE 2.0-1 MAJOR PROGRAM MILESTONES

3.0 EQUIPMENT DESCRIPTION

To assure a high probability of success, the equipment selected was that which met the highest possible standards consistent with return on investment rationale of a 2 year payback. Figure 3.0-1 shows a block diagram of the overall system. The individual items are described below:

1. X-Ray Generator, Model MCN321, Philips Electronic Instruments, Inc.
2. X-Ray-to-Visible Light Converter and Intensifier, Dynascope Models 22D20R and 17D20, Machlett Laboratories, Inc.
3. Video Camera, Model LSV-1, Sierra Scientific Corp.
4. Video Monitor, Model SNA-24R, CONRAC Corp.
5. Image Processor, Model 70/E, Stanford Technology Corp.
6. Image Processor, Computer Interface, Boeing Aerospace Co.
7. Computer System-Model 550, Prime Computer Inc.
8. Parts Control-Computer Interface, Boeing Aerospace Co.
9. Parts Control, Boeing Aerospace Co.

A ROLAND missile booster and sustainer motor section was assembled with inert grains for use during the project. The inert grains were machined from solid polyvinyl chloride (PVC) rods to simulate actual grains in dimensional stability and optical density and uniformity. However, these inert grains were not coated to simulate an inhibitor as this would have been cost prohibitive and that requirement can be met when the system is implemented.

3.1 MECHANICAL EQUIPMENT

The real time X-ray mechanical equipment has been assembled to permit a complete examination of the ROLAND missile. The part manipulation table (Figure 3.1-1) supports the missile at each end. The table was constructed with three degrees of freedom: longitudinal, transverse, and rotational. The longitudinal and transverse movements permit a full scan of the missile over the total length and width while it is continuously rotated. A remotely controlled lead shield (Figure 3.1-2) permits selective examination of small areas and reduces scattered

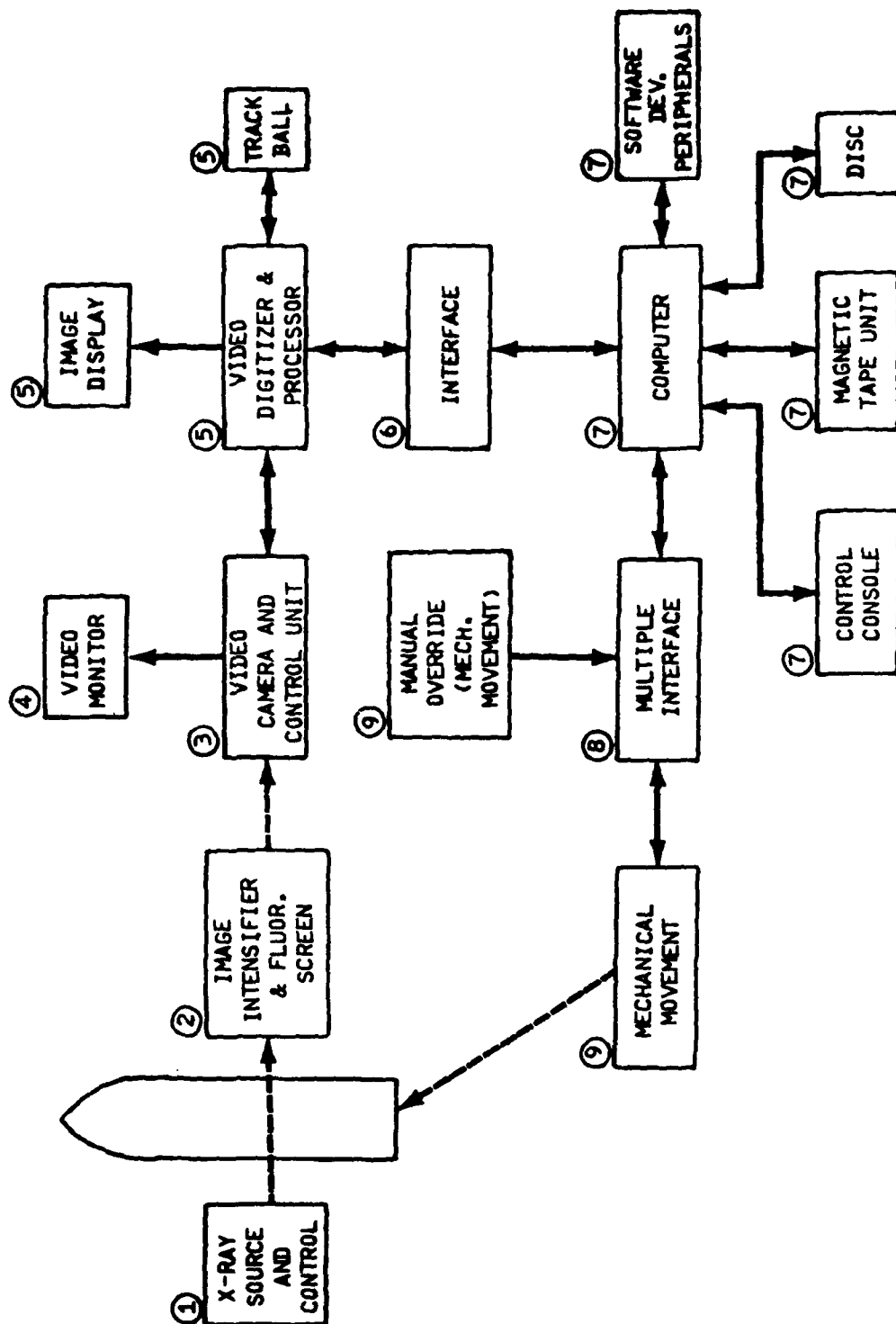


FIGURE 3.0-1 HIGH VOLUME RADIOGRAPHIC INSPECTION SYSTEM

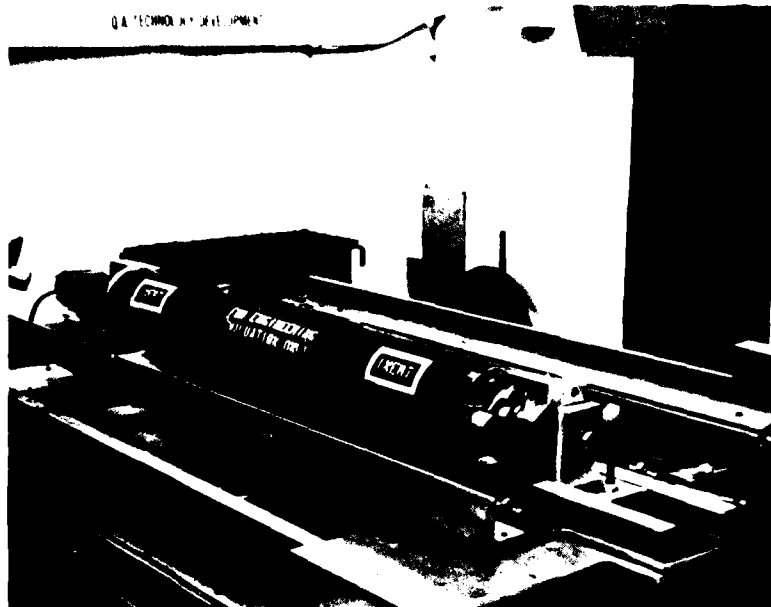


FIGURE 3.1-1 PART MANIPULATION TABLE



FIGURE 3.1-2 LEAD SHIELDING

X-rays, improving sensitivity. The X-ray tube height can be adjusted to achieve a source-to-fluorescent-screen distance of up to 210 cm (Figure 3.1-3). All movements are electrically controlled and can be remotely adjusted.

The prototype system enables the missile to be manipulated, allowing evaluation of all of the defined inspection locations. Motor drive systems were installed to allow automatic positioning of the missile simulating actual production testing operations. Radiography was performed vertically in the prototype system to comply with safety regulations for the particular facility. Modification of the mechanical system will be accomplished during the production implementation to accommodate horizontal radiography.

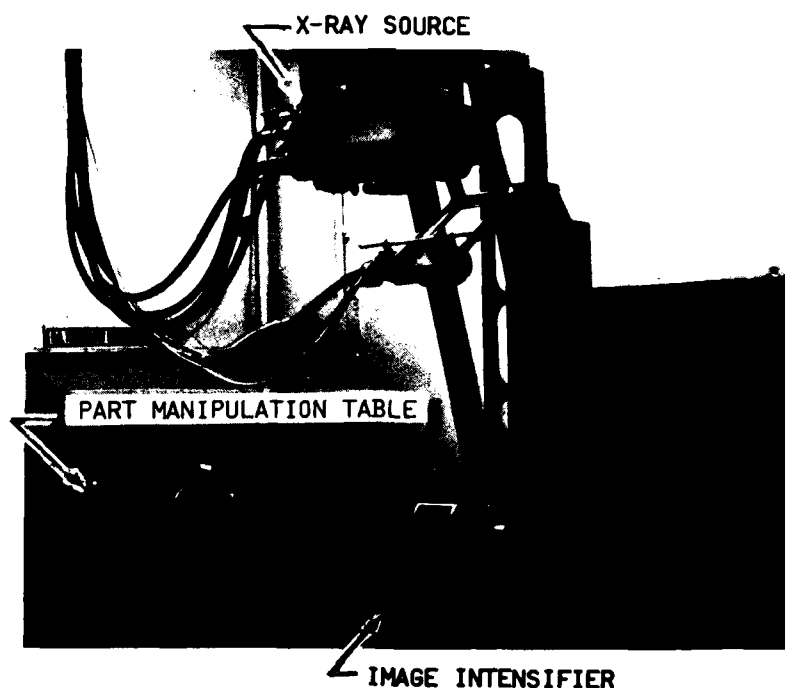


FIGURE 3.1-3 X-RAY SOURCE AND IMAGE INTENSIFIER CONFIGURATION

3.2 REAL TIME RADIOGRAPHIC EQUIPMENT

The equipment necessary for real time radiography, in addition to a source of X-rays, falls into two categories: (1) the conversion of X-ray light to visible light, and (2) the viewing of the converted radiation.

Fluorescent screens convert X-ray light to visible light. They are made from a variety of materials, and are generally designed for specific applications. X-ray image intensifiers contain a fluorescent material but amplify the light output by several thousand times. As a result of the intensification, however, the image contrast is significantly reduced, resulting in a lower dynamic range than that of fluorescent screens.

Direct viewing of a fluorescent screen or image intensifier is not practical because of the geometry, radiation hazard, and low light level. A video camera and remote monitor is a convenient system for viewing the visible image. There are two video tubes commonly used in this application: vidicon and isocon. The vidicon is an inexpensive tube but has low sensitivity and dynamic range and is blinded by high light levels. The much more expensive isocon tube has high sensitivity and a wide dynamic range. A Sierra Scientific Corp. Model LSV-1 video camera with a antimony trisulfide vidicon tube was used during this study because of availability and cost.

Two image intensifiers were utilized at different times, Machlett Laboratories, Inc. Models 22D20R and 17D20. The camera and intensifier were optically coupled by two KOWA fixed focus lenses, both 42 mm @ f/1.1.

3.3 IMAGE PROCESSOR

The minimum requirements established for this system are as follows:

1. Video input: Analog to digital converter, 8-bit resolution, 10 MHz sample rate minimum.
2. Video output: Digital to analog conversion controlled by table lookup and/or bit level manipulation.

3. Memory compatible with video in/out 512 x 512 minimum, 1024 x 1024 desirable. Bits per pixel must be compatible with functional requirements (12 bits if frame summing is provided, 8 bits if only averaging is provided).
4. Digital input/output: Provisions must be available for parallel input or output at rates greater than 500,000 bytes per second.
5. Programming capabilities: Unit must have capability of control of its function via an outside computer. Such functions must include as a minimum:
 - Summation of specified number of frames or as an alternative, averaging of a specified number of frames.
 - Inversion of memory or alternate subtraction of specified number of frames.
 - Overlay via computer control ASCII characters or symbols.
 - Desired option: Ability to manipulate sections of memory via the computer independent of the rest of memory, commonly termed: scroll, zoom, split screen, quartering, window transfer, psuedo color.

A summary of available systems and their acceptability is given in Figure 3.3-1. Two systems met the minimum requirements: the IP-5000 manufactured by DeAnza Systems, Inc., and the Model 70/E manufactured by Stanford Technology Corporation. The Model 70/E image processor was selected because of the following advantages: The input/output transfer rate was four times faster than the IP-5000. The time for conversion from video to digital with 8-bit resolution was four times faster in the Model 70/E. Processing speeds for additions, subtractions, multiplications, etc., were as much as 17 times faster in the Model 70/E. The Model 70/E has hardware histogram generation and hardware min-max pixel intensity calculation as opposed to software generation in the IP-5000. The software package and documentation also was better for the Model 70/E.

3.4 COMPUTER SYSTEM

The computer selection for this project was based on the ability to meet the specific hardware, execution speed, and software requirements to facilitate development and to execute implemented tasks.

VIDEO SYSTEMS	INPUT		MEMORY		PROCESSING (MEMORY)			PROCESSING (OUTPUT)	CPU INTERFACE	ALPHA NUMERIC	CURSOR/ JOYSTICK	PSEUDO COLOR	WINDOW TRANSFER	SOFTWARE CONVERSION	RESC COMMENTS
	DIGITAL IN	VIDEO IN	512 ^a	1024 ^a	AVG	SUN	SUB								
QUARTER CORP DS-20	•	• 512 LINES	•		•	•	•	CONTRAST ENHANCE BIT LEVEL	16 BIT. IEEE 488						
BAITEX CORP	•		•	•	•			PROGRAMMABLE LOOK UP TABLE	16 BIT	•	•	•	•		
CINISHELL SYSTEMS (CA-7)	•	• 512 LINES	•	•	•	•	•	PROGRAMMABLE LOOK UP TABLE	16 BIT	•	•	•	•		SCROLL, ZOOM BYTE-PACKING
CRESCOD COMPUTERS	•		•	•				PROGRAMMABLE LOOK UP TABLE	16 BIT SERIAL DMA	•	•	•			SCROLL, ZOOM
CONVAL CORP VISION ONE	•	•	•	•	•	•	•	PROGRAMMABLE LOOK UP TABLE	16 BIT	•	•	•	•		ZOOM, PAN
MANUNATSU TV CO. LTD		• 1024 LINES			•	•	•	SHOOTING HISTOGRAM CONVERSION	16 BIT						• NO DETAILS
COLORADO VIDEO INC.		• 512 LINES	• 256 x 512												REQUIRES SPECIAL CAMERA
STANFORD TECHNOLOGY SYSTEM 501	•	• 512 LINES	•		•	•	•	PROG. LOOK UP TABLE	16 BIT (DIFF.)	•	•	•	•		NO VIDEO OUTPUT
AYDIN CONTROLS	•									•					ZOOM, SCROLL
GRAPHICS DISPLAY COB															GRAPHICS ONLY - NO GREYSCALE
DE AREA SYSTEMS INC. IP 5000	•	• 512 LINES	•		•	•	•	PROG. LOOK UP TABLE (INPUT)	16 BIT	•	•	•	•		ZOOM, SCROLL
CSP INC.		•	• 160 OR LARGER						16 BIT						SCROLL
LEXIDATA CORP.	•		•					PROG. LOOK UP TABLE	16 BIT	•	•	•			ZOOM, SCROLL
SPATIAL DATA SYSTEMS INC.		•						PROG. LOOK UP TABLE	16 BIT RS232	•	•	•			
PRINCETON ELECTRONIC PRODUCTS PEP 500		•		•											• TWO MOVE ATTENTION ON SCREEN ZOOM

FIGURE 3.3-1 IMAGE PROCESSING SYSTEM COMPARISONS

Also, the memory management system must manage a large records data base, control mechanical motion, and must be capable of executing a large program that manipulates an image in memory. Six systems were examined after preliminary screening. A comparison is given in Figure 3.4-1. Two of these computers, the Hewlett-Packard 1000 and the Prime 550 were examined in detail.

Memory

The Prime 550 memory capacity of 8 MB of real memory and 32 MB of user memory by means of the virtual memory feature exceeds present and probably future requirements. HP memory capability does not meet these requirements, but a means of swapping between programs is provided. This would require development of programs in segments, increasing programming costs and greatly increasing execution time.

Software

The software capabilities of the Prime exceed that of the HP, although the HP does provide the software tools necessary to meet minimum requirements. Prime provides full implementation of FORTRAN IV while HP does not, but HP does provide work-around methods.

Speed

One of the main purposes of the computer is to control the digital image system and the possible manipulation of large data arrays (up to 512 x 512). A benchmark program was written to define the exact speed of the two computers performing the type of calculations encountered in the application. The result of this benchmark test showed a speed advantage of better than 2.1x for Prime over HP.

REVT • MEETS REQTS • B.D. NOT DETERMINED • COVERED ELSEWHERE	DISK SIZE	MAG TAPE	PROG. SIZE	DATA SIZE	SOFTWARE			SPEED OF FIN. SW	COST	SOFTWARE DEV. COSTS	NOTABLE FEATURES	COMMENTS	REASONS FOR SELECTION
REQUIREMENTS	50MB MIN.	800/1600 bp1 SWITCH- ABLE	120MB MIN.	375KB MIN. 2.1MB MAX	MULTI- USERS MIN. 3	FORTRAN IV W/FILE EXTENSIONS	DATA BASE MONT	SPEED RIGHTS BASED ON MOVING LARGE ARRAYS	<100K	COSTS ESTIMATED ON EXPERIENCE W/DATA CHAL. HEWLETT-PACKARD, AND PRIME	ADVANTAGES APPLY TO OVERALL PROJECT SUCCESS		
PRIME 550	64MB	•	32MB (INCL DATA)	• 1	63 MAX	•	•	VERY FAST	99.85K INCL. DISCOUNT	MUCH BELOW AVERAGE	TRUE VIRTUAL MEMORY FORTRAN DEBUG REMOTE DIAGNOS- TICS	1. MEMORY CAPACITY • 8MB PROG + DATA • 32MB VIRTUAL MEM. 2. BENCHMARK TEST SHOWS 2.1X FASTER THAN HP	MEETS MAX. REQTS. HIGHEST PERM. PER COST RATIO
HEWLETT-PACKARD 1000	50MB	800 ¹ bp1 OR 1600 bp1	54KB	1.8MB	64 MAX	Q3	•	FAST	80.86K INCL. DISCOUNT	HIGH	FORTRAN IV NOT FULLY IMPLEMENTED 2. ANSI STD FORTRAN DIFFICULT IMPLE- MENTATION OF LARGE PROGRAMS	1. REQUIRES SUBSTI- TUTION OF OTHER BRAND (UNKNOWN) 2. ANSI STD FORTRAN NOT IMPLEMENTED, OTHER MEANS PROV.	OVERALL COST HIGHER THAN PRIME, DOESN'T MEET MAX REQTS
DATA GENERAL 9/130	50MB	•	64KB INCL DATA	•	2 MAX	•	ADDITIONAL COST	SLOW	90 LESS DISCOUNT	HIGHEST	MEMORY USE VERY DIFFICULT OVER 64KB PROG + DATA		PERFORMANCE TOO LOW
DIGITAL EQUIP. CORP. PDP 11/34	176M	800 bp1	56KB INCL DATA	•	MULTIPLE	•	•	B.D.	101K + PERIPH.	AVERAGE			DOES NOT MEET ALL REQTS
DIGITAL EQUIP. CORP. VAX 11-780	67MB	•	32MB INCL DATA	•	96 MAX	•	•	B.D.	153K + PERIPH.	BETTER THAN AVERAGE			COST TOO HIGH
PRIME 400	80MB	•	32MB INCL DATA	•	63 MAX	•	•	VERY FAST	140K + PERIPH.	MUCH BELOW AVERAGE			COST TOO HIGH

FIGURE 3.4-1 COMPUTER COMPARISONS

Maintainability and Reliability

The selection of the prime computer over the Hewlett-Packard was also made due to the maintainability and reliability of the computer. The Prime computer has proven to be reliable, and the diagnostic capabilities have proven to be effective. In one case, the diagnostic program located a bad memory chip, and since the chip was mounted in a socket, it was easily replaced.

3.5 EQUIPMENT INTERFACING

The interconnection of most subsystems is either inherent in a subsystem or a simple switching control to route signals to manual controls or automatic controls. The exceptions are the interfaces between the Prime 550 computer and both the STC Model 70/E image processor and the motor control.

The image processor interface includes both software and hardware components. The hardware was designed around the requirements of STC Model 70/E hardware and software. The interface software design makes the hardware compatible with the STC user programs. The interface was built on the Prime General Purpose Interface Board (GPB) and installed in the computer. Two cables connect the STC image processor to the GPB. The details of the interface board and software driver are included in Appendices 1 and 2.

An interface between parts movement and computer or operator was designed around joy sticks, the type used in the hobby industry. The design allows either manual or computer control of the part manipulation table. Control is provided for X, Y, and rotational parts movement, X and Y lead mask position, X-ray tube height and X-ray voltage and current settings. The details of the motor control interface design and the software driver are included in Appendices 3 and 4.

4.0 RADIOGRAPHY

Radiography is the standard quality assurance inspection technique used for a wide variety of applications, such as assembly verification and inspection of casts and welds. Excessive costs and flow times have made film-based techniques increasingly prohibitive, causing industry to search for non-film radiographic inspection methods.

4.1 REAL TIME RADIOGRAPHY

Real time radiography is a non-film technique that uses a TV camera and monitor to display the image instead of capturing it on film. In addition to eliminating the cost of film and reducing flow time, real time radiography offers an additional advantage—that of motion.

4.1.1 THE ADVANTAGE OF MOTION

Real time radiography allows movement of the test part during the radiographic inspection. This is one major advantage that real time X-ray has over film radiography for the inspection of most complex items. Motion of the part, especially rotation, allows the observer to integrate the total picture and build a three-dimensional impression, greatly enhancing the ability to detect and identify flaws and perceive their location.

When a part is in motion, small flaws may be easily differentiated from background noise or system imperfections. The ability to view a part in motion provides a rapid means of scanning complex parts at various angles, allowing more reliable identification of any defects.

The lower resolution inherent in real time radiography is compensated for by the advantages of motion. Real time X-ray easily permits 100% inspection of a part, which is impractical with film techniques for most applications.

4.1.2 METHODS OF REDUCING SCATTERED RADIATION

Real time systems do not as yet provide the quality that has been developed with film methods. The system used in this project will resolve thickness changes in aluminum down to about 2% under optimum conditions while film can usually resolve changes much less than 2% (2T). Under less than ideal conditions, resolution and contrast suffer from scattered radiation from complex parts and from the greater distance between the fluorescent screen and the object being tested.

Scattered radiation may be reduced by masking off areas that do not contribute to a specific point of interest. This has been accomplished by remotely controlled lead shields that can be closed from two directions (X & Y) reducing radiation from the sides. This has an added advantage for the vidicon tube, which is easily blinded by areas of high intensity light. For example, if the point of interest is in a thick section of the missile, the adjacent thin sections can be masked off by remote control.

Since lower energy photons tend to be more easily scattered, they may be reduced somewhat by filtration from the X-ray beam. Filtering may be accomplished by placing a thin film of lead over the X-ray tube.

The medical profession uses a collimating diaphragm composed of many thin closely spaced lead sheets sandwiched between transparent spacers. The lead sheets are parallel to the X-ray beam, allowing it to pass, but blocking nearly all scattered radiation. The diaphragm, also called a grid, is placed between the object and the fluorescent screen. This system was evaluated and appears to be of some benefit.

4.1.3 VIEWING OF THE X-RAY IMAGE

There are two basic equipment configurations used for converting the X-rays to a visible image and transmitting that image to a TV monitor. The vidicon camera and an image intensifier tube may be optically coupled, as was done for this project, or an isocon camera may transmit the image from a fluorescent screen.

A brief examination of an isocon camera was made using fluorescent screens designed for medical use. It was apparent that even under crude experimental conditions the isocon system is superior to the vidicon system in at least one respect. The low light level sensitivity and extreme dynamic range obviate the need for the intricate masking necessary with the vidicon camera and image intensifier. The only drawback of the fluorescent screen-isocon camera system was an overall lower sensitivity. With the system examined, full X-ray output was required to equal the image intensifier-vidicon brightness at a much lower X-ray output.

The combination of image intensifier and vidicon camera is very sensitive but suffers from a narrow dynamic range and thus is best suited for objects of relatively uniform density. The combination of fluorescent screen and isocon camera have wide dynamic range but require more X-rays (higher energy or greater intensity) to produce a satisfactory image. Therefore, there is a different best system for each application. At this point in the development task, it appears that a very good system able to handle most jobs would be as follows:

- o High Energy or High Intensity X-Ray Source
- o Selection of Fluorescent Screens
- o Isocon Camera
- o Selection of Lenses

This setup could produce images of complex parts, and still provide the flexibility to examine details where higher resolution is needed. Greater detail would be achieved by a narrower field of view.

If the application requires examination of very thick parts, for example more than 5 cm of aluminum, the fluorescent screen-isocon system may require integration of many frames to obtain adequate brightness. This would remove the advantage of motion in real time.

4.2 IMAGE PROCESSING

Although the real time system described above permits examination of a part while it is in motion, it does not provide the contrast or resolution of film methods. Image quality may be improved considerably through image processing.

Image processing can be accomplished by two basically different approaches, video signal processing and digital image processing. Video signal processing provides a very rapid response to changes in an image, which is tantamount to doing real time image enhancement. However, it is a very limited and inflexible technique. Digital image processing, a more versatile, computer-controlled method, was selected for this project. Digital image processing technology has advanced rapidly in recent years because of the vast amounts of satellite (LANDSAT) data available for land use planning and other applications. Many of these techniques are directly applicable to radiographic image processing. The processes applicable to this study are fundamentally of four types: noise reduction, edge enhancement, contrast manipulation and color representation.

4.2.1 NOISE REDUCTION

A video image is composed of many adjacent lines with the intensity of each line varying continually along its length. On the other hand, a digital image is composed of rows and columns of picture elements (pixels), each of which has a discrete intensity. A video signal contains a significant level of noise, producing a grainy and confusing effect. The magnitude and distribution of the noise is basically random, and therefore, can be reduced by averaging several images (video frames) within a very short time. The noise level is reduced by the square root of the number of frames averaged. If the object is stationary, a large number of frames can be averaged, greatly reducing the noise level. Averaging 256 frames reduces the noise by a factor of 16; at an execution rate of 15 frames per second, this requires only 17 seconds. In this project, all images were produced by averaging 256 frames.

A second method of noise reduction, localized averaging, is accomplished by computing the average of a pixel intensity with that of its neighbors. Although this technique smoothes out the noise, it also blurs the image, making the details unclear. This technique is unsatisfactory for radiographic applications.

4.2.2 EDGE ENHANCEMENT

Edges in an image may be poorly defined, particularly when there is not much contrast between an object and its surroundings. Edge enhancement is a method of increasing the contrast where there are edges, thus making them more visible. For example, if a light grey object is surrounded by dark grey, edge enhancement would make the light grey object lighter at the edge, and the surroundings darker at that edge, improving the contrast between the two. Digital edge enhancement transforms the intensity of each pixel based on the intensity of its neighbors. There are several methods of edge enhancement, each producing different results. Three methods of edge enhancement have been examined here: difference, gradient, and Laplacian. When one of these techniques is applied to an image, we are left with only the edges. By adding this result to the original in about a 1:2 ratio, we obtain an image whose edges have been enhanced. This has proven to provide the most aesthetically pleasing results. Edge enhancement techniques naturally result in an increase in noise. For this reason, it is very important to start with the lowest noise image available before performing edge enhancement functions. Low noise can best be achieved by averaging a large number of video frames.

Difference

The difference edge enhancement technique is the simplest of transformations, whereby the intensity of each pixel is replaced by the difference between its intensity and that of an adjacent pixel. This produces an effect like casting a shadow from one side. The technique is fast, requiring only one video frame to obtain the difference, but increases any noise present in the image. Figure 4.2.2-1, an example of the difference technique, was produced by adding the output to the original in a ratio of 1:2 and displaying it with the original in a split screen format.

Gradient

The gradient of an image is its rate of change of intensity with respect to distance in a specific direction. This is a general case of the difference technique. The



FIGURE 4.2.2-1 DIFFERENCE EDGE ENHANCEMENT



FIGURE 4.2.2-2 GRADIENT EDGE ENHANCEMENT

analytical solution to this problem involves finding the differential equation of each pixel. However, an approximation to this solution may be made by performing a convolution of a small gradient matrix about each pixel. For example, a 3 x 3 north-south gradient matrix would be:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}$$

The convolution can be performed in 17 video frames. The gradient results in a shadowed edge; its direction can be varied and it is less sensitive to noise in the image. Figure 4.2.2-2 is an example of an east gradient with a 3 x 3 convolution matrix.

Laplacian

The Laplacian transform of an image is its rate of change of intensity with respect to distance without regard to direction. As with the gradient technique, a matrix convolution can be performed as an approximation. The Laplacian transformation being non-directional may be represented by a matrix with symmetry about the center:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

The output of the Laplacian transformation shows no shadow and amplifies noise. An example is shown in Figure 4.2.2-3. The non-directional nature of the Laplacian transform amplifies all edges regardless of orientation.

These edge enhancement techniques can improve the quality of an image and the observers ability to visualize the original object. The Laplacian technique is often the most useful, especially in complex images, but when the detail in question is linear, as in screw threads, a gradient technique may be better.



FIGURE 4.2.2-3 LAPLACIAN EDGE ENHANCEMENT

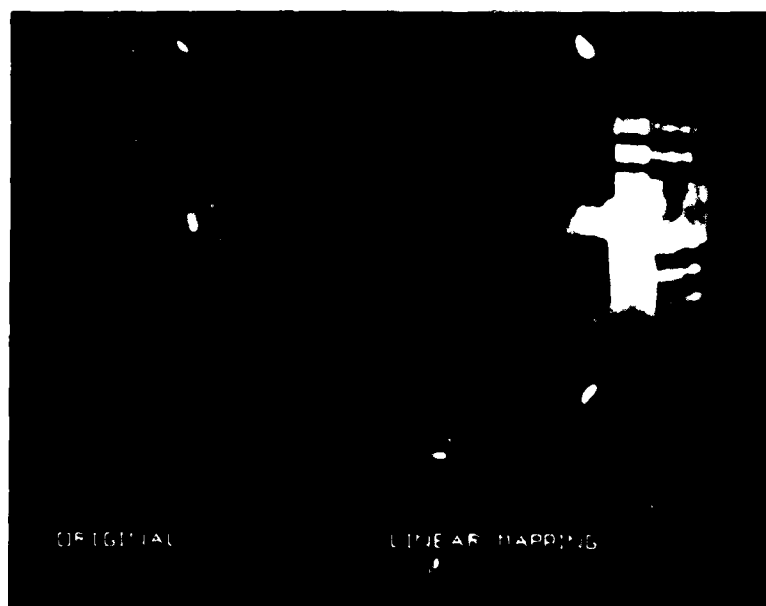


FIGURE 4.2.3-1 LINEAR MAPPING IMAGE ENHANCEMENT

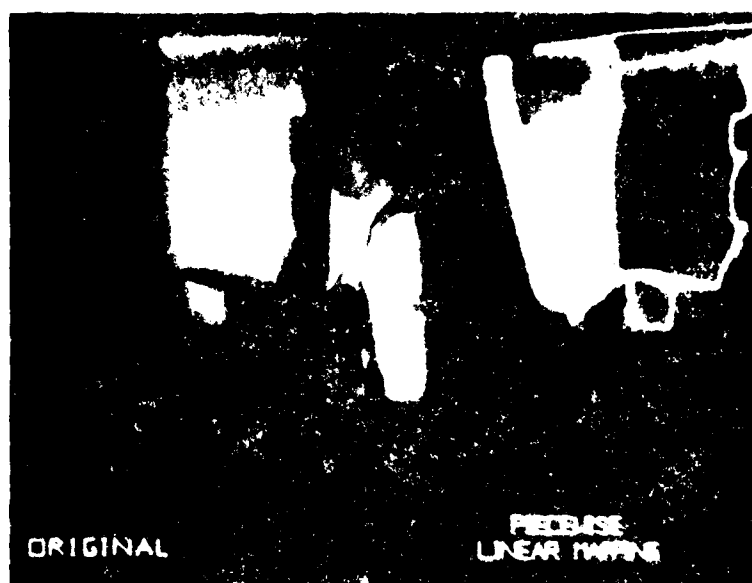


FIGURE 4.2.3-2 PRECISE LINEAR MAPPING IMAGE ENHANCEMENT

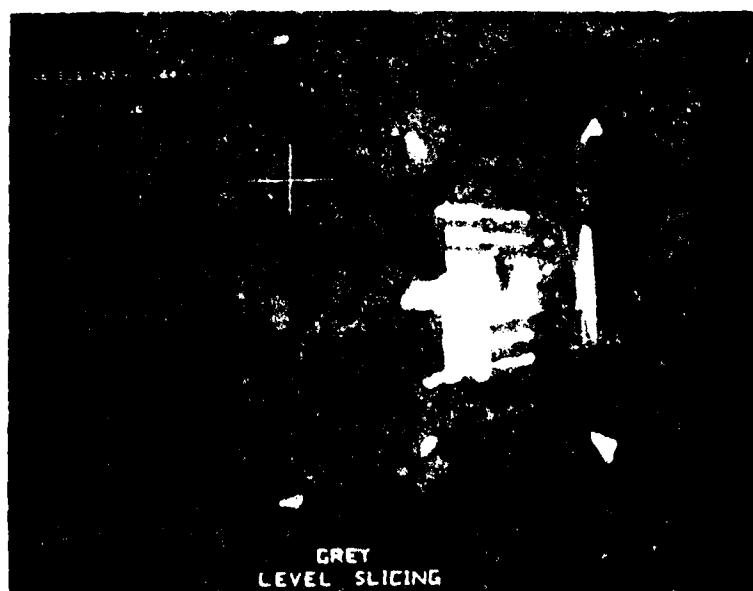


FIGURE 4.2.3-3 COLOR LEVEL SLICING

4.2.3 CONTRAST MANIPULATION

An eight-bit digital image can produce 256 different shades of gray, but the human eye can distinguish only about 64. An image therefore could have detail available that the eye could not see. By contrast manipulation it is possible to stretch the gray scale in one regime at the expense of another. For example, detail may be brought out in the dark areas by a logarithmic rescaling, or detail in the light areas may be brought out by an exponential rescaling.

Digital image processors perform contrast changes by means of a look-up table translation (mapping). Thus a table with 256 entries may translate any original level to any other chosen level.

The most useful means of modifying an image while maintaining the overall information is linear expansion of a certain gray regime. This technique, called piecewise linear mapping, gives the operator a nearly infinite range of contrast modifications. Figure 4.2.3-1 shows a linear mapping of an image that lacked the full range of intensities. Figure 4.2.3-2 shows an extreme modification where all intensities outside the regime of interest were mapped to zero (black).

Gray level slicing is a technique where all intensities below a selected value are set to black and all others set to white, giving a marked delineation at that intensity. Another means of viewing the same information is through color level slicing. In this technique the image is left intact but all intensities within a chosen regime are overlaid with a contrasting color. When this feature is coupled with a track ball, joy stick or other interactive means, it is possible to bring out subtle changes in intensity not visible to the eye. Figure 4.2.3-3 shows an example of color level slicing.

4.2.4 COLOR REPRESENTATION

Although the human eye can differentiate only about 64 different shades of gray, it can distinguish many times that number of shades of colors. For this reason it is often advantageous to add the color dimension to the image, allowing the observer

to see smaller changes in intensity. Pseudocolor is a function which assigns a selected color to each gray level intensity. Figure 4.2.4-1 shows an example of the pseudocolor operation.

4.2.5 SOFTWARE OPERATIONS

The Stanford Technology Corp. Model 70/E image processor has been designed to be very flexible and accept all controls from a computer. STC provides a software package, System 500, which is primarily designed for satellite data processing. Application of the system to this task and in particular to the Prime 550 computer required some software modification and creation of two additional image processing functions.

Software Translation

The System 500 software, while written in FORTRAN IV, necessarily contained significant machine-dependent code. This fell into four general areas: hardware



FIGURE 4.2.4-1 PSEUDOCOLOR IMAGE ENHANCEMENT

drivers (discussed elsewhere), input and output to peripherals, bit manipulation, and program segmentation and swapping. The large size of the System 500 package required the program segmentation to function in a 16-bit computer. These segments are swapped into the computer memory as required to execute a given process. The Prime 550 is designed around virtual memory and an entirely different scheme for executing large programs. The program swapping features were removed, and actual addresses were converted from two byte integers to four byte integers. Other changes were handled by FORTRAN callable subroutines supplied by Prime Computer Inc. These subroutines are described in appendix 7.

Software Additions

Two capabilities necessary for this task were missing from the System 500 software: the ability to accumulate video frames (averaging or time-smoothing), and the ability to easily preserve the output of one operation for use in the next operation. These two functions are described in detail in appendix 6. The first function, subroutines DAVG and DAVGD, was implemented to average a specified number of video frames in the processor's 16-bit accumulator, then convert to an 8-bit image.

The result of an image process exists as the output of a transformation. In order to use this resultant image for further processing, this output must be stored in one of the image planes. The function to save consists of subroutines SAVE and SAVED and allows the result of a process to overwrite the original image or to be saved as a new image in another image plane.

4.3 RADIOGRAPHIC QUALITY

Radiography is a technique analogous to projecting a shadow of a transparent object. All of the parameters affecting the sharpness of the shadow apply to radiography: the relative size of the light source, the distance between shadow and object and light source, smoothness of the surface, etc.

The important features which determine radiographic quality are resolution, sensitivity, and contrast. Silver halide film used in photographic film radiography is very good with respect to all of these properties and the techniques necessary to attain the highest possible quality with film are well understood. The major drawbacks to this technique are the high labor and material costs and long flow times.

4.3.1 PARAMETERS AFFECTING RESOLUTION

The resolution of an image is a function of many parts of the total system. One of the key parameters is geometric unsharpness, which is dependent upon the focal spot size of the x-ray beam and the relative distances from the object to the x-ray source and to the screen (or film).

The focal spot size is measured by locating a lead plate with a pin hole midway between the source and a radiographic film. It is then calculated as the size of the image less twice the size of the pinhole. Measurements made in this manner are summarized in Figure 4.3.1-1. As indicated, no significant change in spot size was observed over the range of experimental conditions. The average values also agree with the values quoted by the supplier.

The geometric unsharpness is calculated using the formula

$$U = F \times (t/D),$$

where: U = Geometric unsharpness

F = Focal spot size

t = distance from object to screen

D = distance from X-ray source to object

For this system, t = 14 cm (5.5 in.) and D = 213 cm (7 ft.). The U calculated for the large (4.0 mm) spot is 0.26 mm and for the small (1.9 mm) spot is 0.13 mm. These values reflect a basic limitation in the X-ray system, which affects both film and

non-film resolution. Thus, one advantage film has over fluoroscopic systems is that "t" can be made very small by placing the film in contact with the object.

Fluoroscopic systems and film both have inherent limitations in resolution. For example, the image intensifier Model 22D20R is advertised to have a maximum resolution of 0.23 mm normal and 0.18 mm high magnification.

The video portion of a real time X-ray system provides several possible sources of resolution limitation. The most significant of these are signal-to-noise ratio, bandwidth, and total number of raster lines. For example, normal video systems display about 480 lines, while image processor systems usually display 512 lines with 512 points per line. The optical coupling between the image intensifier and video camera in this system produces a ratio of screen size to object size of 2.1 for normal and 3.2 for high magnification. This means that the distance between picture elements (pixels) represents 0.24 mm and 0.16 mm at the object for normal and high magnification, respectively.

VOLTAGE, KV	LARGE SPOT, mm	SMALL SPOT, mm
180	4.12, 4.42	1.92
160	3.97, 3.97, 4.12	1.72
150		1.97
140	4.32, 3.87, 3.82	2.02
120	4.17, 3.67, 3.77	1.92, 2.37
100	3.97, 3.62, 3.72	1.87
80	4.07	1.82
60	4.02	1.87
40	4.02	1.92
AVERAGE	3.98	1.93
STANDARD DEVIATION	0.22	0.17
PROBABLE ERROR	0.14	0.11
ADVERTISED	4.0	1.5

FIGURE 4.3.1-1 MEASURED FOCAL SPOT SIZE FOR PHILLIPS ELECTRONICS INSTRUMENTS, INC. MODEL MCN 321

These three limiting factors are summarized in Figure 4.3.1-2. It is evident that an improvement in either geometric unsharpness or image intensifier resolution would provide very little system improvement without a comparable change in the other. On the other hand, the overall system performance could be improved if a different lens coupling were used to magnify the image and provide more pixels to depict each detail. That is, there should be several pixels (TV lines) representing the minimum detectable detail limit of the image intensifier for maximum system resolution. The minimum frequency response necessary to equal the 0.18 mm resolution is 15 MHz at the line rate of 15,750 lines per second for standard video. Therefore, it is necessary to purchase high quality electronics when assembling a real time X-ray system.

RESOLUTION FACTOR	LIMITATIONS	
Geometric Unsharpness	0.13 mm (small spot)	0.26 mm (large spot)
Image Intensifier	0.18 mm (high mag.)	0.23 mm (normal)
Pixel Separation	0.16 mm (high mag.)	0.24 mm (normal)

Figure 4.3.1-2 Resolution Limitation of Prototype System Due to X-Ray Focal Spot, Image Intensifier and Video Display

4.3.2 MEASUREMENT OF RESOLUTION

Quality level, a measurement affected by resolution, sensitivity and contrast, is the parameter radiographers generally use to describe the performance of an X-ray system. The penetrameter is used for this measurement and is a piece of material of specific thickness (2% of material being radiographed) with three holes, the diameters of which are 1, 2, and 4 times the thickness of the penetrameter. The penetrameter thickness used for 2.54 cm (1.0") material is 0.0508 cm (0.02"). The quality level is expressed as the penetrameter thickness and the smallest hole size visible, e.g., 2%-2T = .04" dia. hole in .02" thick penetrameter on material of 1" thickness. With the system described here, using aluminum plate and aluminum

penetrameters, results are a function of object thickness. for example, Figure 4.3.2-1 shows the quality levels of real time X-ray and image enhancement. Gray level slicing, a method of contrast manipulation, was the method of image enhancement used to emphasize the penetrometer holes.

ALUMINUM THICKNESS	QUALITY LEVEL	
	REAL TIME RADIOGRAPHY	IMAGE PROCESSING
3.81 cm (1:5)	4% - 2T	2% - 2T
2.54 cm (1.0)	4% - 2T	2% - 2T
.92 cm (.375)	4% - 2T	4% - 2T

Figure 4.3.2-1 Penetrameter Sensitivity

4.4 RADIOGRAPHIC RECORDS

The elimination of X-ray film necessitates using an alternate method of image storage. The requirements for such a method are: the image must be stored for a long time, it must be easy to retrieve, it must retain the original quality of the image, it must be able to be entered back into the imaging system, and it must have a low cost per image.

The following types of storage methods were examined: video tape, video disc, microfiche, digital disc and digital tape. Research into video tape and video disc recorders determined that they were limited to recording 480 lines of video image, whereas the image processor produced 512 lines of image. In order to use either of these two types of recording methods, a converter would have to be used, which would result in lost information. Also, the bandwidth of video recorders was much lower than that required by the image processor, unless very high quality recorders are used. Image retrieval from video tape is difficult since images have to be read sequentially. Images stored in video form could not be entered into the image processor and therefore could not be used to recreate the image process that leads to the pass/fail decision.

Images could be stored on microfiche in digital format without loss of information. However, once an image is put on microfiche, there is no way to enter the data back into the system.

Digital discs offer fast storage and easy retrieval, but have an extremely high cost per image. In contrast, digital magnetic tape has a very low cost per image. Digital storage on tape or disc permits easy re-entry of the image into the system and retains all information. Digital tape has moderate retrieval time.

Thus, it is felt that image storage on digital tape best meets storage requirements.

5.0 COMPUTER-AIDED INSPECTION

One of the goals of this project has been to demonstrate a prototype computer-aided inspection system. In the automated non-film radiographic system, the computer controls parts positioning, records management, and the functions of an inspection plan.

5.1 CONCEPT

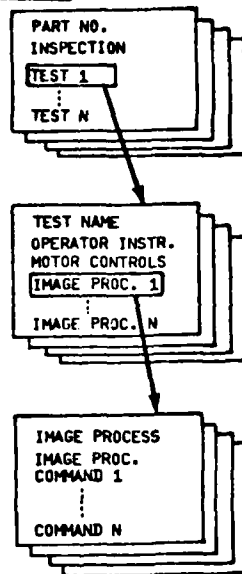
The automated system provides computer control of the manual functions that are labor intensive or sensitive to human error. These primarily fall into the category of data recording and retrieval. Examples of this are: inspection plan preparation and dissemination, inspection results recording and retrieval, and radiograph (image) storage and retrieval. A successful system must have certain basic capabilities. Among these are: simple plan preparation and modification procedures, minimum input from the operator (radiographer), and maximum traceability (retrieval of records, re-creation of inspection, etc.).

5.2 SOFTWARE DESIGN

The automated inspection system software design is detailed in Appendix 5. The program, termed CAI, for computer aided inspection, has three basic functions: inspection, planning and data retrieval.

The key to a useful computer-aided system is in the design of the software - its flexibility and ease of operation by the user. The easier it is for the user, the more sophisticated the software must be. This prototype system has been designed with enough sophistication for demonstration purposes but would require minor improvements for use in a real environment. An overview of the data base design is shown in Figure 5.2-1. The inspection plans are actually composed of three separate files: first, the plan main records, each containing primarily a list of test names; second, the plan test file, containing records with the basic data for a specific test such as motor and position parameters, special instructions for the inspector and names of image processing functions; and third, image processing command file with each record containing a series of image processing commands.

PLAN DATA



FILE 1: PLAN MAIN

EXAMPLE: ROLAND, ASSEMBLY VERIFICATION

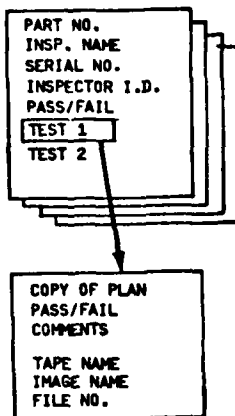
FILE 2: TEST NAME

EXAMPLE: GAS FLOW SHIM

FILE 3: IMAGE PROCESS NAMES

EXAMPLE: LAPLACIAN

RESULTS DATA



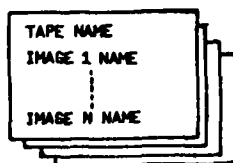
FILE 1: RESULTS MAIN

EXAMPLE: ROLAND/FINAL/SN9-153 012880135003

FILE 2: TEST RESULTS
COPY OF PLAN AT INSPECTION

EXAMPLE: ROLAND/FINAL/SN9-153/WIRES
012880135003

HISTORICAL DATA



FILE 1: TAPE

EXAMPLE: ROLAND/FINAL/SN9-153/WIRES
012880135003

FIGURE 5.2-1 COMPUTER-AIDED INSPECTION DATABASE

This data base structure allows the planner to create a new plan by merely assembling a list of tests in the order desired. The computer then accesses those tests by name from the various files at the time the inspection is performed. As the inspection is performed, the test results are saved (pass-fail and inspector's comments) along with a copy of the test plan. The last file maintained by the system is the image storage file, a cross reference of magnetic tape names, image names, and unique inspection names.

An overview of the CAI system is shown in Figure 5.2-2.

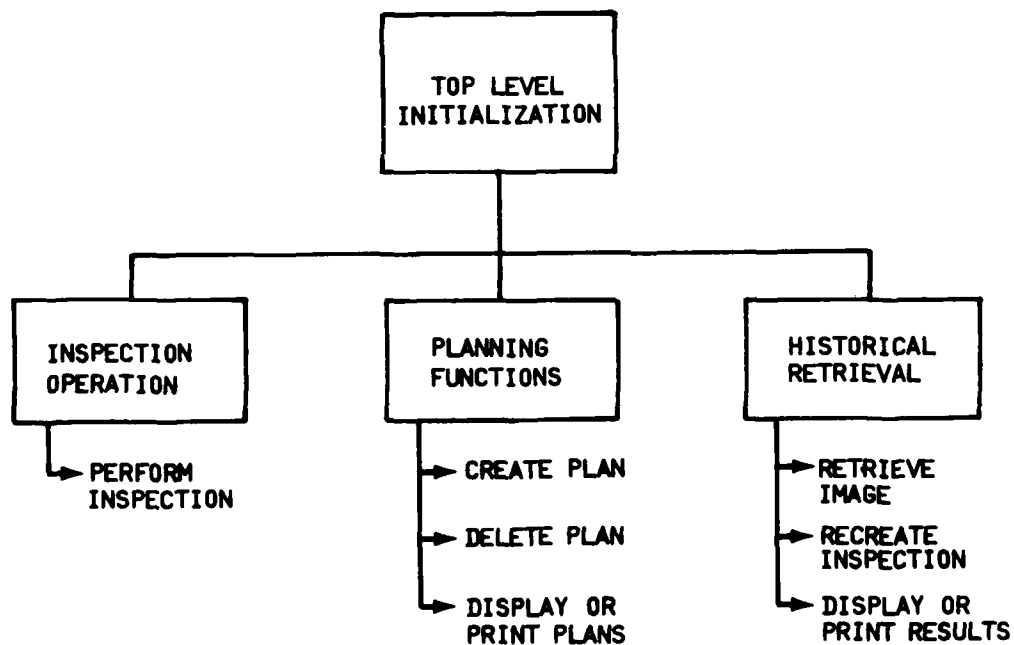


FIGURE 5.2-2 COMPUTER-AIDED INSPECTION SYSTEM

5.2.1 INSPECTION

The inspection function of CAI is designed around a data base management system (DBMS), in this case Prime's multiple index data access system (MIDAS). A CAI system was designed to interact with an inspector who has no knowledge of image processing. The operator need only enter, at a keyboard, his identification (Stamp No.), the Part No. to be inspected, name of the inspection, and the part's serial number. The computer will access a data base containing the inspection plan for that part. That plan will contain all of the necessary data to run the test: coordinates to position the part, functions and parameters to control the image processor, and instructions for the inspector indicating the key inspection criteria. The inspector then enters a pass/fail decision and any observations for future reference. These data are then saved in the historical data file and a copy of the image is recorded on digital magnetic tape.

An overview of the inspection function of the system is shown in Figure 5.2.1-1.

5.2.2 PLANNING

The planning functions are, like the inspection functions, designed around a data base management system. These tasks are greatly enhanced by a screen editing capability. Screen utilities are a software package designed to allow interactive data entry. In this case a planner sits at a CRT terminal and modifies existing plans or creates new plans by entering data into specified fields. The data transferred to the data base is controlled by the screen format and the application program (software). Examples of these screens are shown in Figure 5.2.2-1.

The planning package provides the following operations:

1. Create new plan records.
2. Modify existing plan records.
3. Make new records from old.
4. Delete plan records.
5. Display or print plan or result records.
6. Display or print lists of existing records.

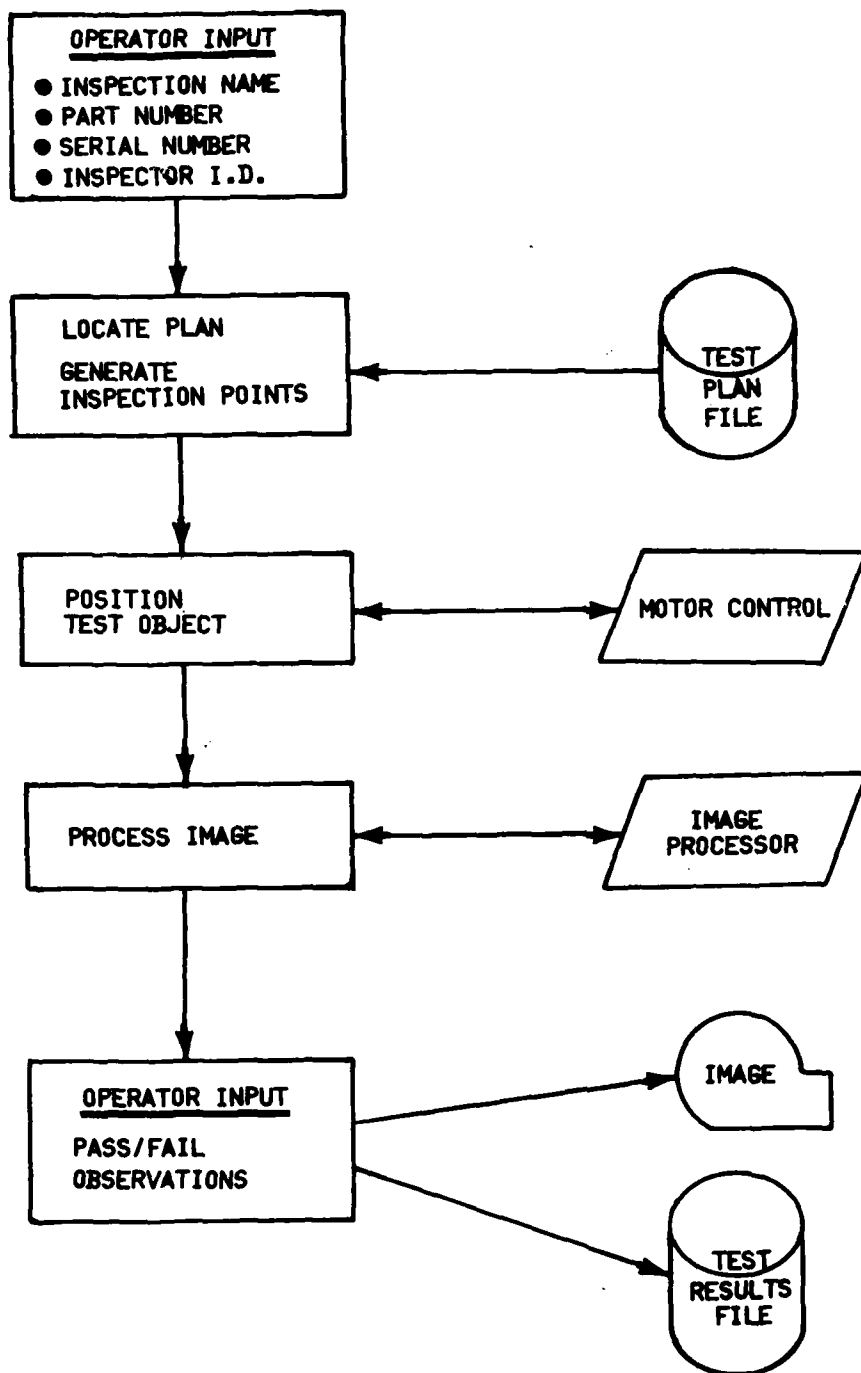


FIGURE 5.2.1-1 INSPECTION SUBSYSTEM

INSPECTION TEST PLAN

```

PART NUMBER: -----
INSPECTION NAME: -----
NUMBER OF TESTS: --
TEST 1 NAME: -----
TEST 2 NAME: -----
TEST 3 NAME: -----
TEST 4 NAME: -----
TEST 5 NAME: -----
TEST 6 NAME: -----
TEST 7 NAME: -----
TEST 8 NAME: -----
TEST 9 NAME: -----
TEST 10 NAME: -----
  
```

```

TEST NAME: ----- DESCRIPTION: -----
-----
-----
-----
MOTOR MASK: X POSITION --- X SPEED --- Y POSITION --- Y SPEED ---
MOTOR TABLE: X POSITION --- X SPEED --- Y POSITION --- Y SPEED ---
MOTOR ROTATION POSITION --- SPEED ---
IMAGE PROCESS 1 --- CODE: ---
IMAGE PROCESS 2 --- CODE: ---
IMAGE PROCESS 3 --- CODE: ---
IMAGE PROCESS 4 --- CODE: ---
IMAGE PROCESS 5 --- CODE: ---
IMAGE PROCESS 6 --- CODE: ---
IMAGE PROCESS 7 --- CODE: ---
IMAGE PROCESS 8 --- CODE: ---
  
```

FIGURE 5.2.2-1 PLANNING SCREEN FORMATS

5.2.3 RETRIEVAL

In order that the data be of value, there must be a means of retrieval of all data and re-creation of the original conditions that allowed the inspector to make a pass or fail decision. To provide these capabilities, it is necessary to not only save the image and results of inspection but also a copy of the inspection plan at the time of inspection. This is necessary because of the possibility that the inspection plan records may be modified or deleted between the time the inspection is performed and the time when it is recreated. These minimum capabilities have been included in the prototype computer aided inspection system. The full details are included in Appendix 5.

5.3 APPLICATION TO ROLAND FINAL INSPECTION

A test plan was made for final assembly verification of the ROLAND propulsion unit, based on the evaluation of image processing techniques at each inspection point. A summary of the optimum image processing sequences is given in Figure 5.3-1. All of the inspection points could be seen with real time X-ray, but image processing greatly improves the operators ability to interpret the images. Full grey scale utilization is useful at all inspection points, and can be achieved by contrast modification. Some areas, such as those with screw threads, can be made sharper by edge enhancement. Missile position was controlled by the test plan.

Lead mask position and X-ray energy can be part of the plan, also. Use of the prototype test plan provided data for flow time estimates. The inspection of each missile should take approximately 20 minutes. Since the ROLAND missile was still in the development stage during this project, inspection requirements may change before full rate production is initiated. The test plan would then be changed accordingly.

INSPECTION AREA	PROCESSES PERFORMED	STC IMAGE PROCESSOR COMMANDS
1. Inhibited Grains	Contrast Modification	\$A>SCALE;
2. O Rings	Contrast Modification	\$A>PLIM(BR=40 0 170 255);
3. Gas Flow Shim	Contrast Modification Gradient Edge Enhancement	\$A>SCALE; \$A>SAV>\$B; \$A>SAV>\$C; \$C>CONV(NR=3 NC=3 W=-1 1 1 -1 -2 1 -1 1 1); \$A \$C>ADD(*MINMAX); \$A \$C>SAV; \$A>SCALE;
4. Snap Ring	Contrast Modification	\$A>SCALE;
5. Igniter Wires	Contrast Modification Laplacian Edge Enhancement	\$A>SCALE; \$A>SAV>\$B; \$A>SAV>\$C; \$C>CONV(NR=3 NC=3 W=-1 -1 -1 -1 8 -1 -1 -1 -1 \$A \$C>ADD(*MINMAX); \$A \$C>SAV; \$A>SCALE;
6. Nozzles	Contrast Modification Gradient Edge Enhancement	\$A>SCALE; \$A>SAV>\$B; \$A>SAV>\$C; \$C>CONV(NR=3 NC=3 W=-1 1 1 -1 -2 1 -1 1 1); \$A \$C>ADD(*MINMAX); \$A \$C>SAV; \$A>SCALE;
	Contrast Modification	\$A>SCALE;

Figure 5.3-1 ROLAND Image Processing Sequence

6.0 COST - BENEFIT

Real time X-ray radiography would be of benefit to nearly any application. The cost and flow time can be reduced significantly compared to film radiographic techniques. A possible cost reduction of 10:1 in high volume applications is anticipated. Computer aided inspection would benefit the inspection of parts from high volume production, providing a potential for even greater savings.

The final assembly verification of the ROLAND missile may be improved by a cost reduction of about 8:1 by the implementation of real time X-ray. The capital costs of implementation of the system described in this project are listed in Figure 6.0-1. Using these values and applying methods of investment analysis it is possible to draw conclusions about the feasibility in a specific application. Figures 6.0-2, and 6.0-3 illustrate the investment analysis results. The assumptions used in the analysis are as follows: 8 year depreciation, 10% tax credit, 5.4% sales tax and 24 month payback (44.5% rate of return).

This analysis predicts that the annual savings must be greater than \$45,000 in order to achieve a two year payback of the \$70,000 capital investment required for real time X-ray equipment only. The complete system, providing image processing and computer aided inspection as well as real time radiography costs \$300,000. This expenditure could be justified if an annual cost savings of \$193,000 could be realized.

EQUIPMENT	CAPITAL THOUSANDS	TOTAL
I. Real-Time-X-Ray		
1. X-Ray to Visible Light Conversion and Camera	44	
2. Monitor	5	
3. Room Monitor	1	
4. Part Manipulator	<u>20</u>	
SUBTOTAL		70
II. Image Processing and Computer Aided Inspection		
1. Prime 550 Computer with 1 Mbyte Memory 64 Mbyte Disk Dual Density Tape Drive	150	
2. Computer Interfaces for Image Processor and Part Manipulator	10	
3. Image Processor	<u>70</u>	
GRAND TOTAL		<u>230</u> 300

Figure 6.0-1 Real Time X-Ray System Capital Costs

INVESTMENT ANALYSIS No. _____ PRELIMINARY / FINAL

Title LOW COST HIGH VOLUME RADIOGRAPHY

Date 1 SEPT 80

Proposed Investment Item(s)

Item 1. Real Time X-Ray System \$ 70,000 Life 8 Years Salvage Value of Existing Eqmt 0
 2. _____ \$ _____ Life _____ Investment Tax Credit 7,000
 3. _____ \$ _____ Life _____ Sales Tax 3,780

Acquisition Date 1981

Other Procurement/Installation Expense 0

Total \$ _____

Year	(1) Operating Costs - Existing Equipment	(2) Operating Costs - New Equipment	(3) Gross Savings Before Taxes	(4) Depreciation	(5) Net Book Value of New Equipment	(6) Gross CFC	(7) Recoverable CFC	(8) Before Tax Profit or (Loss)	(9) After Tax Profit or (Loss)	(10) Net Proceeds From Sale of Existing Equipment	(11) New Equipment Capitalized Cost	(12) Net Cash Flow	(13) Cum Net Cash Flow
0	0	3,780	-3,780	0	0	0	0	-3,780	-1,966	0	70,000	-71,966	-71,966
1	51,429	6,429	45,000	7,778	62,222	9,644	5,787	43,009	29,365	0	0	37,142	-34,823
2	51,429	6,429	45,000	14,583	97,639	7,384	4,430	34,847	18,120	0	0	32,704	-2,119
3	51,429	6,429	45,000	12,639	35,000	5,425	3,526	35,887	18,661	0	0	31,300	29,181
4	51,429	6,429	45,000	10,694	24,306	3,767	2,449	36,754	19,112	0	0	29,800	58,988
5	51,429	6,429	45,000	8,750	15,556	2,411	1,567	37,817	19,665	0	0	28,415	87,403
6	51,429	6,429	45,000	6,806	8,750	1,356	882	39,076	20,320	0	0	27,125	114,528
7	51,429	6,429	45,000	4,861	3,889	603	392	40,531	21,076	0	0	25,937	140,465
8	51,429	6,429	45,000	2,917	972	151	98	42,181	21,934	0	0	24,851	165,316
9	51,429	6,429	45,000	972	0	0	0	44,028	22,894	0	0	23,867	189,182
10													
Total	462,861	61,641	401,220	70,000		30,742	19,131	350,351	189,182	0	70,000	189,182	

PAYBACK PERIOD 2 YEAR(S) 0 MONTH(S)

RATE OF RETURN ON INVESTMENT 44.65 % (After tax)

FIGURE 6.0-2 INVESTMENT ANALYSIS: REAL TIME X-RAY

INVESTMENT ANALYSIS No. _____ PRELIMINARY / FINAL

Title LOW COST HIGH VOLUME RADIOGRAPHY

Date 1 SEPT 80

Proposed Investment Item(s)

Item 1. _____ \$ 300,000 Life 8 Years Activation Date 1981
 2. _____ \$ _____ Life _____ Salvage Value of Existing Equipmt 0
 3. _____ \$ _____ Life _____ Investment Tax Credit 29,999
 Sales Tax 16,200
 Total \$ _____ Other Procurement/Installation Expense 0

Year	(1) Operating Costs- Existing Equipment	(2) Operating Costs- New Equipment	(3) Gross Savings Before Taxes	(4) Depreciation	(5) Net Book Value of New Equipment	(6) Gross CFC	(7) Recoverable CFC	(8) Before Tax Profit or (Loss)	(9) After Tax Profit or (Loss)	(10) Net Proceeds From Sale of Existing Equipment	(11) New Equipment Capitalized Cost	(12) Net Cash Flow	(13) Cum Net Cash Flow
0	220,571	16,200	-16,200	0	0	0	0	-16,200	-8,424	0	300,000	-308,424	-308,424
1	220,571	27,571	193,000	33,333	266,667	41,333	24,800	184,467	125,923	0	0	159,256	-149,168
2	220,571	27,571	193,000	62,500	204,167	31,666	18,897	149,447	77,733	0	0	140,233	-8,934
3	220,571	27,571	193,000	54,167	150,000	23,250	15,112	153,946	80,052	0	0	134,218	125,284
4	220,571	27,571	193,000	45,833	104,167	16,146	10,495	157,661	81,954	0	0	127,817	253,101
5	220,571	27,571	193,000	37,500	66,667	10,333	6,717	162,217	84,353	0	0	121,853	374,954
6	220,571	27,571	193,000	29,167	37,500	5,812	3,778	167,611	87,158	0	0	116,325	491,279
7	220,571	27,571	193,000	20,833	16,667	2,583	1,679	173,846	90,400	0	0	111,233	602,512
8	220,571	27,571	193,000	12,500	4,167	646	420	180,920	94,078	0	0	106,578	709,090
9	220,571	27,571	193,000	4,167	0	0	0	188,833	98,193	0	0	102,360	811,450
10													
Total	1,985,139	264,339	1,720,800	300,000		131,750	81,989	1,502,788	811,450	0	300,000	811,450	

PAYBACK PERIOD 2 YEARS(Y) 0 MONTH(S)

RATE OF RETURN ON INVESTMENT 44.68 % (After tax)

FIGURE 6.0-3 NON-FILM RADIOGRAPHY WITH COMPUTER-AIDED INSPECTION

7.0 CONCLUSIONS AND RECOMMENDATIONS

7.1 REAL TIME X-RAY

7.1.1 EQUIPMENT

The recommended viewing system consists of a high energy X-ray source, a selection of fluorescent screens, an isocon camera, and a selection of lenses. A part manipulation table and remote control capability must be provided to realize the advantages of real time radiography. If computer-aided inspection is to be implemented, then computer control of part positioning is required. The best image storage and retrieval medium is digital magnetic tape for systems which include a computer (for either image processing or computer-aided inspection). Otherwise, a high quality video tape system is satisfactory.

7.1.2 APPLICATIONS

Real time X-ray can easily be applied to assembly configuration inspections of large, complex parts. The assembly verification of the ROLAND missile can be performed by non-film radiography for 80% less cost than by film methods. Other applications, such as inspection of castings and welds, require higher resolution than the present system can provide.

7.2 IMAGE PROCESSING

The most useful image processing technique is contrast modification. Expanding the range of intensities will improve virtually any real time image. Edge enhancement methods emphasize details in some of the inspection areas, and the preferred type of edge enhancement depends upon the orientation of the detail. Gray level slicing is an effective way to determine the parameters for contrast modification. The pseudocolor operation is not particularly useful for this application.

The image processing techniques were applied to each ROLAND missile inspection point to determine the optimum sequence for assembly verification of the missile. Although there were no critical areas that could not be seen in real time, image processing did improve the quality of the images. For applications in which image processing allows the operator to see a detail that can not be seen in real time, then image processing is a necessary part of the non-film X-ray system.

7.3 COMPUTER-AIDED INSPECTION

The computer-aided inspection scheme saves time by positioning the part and performing the image processing sequence automatically. It also may be used for data base management of records and storage and retrieval of images. This type of automation is cost-effective when a computer is required for the image processing, and when working with high volume production.

7.4 ACCOMPLISHMENTS AND RECOMMENDATIONS FOR FUTURE WORK

This project has produced a system that is capable of meeting the inspection requirements for assembly verification of large, complex parts. Real time X-ray will be implemented for inspection of the ROLAND missile propulsion unit. Cost savings of \$1.1 million per 10,000 missiles could be realized.

Future efforts should be directed toward expanding the applicability of non-film radiography. Assembly of a real time system with sufficient resolution to inspect castings and welds should be accomplished expeditiously. Another possible application is evaluation of composites. Further development of non-film X-ray could lead to the elimination of most film-based radiography. The potential cost benefits of replacement of film radiography are so significant that development continuation is strongly recommended to achieve this objective.

8.0 BIBLIOGRAPHY

CSOBA, ILLES P. "MODULATION TRANSFER FUNCTION OF IMAGE TUBE LENSES", APPLIED OPTICS, VOL. 16, 1977, P. 2647

CUNNINGHAM, R. C., "UPDATE ON DIGITAL IMAGE PROCESSING", ELECTRO-OPTICAL SYSTEMS DESIGN VOL. 11, JULY 1979, P. 34

DAHLBERG, ROGER C. "REAL-TIME DIGITAL IMAGE PROCESSING III", PROC. OF THE SOCIETY OF PHOTO-OPTICAL INSTRUMENTATION ENGINEERS, VOL. 207, 1979, P. 104

DAHLBERG, ROGER C. "DIGITAL IMAGE PROCESSOR LINKS TV SIGNAL SOURCES TO COMPUTER", COMPUTER DESIGN, OCT. 1977, P. 115

DAVIS, GORDON B. "AN INTRODUCTION TO ELECTRONIC COMPUTERS", MCGRAW-HILL, NEW YORK, 1965

DYALL, WILLIAM T., "WHERE DO THE LITTLE PIXELS AND BITS COME FROM?", ELECTRO-OPTICAL SYSTEMS DESIGN, VOL. 12, JULY 1980, P. 31

GONZALEZ, RAFAEL C. "DIGITAL IMAGE PROCESSING", ADDISON-WESLEY, READING, 1977

HALMSHAW, R. "RESEARCH TECHNIQUES IN NONDESTRUCTIVE TESTING", ACADEMIC PRESS, LONDON, 1970, CH. 8, P. 241

JANNEY, D. H. "DIGITAL ENHANCEMENT OF VERY LOW CONTRAST RADIOGRAPHS", MATERIALS EVALUATION, VOL. 34, AUG. 1976, P. 182

LYONS, ESMOND C., JR. "NEW DIMENSIONS IN EARTH RESOURCES STUDIES", DIGITAL DESIGN, JUNE 1978, P. 38

MORIKAWA, YOSHITAKA AND HAMADA, HIROSHI, "A TV TYPE OF ANALOGUE IMAGE PROCESSOR", OPTICA ACTA, VOL. 27, 1980 P. 1411

PRATT, WILLIAM K. "DIGITAL IMAGE PROCESSING", JOHN WILEY & SONS, NEW YORK, 1978

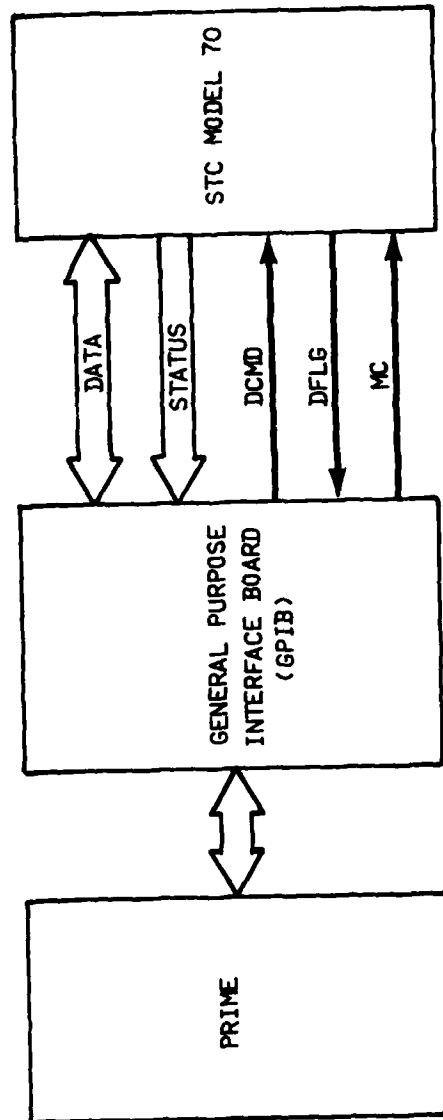
SCHLOSSER, M. S. "RADIOGRAPHIC ENHANCEMENT AND ANALYSIS SYSTEM", MATERIALS EVALUATION, NO. 1977, P. 16

TAYLOR, C. J., PULLAN, B. R. "RESEARCH TECHNIQUES IN NONDESTRUCTIVE TESTING", ACADEMIC PRESS, LONDON, 1973, CH. 3, P. 65

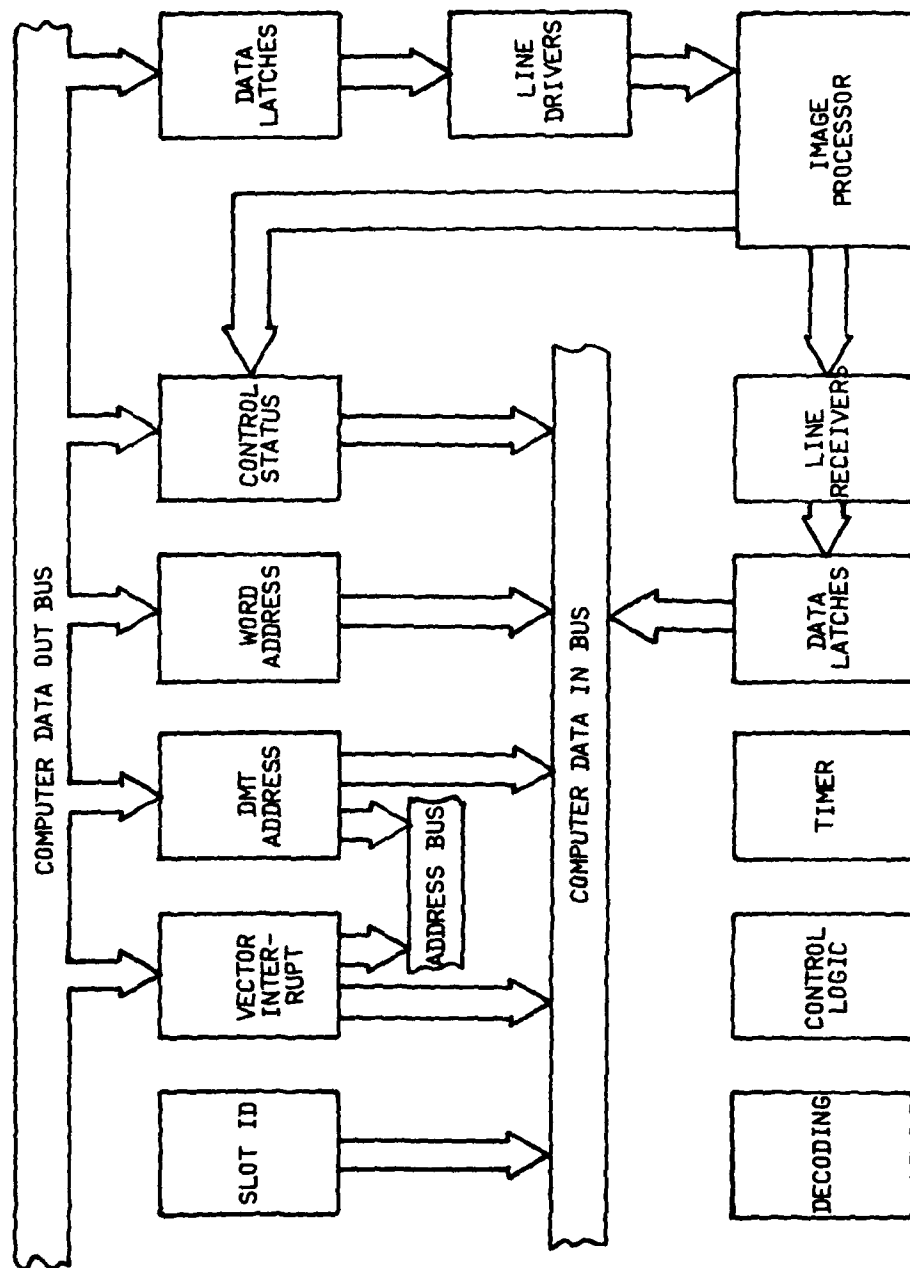
APPENDIX 1

STC M70 I/O SCHEMATICS

MODEL 70 CONTROL SYSTEM



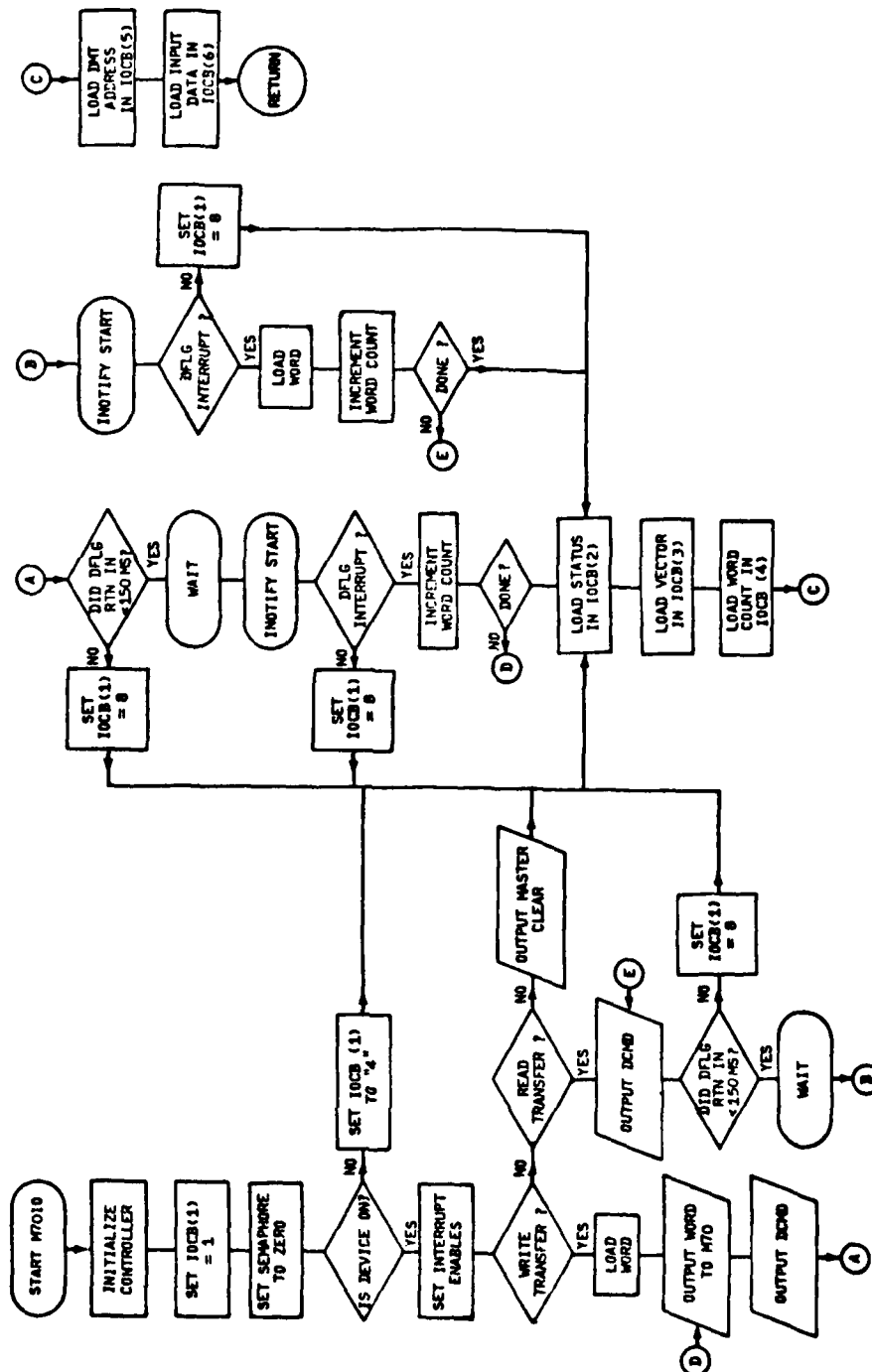
GENERAL PURPOSE INTERFACE BOARD (GPIB)



APPENDIX 2

IMAGE PROCESSOR SOFTWARE, LOGIC AND CODE

M7010 - COMPUTER-PROCESSOR COMMUNICATION



* M70IO: MODEL 70 IO DRIVER

```

*****
*                                     *
*      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
*      VERSION 1.0      JUNE 1,1980                      *
*      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
*                                     *
*****

```

* M70IO 11FEB1980 11:00
*CALL M70IO(IOCB,FUNC,BUFFER,COUNT)

* IOCB IS AN INTEGER ARRAY, WHERE :

* IOCB(1) = AN INTEGER TO RETURN TO CALLING PROGRAM

```

*      IOCB(1)=1  IS A GOOD RETURN
*      IOCB(1)=2  IS NO READY SIGNAL ON GPIB
*      IOCB(1)=4  IS NO 5 VOLTS FROM MODEL 70
*      IOCB(1)=8  IS INTERRUPT HAS OCCURRED

```

* IOCB(2) = THE INTERRUPT ENABLES

```

*      BIT 1=
*      BIT 2=
*      BIT 3=
*      BIT 4=
*      BIT 5=
*      BIT 6=
*      BIT 7=
*      BIT 8=
*
*      BIT 9=  DONE
*      BIT 10= BUTTONPRESSED
*      BIT 11= CURSOR MOVED
*      BIT 12= VERTICAL INTERVAL
*      BIT 13= THING COUNT=0
*      BIT 14= ATTENTION
*      BIT 15= TIMEOUT
*      BIT 16= +5VOLTS FROM M70

```

```

*      IOCB(3) = VECTOR ADDRESS
*      IOCB(4) = WORD COUNT
*      IOCB(5) = DMT ADDRESS
*      IOCB(6) =

```

* FUNC IS THE OPERATION TO BE PERFORMED

```

*      FUNC      = 0 IMPLIES WRITE  (COMPUTER --> M70)
*      FUNC      = 1 IMPLIES READ   (M70 -->COMPUTER )
*      FUNC      = 2 IMPLIES M70 MASTER CLEAR

```

* BUFFER IS AN INTERGER ARRAY THAT EITHER CONTAINS THE HEADER TO BE
* TRANSFERRED OR CONTAINS THE DATA TO OR FROM THE M70

* COUNT IS AN INTEGER REPRESENTING THE NUMBER OF WORDS TO BE SENT OR
* RECEIVED BY THE M70

```

*
*      ENT  M70IO,M70ECB
*      EXT  SP2SEM
*      SEG
*      D64V
M70IO  EQU  *
*      ARG1

```

* INITIALIZE THE CONTROLLER

* OCF '1760 CLEAR ALL THE REGISTERS ON THE BOARD

```

*
*
*      OCF  '260      ENABLE INTERRUPTS
*LOAD THE INTERRUPT VECTOR
*SP2INT  XAC  SP2INT      GET ADDRESS OF INTERRUPT IN SEG4
*      LDA  SP2INT
*      OTA  '1660      OUTPUT IT TO VECTOR ADDRESS REGISTER
*      BCNE ERR
*      LD  =0
*      STL  SP2SEM
*SET IOCB(1) EQUAL TO 1
*
*      LT
*      STA  IOCB,*1
*CHECK IF DEVICE IS ON
*
*      SKS  '60
*      BCNE DOFF      BRANCH TO DOFF IF DEVICE HAS NO +5 VOLTS
*SET UP INTERRUPT ENABLES
*
*      LDX  =1
*      LDA  IOCB,*1      LOAD IOCB(2)
*      OTA  '560      OUTPUT IT TO CONTROL/STATUS REG
*      BCNE ERR
*      STA  CSREG
*DETERMINE IF READ OR WRITE (IF NEITHER THEN MASTER CLEAR)
*
*      LDA  FUNC,*      LOAD FUNC
*      STA  FNTN
*      BEQ  WRIT      BRANCH TO WRIT IF A = 0
*      ARS  1
*      BEQ  READ      BRANCH TO READ IF A = 1
*      JMP  MC      OTHERWISE JUMP TO MASTER CLEAR
*WRITE OPERATION (COMPUTER TO M70)
*WRIT  LDA  COUNT,*      LOAD NUMBER OF WORDS TO TRANSFER
*      STA  INCT
*      TCA
*      STA  WDCTM      2'S COMPLEMENT
*      OTA  '360      OUTPUT TO WC REGISTER
*      BCNE ERR
*      LDX  =0      LOAD INDEX REGISTER
*WLOOP STX  XSAV
*      LDA  BUFFER,*1      LOAD NEXT WORD IN BUFFER
*      INH
*      OTA  '1060      OUTPUT DATA
*      BCNE ERR
*      WAIT SP2SEM      SUSPEND
*      SKS  '1060      SKIP IF DFLG IS HIGH
*      BCNE INTERR
*      LDX  XSAV
*      LDA  WDCTM
*      A1A      ADD 1 TO WDCTM
*      STA  WDCTM
*      IRX      INCREMENT INDEX
*      STX  XSAV
*      BLT  WLOOP      BRANCH IF WDCTM < 0
*      JMP  DONE
*READ OPERATION (M70 TO COMPUTER)
*READ  LDA  COUNT,*      LOAD NUMBER OF WORDS TO READ
*      STA  INCT
*      TCA      2'S COMPLEMENT
*      STA  WDCTM
*      OTA  '360      OUTPUT TO WORD COUNT REGISTER

```

```

BCNE ERR
LDX =0          SET INDEX =0
STX XSAV
EQU *
RLOOP INH
OCF '1060      SEND DCMD
WAIT SP2SEM    SUSPEND
SKS '1060      SKIP IF DFLG IS HIGH
BCNE INTERR    SOMETHING ELSE CAUSED INTERRUPT
INA '1060      INPUT DATA
BCNE ERR
LDX XSAV
STA BUFFER,*1  STORE IT IN BUFFER
LDA WDCTM
AIA           ADD 1 TO WDCTM
STA WDCTM
IRX           INCREMENT INDEX
STX XSAV
BLT RLOOP     BRANCH IF WDCTM < 0
JMP DONE

```

*MASTER CLEAR OPERATION

```

MC EQU *
OCF '1160      OUTPUT MASTER CLEAR TO M70
JMP DONE

```

*DONE ROUTINE

```

DONE EQU *
OCF '460      DISABLE INTERRUPTS
LDX =1
INA '560      INPUT STATUS REGISTER
BCNE ERR
STA IOCB,*1    IOCB(2)
IRX
INA '1560     INPUT VECTOR ADDRESS
BCNE ERR
STA IOCB,*1    IOCB(3)
IRX
INA '360      INPUT WORD COUNT
BCNE ERR
TCA           2'S COMPLEMENT
STA IOCB,*1    IOCB(4)
IRX
INA '160      INPUT DMT ADDRESS
BCNE ERR
STA IOCB,*1    IOCB(5)
IRX
INA '1060     INPUT INPUT DATA
BCNE ERR
STA IOCB,*1    IOCB(6)

```

```

*
*
PRTN
DOFF EQU *      DEVICE NOT ON!!!!!!
LDX =0
LDA =4          SET FLAG TO 4
STA IOCB,*1     PUT IT IN IOCB(1)
JMP DONE

```

*THIS ERROR MEANS NO READY SIGNAL ON GPIB FOR INA OR OTA

```

ERR EQU *
LDX =0
LDA =2          SET FLAG TO 2
STA IOCB,*1     PUT IT IN IOCB(1)
PRTN

```

*INTERRUPT ERROR ROUTINE

```

*
INTERR EQU *    AN INTERRUPT HAS OCCURED
LDX =0

```

```

        LDA    =8      SET FLAG TO 8
        STA    IOCB,*1  PUT IT IN IOCB(1)
        JMP    DONE

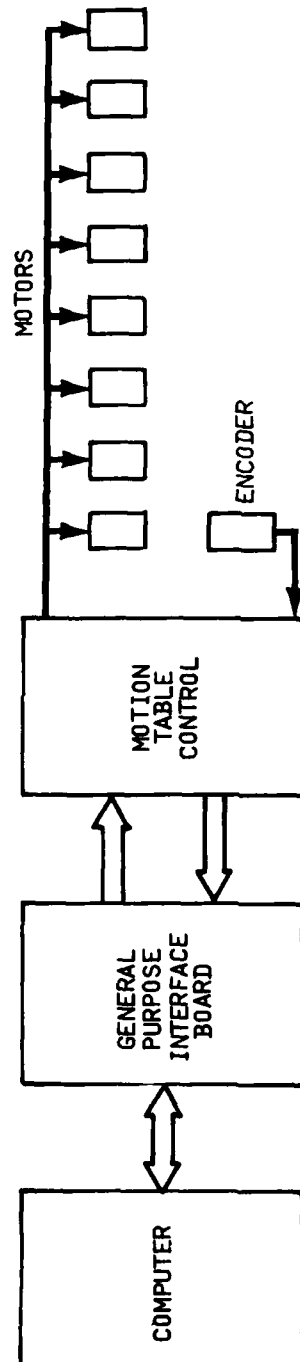
*DATA DEFINITION
*
WDCTM      0           TWOS COMPLEMENT OF COUNT
CSREG      0
FNIN       0
INCT       0
XSAV       0
        DYNM  IOCB(3),FUNC(3),BUFFER(3),COUNT(3)
*
M70ECB     LINK
           ECB  M70IO,,IOCB,4
           END

```

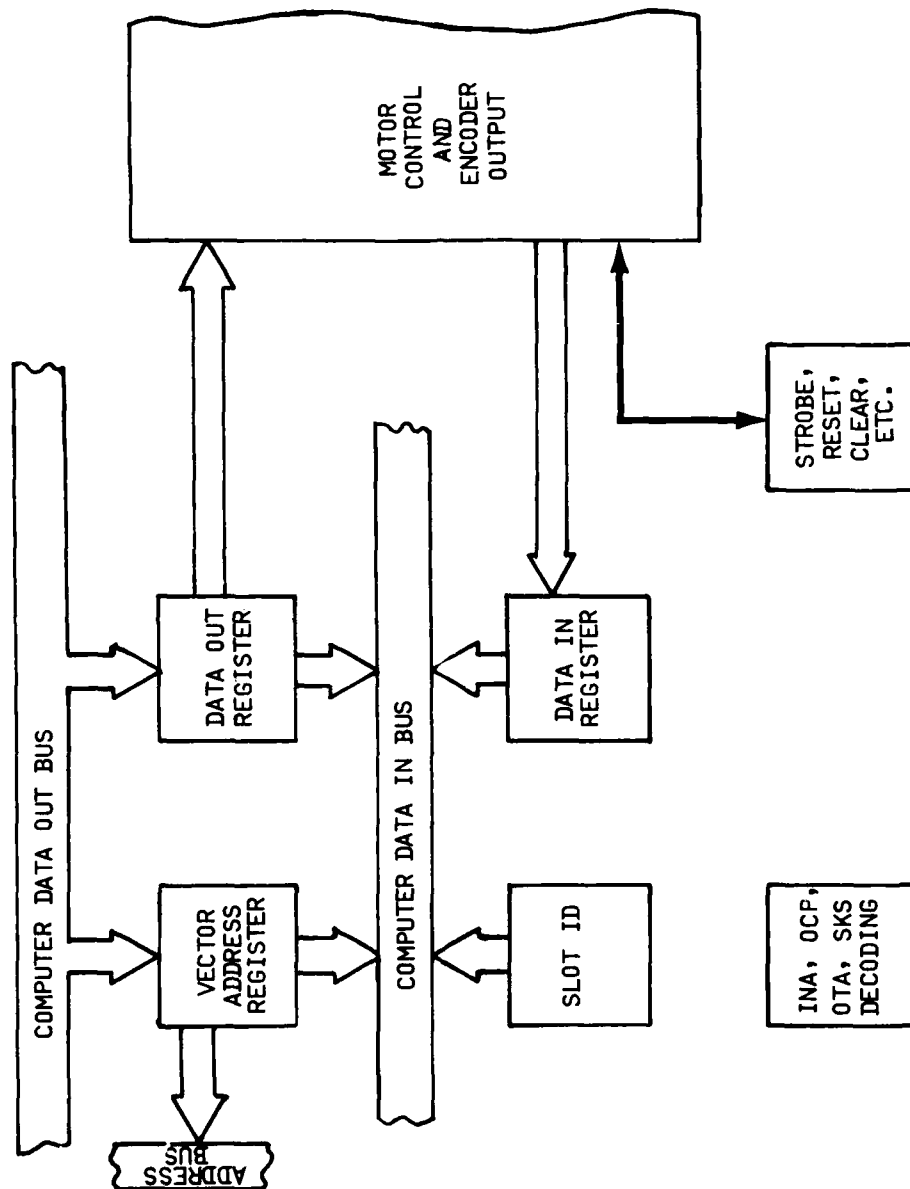
APPENDIX 3

MOTOR CONTROL SCHEMATICS

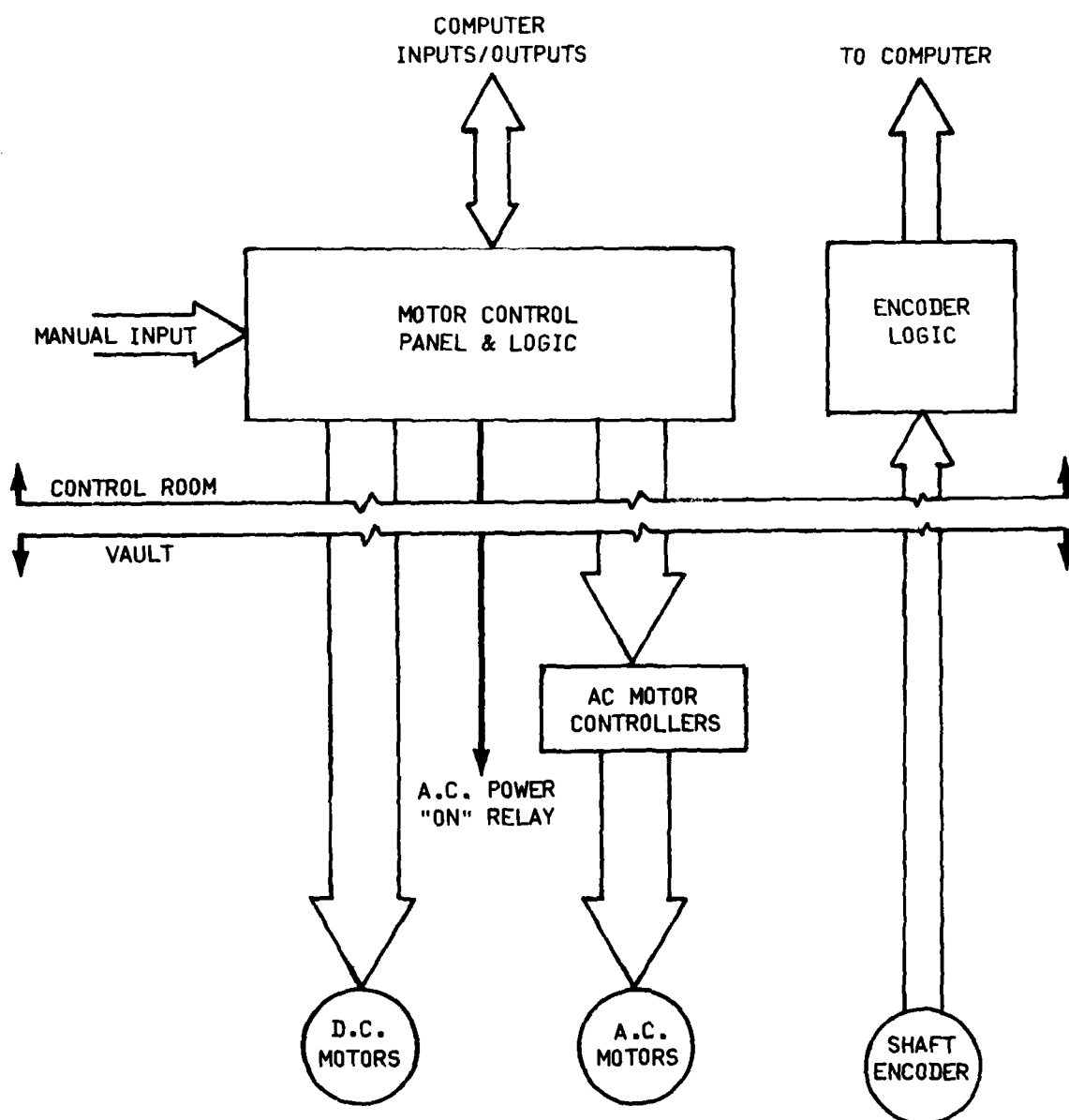
MOTION CONTROL SYSTEM



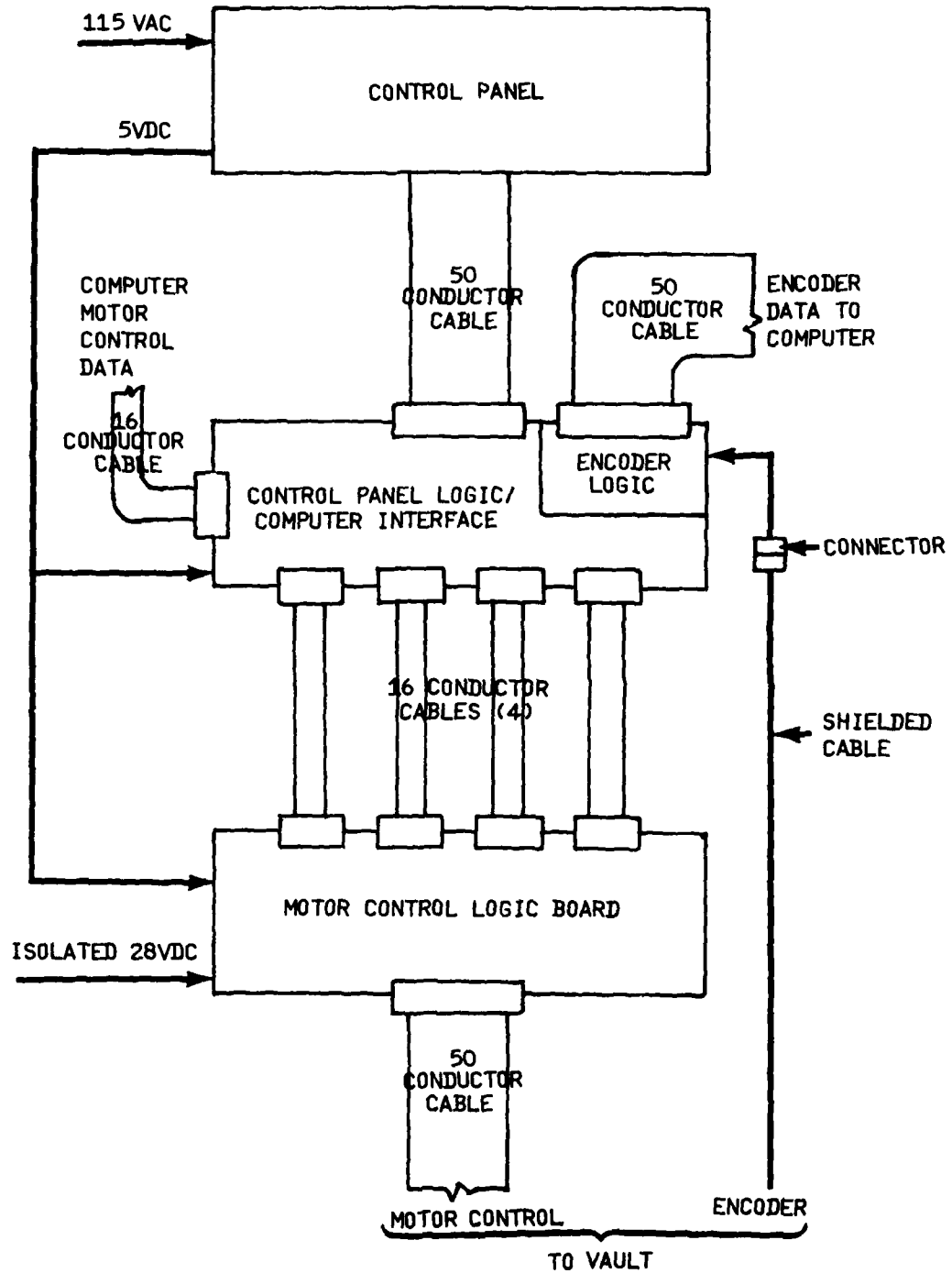
MOTION CONTROL INTERFACE BOARD



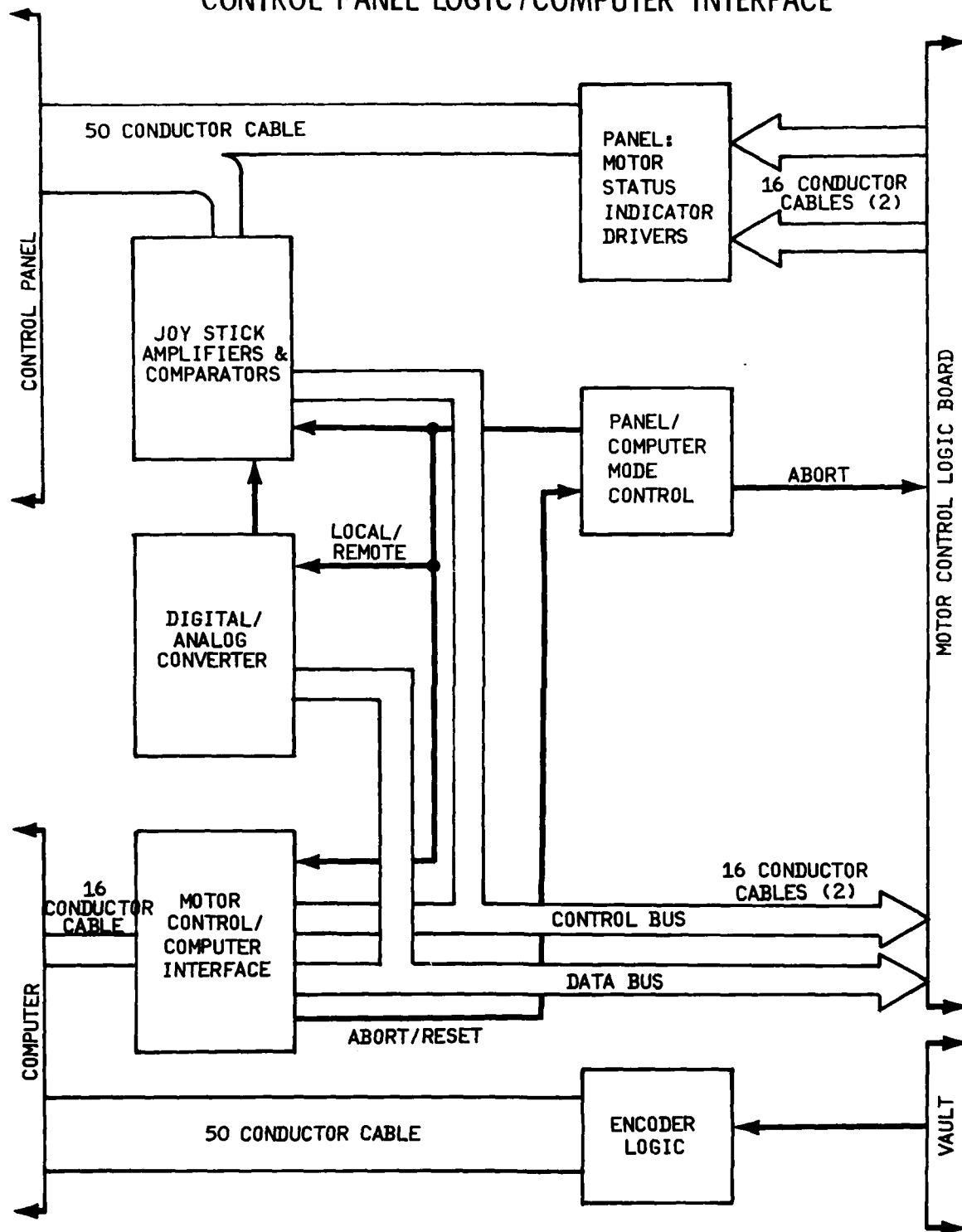
MOTOR CONTROL SYSTEM



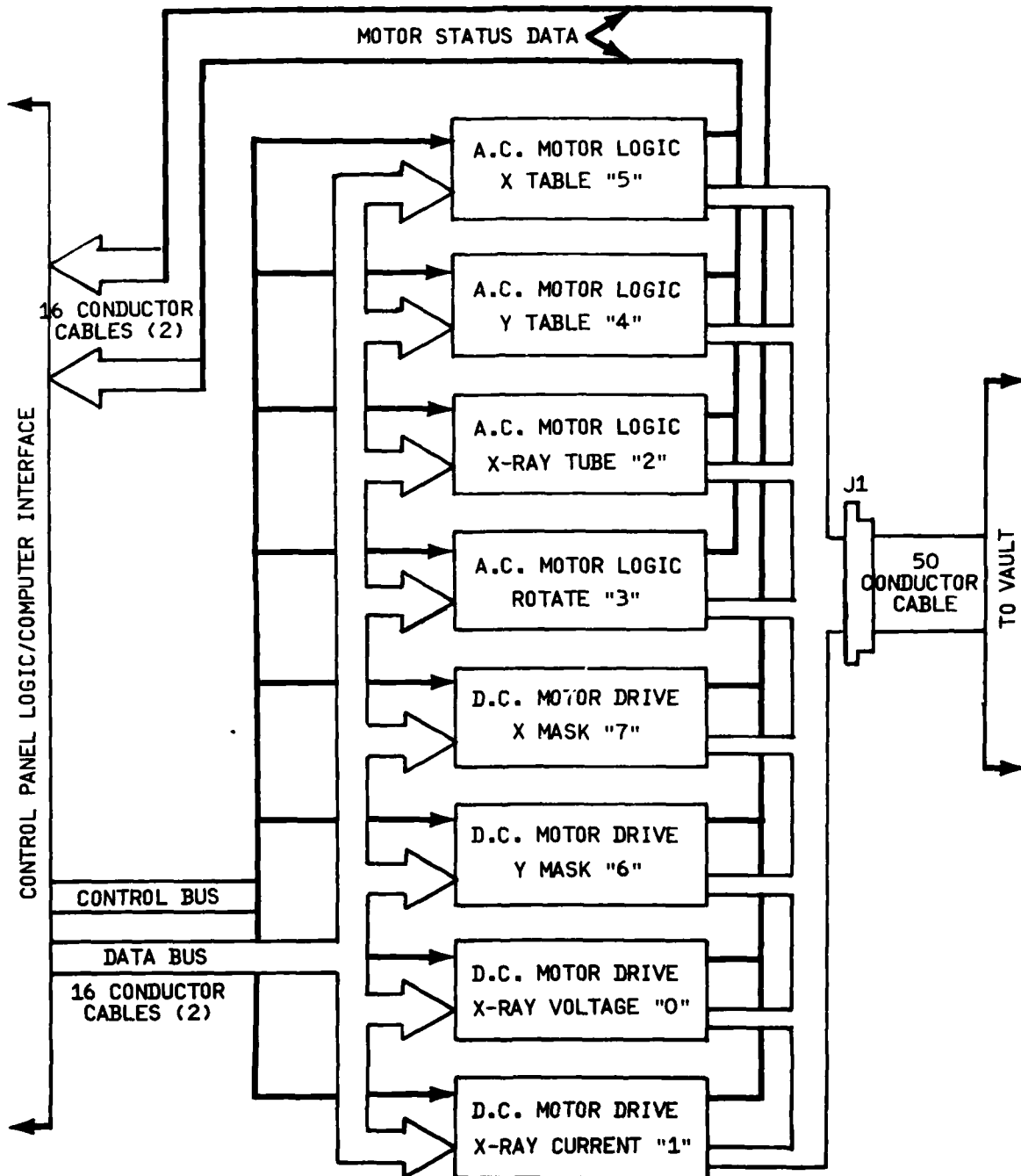
MOTOR CONTROL PANEL & LOGIC



CONTROL PANEL LOGIC/COMPUTER INTERFACE



MOTOR CONTROL LOGIC BOARD

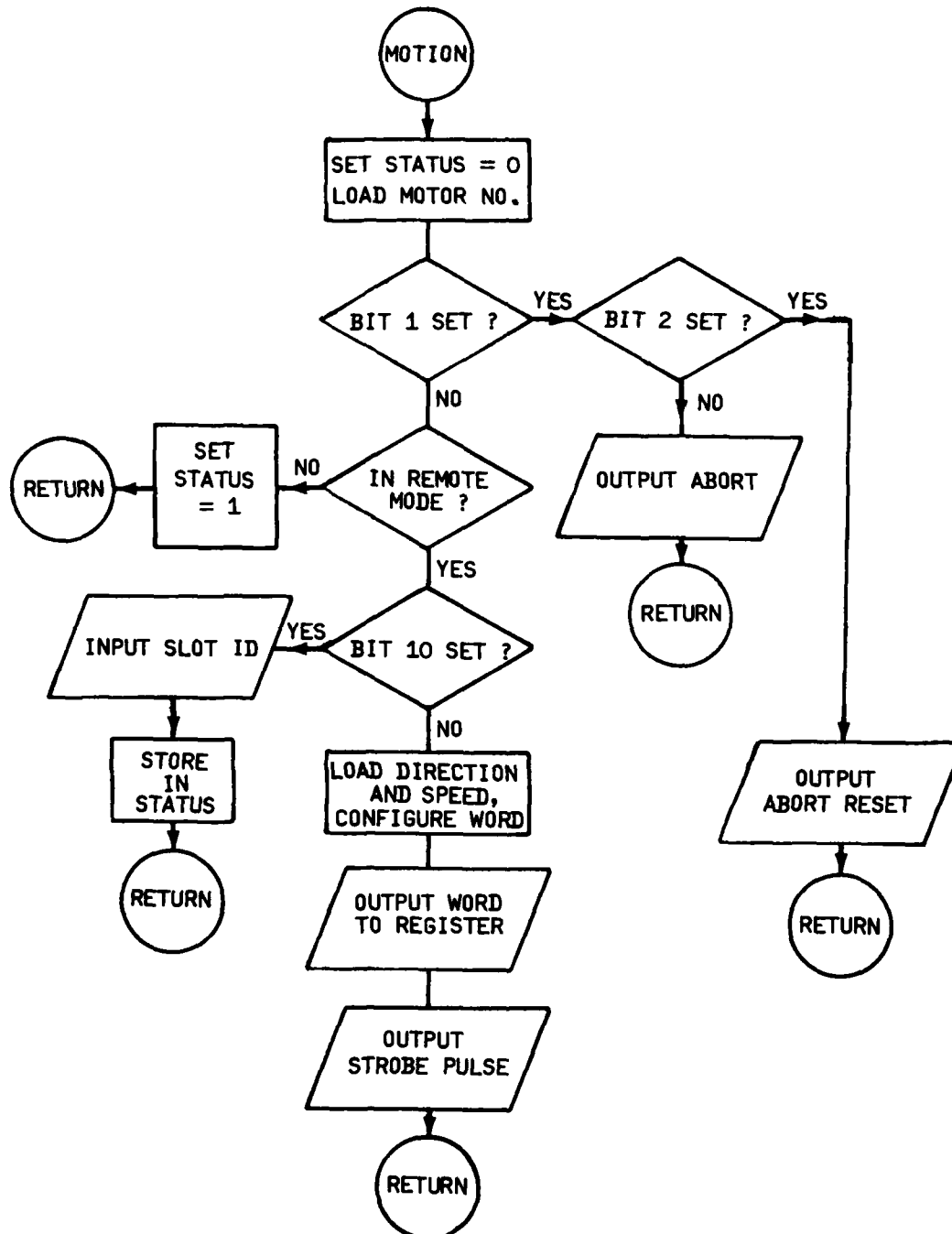


A3-7

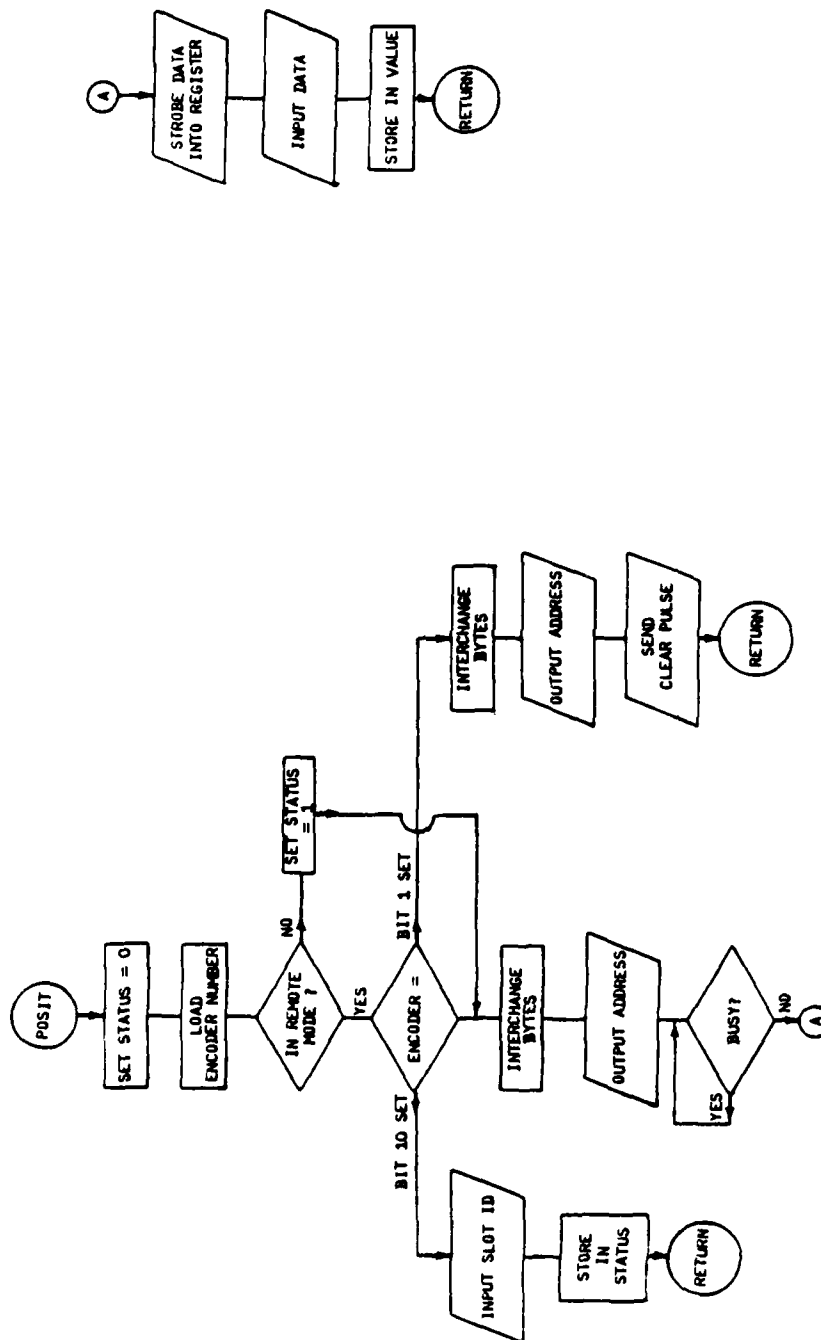
APPENDIX 4

MOTOR CONTROL SOFTWARE, LOGIC AND CODE

MOTION - MOTOR(S) CONTROL



POSIT - READ ENCODER POSITION



* MOTION: ENTRY TO MOTOR DRIVER IN SYSTEM PAGE 0001

* MOTION: ENTRY TO MOTOR DRIVER IN SYSTEM

```
*
* *****
* REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM
* VERSION 1.0 JUNE 1,1980
* BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY
* *****
* CALL MOTION(MOTORN,DIRECTION,SPEED,STATUS)
* MOTORN= 0 TO 7
* IF BIT 1 SET THEN ISSUE ABORT
* IF BIT 1 AND BIT 2 SET THEN ISSUE ABORT RESET
* :100 IS SLOT ID INQUIRY
*
* DIRECTION
* 0 =ONE WAY
* 1 =THE OTHER WAY
*
* SPEED
* 0 THRU :177 (MUST BE >= :20 FOR MOTOR TO GO ON)
*
* STATUS
* 0= OK
* 1= NOT IN REMOTE
* 2= NO READY SIGNAL ON GPIB
*
* *CALLED BY: POSINT, MOTOR
* *FUNCTION: MOVE A MOTOR
* *WRITTEN IN ASSEMBLY LANGUAGE
*
* DYNT MOTION
* SEG
* END
```

* POSIT: ENTRY FOR ENCODER READOUT

* CALL POSIT(ENCODERN,VALUE,STATUS) 29MAY1980

```

*      ENCODERN#      0 TO 7 FOR THE MOTORS
*
*      '100 IS REQUEST FOR SLOT ID, WHICH IS RETURNED
*      IN STATUS
*
*      IF BIT 1 (MSB) IS SET THEN CLEAR THE ENCODER
*
*      VALUE=         THE RETURNED VALUE FROM THE ENCODER
*                      (UNLESS A REQUEST WAS MADE FOR SLOT ID)
*
*
*      STATUS=        0 = A GOOD READ
*
*                      1 = NOT IN REMOTE MODE
*
*                      2 = NO READY SIGNAL ON GPIB
*

```

```

*      *CALLED BY: POSINT AND PING AND MOTOR      *
*      *FUNCTION: READ POSITION OF ENCODER          *
*      *WRITTEN IN ASSEMBLY LANGUAGE              *

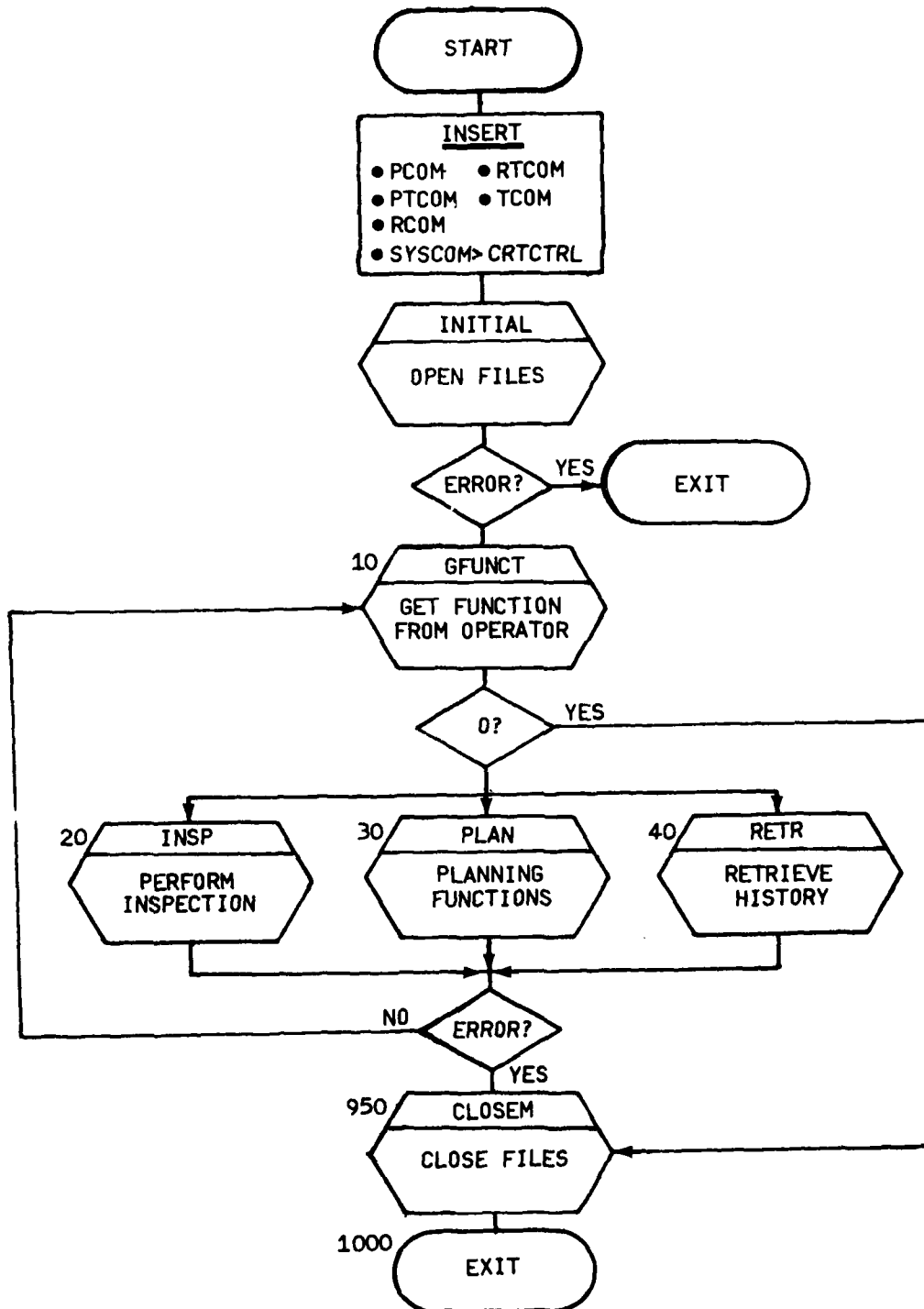
```

DYNT POSIT
SEG
END

APPENDIX 5

COMPUTER-AIDED INSPECTION SOFTWARE. LOGIC AND CODE

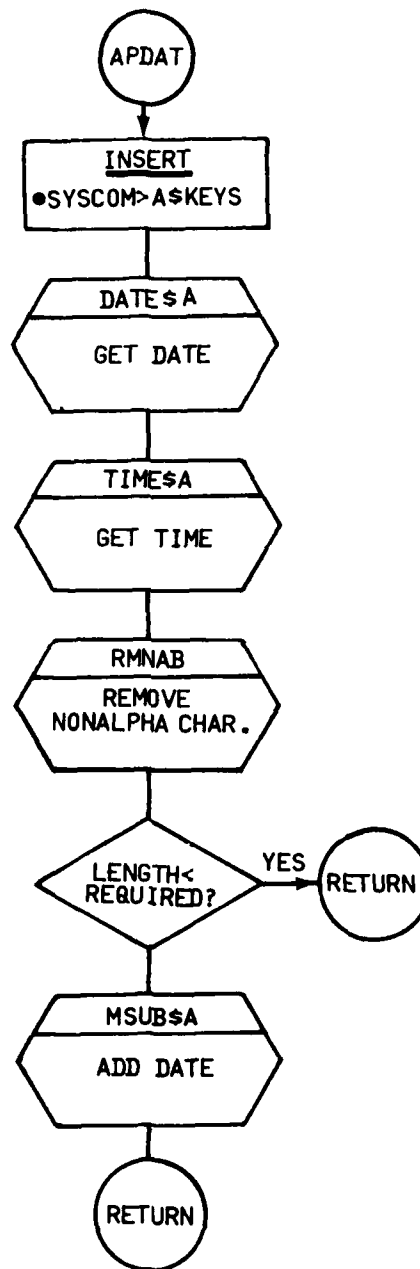
AI - AUTOMATED INSPECTION (TOP LEVEL)



APDAT - APPEND DATE TO KEYWORD

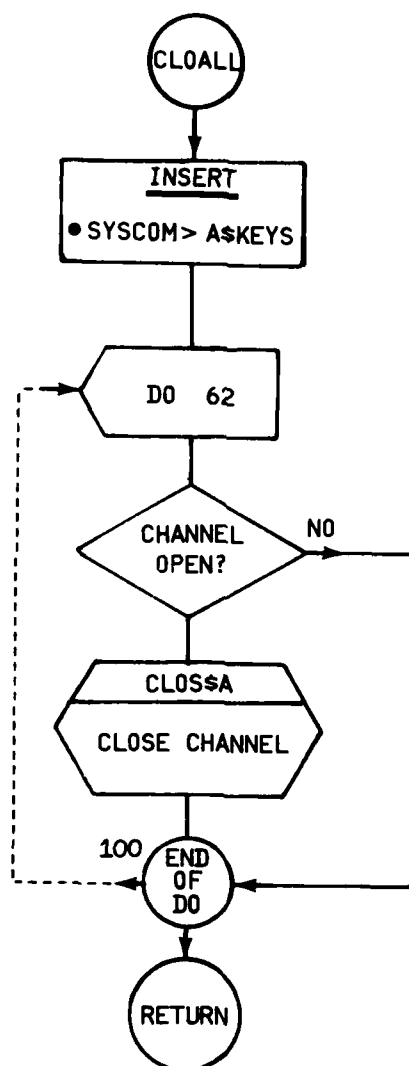
ARGUMENTS: 2

- STRING: KEY
- INTEGER: LENGTH



CLOALL - CLOSE ALL OPEN CHANNELS

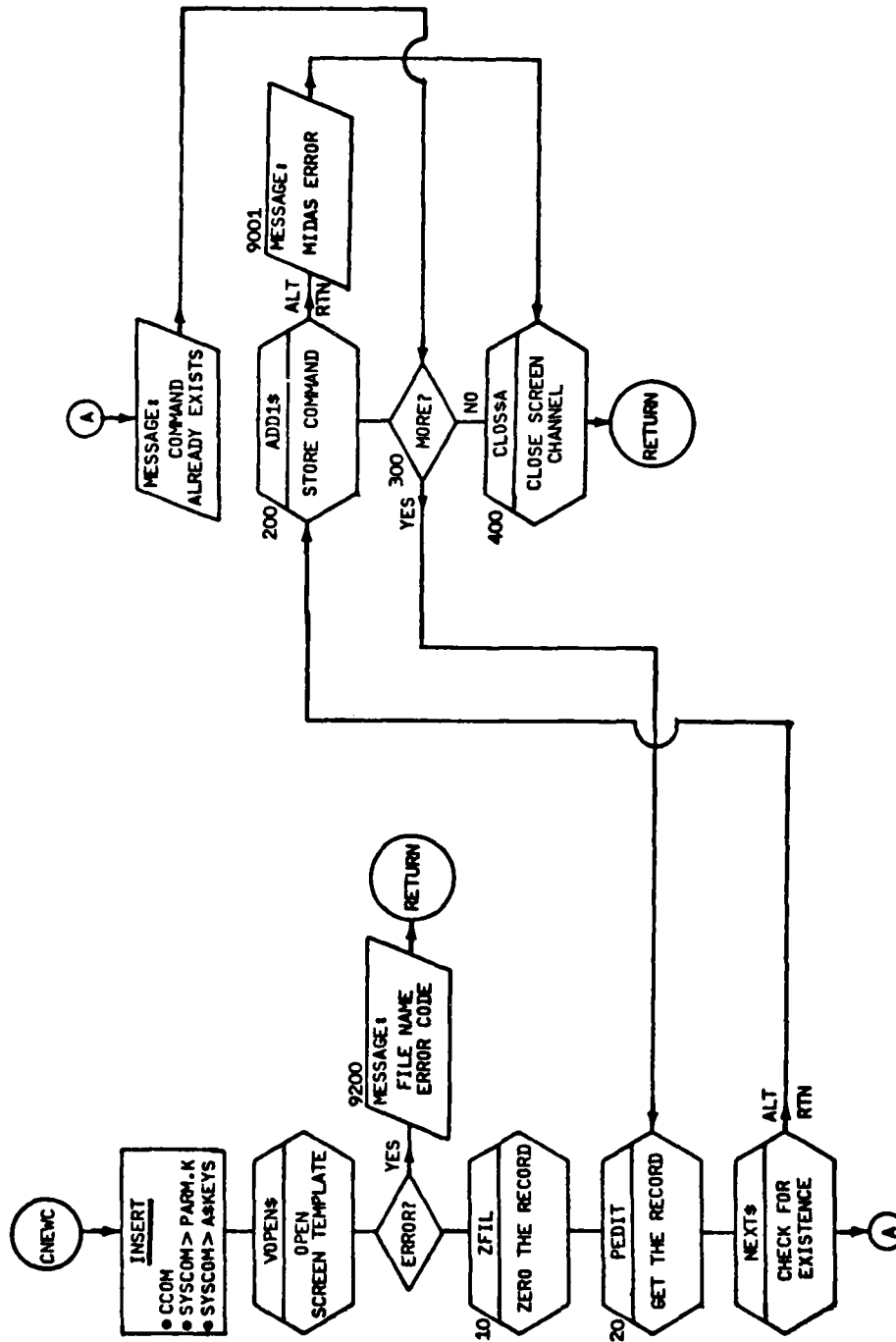
ARGUMENTS: NONE



CNEWC - CREATE NEW IMAGE PROCESSOR COMMAND

ARGUMENTS: 1

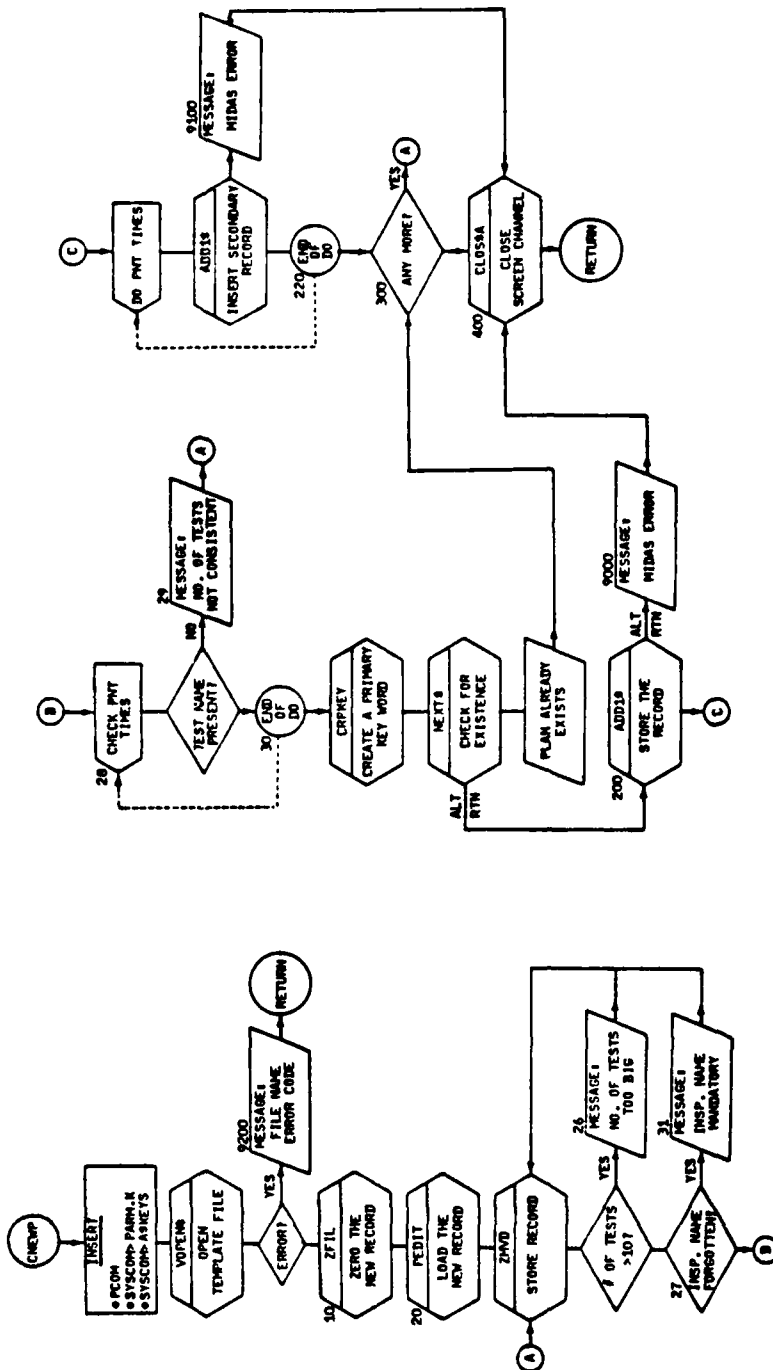
● INTEGER: ERROR



CNEWP - CREATE NEW PLAN

ARGUMENTS: 1

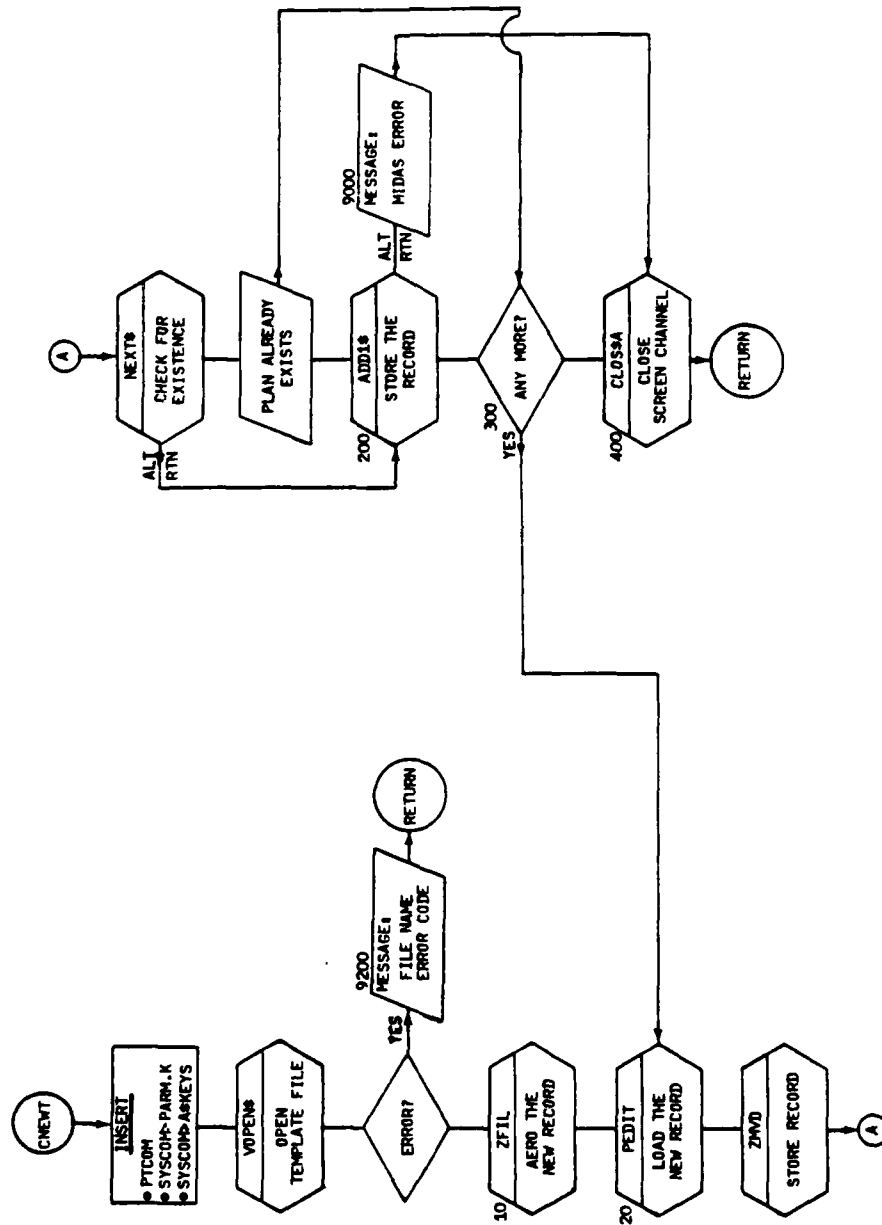
● INTEGER: ERROR



CNEWT - CREATE NEW TEST ENTRY

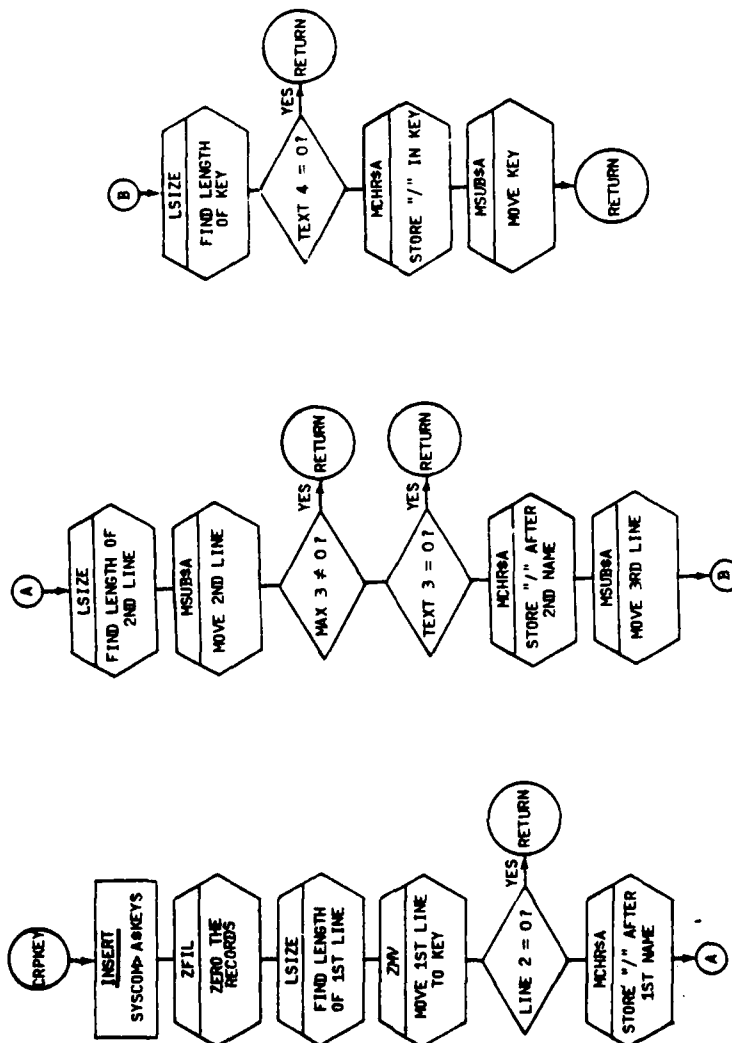
ARGUMENTS: 1

● INTEGER: ERROR



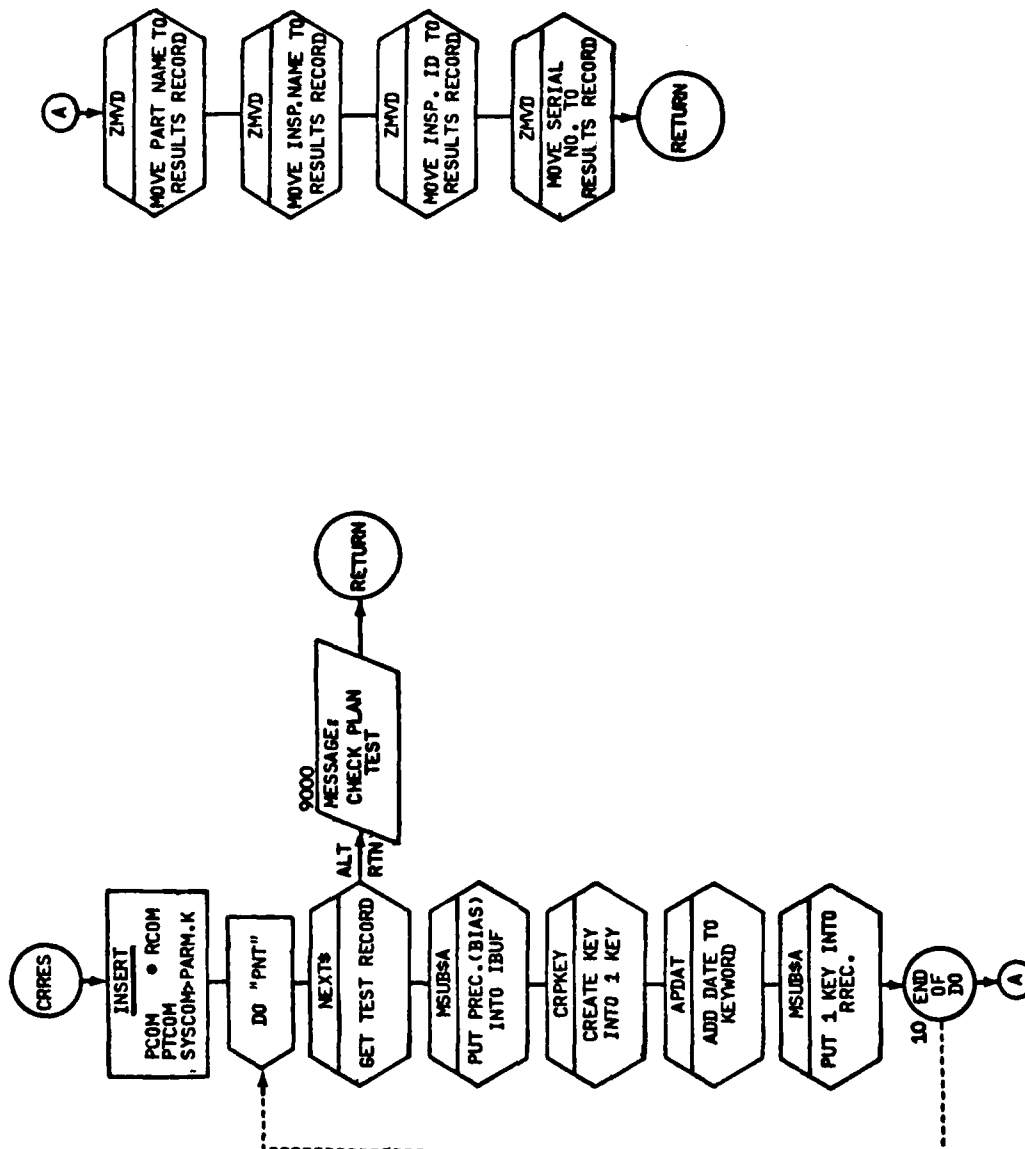
CRPKEY - CREATE PLAN KEYWORD

ARGUMENTS: 10
 ● STRING: 1ST LINE OF TEXT
 ● INTEGER: LENGTH OF 1ST LINE
 ● STRING: 2ND LINE OF TEXT
 ● INTEGER: LENGTH OF 2ND LINE
 ● STRING: 3RD LINE OF TEXT
 ● INTEGER: LENGTH OF 3RD LINE
 ● STRING: 4TH LINE OF TEXT
 ● INTEGER: LENGTH OF 4TH LINE
 ● STRING: KEYWORD
 ● INTEGER: LENGTH OF KEYWORD



CRRES - CREATE RESULTS RECORD

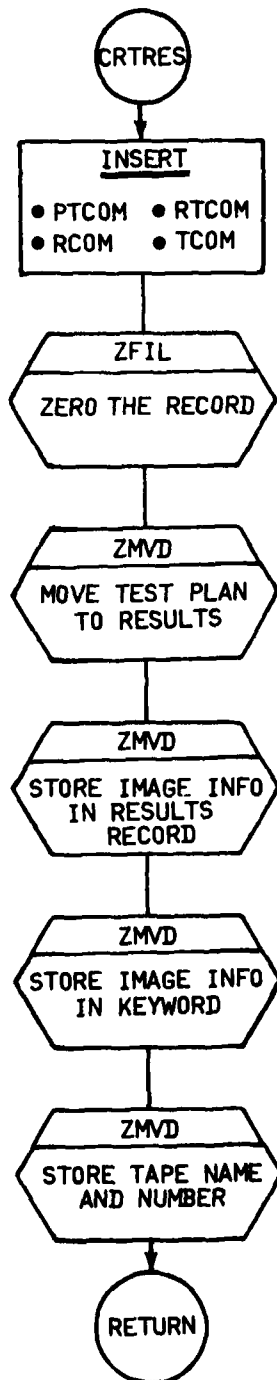
ARGUMENTS: 3
 ● STRING: SERIAL NUMBER
 ● STRING: INSPECTOR I.D.
 ● INTEGER: ERROR



CRTRES - CREATE A TEST RESULTS RECORD

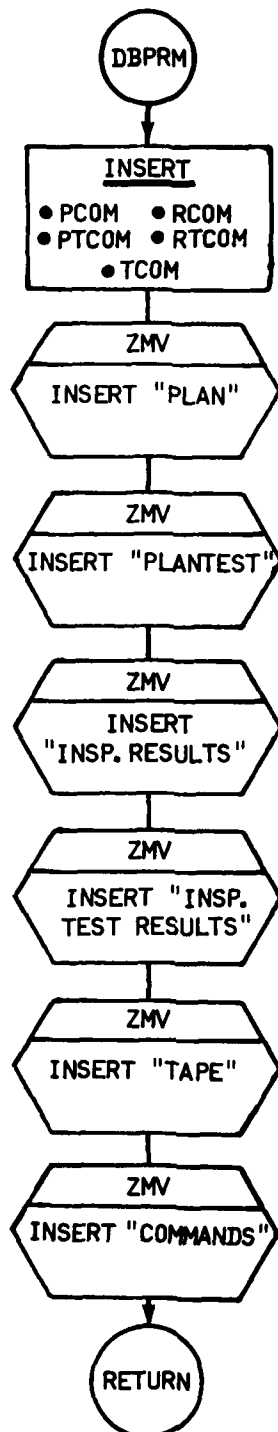
ARGUMENTS: 2

- INTEGER: TEST SEQUENCE NO.
- STRING: RESULTS KEYWORD



DBPRM - INSERT DATA BASE PARAMETERS

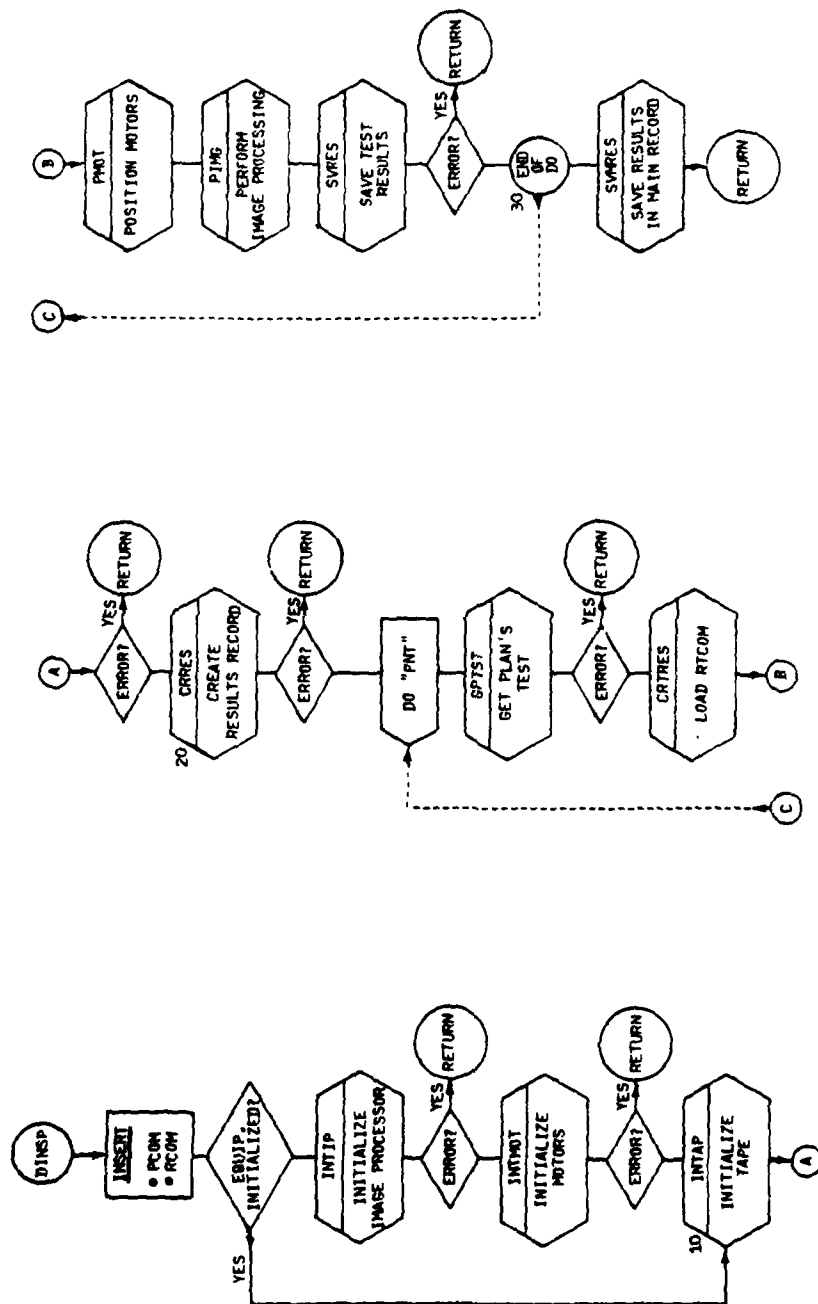
ARGUMENTS: NONE



DINSP - DO THE INSPECTION PER PLAN

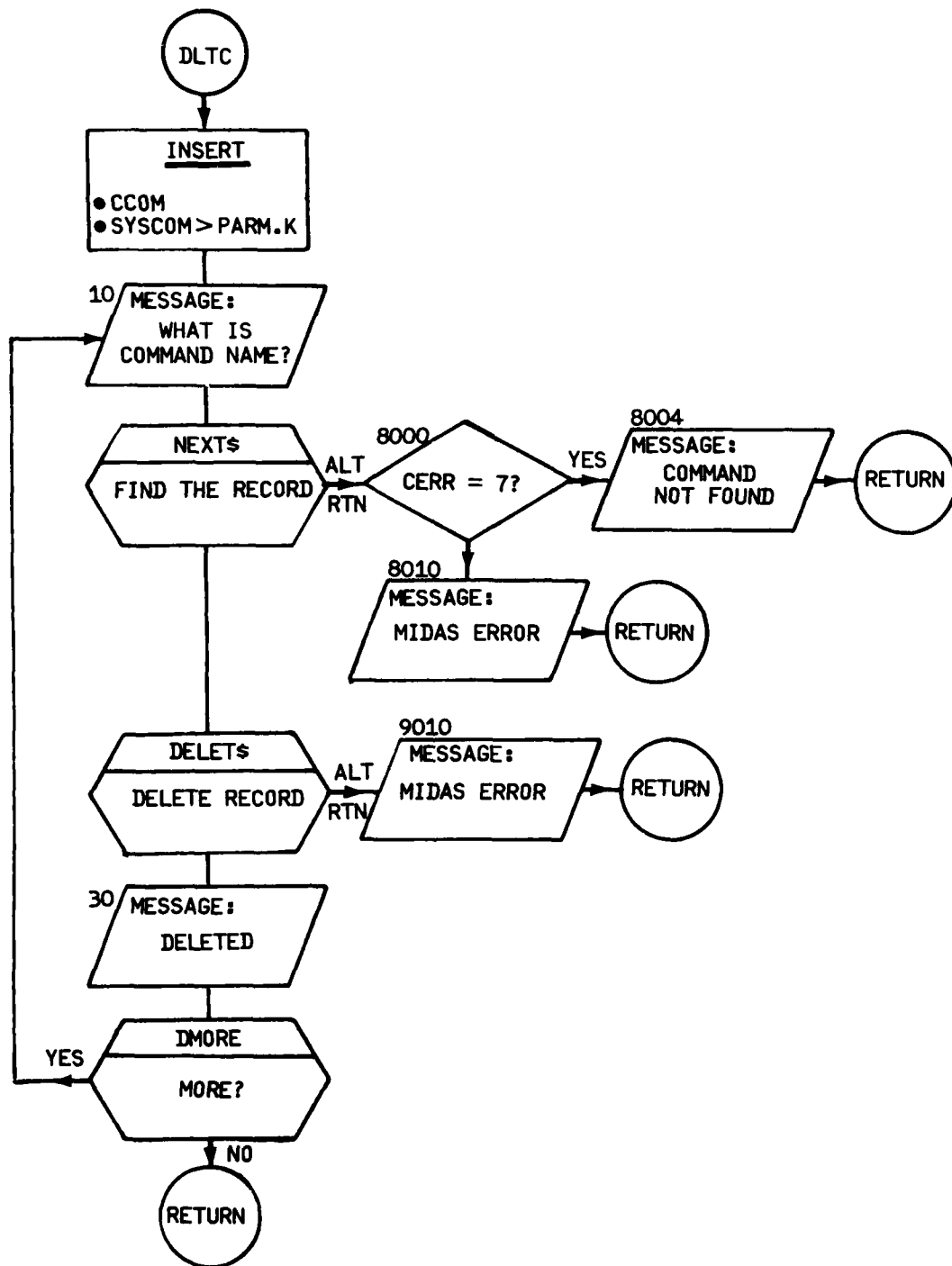
ARGUMENTS: 5

- STRING: INSPECTOR I.D.
- STRING: PART NUMBER
- STRING: TEST I.D.
- STRING: SERIAL NUMBER
- INTEGER: ERROR



DLTC - DELETE IMAGE PROCESSOR COMMAND

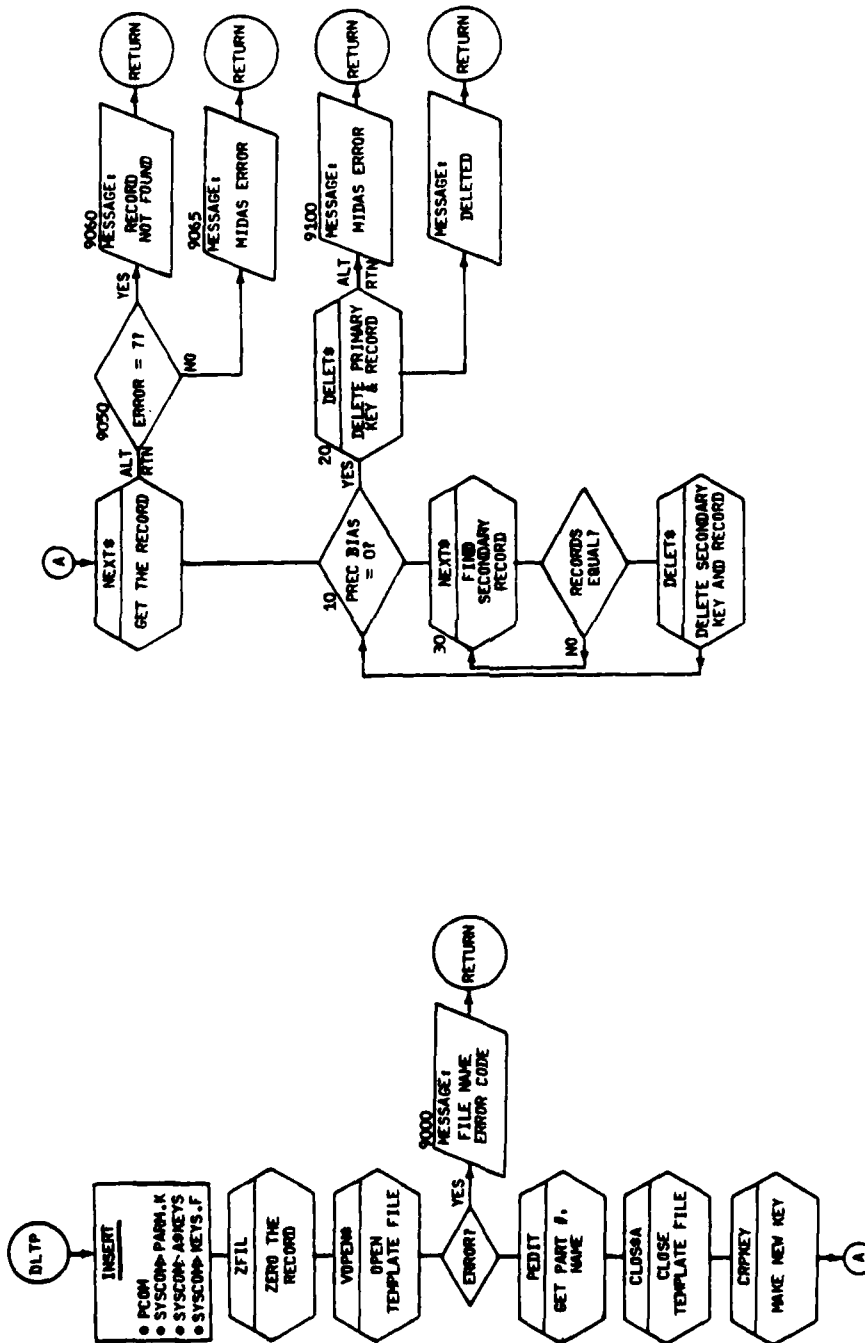
ARGUMENTS: 1
 • INTEGER: ERROR



DLTP - DELETE A PLAN

ARGUMENTS: 1

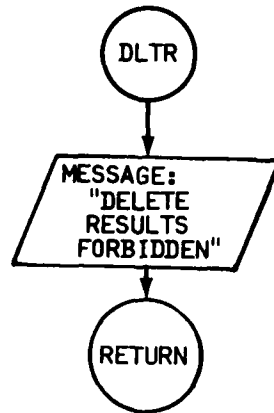
● INTEGER: ERROR



DLTR - DELETE RESULTS

ARGUMENTS: 1

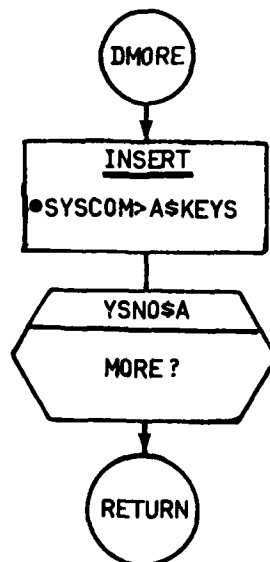
• INTEGER: ERROR



DMORE - DO YOU WANT TO DO MORE ?

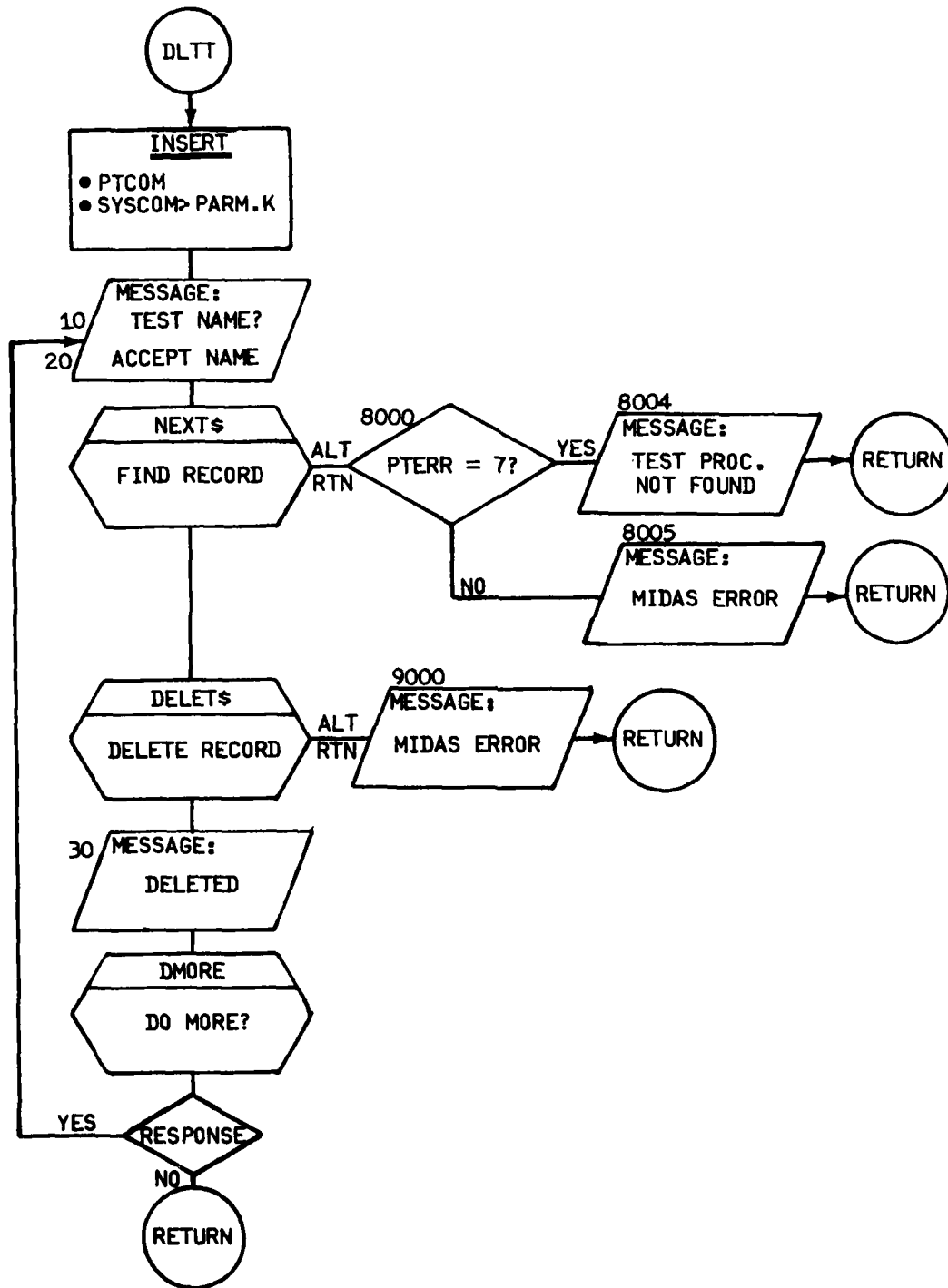
ARGUMENTS: 1

• LOGICAL: MORE



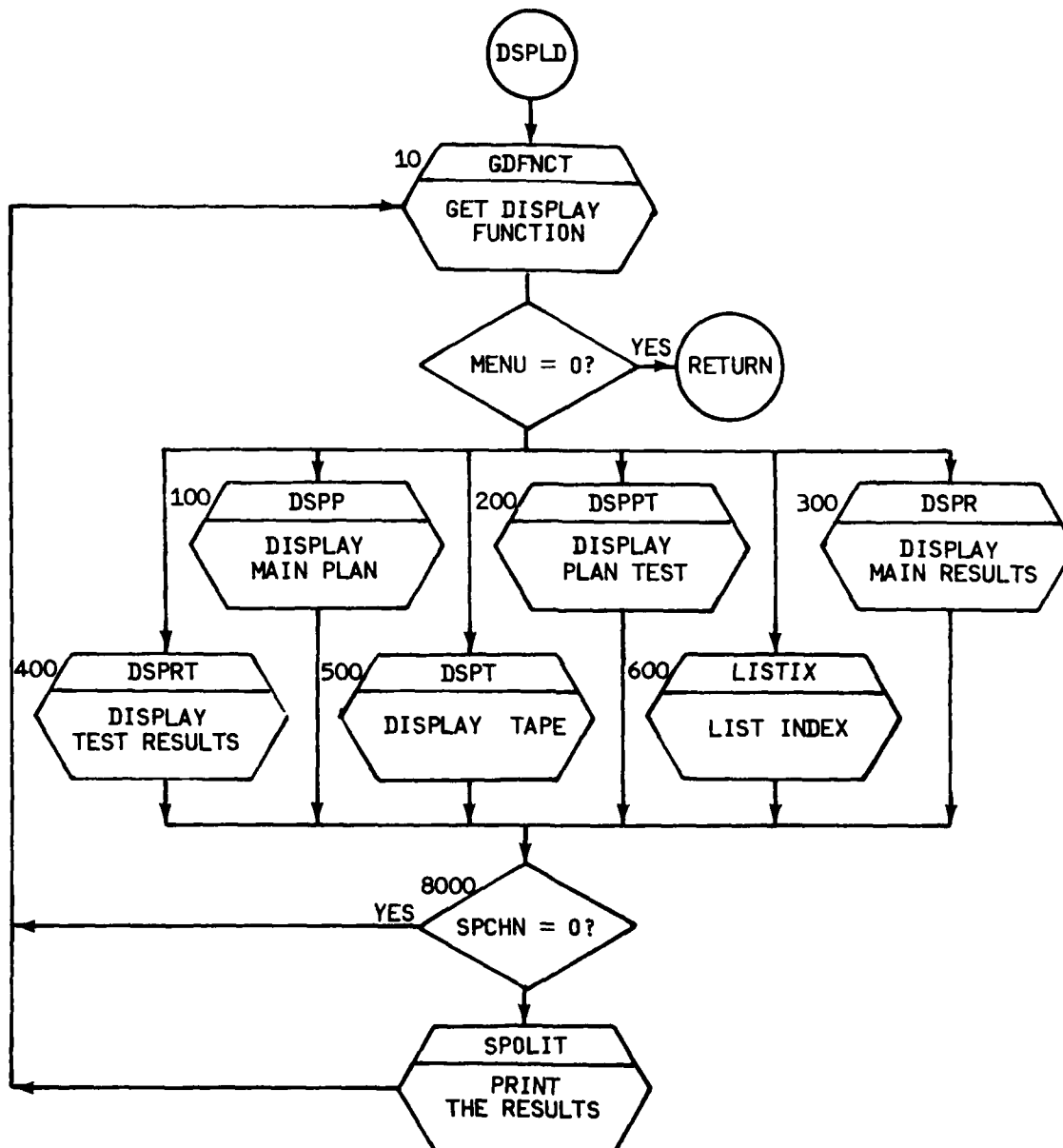
DLTT - DELETE TEST PLAN

ARGUMENTS: 1
 • INTEGER: ERROR



DSPLD - DISPLAY OR PRINT PLAN AND RESULT DATA

ARGUMENTS: 1
• INTEGER: ERROR



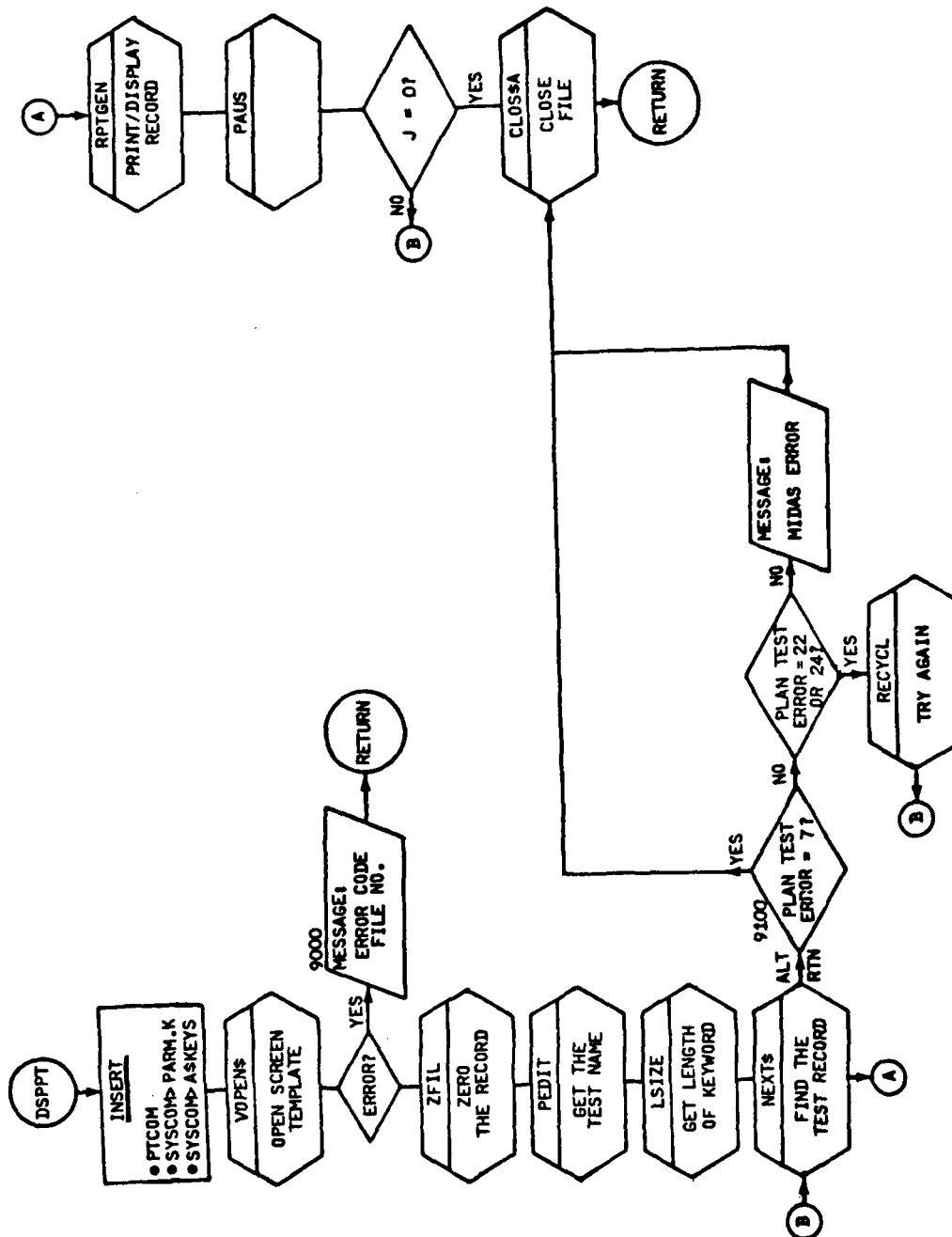
● INTEGER: REPORT CHANNEL



DSPPT - DISPLAY ONE TEST PLAN

ARGUMENTS: 1

● INTEGER: SPOOL CHANNEL



AD-A098 658

BOEING AEROSPACE CO SEATTLE WA
LOW COST HIGH VOLUME RADIOGRAPHIC INSPECTION.(U)

F/6 13/8

UNCLASSIFIED

JAN 81 N M LOWRY, H J ABPLANALP, J M TANKE
D180-26159-1

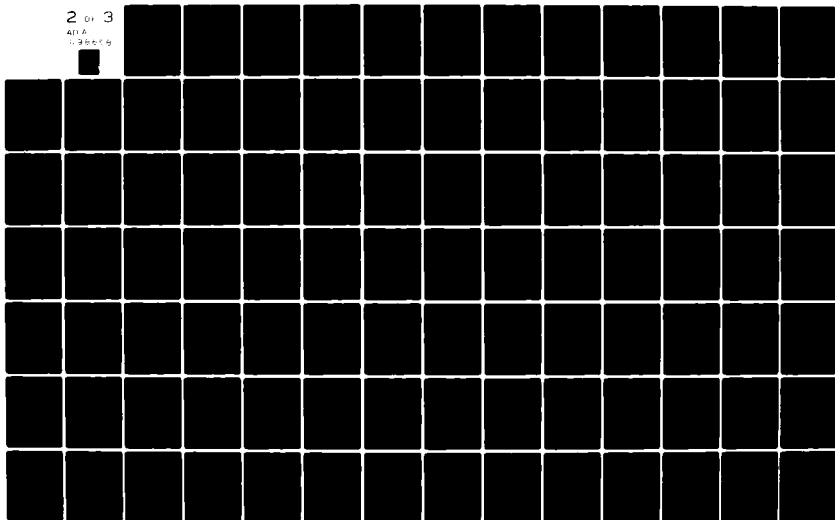
DAAK40-78-C-0197

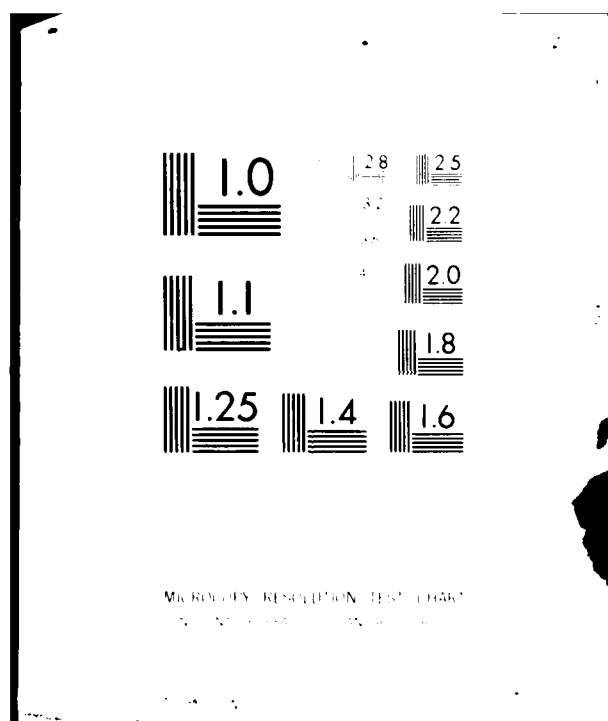
DRSMI/RS-CR-81-1

NL

2 of 3

411 A
0.00000

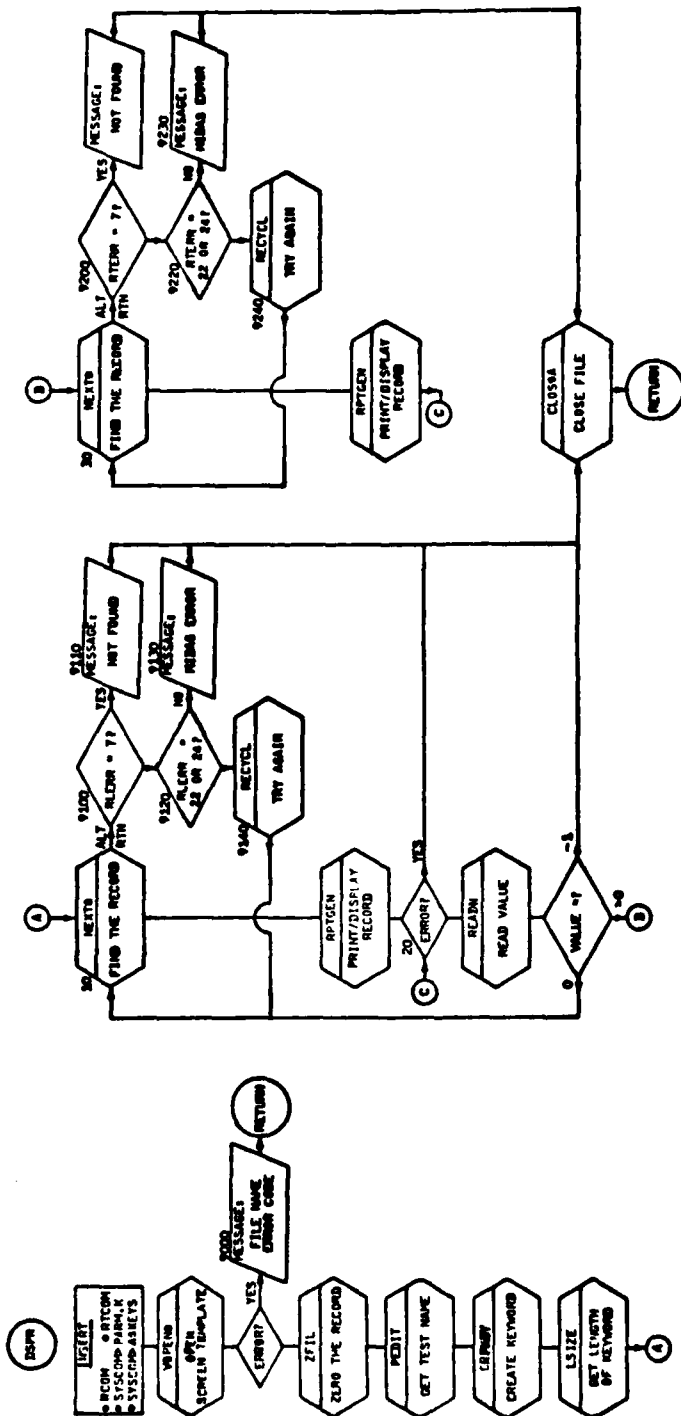




DSPR - DISPLAY RESULTS

ARGUMENTS: 1

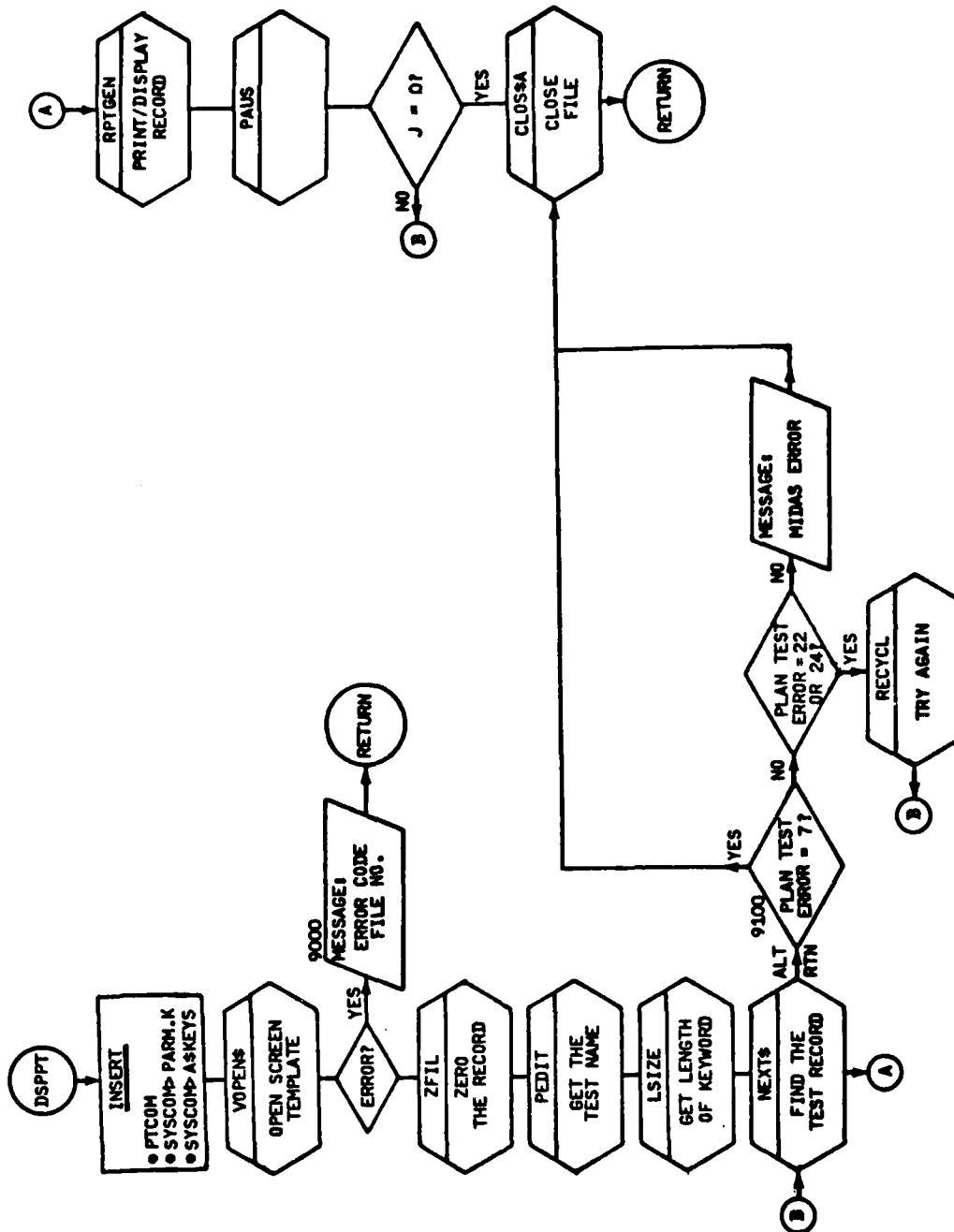
● INTEGER: REPORT CHANNEL



DSPPT - DISPLAY ONE TEST PLAN

ARGUMENTS: 1

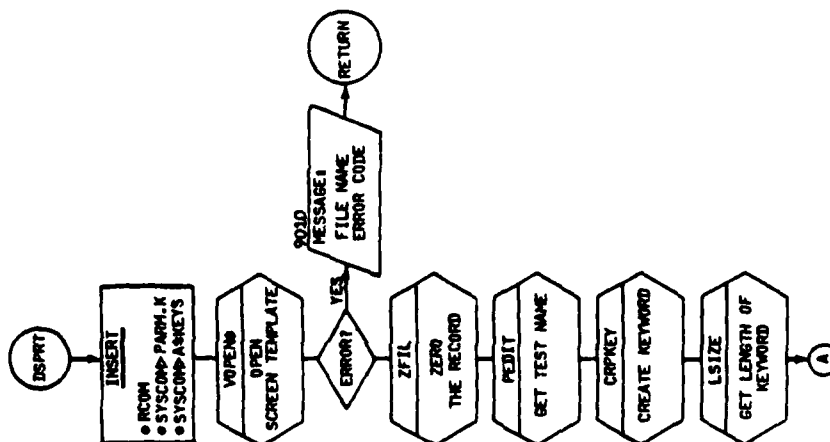
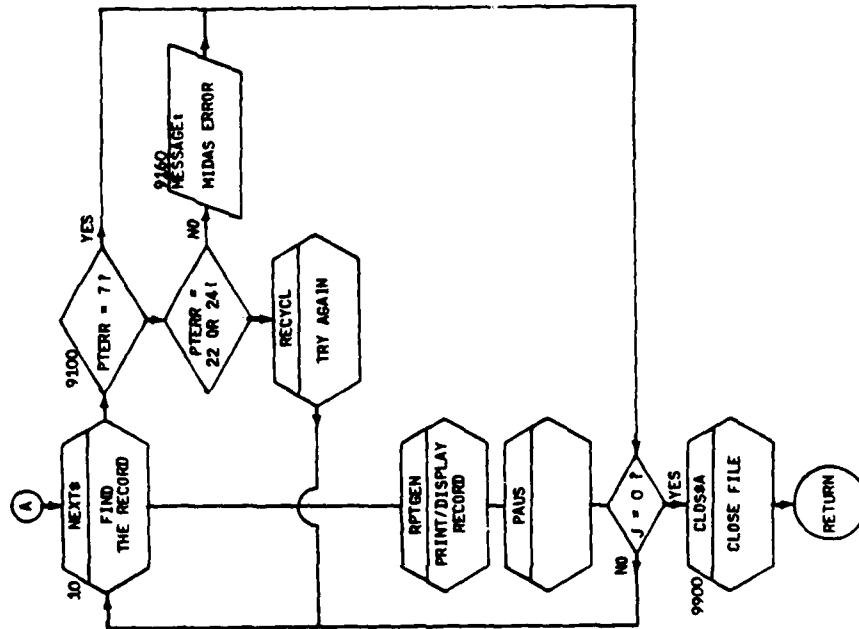
● INTEGER: SPOOL CHANNEL



DSPRT - DISPLAY RESULTS OF ONE TEST

ARGUMENTS: 1

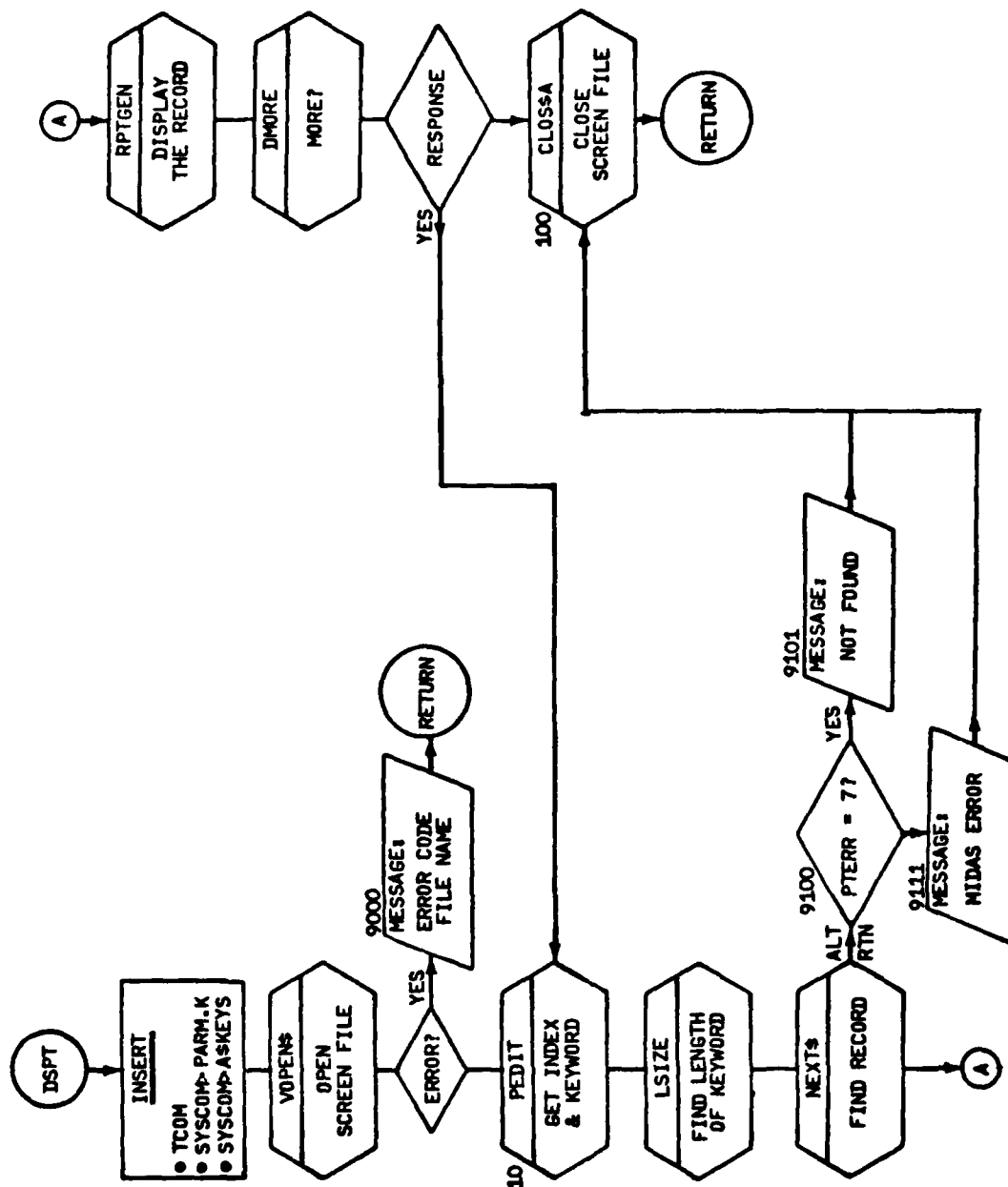
● INTEGER: SPOOL CHANNEL



DSPT - DISPLAY TAPE DATA

ARGUMENTS: 1

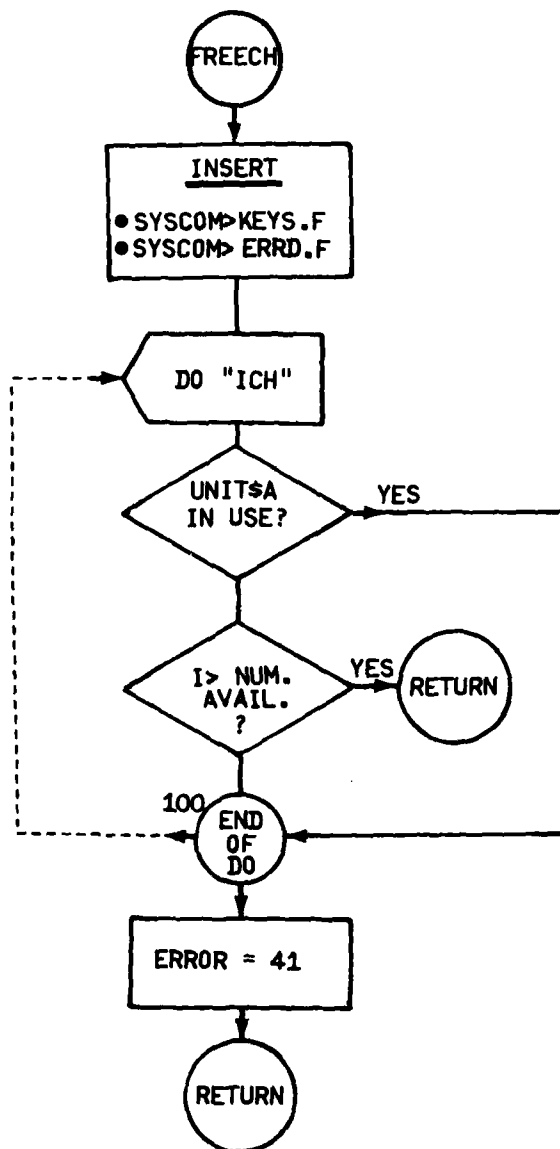
● INTEGER: SPOOL FILE CHANNEL



FREECH - RETURN THE NUMBER OF AVAILABLE PRIMOS CHANNELS

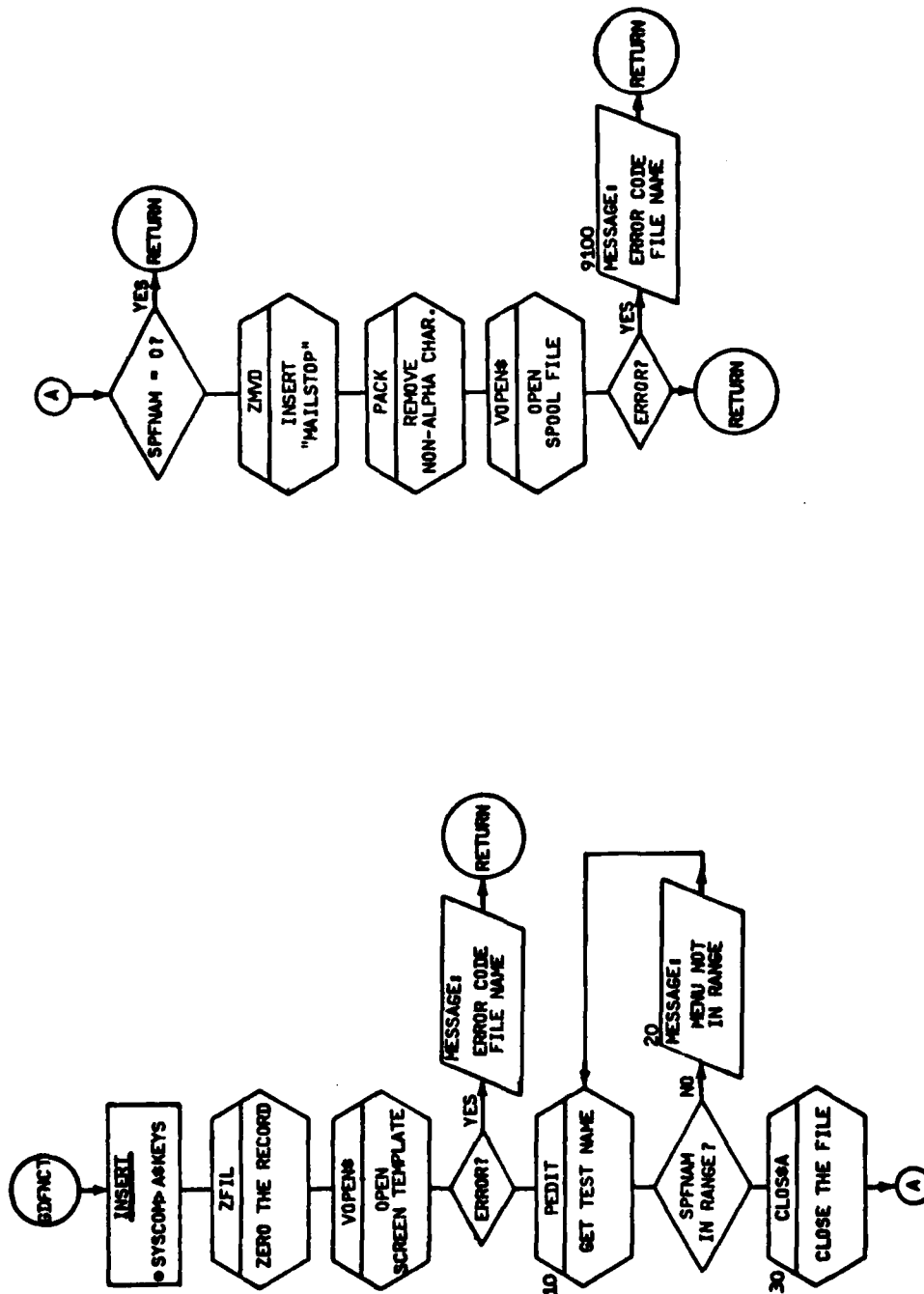
ARGUMENTS: 2

- INTEGER: FREE CHANNELS REQUESTED
- INTEGER: ARRAY SIZE



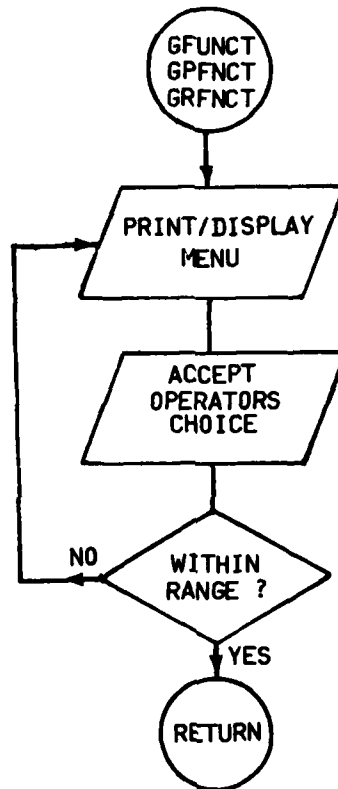
GDFNCT - GET DISPLAY FUNCTION

ARGUMENTS: 3
 ● INTEGER: MENU
 ● INTEGER: SPOOL CHANNEL NUMBER
 ● INTEGER: SPOOL FILE NAME



GFUNCT - GET FUNCTION, MAIN MENU
GPFUNCT - GET PLAN FUNCTION
GRFNCT - GET RETRIEVAL FUNCTION

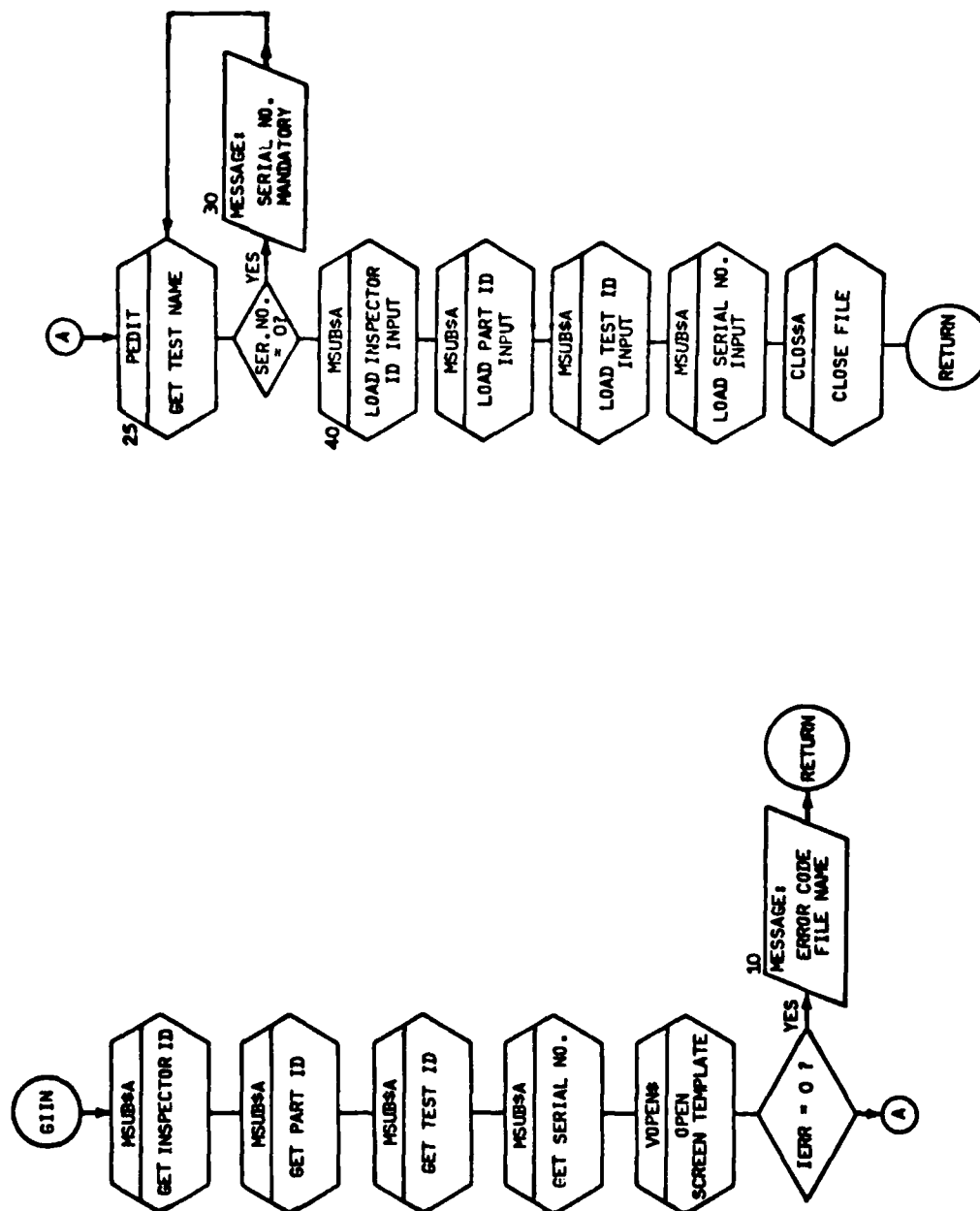
ARGUMENTS: 1
• INTEGER: MENU



GIIN - GET INSPECTOR INPUT

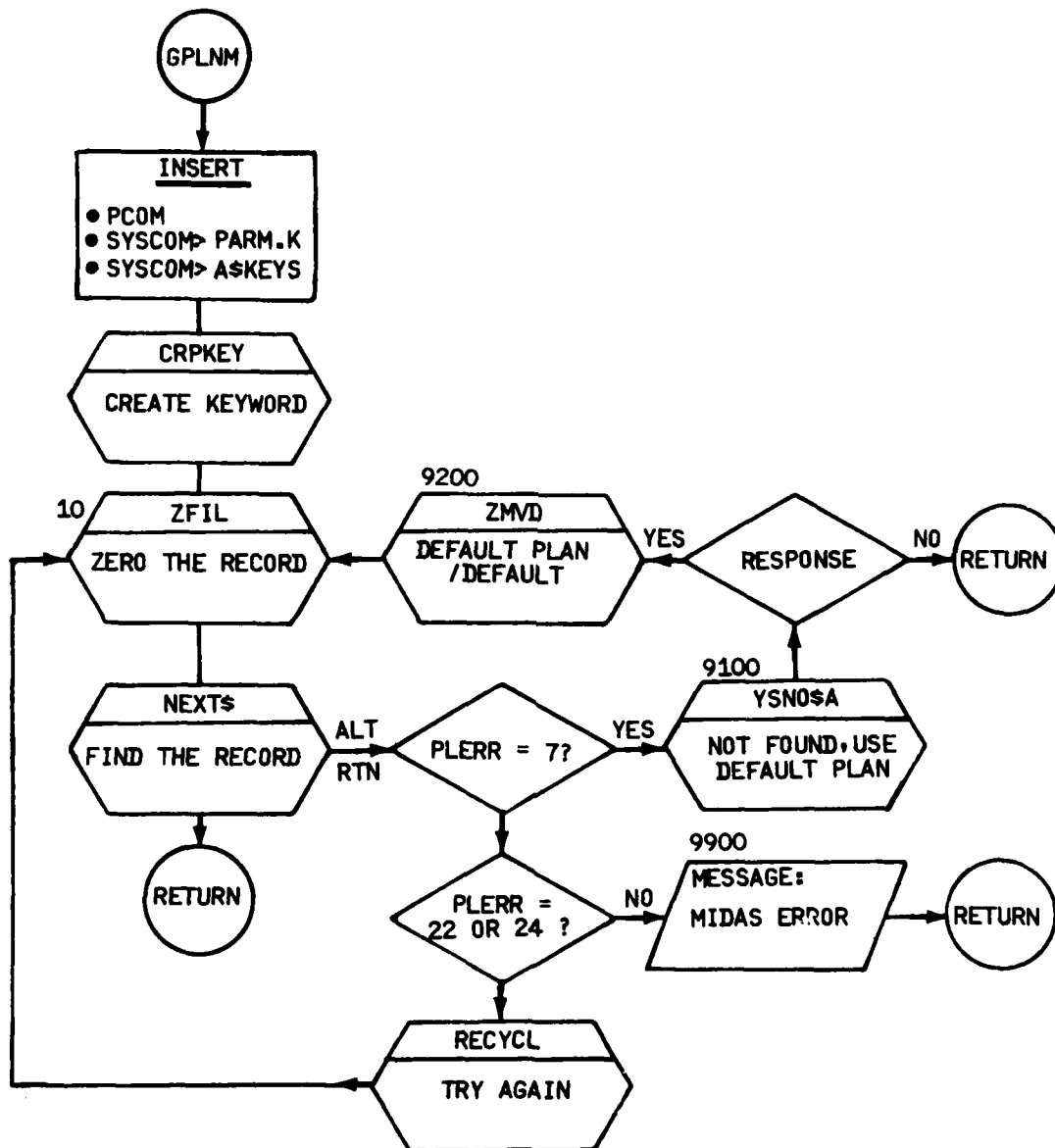
ARGUMENTS: 5

- STRING: INSPECTOR I.D.
- STRING: SERIAL NUMBER
- LOGICAL: EQUIP. INIT. FLAG
- LOGICAL: DEFAULT PLAN FLAG
- INTEGER: ERROR CODE



GPLNM - GET PLAN RECORD

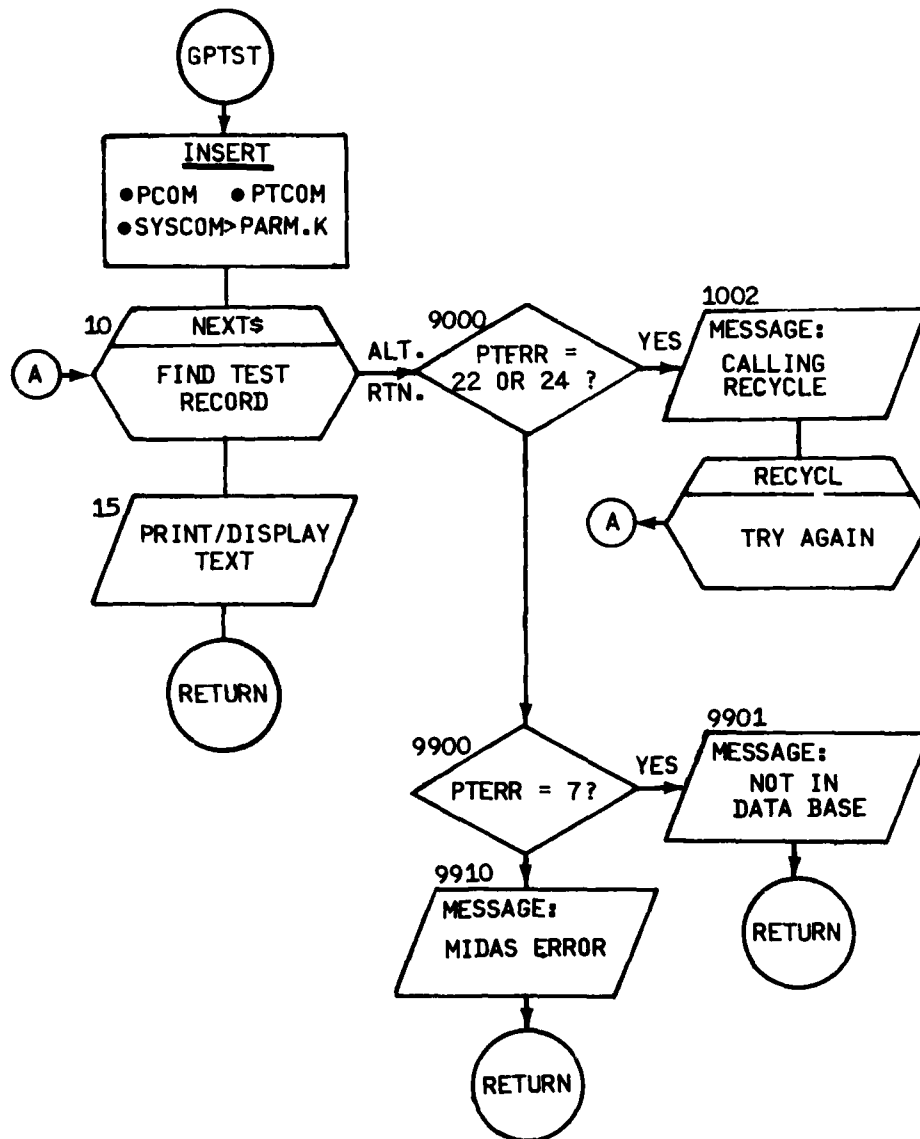
ARGUMENTS: 4
 • INTEGER: PART NO.
 • INTEGER: TEST ID
 • LOGICAL: DEFAULT PLAN
 • INTEGER: ERROR



GPTST - GET A TEST PLAN

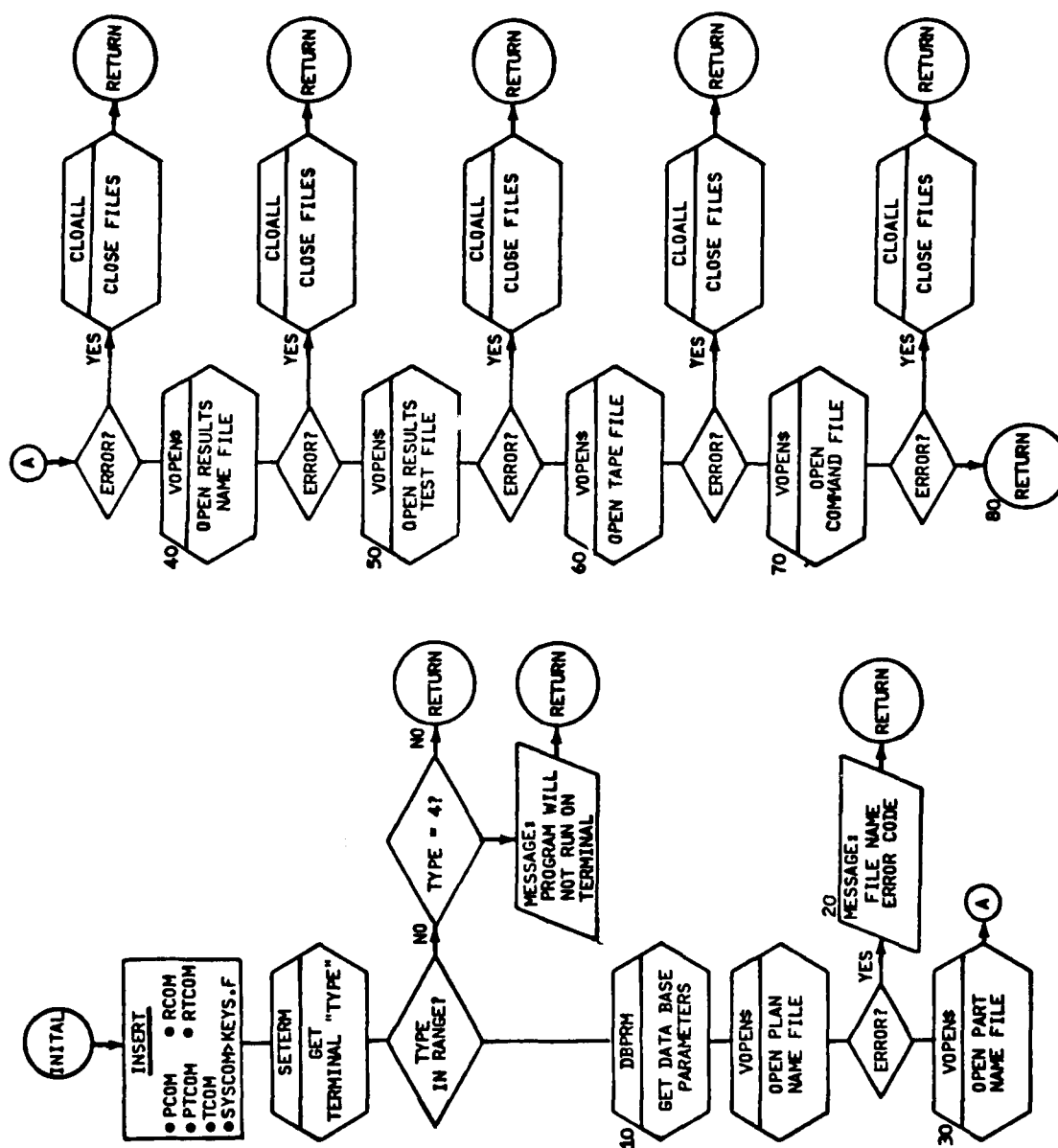
ARGUMENTS: 2

- INTEGER: TEST NO.
- INTEGER: ERROR



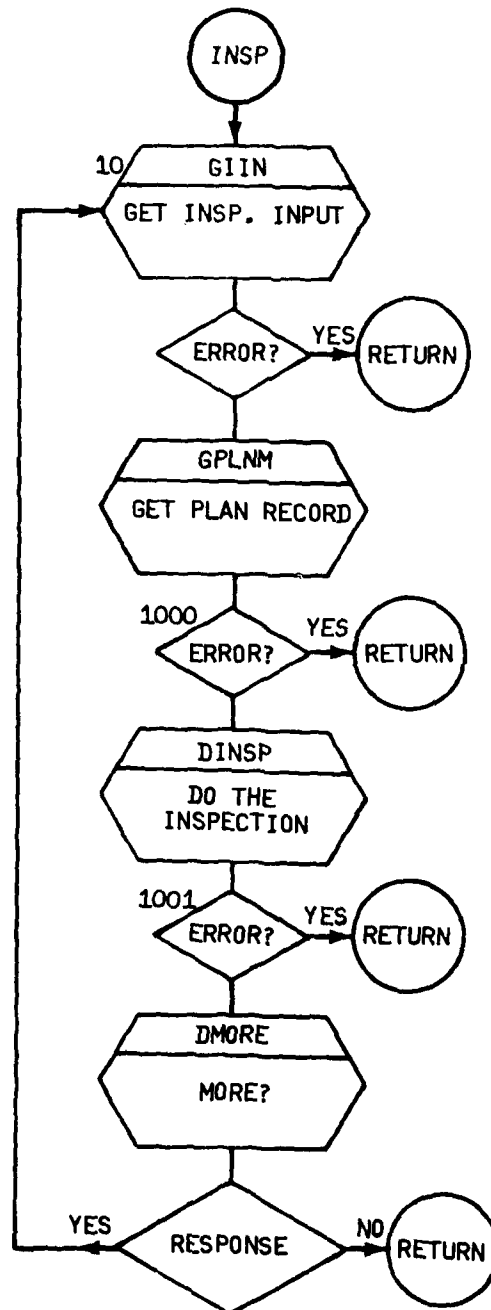
INITIAL - SET UP TERMINAL CODES AND OPEN DATA BASE FILES

ARGUMENTS: 1
● **INTEGER:** ERROR



INSP - PERFORM AN INSPECTION

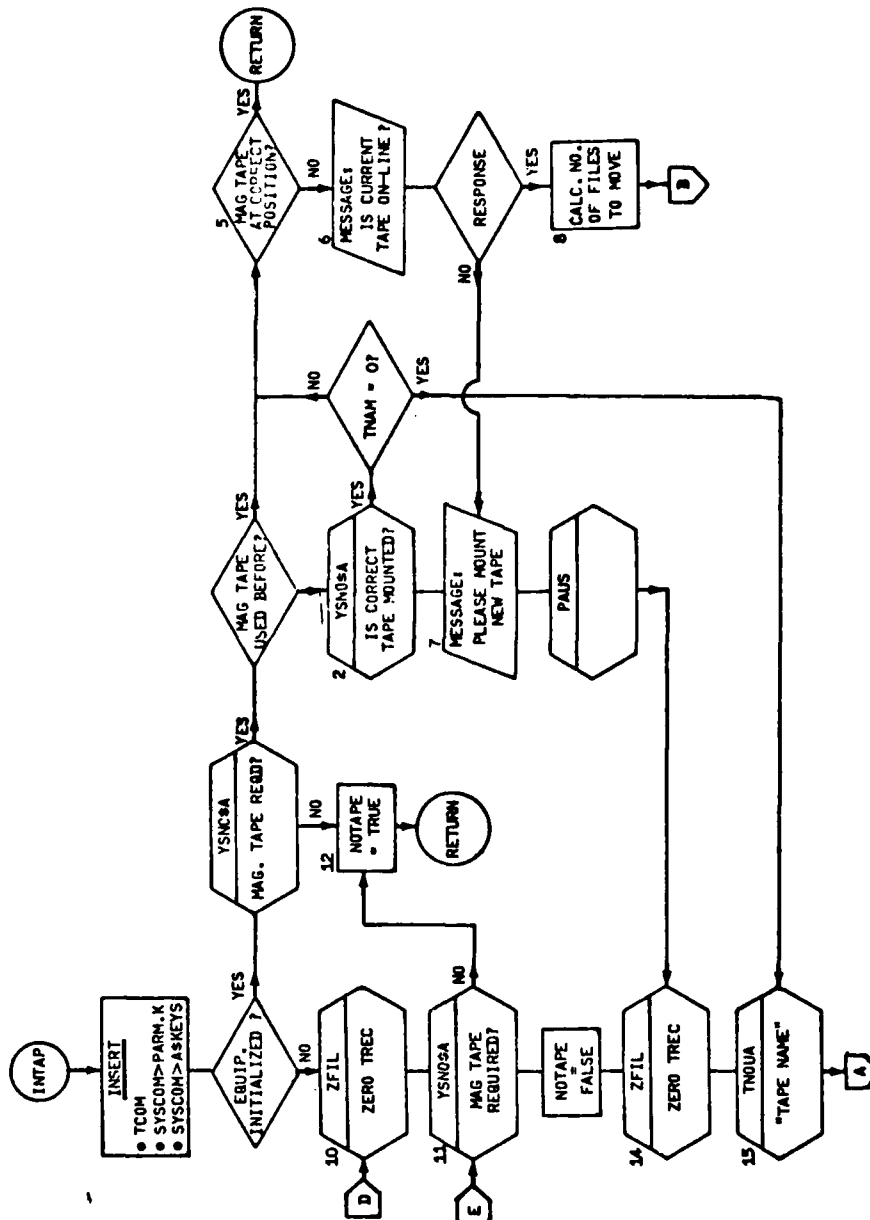
ARGUMENTS: 2
LOGICAL: EQUIPMENT
INTEGER: ERROR



INTAP - INITIALIZE TAPE

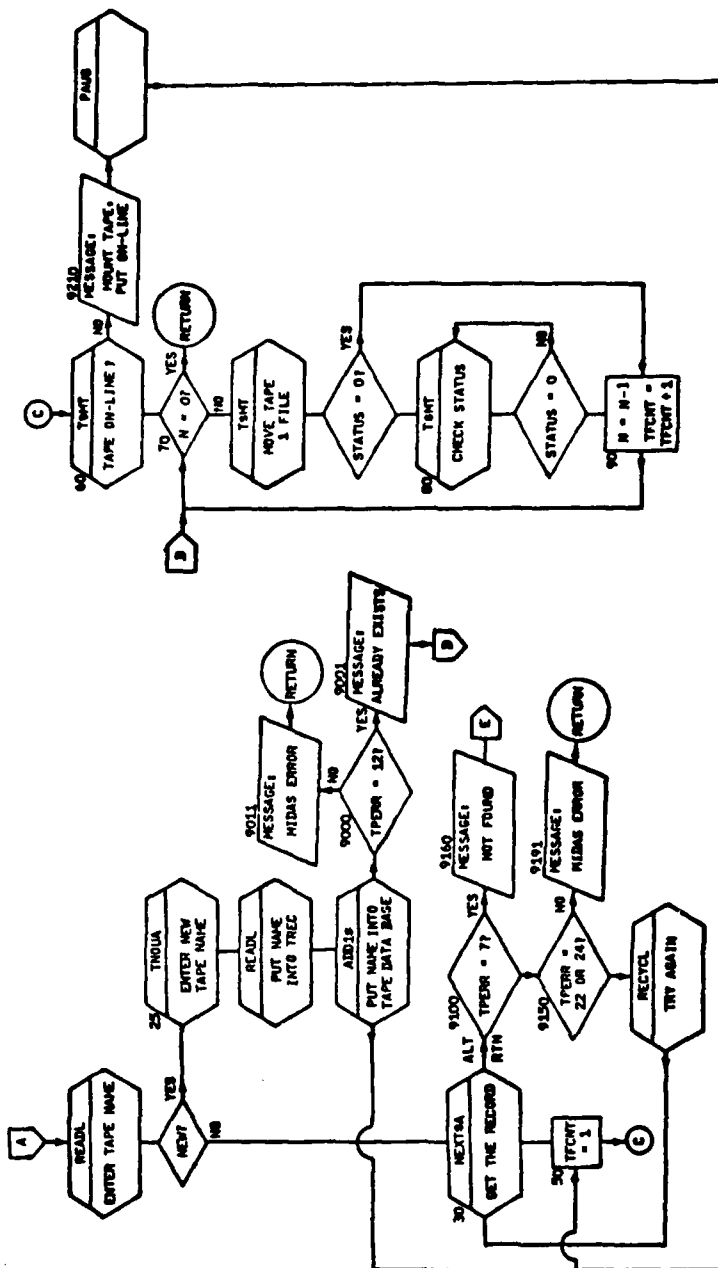
ARGUMENTS: 3

- INTEGER: EQUIPMENT INIT. FLAG
- INTEGER: EQUIP. IN USE FLAG
- INTEGER: ERROR



CONTINUED ON NEXT PAGE

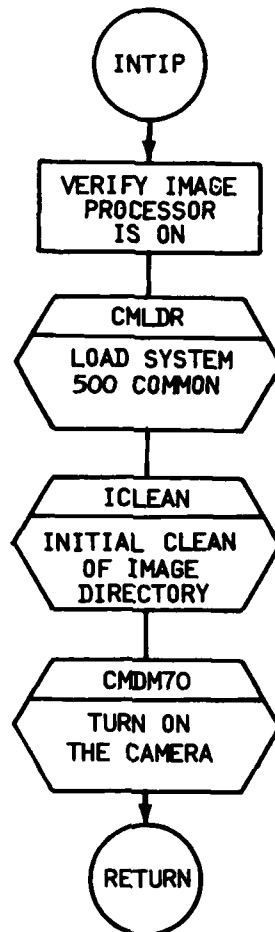
INTAP - INITIALIZE TAPE (CONT'D)



INTIP - INITIALIZE THE IMAGE PROCESSOR

ARGUMENTS: 1

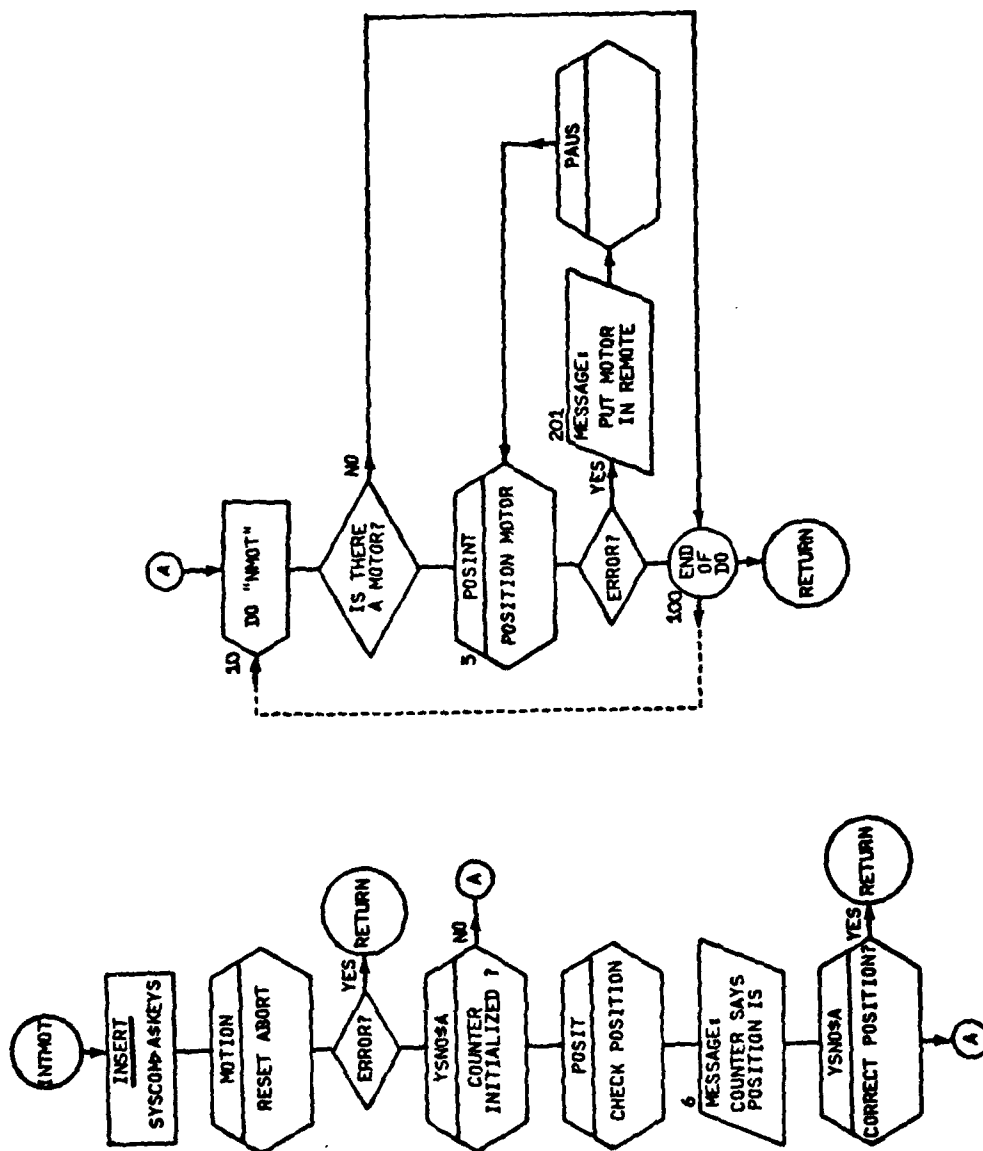
• INTEGER: ERROR



INTMOT - INITIALIZE MOTORS

ARGUMENTS: 1

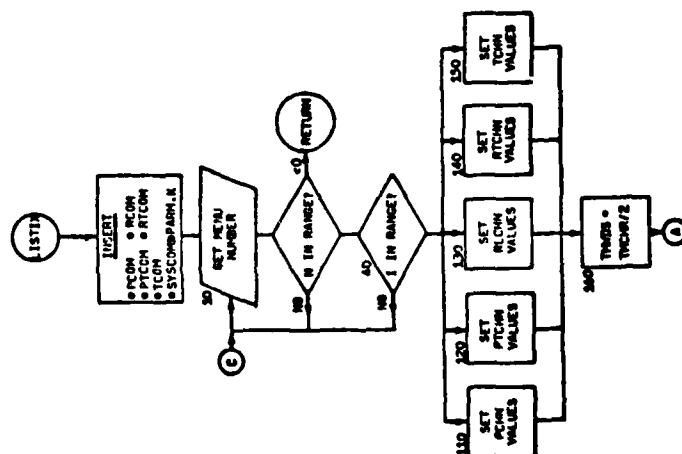
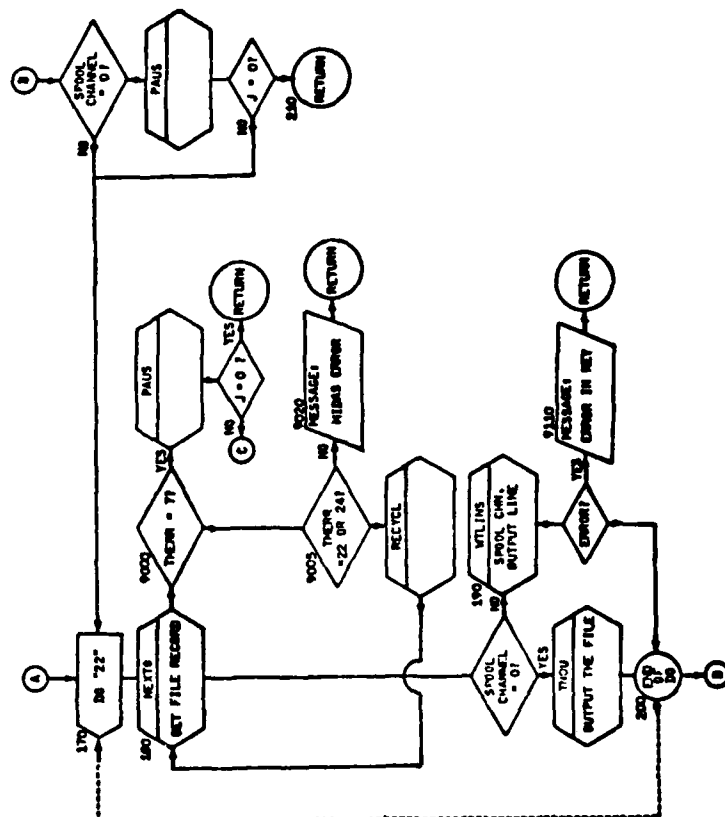
● INTEGER: ERROR



LISTIX - LIST KEY NAMES BY INDEX

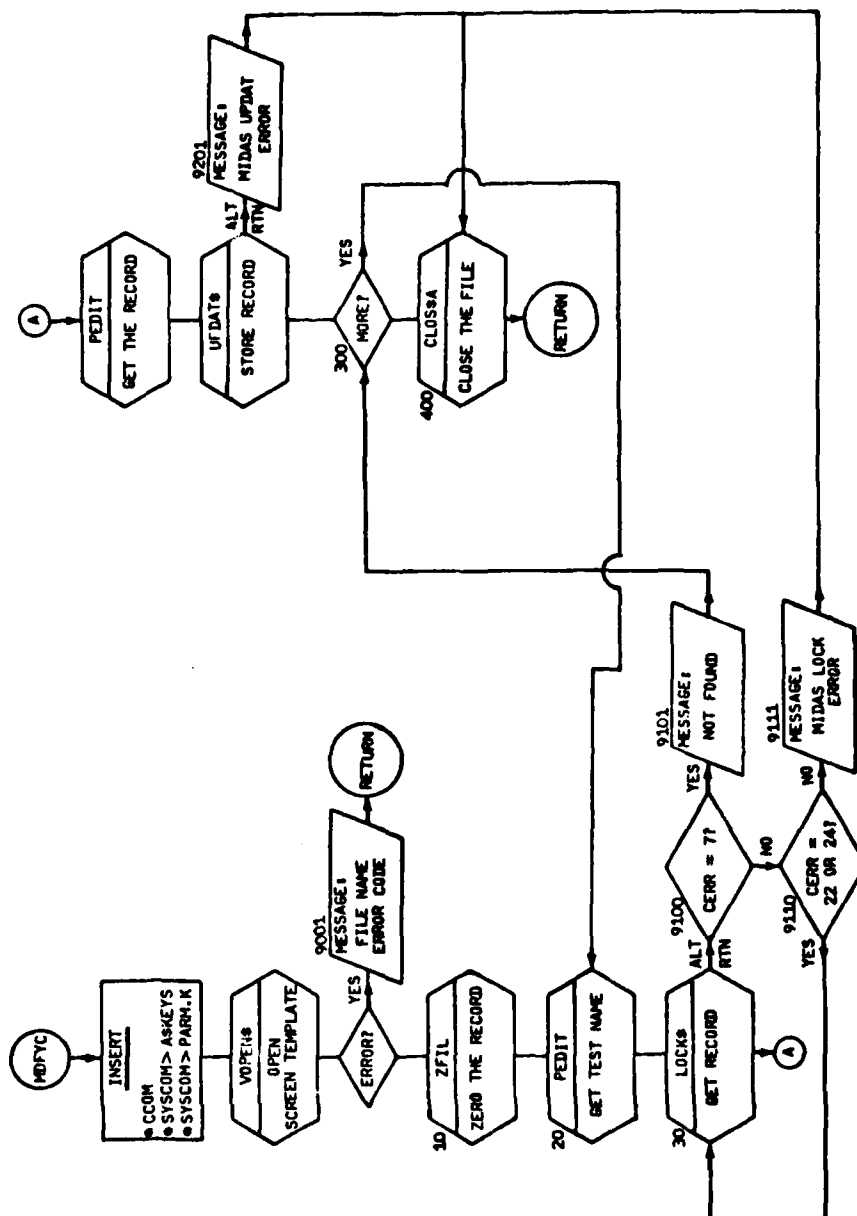
ARGUMENTS: 1

● INTEGER: SPOOL CHANNEL NUMBER



MDFYC - MODIFY IMAGE PROCESSOR COMMAND

ARGUMENTS: 1
● INTEGER: ERROR



MDFYP - MODIFY PLAN
MKFMC - MAKE IMAGE COMMAND FROM OLD
MKFMP - MAKE FROM PLAN
MKFMT - MAKE FROM TEST

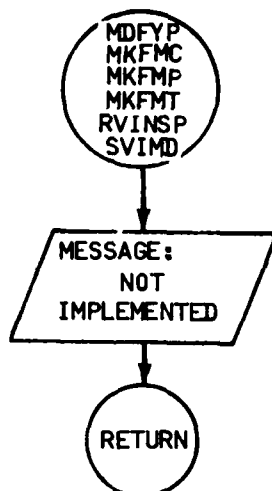
ARGUMENTS: 1
• INTEGER: ERROR

RVINSP - RECREATE AN INSPECTION

ARGUMENTS: NONE

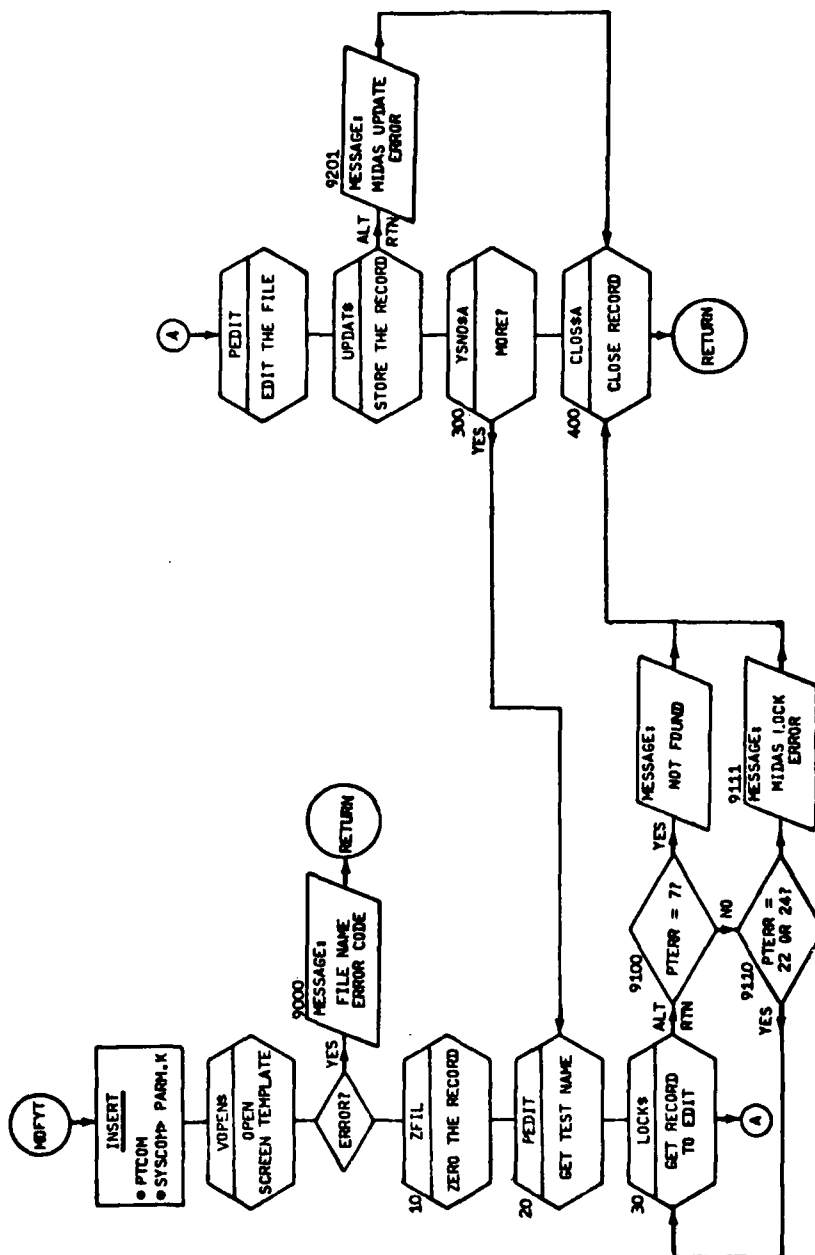
SVIMD - SAVE IMAGE ON DISK

ARGUMENTS: 1
• STRING: NAME



MDFYT - MODIFY TEST PROCEDURE

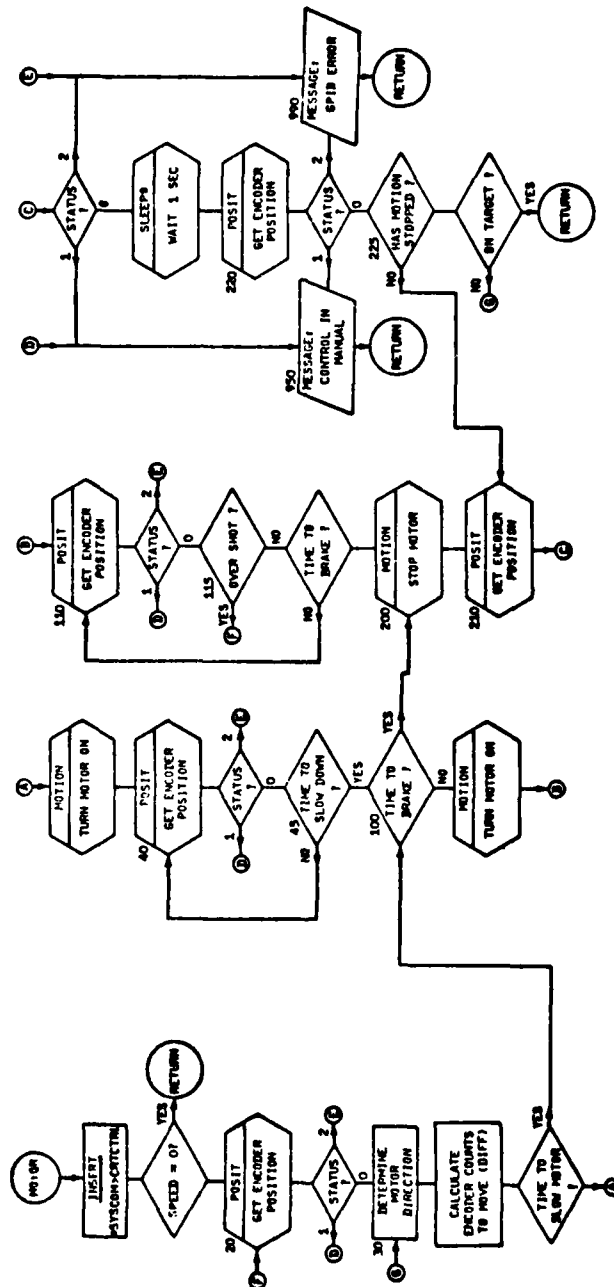
ARGUMENTS: 1
● INTEGER: ERROR



MOTOR - CONTROL MOTION OF A MOTOR

ARGUMENTS: 4

- INTEGER: MOTOR NUMBER
- INTEGER: MOTOR SPEED
- INTEGER: POSITION
- INTEGER: ERROR



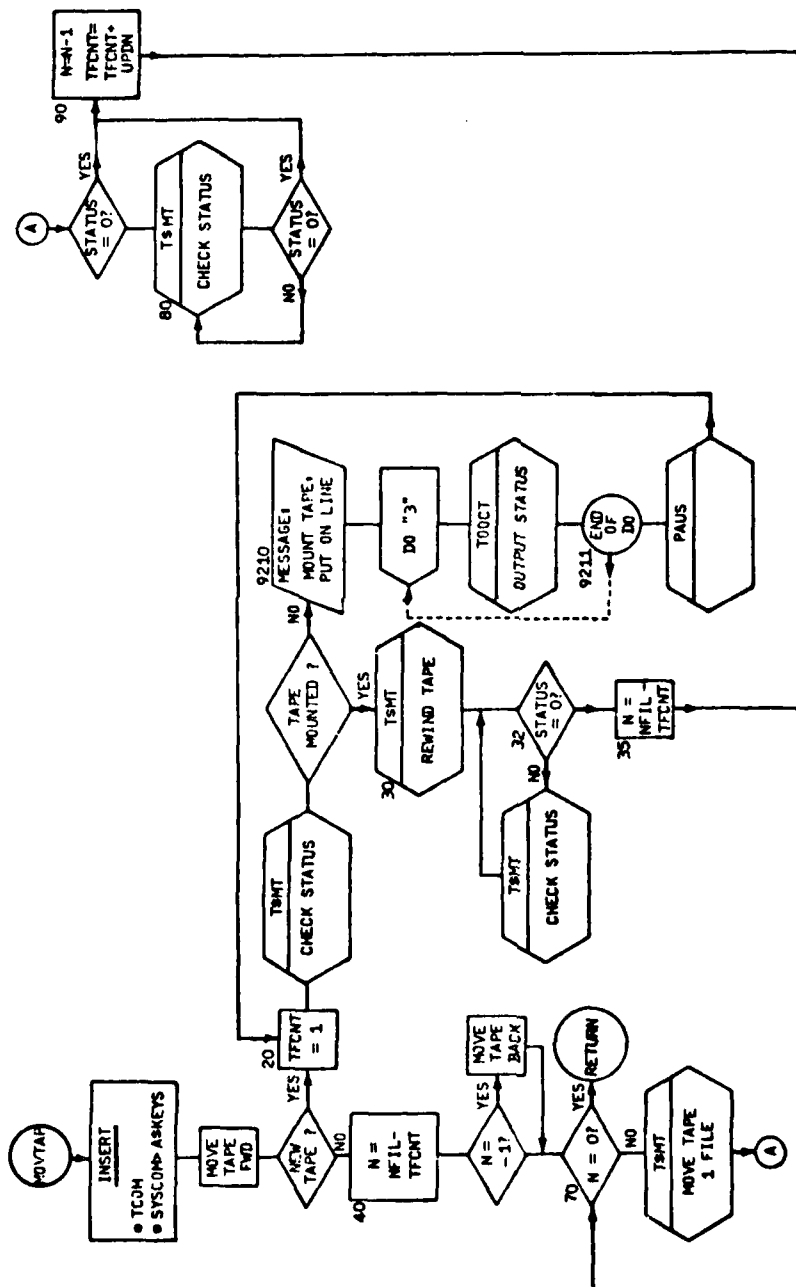
MOVTAP - VERIFY AND MOVE TAPE

ARGUMENTS: 3

● LOGICAL: NEW TAPE

● INTEGER: FILE NO. TO POSITION TAPE

● INTEGER: ERROR



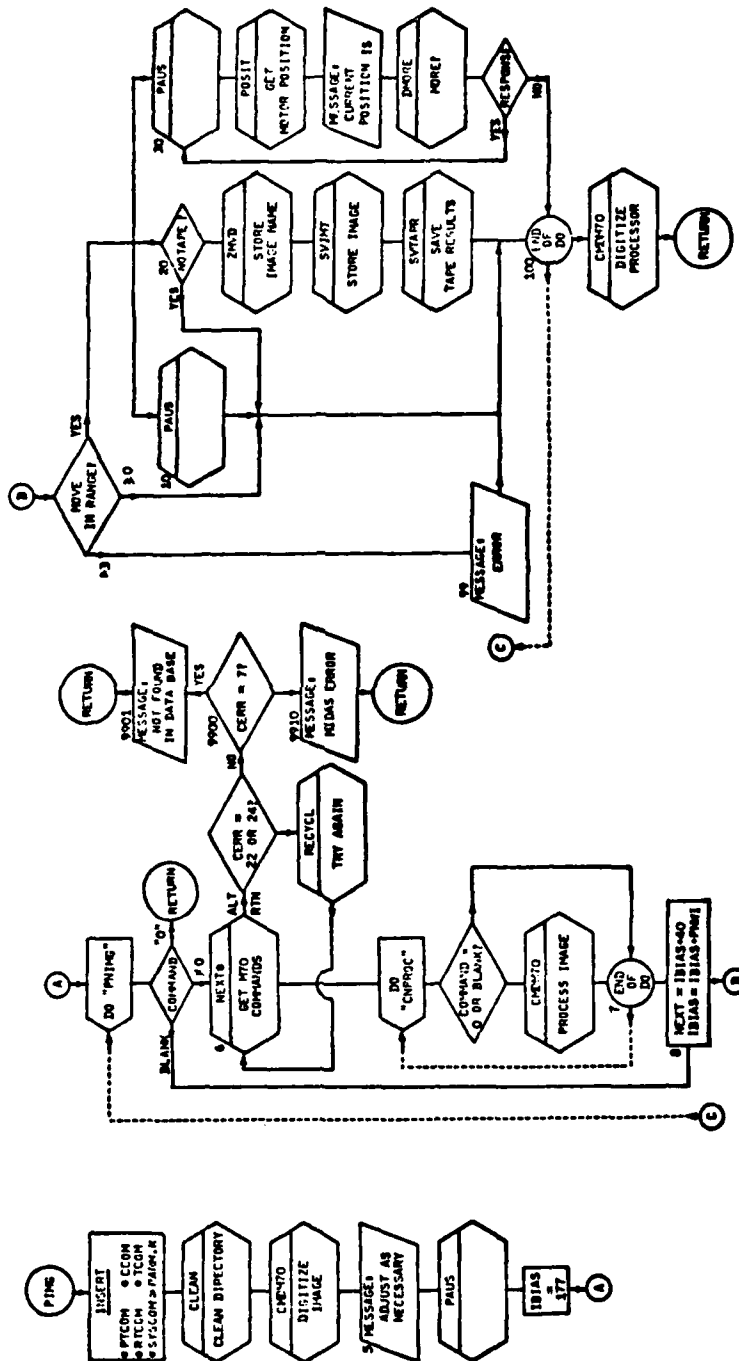
PIMG - PROCESS THE IMAGE

ARGUMENTS: 3

● INTEGER: TEST SEQUENCE NO.

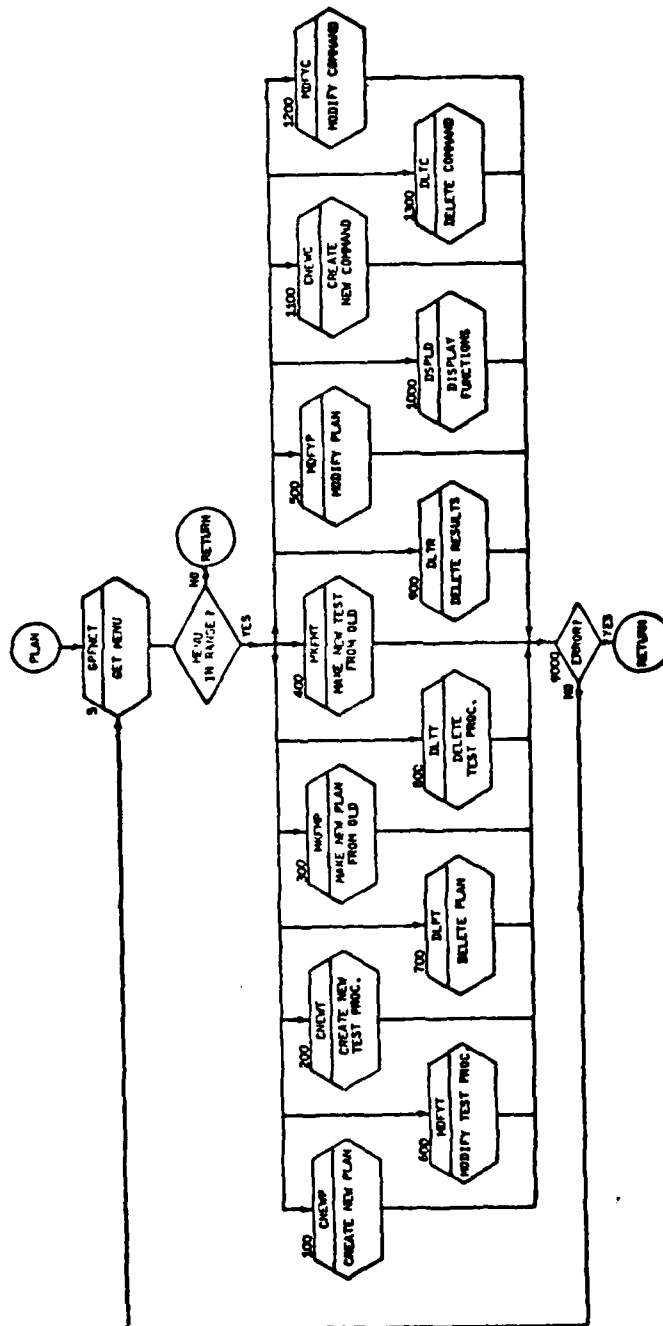
● LOGICAL: NO TAPE FLAG

● INTEGER: ERROR



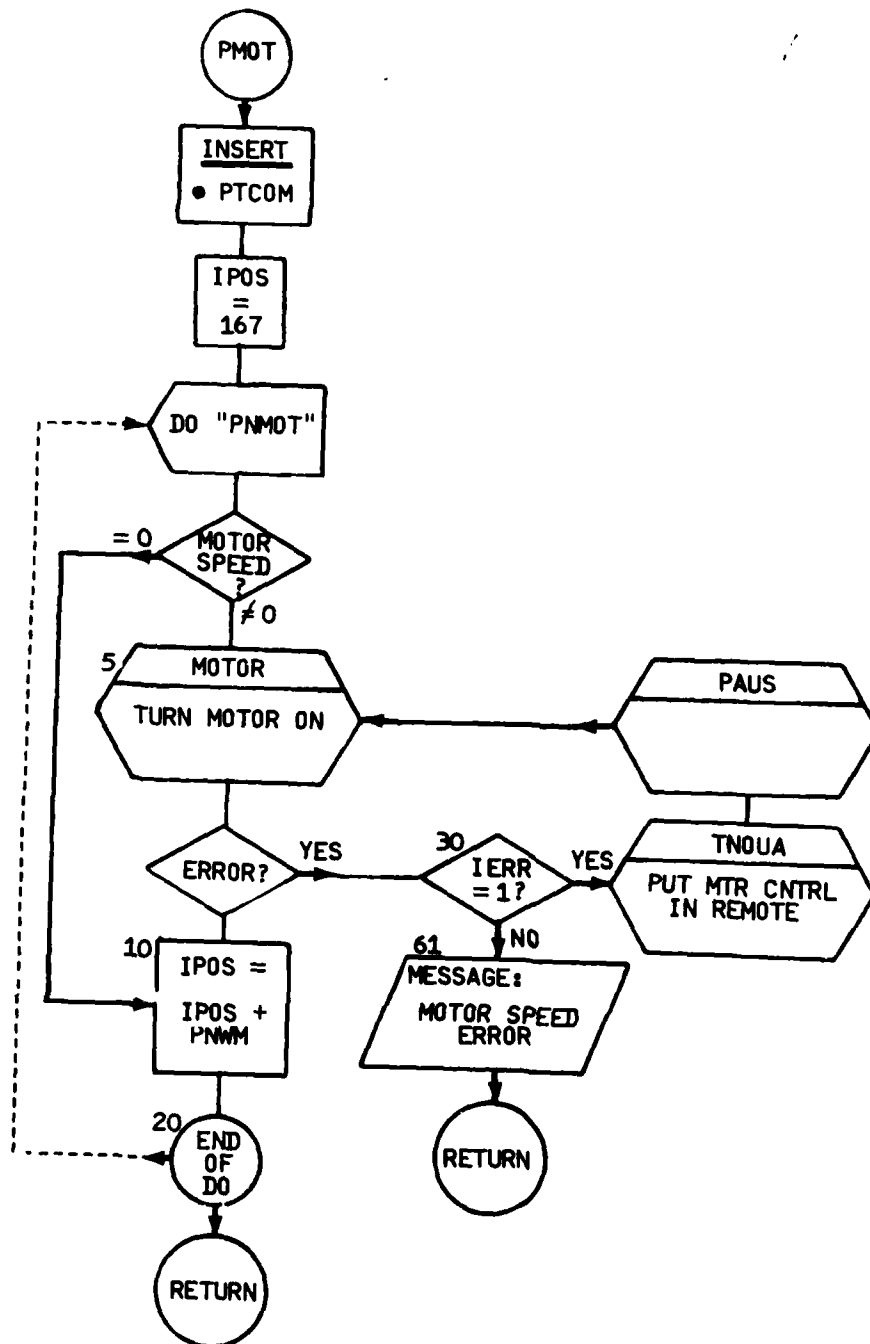
PLAN - MAIN PLANNING ROUTINE

ARGUMENTS: 1
 ● INTEGER: ERROR



PMOT - POSITION MOTORS

ARGUMENTS: NONE

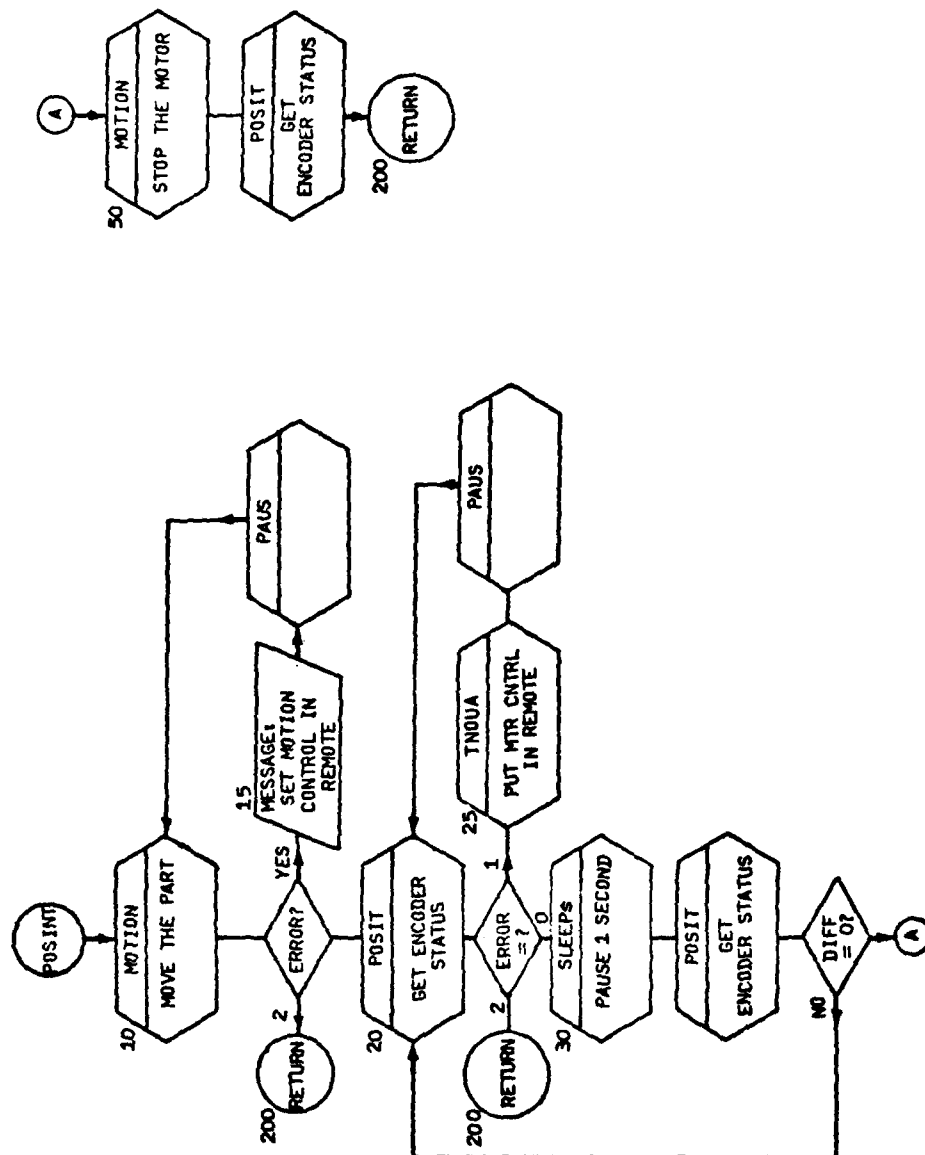


POSINT - POSITION MOTOR & INITIALIZE ENCODER

ARGUMENTS: 2

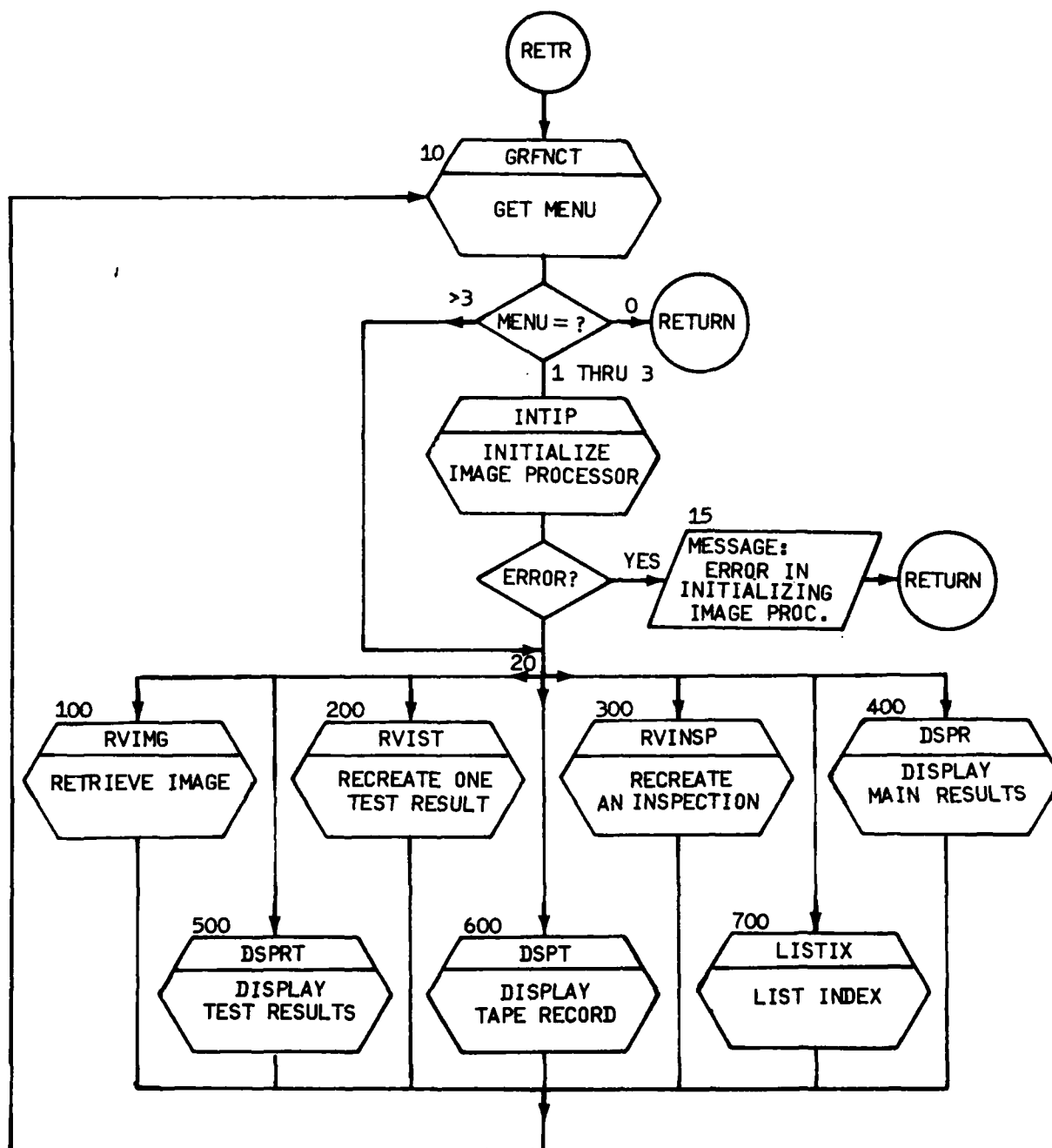
● INTEGER: ENCODER

● INTEGER: ERROR



RETR - RETRIEVE HISTORICAL RECORD

ARGUMENTS: 2
 INTEGER: ERROR
 LOGICAL: EQUIPMENT

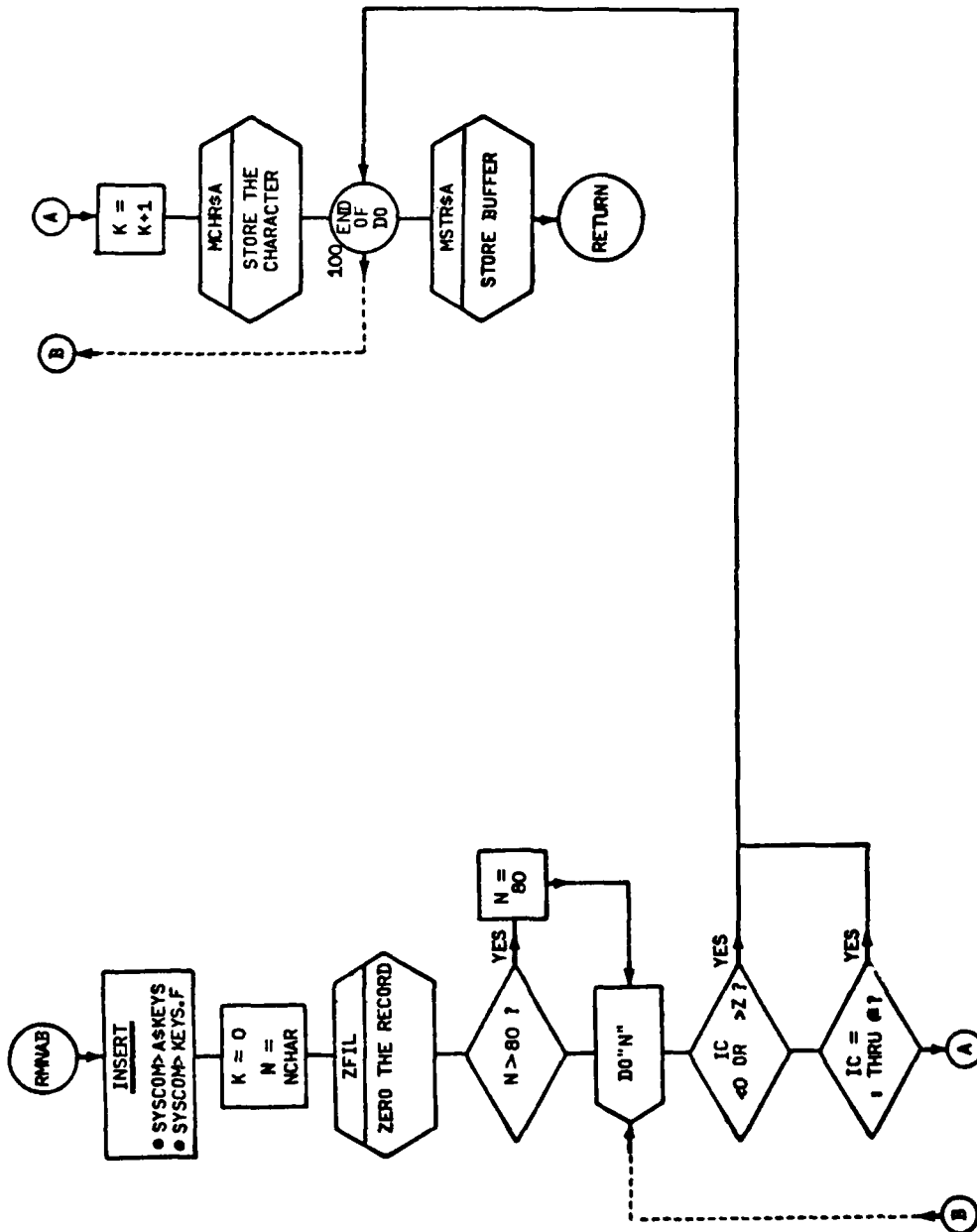


RMNAB - REMOVE NON-ALPHAHBETICAL CHARACTERS

ARGUMENTS: 2

● **STRING:** TEXT

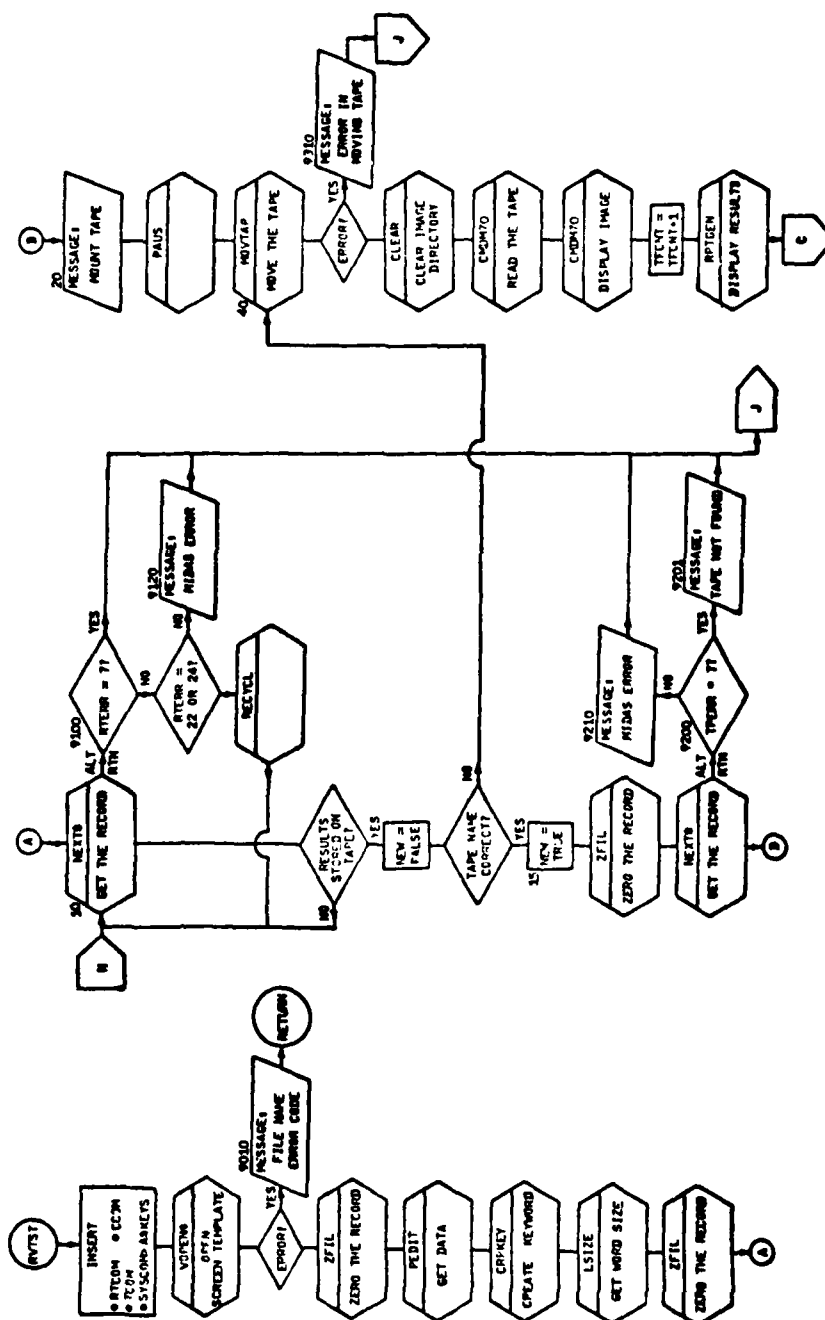
● INTEGER: NO. OF CHARACTERS



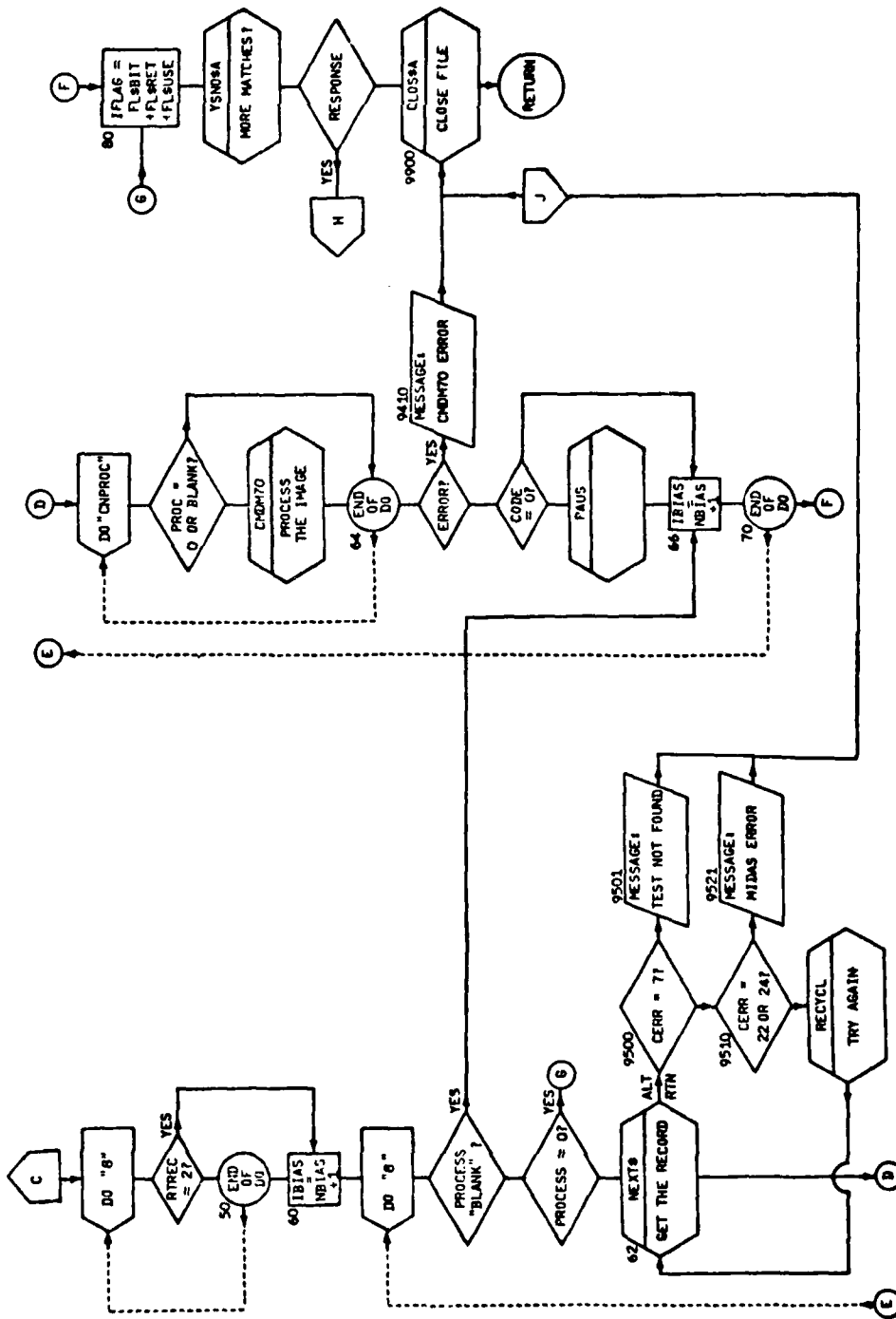
ARGUMENTS: NONE



ARGUMENTS: NONE



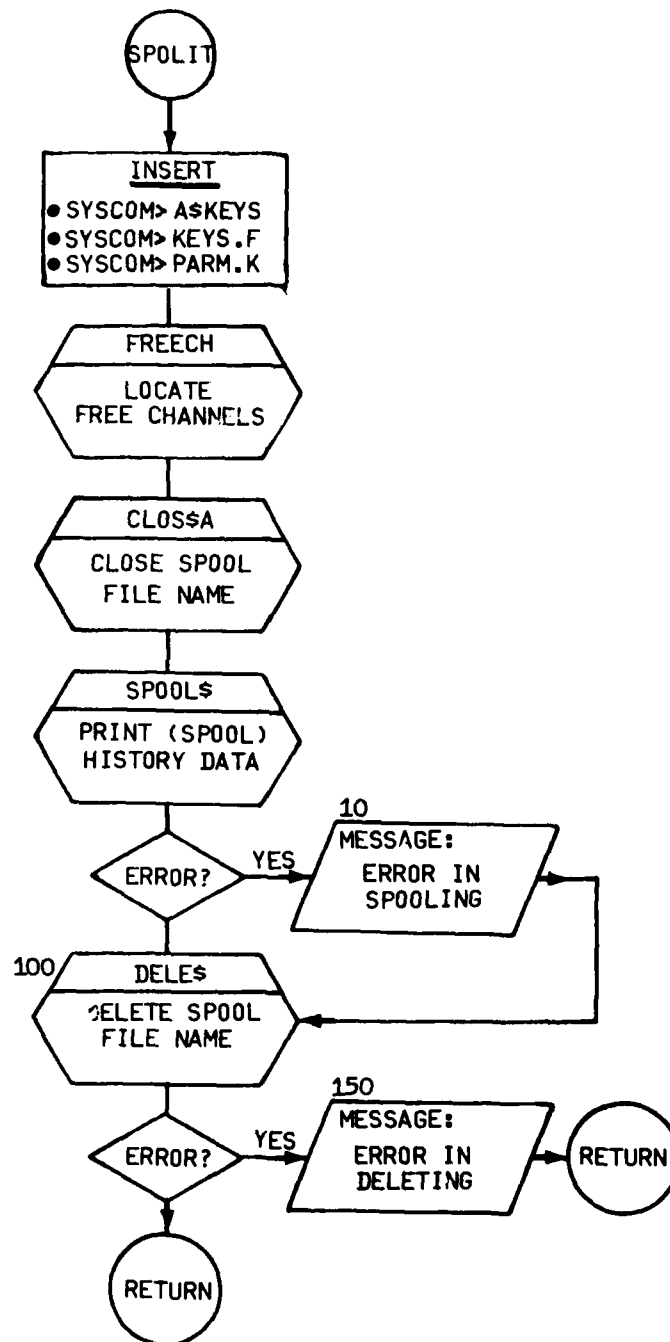
CONTINUED ON
NEXT PAGE

RVTST - RECREATE ONE TEST RESULT (CONT'D)

SPOLIT - SPOOL RETRIEVAL INFORMATION

ARGUMENTS: 2

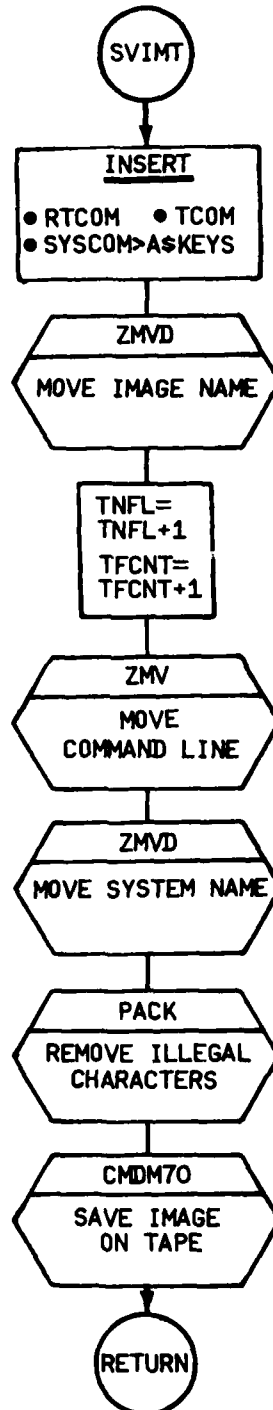
- INTEGER: SPOOL CHANNEL
- STRING: SPOOL FILE NAME



SVIMT - SAVE IMAGE ON TAPE

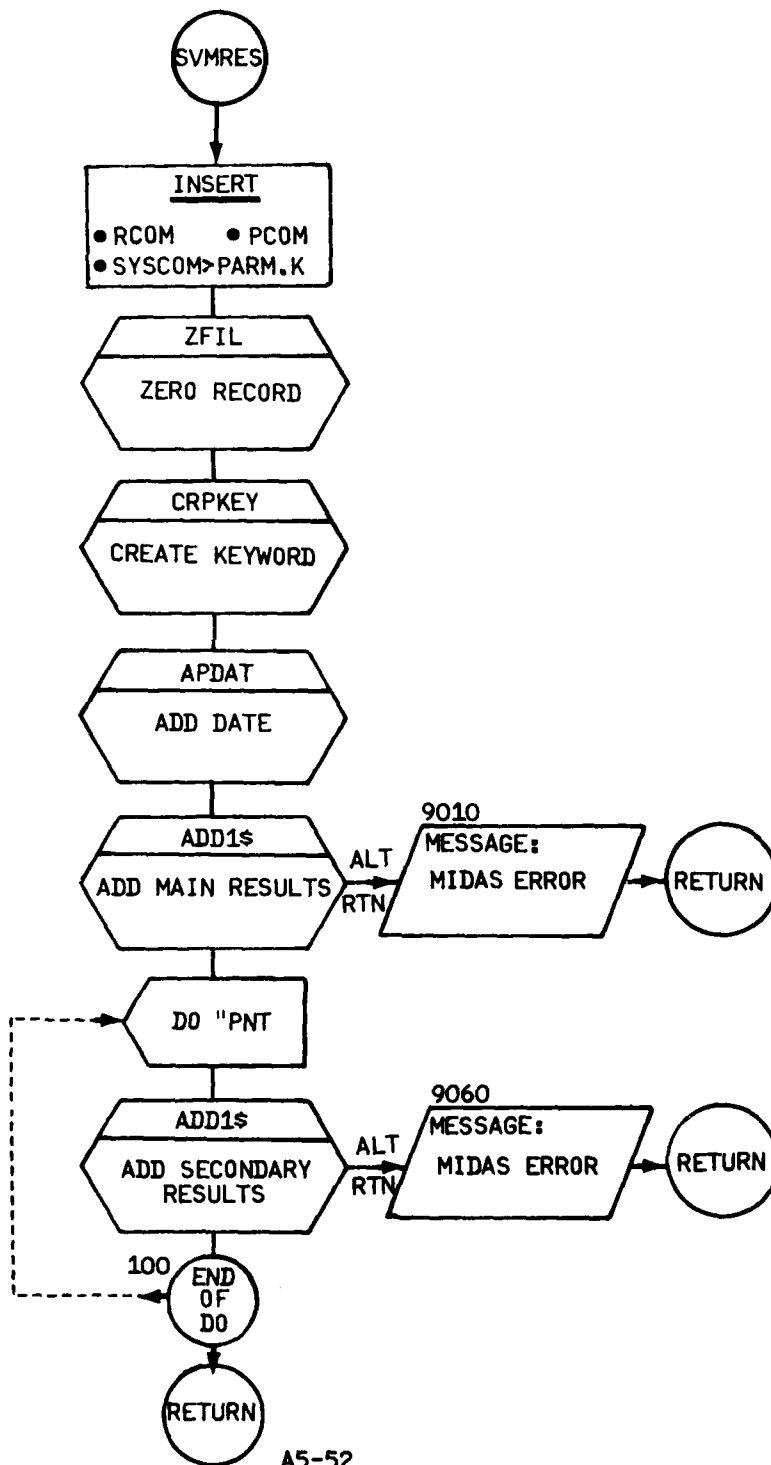
ARGUMENTS: 1

• STRING: SYSTEM IMAGE NAME



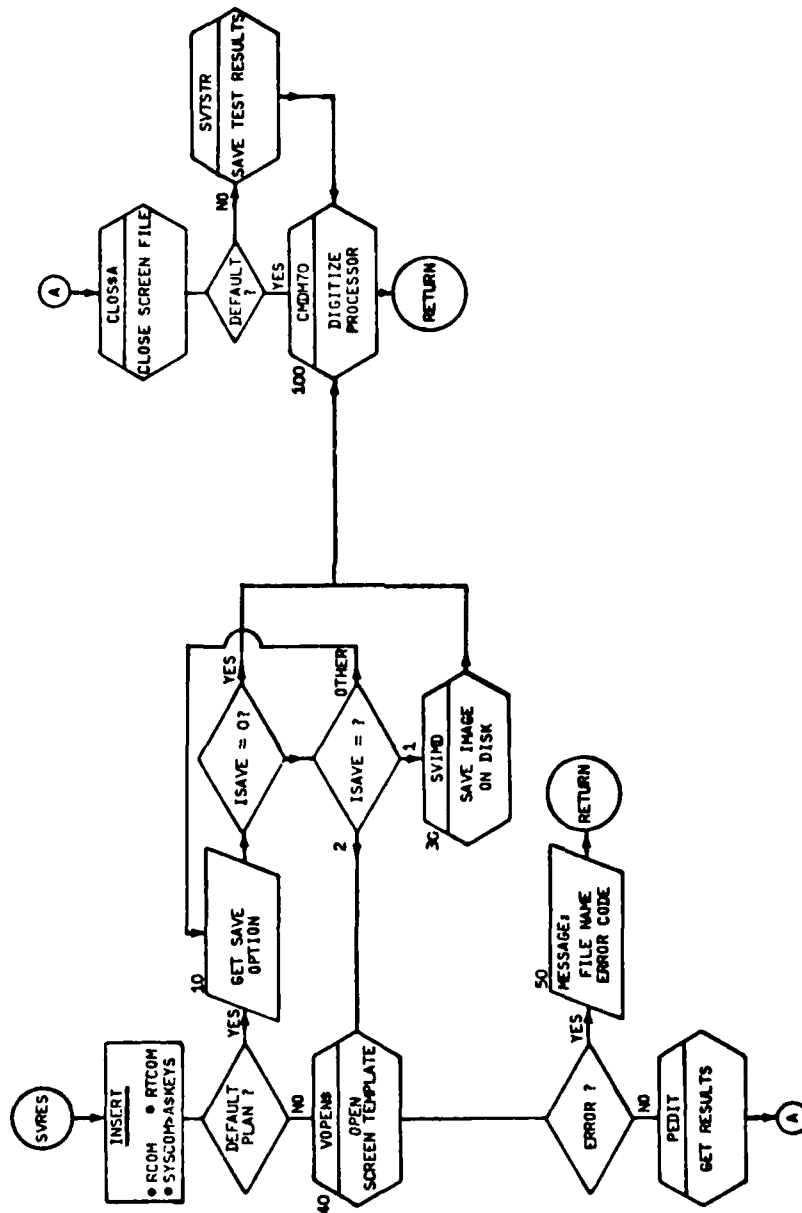
SVMRES - SAVE RESULTS - MAIN RECORD

ARGUMENTS: 1
 • INTEGER: ERROR



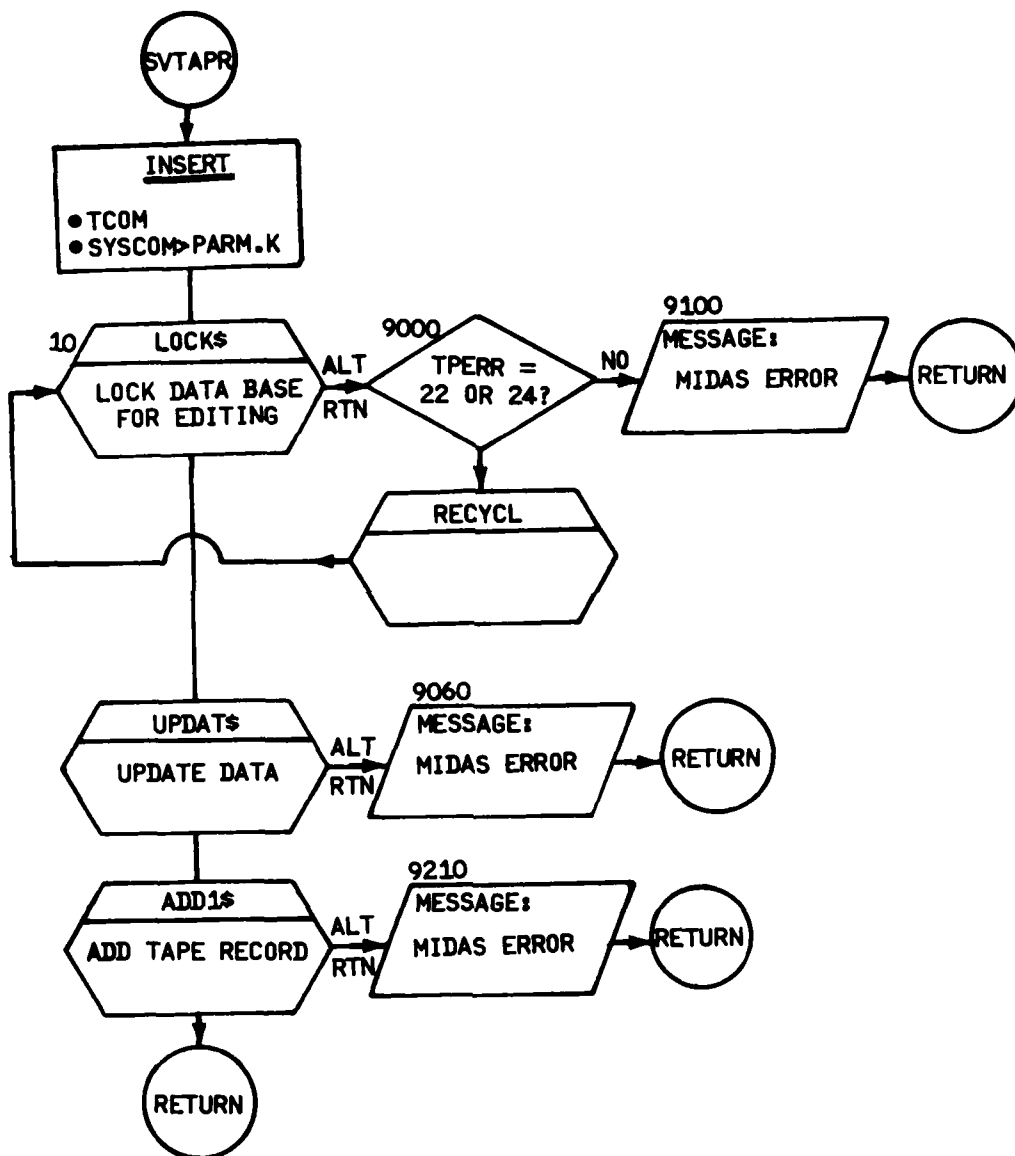
SVRES - SAVE RESULTS OF INSPECTION

ARGUMENTS: 5
 ● LOGICAL: DEFAULT
 ● LOGICAL: NO TAPE
 ● INTEGER: LOOP
 ● INTEGER: RTKEY
 ● INTEGER: ERROR



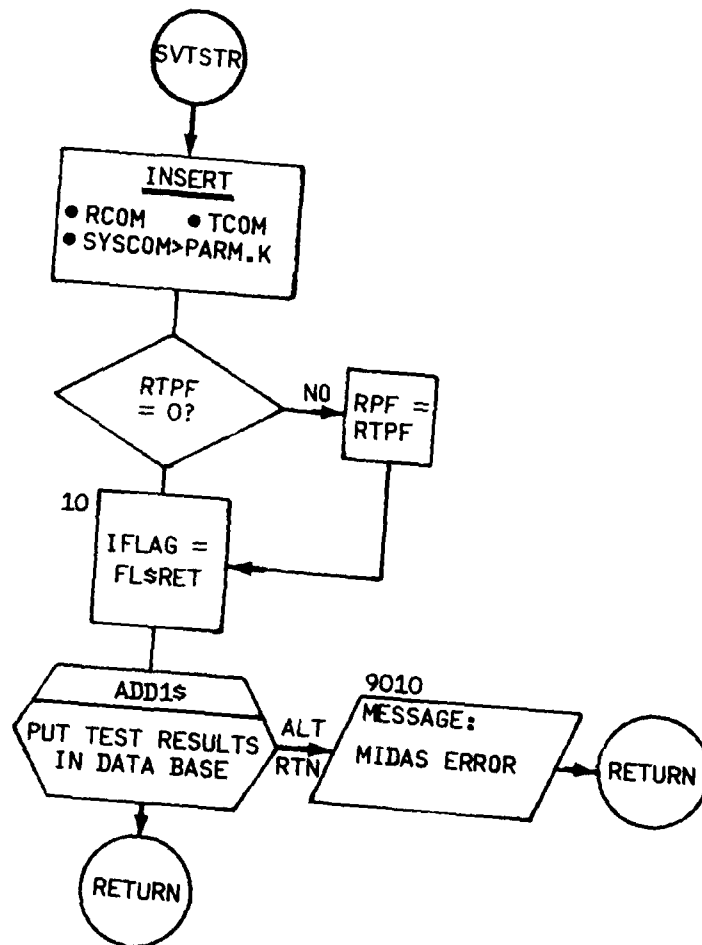
SVTAPR - SAVE THE TAPE RECORD (MIDAS)

ARGUMENTS: 1
• INTEGER: ERROR



SVTSTR - SAVE TEST RESULTS

ARGUMENTS: 2
 • INTEGER: KEY
 • INTEGER: ERROR



C PCOM: COMMON FILE FOR PLAN DATA-BASE.

```
*****
*
*   REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM
*   VERSION 1.0      JUNE 1, 1980
*   BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY
*
*****
```

```

INTEGER PSZW,PSZB,PNWT
PARAMETER PNWT=10 /* No. OF WORDS IN TEST NAME
PARAMETER PSZW=10*PNWT+21 /* PLAN RECORD SIZE IN WORDS
PARAMETER PSZB=PSZW*2 /* PLAN RECORD SIZE IN BYTES

```

```
INTEGER PREC(FSZW)      /* PLAN RECORD
```

```

INTEGER PPN(10)          /* PLAN PART No.
INTEGER PIN(10)          /* " INSPECTION NAME
INTEGER PNT              /* " No. TESTS
INTEGER PTNO1(PNWT)      /* TEST 01 NAME
INTEGER PTNO2(PNWT)      /* TEST 02 NAME
INTEGER PTNO3(PNWT)      /* TEST 03 NAME
INTEGER PTNO4(PNWT)      /* TEST 04 NAME
INTEGER PTNO5(PNWT)      /* TEST 05 NAME
INTEGER PTNO6(PNWT)      /* TEST 06 NAME
INTEGER PTNO7(PNWT)      /* TEST 07 NAME
INTEGER PTNO8(PNWT)      /* TEST 08 NAME
INTEGER PTNO9(PNWT)      /* TEST 09 NAME
INTEGER PTNO10(PNWT)     /* TEST 10 NAME

```

PRIMARY KEY: CONDENSATION OF PART-NUMBER AND INSPECTION NAME
WITH / SEPARATING THEM. 30 CHARACTERS LONG.
SECONDARY KEY: TEST NAMES.

EQUIVALENCE	(PREC(1),	PPN)	/* PLAN PART No.	
EQUIVALENCE	(PREC(11),	PIN)	/* " INSPECTION	NAME
EQUIVALENCE	(PREC(21),	PNT)	/* No. TESTS	
EQUIVALENCE	(PREC(22),	PTN01)	/* TEST 01	NAME
EQUIVALENCE	(PREC(32),	PTN02)	/* TEST 02	NAME
EQUIVALENCE	(PREC(42),	PTN03)	/* TEST 03	NAME
EQUIVALENCE	(PREC(52),	PTN04)	/* TEST 04	NAME
EQUIVALENCE	(PREC(62),	PTN05)	/* TEST 05	NAME
EQUIVALENCE	(PREC(72),	PTN06)	/* TEST 06	NAME
EQUIVALENCE	(PREC(82),	PTN07)	/* TEST 07	NAME
EQUIVALENCE	(PREC(92),	PTN08)	/* TEST 08	NAME
EQUIVALENCE	(PREC(102),	PTN09)	/* TEST 09	NAME
EQUIVALENCE	(PREC(112),	PTN10)	/* TEST 10	NAME

```
DATA-BASE NAME AND MIDAS INDEX POINTER ARRAY
INTEGER PLNAM(16)      /* PLAN DATA FILE NAME
INTEGER PLEN           /* NAME LENGTH
INTEGER PCHN           /* CHAN. TO DATA BASE FILE
INTEGER PLARR(14)      /* MIDAS INFO ARRAY
INTEGER PLERR          /* MIDAS ERROR CODE
```

EQUIVALENCE (FLARR,FLERR)
COMMON /PCOM/ PREC,PLNAM,PLLEN,PCHN,PLARR

C PTCOM: PLAN TEST COMMON FILE

```

*****
*                                     *
*      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
*      VERSION 1.0      JUNE 1, 1980                    *
*      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
*                                     *
*****

```

```

INTEGER PTSZW, PTSZB, PNWM, PNWI, FNMOT, FNIMG
PARAMETER FNMOT=5          /* No. OF MOTOR COMMANDS
PARAMETER FNIMG=10         /* No. OF IMAGE COMMANDS
PARAMETER PNWM=2           /* No. OF WORDS PER MOTOR COMMAND
PARAMETER PNWI=41          /* No. OF WORDS PER IMAGE COMMAND
PARAMETER PTSZW=FNMOT*PNWM+FNIMG*PNWI+166
PARAMETER PTSZB=PTSZW*2    /* No. OF BYTES IN PLAN TEST RECORD

```

C INTEGER PTREC(PTSZW)

```

INTEGER PTNM(10)          /* TEST NAME
INTEGER PDA(39)           /* DESCRIPTION LINE A
INTEGER PDB(39)           /* DESCRIPTION LINE B
INTEGER PDC(39)           /* DESCRIPTION LINE C
INTEGER PDD(39)           /* DESCRIPTION LINE D
INTEGER PKXP              /* MASK X POSITION
INTEGER PKXS              /* MASK X SPEED
INTEGER PKYP              /* MASK Y POSITION
INTEGER PKYS              /* MASK Y SPEED
INTEGER PMXP              /* TABLE X POSITION
INTEGER PMXS              /* TABLE X SPEED
INTEGER PMYP              /* TABLE Y POSITION
INTEGER PMYS              /* TABLE Y SPEED
INTEGER PMRP              /* ROTATE POSITION
INTEGER PMRS              /* ROTATE SPEED
INTEGER PIPA(40)          /* IMAGE PROCESS A
INTEGER PNXA              /* CODE FOR NEXT OPERATION
INTEGER PIPB(40)          /* IMAGE PROCESS B
INTEGER PNXB              /* CODE FOR NEXT OPERATION
INTEGER PIPC(40)          /* IMAGE PROCESS C
INTEGER PNXC              /* CODE FOR NEXT OPERATION
INTEGER PIPD(40)          /* IMAGE PROCESS D
INTEGER PNXD              /* CODE FOR NEXT OPERATION
INTEGER PIPE(40)          /* IMAGE PROCESS E
INTEGER PNXE              /* CODE FOR NEXT OPERATION
INTEGER PIPF(40)          /* IMAGE PROCESS F
INTEGER PNXF              /* CODE FOR NEXT OPERATION
INTEGER PIPG(40)          /* IMAGE PROCESS G
INTEGER PNXG              /* CODE FOR NEXT OPERATION
INTEGER PIPH(40)          /* IMAGE PROCESS H
INTEGER PNXH              /* CODE FOR NEXT OPERATION

```

```

C PRIMARY KEY = PTNM (1st. ELEMENT, 20 CHAR.)
C SECONDARY KEY = NONE.

```

```

EQUIVALENCE (PTREC(1), PTNM) /* TEST NAME
EQUIVALENCE (PTREC(11), PDA) /* DESCRIPTION LINE A
EQUIVALENCE (PTREC(50), PDB) /* DESCRIPTION LINE B
EQUIVALENCE (PTREC(89), PDC) /* DESCRIPTION LINE C
EQUIVALENCE (PTREC(128), PDD) /* DESCRIPTION LINE D
EQUIVALENCE (PTREC(167), PKXP) /* MASK X POSITION
EQUIVALENCE (PTREC(168), PKXS) /* MASK X SPEED
EQUIVALENCE (PTREC(169), PKYP) /* MASK Y POSITION
EQUIVALENCE (PTREC(170), PKYS) /* MASK Y SPEED
EQUIVALENCE (PTREC(171), PMXP) /* TABLE X POSITION
EQUIVALENCE (PTREC(172), PMXS) /* TABLE X SPEED
EQUIVALENCE (PTREC(173), PMYP) /* TABLE Y POSITION
EQUIVALENCE (PTREC(174), PMYS) /* TABLE Y SPEED
EQUIVALENCE (PTREC(175), PMRP) /* ROTATE POSITION
EQUIVALENCE (PTREC(176), PMRS) /* ROTATE SPEED
EQUIVALENCE (PTREC(177), PIPA) /* IMAGE PROCESS A
EQUIVALENCE (PTREC(217), PNXA) /* CODE FOR NEXT OPERATION
EQUIVALENCE (PTREC(218), PIPB) /* IMAGE PROCESS B
EQUIVALENCE (PTREC(258), PNXB) /* CODE FOR NEXT OPERATION
EQUIVALENCE (PTREC(259), PIPC) /* IMAGE PROCESS C

```

```
EQUIVALENCE (PTREC(299),PNXC) /* CODE FOR NEXT OPERATION
EQUIVALENCE (PTREC(300),PIPD) /* IMAGE PROCESS D
EQUIVALENCE (PTREC(340),PNXD) /* CODE FOR NEXT OPERATION
EQUIVALENCE (PTREC(341),PIPE) /* IMAGE PROCESS E
EQUIVALENCE (PTREC(381),PNXE) /* CODE FOR NEXT OPERATION
EQUIVALENCE (PTREC(382),PIPF) /* IMAGE PROCESS F
EQUIVALENCE (PTREC(422),PNXF) /* CODE FOR NEXT OPERATION
EQUIVALENCE (PTREC(423),PIPG) /* IMAGE PROCESS G
EQUIVALENCE (PTREC(463),PNXG) /* CODE FOR NEXT OPERATION
EQUIVALENCE (PTREC(464),PIPH) /* IMAGE PROCESS H
EQUIVALENCE (PTREC(504),PNXH) /* CODE FOR NEXT OPERATION
```

C

```
C DATA-BASE NAME AND MIDAS INDEX POINTER ARRAY
  INTEGER PTNAM(16) /* PLAN TEST DATA FILE NAME
  INTEGER PTLEN /* NAME LENGTH
  INTEGER PTCHN /* CHAN. TO DATA BASE FILE
  INTEGER PTARR(14) /* MIDAS INFO ARRAY
  INTEGER PTERR /* MIDAS ERROR CODE
```

C

```
EQUIVALENCE (PTARR,PTERR)
COMMON /PTCOM/ PTREC,PTNAM,PTLEN,PTCHN,PTARR
```

C CCOM: COMMAND LIBRARY COMMON FILE

```
*****
*                                     *
*      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
*      VERSION 1.0      JUNE 1,1980                      *
*      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
*                                     *
*****
```

```
INTEGER CNPROC,CNWI,CSZW,CSZB
PARAMETER CNPROC=10      /* NO. OF PROCESSES FOR THIS COMMAND
PARAMETER CNWI=40        /* NO. OF WORDS PER PROCESS LINE
PARAMETER CSZW=CNPROC*CNWI+10 /* NO. OF WORDS PER RECORD
PARAMETER CSZB=CSZW*2    /* NO. OF BYTES PER RECORD
```

INTEGER CREC(CSZW)

```
INTEGER CNAM(10)      /*COMMAND NAME
INTEGER PROC1(CNWI)   /* FIRST PROCESS COMMAND LINE
INTEGER PROC2(CNWI)   /* 2ND PROCESS COMMAND LINE
INTEGER PROC3(CNWI)   /* 3RD PROCESS COMMAND LINE
INTEGER PROC4(CNWI)   /* 4TH PROCESS COMMAND LINE
INTEGER PROC5(CNWI)   /* 5TH PROCESS COMMAND LINE
INTEGER PROC6(CNWI)   /* 6TH PROCESS COMMAND LINE
INTEGER PROC7(CNWI)   /* 7TH PROCESS COMMAND LINE
INTEGER PROC8(CNWI)   /* 8TH PROCESS COMMAND LINE
INTEGER PROC9(CNWI)   /* 9TH PROCESS COMMAND LINE
INTEGER PROC10(CNWI)  /*10TH PROCESS COMMAND LINE
```

PRIMARY KEY = CNAM (1st. ELEMENT, 20 CHAR.)
SECONDARY KEY = NONE.

```
EQUIVALENCE (CREC(1), CNAM)      /* COMMAND NAME
EQUIVALENCE (CREC(11), PROC1)    /* FIRST PROCESS COMMAND LINE
EQUIVALENCE (CREC(51), PROC2)    /* 2ND PROCESS COMMAND LINE
EQUIVALENCE (CREC(91), PROC3)    /* 3RD PROCESS COMMAND LINE
EQUIVALENCE (CREC(131), PROC4)   /* 4TH PROCESS COMMAND LINE
EQUIVALENCE (CREC(171), PROC5)   /* 5TH PROCESS COMMAND LINE
EQUIVALENCE (CREC(211), PROC6)   /* 6TH PROCESS COMMAND LINE
EQUIVALENCE (CREC(251), PROC7)   /* 7TH PROCESS COMMAND LINE
EQUIVALENCE (CREC(291), PROC8)   /* 8TH PROCESS COMMAND LINE
EQUIVALENCE (CREC(331), PROC9)   /* 9TH PROCESS COMMAND LINE
EQUIVALENCE (CREC(371), PROC10)  /* 10TH PROCESS COMMAND LINE
```

DATA-BASE NAME AND MIDAS INDEX POINTER ARRAY

```
INTEGER CNNAM(16)      /* COMMAND DATA FILE NAME
INTEGER CLEN          /* NAME LENGTH
INTEGER CCHN          /* CHAN. TO DATA BASE FILE
INTEGER CARR(14)      /* MIDAS INFO ARRAY
INTEGER CERR          /* MIDAS ERROR CODE
```

```
EQUIVALENCE (CARR,CERR)
COMMON /CCOM/ CREC,CNNAM,CLEN,CCHN,CARR
```

C RCOM: COMMON FILE FOR RESULTS DATA-BASE PAGE 0001

C RCOM: COMMON FILE FOR RESULTS DATA-BASE.

*
* REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM *
* VERSION 1.0 JUNE 1,1980 *
* BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY *
*

INTEGER RSZW,RSZB,RNWT,RNT
PARAMETER RNT=10 /* No. TESTS
PARAMETER RNWT=25 /* No. OF WORDS IN TEST NAME
PARAMETER RSZW=RNT*RNWT+42 /* RESULTS RECORD SIZE IN WORDS
PARAMETER RSZB=RSZW*2 /* RESULTS RECORD SIZE IN BYTES

INTEGER RREC(RSZW) /* RESULTS RECORD

INTEGER RPN(10) /* RESULTS PART No.
INTEGER RIN(10) /* INSPECTION NAME
INTEGER RSN0(10) /* SERIAL No.
INTEGER RIID(5) /* INSPECTOR'S ID
INTEGER RDAT1(6) /* DATE-TIME
INTEGER RPF /* PASS-FAIL CODE
INTEGER RTN01(RNWT) /* TEST 01 NAME
INTEGER RTN02(RNWT) /* TEST 02 NAME
INTEGER RTN03(RNWT) /* TEST 03 NAME
INTEGER RTN04(RNWT) /* TEST 04 NAME
INTEGER RTN05(RNWT) /* TEST 05 NAME
INTEGER RTN06(RNWT) /* TEST 06 NAME
INTEGER RTN07(RNWT) /* TEST 07 NAME
INTEGER RTN08(RNWT) /* TEST 08 NAME
INTEGER RTN09(RNWT) /* TEST 09 NAME
INTEGER RTN10(RNWT) /* TEST 10 NAME

PRIMARY KEY: CONDENSATION OF PART-NUMBER, INSP. NAME, & SERIAL NO.
SEPARATED BY "/". 40 CHARACTERS LONG.
SECONDARY KEY: TEST NAMES

EQUIVALENCE (RREC(1), RPN) /* RESULTS PART No.
EQUIVALENCE (RREC(11), RIN) /* INSPECTION NAME
EQUIVALENCE (RREC(21), RSN0) /* SERIAL No.
EQUIVALENCE (RREC(31), RIID) /* INSPECTOR'S ID
EQUIVALENCE (RREC(36), RDAT1) /* DATE-TIME
EQUIVALENCE (RREC(42), RPF) /* PASS-FAIL CODE
EQUIVALENCE (RREC(43), RTN01) /* TEST 01 NAME
EQUIVALENCE (RREC(68), RTN02) /* TEST 02 NAME
EQUIVALENCE (RREC(93), RTN03) /* TEST 03 NAME
EQUIVALENCE (RREC(118), RTN04) /* TEST 04 NAME
EQUIVALENCE (RREC(143), RTN05) /* TEST 05 NAME
EQUIVALENCE (RREC(168), RTN06) /* TEST 06 NAME
EQUIVALENCE (RREC(193), RTN07) /* TEST 07 NAME
EQUIVALENCE (RREC(218), RTN08) /* TEST 08 NAME
EQUIVALENCE (RREC(243), RTN09) /* TEST 09 NAME
EQUIVALENCE (RREC(268), RTN10) /* TEST 10 NAME

DATA-BASE NAME AND MIDAS INDEX POINTER ARRAY

INTEGER RLNAM(16) /* RESULTS DATA FILE NAME
INTEGER RLLEN /* NAME LENGTH
INTEGER RLCHN /* CHAN. TO DATA BASE FILE
INTEGER RLARR(14) /* MIDAS INFO ARRAY
INTEGER RLERR /* MIDAS ERROR CODE

EQUIVALENCE (RLARR,RLERR)
COMMON /RCOM/ RREC,RLNAM,RLLEN,RLCHN,RLARR

C RTCOM: TEST RESULTS COMMON FILE

```
*****
*                                     *
*      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
*      VERSION 1.0      JUNE 1, 1980                    *
*      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY     *
*                                     *
*****
```

```

INTEGER RTSZW,RTSZB,RNWM,RNWI,RNMOT,RNING
PARAMETER RNMOT=5           /* No. OF MOTOR COMMANDS
PARAMETER RNING=10          /* No. OF IMAGE COMMANDS
PARAMETER RNWM=2            /* No. OF WORDS PER MOTOR COMMAND
PARAMETER RNWI=41           /* No. OF WORDS PER IMAGE COMMAND
PARAMETER RTSZW=RNMOT*RNWM+RNING*RNWI+298
PARAMETER RTSZB=RTSZW*2     /* No. OF BYTES IN PLAN TEST RECORD

```

INTEGER RTREC(RTSZW)

```

INTEGER RTNM(10)          /* TEST NAME
INTEGER RDA(39)           /* DESCRIPTION LINE A
INTEGER RDB(39)           /* DESCRIPTION LINE B
INTEGER RDC(39)           /* DESCRIPTION LINE C
INTEGER RDD(39)           /* DESCRIPTION LINE D
INTEGER RKXP              /* MASK X POSITION
INTEGER RKXS              /* MASK X SPEED
INTEGER RKYP              /* MASK Y POSITION
INTEGER RKYS              /* MASK Y SPEED
INTEGER RMXP              /* TABLE X POSITION
INTEGER RMXS              /* TABLE X SPEED
INTEGER RMYP              /* TABLE Y POSITION
INTEGER RMYS              /* TABLE Y SPEED
INTEGER RMRP              /* ROTATE POSITION
INTEGER RMRS              /* ROTATE SPEED
INTEGER RIPA(40)          /* IMAGE PROCESS A
INTEGER RNXA              /* CODE FOR NEXT OPERATION
INTEGER RIPB(40)          /* IMAGE PROCESS B
INTEGER RNXB              /* CODE FOR NEXT OPERATION
INTEGER RIPC(40)          /* IMAGE PROCESS C
INTEGER RNXC              /* CODE FOR NEXT OPERATION
INTEGER RIPO(40)          /* IMAGE PROCESS D
INTEGER RNXD              /* CODE FOR NEXT OPERATION
INTEGER RIPE(40)          /* IMAGE PROCESS E
INTEGER RNXE              /* CODE FOR NEXT OPERATION
INTEGER RI PF(40)         /* IMAGE PROCESS F
INTEGER RNXF              /* CODE FOR NEXT OPERATION
INTEGER RI PG(40)         /* IMAGE PROCESS G
INTEGER RNXG              /* CODE FOR NEXT OPERATION
INTEGER RI PH(40)         /* IMAGE PROCESS H
INTEGER RNXH              /* CODE FOR NEXT OPERATION
INTEGER RTRKY(20)         /* RESULTS KEY
INTEGER RTPF              /* PASS-FAIL CODE
INTEGER RTCM1(40)         /* COMMENT LINE 1
INTEGER RTCM2(40)         /* COMMENT LINE 2
INTEGER RTTNM(10)         /* TAPE NAME
INTEGER RTFNM             /* FILE No.
INTEGER RTIMN(25)         /* IMAGE NAME

```

PRIMARY KEY = SAME AS TEST NAME FROM RREC, AND RTIME
SECONDARY KEYS = NONE.

EQUIVALENCE	(RTREC(1), RTNM)	/* TEST NAME	
EQUIVALENCE	(RTREC(11), RDA)	/* DESCRIPTION LINE	A
EQUIVALENCE	(RTREC(50), RDB)	/* DESCRIPTION LINE	B
EQUIVALENCE	(RTREC(89), RDC)	/* DESCRIPTION LINE	C
EQUIVALENCE	(RTREC(128), RDD)	/* DESCRIPTION LINE	D
EQUIVALENCE	(RTREC(167), RKXP)	/* MASK X POSITION	
EQUIVALENCE	(RTREC(168), RKXS)	/* MASK X SPEED	
EQUIVALENCE	(RTREC(169), RKYP)	/* MASK Y POSITION	
EQUIVALENCE	(RTREC(170), RKYS)	/* MASK Y SPEED	
EQUIVALENCE	(RTREC(171), RMXP)	/* TABLE X POSITION	
EQUIVALENCE	(RTREC(172), RMXS)	/* TABLE X SPEED	
EQUIVALENCE	(RTREC(173), RMYP)	/* TABLE Y POSITION	
EQUIVALENCE	(RTREC(174), RMLS)	/* TABLE Y SPEED	

```
EQUIVALENCE (RTREC(175),RMRP) /* ROTATE POSITION
EQUIVALENCE (RTREC(176),RMRS) /* ROTATE SPEED
EQUIVALENCE (RTREC(177),RIPA) /* IMAGE PROCESS A
EQUIVALENCE (RTREC(217),RNXA) /* CODE FOR NEXT OPERATION
EQUIVALENCE (RTREC(218),RIPB) /* IMAGE PROCESS B
EQUIVALENCE (RTREC(258),RNXB) /* CODE FOR NEXT OPERATION
EQUIVALENCE (RTREC(259),RIPC) /* IMAGE PROCESS C
EQUIVALENCE (RTREC(299),RNXC) /* CODE FOR NEXT OPERATION
EQUIVALENCE (RTREC(300),RIPD) /* IMAGE PROCESS D
EQUIVALENCE (RTREC(340),RNXD) /* CODE FOR NEXT OPERATION
EQUIVALENCE (RTREC(341),RIPE) /* IMAGE PROCESS E
EQUIVALENCE (RTREC(381),RNXE) /* CODE FOR NEXT OPERATION
EQUIVALENCE (RTREC(382),RIPF) /* IMAGE PROCESS F
EQUIVALENCE (RTREC(422),RNXF) /* CODE FOR NEXT OPERATION
EQUIVALENCE (RTREC(423),RIPG) /* IMAGE PROCESS G
EQUIVALENCE (RTREC(463),RNXG) /* CODE FOR NEXT OPERATION
EQUIVALENCE (RTREC(464),RIPH) /* IMAGE PROCESS H
EQUIVALENCE (RTREC(504),RNXH) /* CODE FOR NEXT OPERATION
EQUIVALENCE (RTREC(505),RTRKY) /* KEY FOR RESULTS RECORD
EQUIVALENCE (RTREC(525),RTPF) /* PASS-FAIL CODE
EQUIVALENCE (RTREC(526),RTCM1) /* COMMENT 1
EQUIVALENCE (RTREC(566),RTCM2) /* COMMENT 2
EQUIVALENCE (RTREC(606),RTTNM) /* TAPE NAME
EQUIVALENCE (RTREC(616),RTFNM) /* FILE #
EQUIVALENCE (RTREC(617),RTIAN) /* IMAGE NAME
```

C
C

```
DATA-BASE NAME AND MIDAS INDEX POINTER ARRAY
INTEGER RTNAM(16) /* PLAN TEST DATA FILE NAME
INTEGER RTLEN /* NAME LENGTH
INTEGER RTCHN /* CHAN. TO DATA BASE FILE
INTEGER RTARR(14) /* MIDAS INFO ARRAY
INTEGER RTERR /* MIDAS ERROR CODE
```

C

```
EQUIVALENCE (RTARR,RTERR)
COMMON /RTCOM/ RTREC,RTNAM,RTLEN,RTCHN,RTARR
```

C TCOM: COMMON FILE FOR TAPE DATA-BASE.

```
C      INTEGER TREC(TSZW)      /* TAPE RECORD
```

```

C
C
C
C
PRIMARY KEY: TAPE NAME
SECONDARY KEY: IMAGE NAME = TEST-RESULTS KEYWORD.

```

```

C
C
C DATA-BASE NAME AND MIDAS INDEX POINTER ARRAY

```

```

EQUIVALENCE (TPARR,TPERR)
COMMON /TCON/ TREC,TPNAM,TPLEN,TCHN,TPARR,TECNT

```

C AI: AUTOMATED INSPECTION - TOP LEVEL

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM
C *      VERSION 1.0      JUNE 1,1980
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY
C *
C *****
C *THIS IS THE MAIN PROGRAM FOR THE AUTOMATED INSPECTION SYSTEM *
C *FUNCTION: 1. OPEN DATA BASE FILES *
C *          2. INITIATE, INSPECTION, PLANNING, OR RETRIEVAL *
C *          3. OR CLOSE DATA BASE FILES AND EXIT *
C
C $INSERT PCOM
C $INSERT PTCOM
C $INSERT RCOM
C $INSERT RTCOM
C $INSERT TCOM
C $INSERT SYSCOM>CRTCTRL
C
C      COMMON/OREZ/ZERO(8191)
C
C      LOGICAL EQUIP
C
C      EQUIP = .FALSE.
C
C SET UP TERMINAL CODES, OPEN DATA BASE FILES
C-----
C      CALL INITAL(IERR)
C      IF (IERR .NE. 0) CALL EXIT
C
C GET FUNCTION: INSPECT,PLAN,RETREIVE HISTORY,CLOSE FILES
C-----
10      CALL GFUNCT(MENU)
C      IF (MENU .EQ. 0) GOTO 950
C      GOTO (20,30,40), MENU
C      GOTO 950
C
C PERFORM INSPECTION
C-----
20      CALL INSP(EQUIP,IERR)
C      IF (IERR) 950,10,950
C
C PLAN
C-----
30      CALL PLAN(IERR)
C      IF (IERR) 950,10,950
C
C RETRIEVE HISTORY
C-----
40      CALL RETR(EQUIP,IERR)
C      IF (IERR) 950,10,950
C
C CLOSE DATA BASE FILES
C-----
950      CALL CLOALL
C
C EXIT
C-----
1000     CALL EXIT
C      END

```

C APDAT: APPEND DATE TO KEYWORD

C APDAT: APPEND DATE TO KEYWORD

```
*****
*                                     *
*      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
*      VERSION 1.0      JUNE 1,1980                      *
*      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
*      *****
*GENERAL PURPOSE SUBROUTINE TO APPEND DATE AND TIME TO A STRING *
*DATE/TIME IS FORM: MMDDYYHHMMSS                          *
*
      SUBROUTINE APDAT(KEY,LEN)
C $INSERT SYSCOM>A$KEYS
      REAL*8 DD,DT,DUMMY(2)
      DIMENSION IDT(8)
      INTEGER BIAS,BI
      EQUIVALENCE (IDT,DD)
      EQUIVALENCE (IDT(5),DT)
C
      DD = DATE$(DUMMY)      /*GET DATE
      TI = TIME$(DT)         /*GET TIME
      CALL RMNAB(IDT,16)     /* REMOVE NON-ALPHABETICAL CHARACTERS
      IE = LEN
      IS = IE - 11
      IF (IS .LE. 0) RETURN
      CALL MSUB$(IDT,12,1,12,KEY,LEN,IS,IE) /* APPEND THE DATE
      RETURN
      END
```

C CLOALL: CLOSE ALL OPEN CHANNELS

C CLOALL: CLOSE ALL OPEN CHANNELS

```
*****
*                                     *
*      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
*      VERSION 1.0      JUNE 1,1980                      *
*      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
*      *****
*
      SUBROUTINE CLOALL
C $INSERT SYSCOM>A$KEYS
      DO 100 ICH=1,62
      IF (.NOT. UNIT$(ICH)) GOTO 100
      IF (.NOT. CLOS$(ICH)) GOTO 100
100  CONTINUE
      RETURN
      END
```

C CNEWC: CREATE NEW IMAGE PROCESSOR CO PAGE 0001

C CNEWC: CREATE NEW IMAGE PROCESSOR COMMAND

C *****
C *
C * REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM *
C * VERSION 1.0 JUNE 1,1980 *
C * BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY *
C *****

C *CALLED FROM: PLAN *
C * 1. INSERT COMMAND FILE IN DATA BASE *

C SUBROUTINE CNEWC(IER)

C \$INSERT CCOM
C \$INSERT SYSCOM>PARAM.K
C \$INSERT SYSCOM>A\$KEYS

C INTEGER TMPR(CSZM),NAME(16)

C DATA NAME /'DEMO.SCREEN.FT' //

C CALL TONL

C Open a channel to the screen template file

C CALL VOPEN\$(NAME,32,1,IFCH,IER) /* OPEN FOR READ
C IF (IER .NE. 0) GOTO 9200

C IS = 318
C IE = 330
C NEWSR = 2 /* DO NOT ERASE THE SCREEN TO START

C Clear record buffer

C 10 CALL ZFIL(TMPR,CSZB,0)

C 20 CALL PEDIT(IFCH,TMPR,IS,IE,NEWSR)
C CALL ZMVD(TMPR,CREC,CSZB)

C Check to see if record already exists.

C IFLAG = FL\$RET
C CALL NEXT\$(CCHN,CREC,CNAM,CARR,IFLAG,\$200,0,0,0,0)
C CALL TONL
C CALL TNOU ('COMMAND ALREADY EXISTS',24)
C CALL TONL
C GO TO 300

C Store the record

C 200 CALL ADD1\$(CCHN,CREC,CNAM,CARR,IFLAG,\$9000,0,0,0,0)

C MORE?

C IER = 0
C 300 IF (YSN0\$(A('More',4,A\$DNO)) GOTO 20

C CLOSE THE SCREEN CHANNEL NO.

C 400 CALL CLOS\$(IFCH)
C RETURN

C 9000 IER = CERR
C WRITE (1,9001) CERR,CNAM
C 9001 FORMAT('MIDAS ERROR =',I3,' KEY = ',I5A2)
C GOTO 400

C 9200 WRITE (1,9201) IER,NAME
C 9201 FORMAT('ERROR ',I3,', OPENING FILE ',I6A2)
C RETURN

C CNENC: CREATE NEW IMAGE PROCESSOR C0 PAGE 0002

C END

C CNEWP: CREATE NEW PLAN ENTRY

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0      JUNE 1,1980                      *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C *****
C *
C *      *CALLED BY: PLAN                                     *
C *      *FUNCTION: 1. GET PART#,INSP NAME,TEST NAMES FROM OPERATOR *
C *                  2. CREATE PLAN KEYWORD                  *
C *                  3. STORE PLAN IN DATA BASE             *
C *

```

SUBROUTINE CNEWP(IER)

```

$INSERT PCOM
$INSERT SYSCOM>PARM.K
$INSERT SYSCOM>A$KEYS

```

```

C      INTEGER TMPR(PSZW),NAME(16)
C      INTEGER KEYO(15)      /* PRIMARY KEY
C
C      DATA NAME /'DEMO.SCREEN.FT      '/'
C
C      CALL TONL
C
C      Open a channel to the screen template file
C-----
C      CALL VOPEN$(NAME,32,1,IFCH,IER)      /* OPEN FOR READ
C      IF (IER .NE. 0) GOTO 9200
C
C      IS = 50
C      IE = 66
C      NEWSCR = 2      /* DO NOT ERASE THE SCREEN TO START
C
C      Clear record buffer
C-----
C      CALL ZFIL(TMPR,PSZB,0)
C
C      CALL PEDIT(IFCH,TMPR,IS,IE,NEWSCR)
C      CALL ZAVD(TMPR,PREC,PSZB)
C
C      Check for internal consistency
C-----
C      IF (PNT .LE. 10) GOTO 27
C      WRITE (1,26)
C      FORMAT(/,'10 IS MAXIMUM NUMBER OF TESTS!','/)
C      GOTO 20
C
C      IF (TMPR(11) .NE. 0) GOTO 28
C      WRITE(1,31)
C      FORMAT(1X,'INSPECTION NAME MANDATORY!')
C      GOTO 20
C
C      DO 30 I=1,PNT
C      IBIAS = 22 + (I-1)*PNWT
C      IF (TMPR(IBIAS) .NE. 0) GOTO 30
C      WRITE (1,29)
C      FORMAT(/,'NUMBER OF TESTS NOT CONSISTENT!','/)
C      GOTO 20
C      CONTINUE
C
C      Create a primary keyword.
C-----
C      CALL CRPKEY(PPN,20,PIN,20,0,0,0,0,KEYO,30)
C
C      LEN = LSIZE(KEYO,30)
C
C      Check to see if record already exists.
C-----

```

PAGE 0002

2020

CCCD

CC

5

0

5

i

1

1

C CNEWT: CREATE NEW TEST ENTRY PAGE 0001

C CNEWT: CREATE NEW TEST ENTRY

```
C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0 JUNE 1,1980                          *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C * *****
C *
C * CALLED FROM: PLAN *
C * FUNCTION: 1. GET TEST NAME, DESCRIPTION, MOTOR POSITIONS, *
C *           AND IMAGE PROCESSES FORM PLANNER *
C *           2. INSERT TEST IN DATA BASE *
C *
```

SUBROUTINE CNEWT(IER)

\$INSERT PTCOM
\$INSERT SYSCOM>PARN,K
\$INSERT SYSCOM>A\$KEYS

```
C      INTEGER TMPR(PTSZW),NAME(16)
C      INTEGER KEYD(10)      /* PRIMARY KEY
C
C      DATA NAME /'DEMO.SCREEN.FT      '/'
C
C      CALL TONL
C      Open a channel to the screen template file
C-----
C      CALL VOPEN$(NAME,32,1,IFCH,IER)      /* OPEN FOR READ
C      IF (IER.NE. 0) GOTO 9200
C
C      IS = 75
C      IE = 98
C      NEWSCR = 2      /* DO NOT ERASE THE SCREEN TO START
C
C      Clear record buffer
C-----
C      CALL ZFIL(TMPR,PTSZB,0)
C
C      CALL PEDIT(IFCH,TMPR,IS,IE,NEWSCR)
C      CALL ZAVD(TMPR,PTREC,PTSZB)
C      WRITE(1,1098)PTNM      /* ***CHECKOUT***
C      1098 FORMAT(1X,'PTNM FROM CNEWT=',10A2)
C
C      Check to see if record already exists.
C-----
C      IFLAG = FL$RET
C      CALL NEXT$(PTCHN,PTREC,PTNM,PTARR,IFLAG,$200,0,0,0,0)
C      CALL TONL
C      CALL TNOU ('PLAN TEST ALREADY EXISTS',24)
C      CALL TONL
C      GO TO 300
C
C      Store the record
C-----
C      200 CALL ADD1$(PTCHN,PTREC,PTNM,PTARR,IFLAG,$9000,0,0,0,0)
C      CALL TDUMP(PTREC,PTSZB)
C
C      MORE?
C-----
C      IER = 0
C      300 IF (YSNO$(A('More',4,A$DN0)) GOTO 20
C
C      CLOSE THE SCREEN CHANNEL NO.
C-----
C      400 CALL CLOS$(A(IFCH)
C      RETURN
C
C      9000 IER = PTERR
C      WRITE (1,9001) PTERR,PTNM
C      9001 FORMAT('MIDAS ERROR =',13,' KEY = ',15A2)
```

A5-70

GOTO 400

```

C
C
9200 WRITE (1,9201) IER,NAME
9201 FORMAT('ERROR ',I3,', ' OPENING FILE ',I6A2)
      RETURN
C
C
9900 FORMAT('ERROR = ',I3)
      END
C

```

C CRPKEY: CREATE PLAN KEYWORD

```

C
C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM
C *      VERSION 1.0      JUNE 1,1980
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY
C *
C *****
C *GENERAL PURPOSE SUBROUTINE TO CREATE A KEYWORD
C *FUNCTION: 1. CONCATENATE TEXT STRINGS WITH A '/' INBETWEEN
C *           2. APPENDS DATE AND TIME TO END
C
C SUBROUTINE CRPKEY(TEXT1,MAX1,TEXT2,MAX2,TEXT3,MAX3,TEXT4,MAX4,
C KEY,MAXKEY)
C
C $INSERT SYSCOM>A$KEYS
C
C CALL ZFIL(KEY,MAXKEY,' ')
C LEN1 = LSIZE(TEXT1,MAX1) /*FIND LENGTH OF FIRST STRING
C CALL ZMV(TEXT1,LEN1,KEY,MAXKEY) /*MOVE FIRST STRING INTO KEY
C IF (TEXT2 .EQ. 0) GOTO 1000
C IPOS = LEN1 + 1
C CALL MCHR$(KEY,IPOS,'/',1) /*INSERT "/" INTO KEY
C IPOS = IPOS + 1
C LEN2 = LSIZE(TEXT2,MAX2) /*FIND LENGTH OF SECOND STRING
C CALL MSUB$(TEXT2,MAX2,1,LEN2,KEY,MAXKEY,IPOS,MAXKEY)
C IF (MAX3 .LE. 0) GOTO 1000
C IF (TEXT3 .EQ. 0) GOTO 1000
C
C IPOS = LSIZE(KEY,MAXKEY) + 1
C CALL MCHR$(KEY,IPOS,'/',1) /*INSERT "/" INTO KEY
C IPOS = IPOS + 1
C CALL MSUB$(TEXT3,MAX3,1,MAX3,KEY,MAXKEY,IPOS,MAXKEY)
C IF (MAX4 .LE. 0) GOTO 1000
C
C IPOS = LSIZE(KEY,MAXKEY) + 1
C IF (TEXT4 .EQ. 0) GOTO 1000
C CALL MCHR$(KEY,IPOS,'/',1) /*INSERT "/" INTO KEY
C IPOS = IPOS + 1
C CALL MSUB$(TEXT4,MAX4,1,MAX4,KEY,MAXKEY,IPOS,MAXKEY)
1000 RETURN
      END

```

C CRRES: CREATE RESULTS RECORD

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0      JUNE 1,1980                      *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C * *****
C
C *CALLED BY: DINSP                                         *
C *FUNCTION:  1. CREATE TEST RESULT NAMES (CRPKEY)          *
C *           2. FILL MAIN RESULT RECORD FROM PLAN (ZAVD)   *
C
C      SUBROUTINE CRRES(SERIAL,IID,IERR)
C
C      $INSERT FCOM
C      $INSERT PTCOM
C      $INSERT RCOM
C      $INSERT SYSCOM>PARM.K
C
C      INTEGER BIAS,BI,IBUF(10),IKEY(RNWT),BIAS2,BI2,TBIAS
C
C      NSIZ = RNWT*2
C
C      TBIAS = 22                      /* TEST NAME BIAS
C
C      DO 10 I = 1,PNT                  /*CREATE TESTNAMES FOR RCOM
C      CALL NEXT$(PTCHN,PTREC,PREC(TBIAS),PTARR,FL$RET,$9000,0,0,0,0)
C      TBIAS = TBIAS + PNWT
C      BIAS = 2*(22 + (I-1)*PNWT) - 1
C      BIAS2 = BIAS + 2*PNWT - 1
C      BI = 2*(43 + (I-1)*RNWT) - 1
C      BI2 = BI + 2*RNWT - 1
C
C      PUT PREC(BIAS) INTO IBUF
C
C      CALL MSUB$(PREC,PSZB,BIAS,BIAS2,IBUF,20,1,20)
C
C      CREATE KEY INTO IKEY
C
C      CALL CRPKEY(PPN,20,PIN,20,SERIAL,20,IBUF,20,IKEY,NSIZ)
C      CALL APDAT(IKEY,NSIZ)
C
C      MOVE IKEY INTO RREC(BI)
C
C      CALL MSUB$(IKEY,NSIZ,1,NSIZ,RREC,RSZB,BI,BI2)
10  CONTINUE
C
C      CALL ZAVD(PPN,RPN,20)
C      CALL ZAVD(PIN,RIN,20)
C      CALL ZAVD(IID,RIID,10)
C      CALL ZAVD(SERIAL,RSNO,20)
C      RETURN
C
9000  IERR = PTERR
      WRITE(1,9010)IERR,TBIAS
9010  FORMAT(1X,'ERROR ',I2,' BIAS = ',I3,' IN CRRES CHECK PLANTEST')
      RETURN
      END

```

C CRTRES: CREATE A TEST RESULTS RECORD. PAGE 0001

C CRTRES: CREATE A TEST RESULTS RECORD.

```
C *****
C *
C *          REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM          *
C *          VERSION 1.0 JUNE 1,1980                               *
C *          BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY         *
C *****
C *CALLED BY: DINSP
C *FUNCTION: 1. MOVE TEST PLAN RECORD TO TEST RESULTS RECORD      *
C *          2. MOVE IMAGE NAME TO TEST RESULTS RECORD           *
C *          3. MOVE TAPE NAME TO TEST RESULTS RECORD             *
```

C SUBROUTINE CRTRES(I,RTKEY)

C I = TEST SEQUENCE NO.

```
C $INSERT PTCOM
C $INSERT RCOM
C $INSERT RTCOM
C $INSERT TCOM
```

C IBIAS = 43 + (I-1)*RNWT
C IMGSIZ = TNWT*2

C ZERO THE RECORD

C CALL ZFIL(RTREC,RTSZB,0)

C MOVE A COPY OF TEST PLAN TO RESULTS RECORD

C CALL ZMVD(PTREC,RTREC,PTSZB)

C STORE THE IMAGE NAME

```
C CALL ZMVD(RREC(IBIAS),RTIMN,IMGSIZ)
C CALL ZMVD(RTIMN,RTKEY,IMGSIZ)
```

C PLUS THE TAPE NAME & NUMBER

```
C CALL ZMVD(TNAM,RTTNM,20)
C RTFNM = TFCNT
```

C RETURN
END

C DBPRM: INSERT DATA BASE PARAMETERS

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0      JUNE 1,1980                      *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C *
C *****
C * CALLED BY:  INITIAL                                     *
C * FUNCTION:  LOAD DATA BASE NAMES, NAMELENGTHS INTO COMMON FILES *
C
C      SUBROUTINE DBPRM
C
C      $INSERT FCOM
C      $INSERT PTCOM
C      $INSERT RCOM
C      $INSERT RTCOM
C      $INSERT TCOM
C      $INSERT CCOM
C
C      INSERT NAMES
C      PLEN = 8
C      CALL ZMV('PLAN.DB',PLEN,PLNAM,32)
C      PTLEN = 11
C      CALL ZMV('PLNTEST.DB',PTLEN,PTNAM,32)
C      RLEN = 10
C      CALL ZMV('INSRES.DB',RLEN,RLNAM,32)
C      RTLEN = 12
C      CALL ZMV('INSTESTR.DB',RTLEN,RTNAM,32)
C      TPLEN = 8
C      CALL ZMV('TAPE.DB',TPLEN,TPNAM,32)
C      CLEN = 11
C      CALL ZMV('COMMAND.DB',CLEN,CNNAM,32)
C      RETURN
C      END

```

C DINSF: DO THE INSPECTION

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM
C *      VERSION 1.0      JUNE 1,1980
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY
C *
C *****
C
C *CALLED BY: INSP
C *FUNCTION: 1. INITIALIZE IMAGE PROCESSOR, IF REQUIRED (INTIP)
C *          2. INITIALIZE MOTORS, IF REQUIRED (INTMOT)
C *          3. INITIALIZE MAG TAPE, IF REQUIRED (INTAP)
C *          4. CREATE A MAIN RESULT RECORD (CRRES)
C *          5. DO THE FOLLOWING FOR EACH TEST:
C *              A. GET TEST PLAN (GPTST)
C *              B. CREATE TEST RESULT RECORD (CRTRES)
C *              C. POSITION MOTORS (PMOT)
C *              D. PROCESS THE IMAGE (PING)
C *              E. SAVE THE TEST RESULTS (SVRES)
C *          6. SAVE THE MAIN TEST RESULTS (SVMRES)
C
C      SUBROUTINE DINSF(IID,SERIAL,EQUIP,DEFALT,IERR)
C
C $INSERT PCOM
C $INSERT RCOM
C
C      LOGICAL EQUIP
C      INTEGER RTKEY(RNWT)
C
C      IF (EQUIP) GOTO 10
C      INITIALIZE IMAGE PROCESSOR
C-----
C      CALL INTIP(IERR)
C      IF (IERR.NE. 0) RETURN
C      INITIALIZE MOTORS
C-----
C      CALL INTMOT(IERR)
C      IF (IERR.NE. 0) RETURN
C      INITIALIZE TAPE
C-----
C 10      CALL INTAP(EQUIP,NOTAPE,IERR)
C      IF (IERR.NE. 0) RETURN
C      EQUIP = .TRUE.
C      SET OVERALL PASS-FAIL CODE TO ZERO
C-----
C      RPF = 0
C      CREATE THE RESULTLS MAIN RECORD
C-----
C 20      CALL CRRES(SERIAL,IID,IERR)
C      IF (IERR.NE. 0) RETURN
C
C      DO FOR EACH TEST
C
C      DO 30 I = 1,PNT
C      GET PLAN'S TEST
C-----
C 21      CALL GPTST(I,IERR)
C      IF (IERR.NE. 0) RETURN
C      LOAD RTCOM
C-----
C      CALL CRTRES(I,RTKEY)
C      POSITION MOTORS
C-----
C      CALL PMOT
C      PERFORM IMAGE PROCESSING PER TEST
C-----
C      CALL PING(I,NOTAPE,IERR)
C      SAVE THE INSPECTION TEST RESULTS
C-----
C      CALL SVRES(DEFALT,NOTAPE,I,RTKEY,IERR)
C      IF (IERR.NE. 0) RETURN
C 30      CONTINUE
C

```

C DINSF: DO THE INSPECTION

PAGE 0002

C SAVE THE RESULTS MAIN RECORD

C-----
CALL SVMRES(1ER)
RETURN
END

C DLTC: DELETE IMAGE PROCESSOR COMMAND

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0   JUNE 1,1980                        *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C *
C *****
C *CALLED BY: PLAN
C *FUNCTION: 1. GET COMMAND NAME FROM OPERATOR
C *          2. FIND RECORD IN COMMAND DATA BASE
C *          3. DELETE THE COMMAND RECORD
C
C ARG 1: IERR      INTEGER
C
C      SUBROUTINE DLTC(IERR)
C
C $INSERT CCOM
C $INSERT SYSCOM>PARM.K
C
C      INTEGER CNAME(10)
C      LOGICAL MORE
C
C GET TEST NAME
C
C 10      WRITE(1,20)
C 20      FORMAT(1X,'COMMAND NAME?')
C      READ(1,25,ERR = 10) CNAME
C 25      FORMAT(10A2)
C
C FIND THE RECORD
C
C      IFLAG = FL$RET
C      CALL NEXT$(CCHN,CREC,CNAME,CARR,IFLAG,$8000,0,0,0,0)
C
C DELETE THE RECORD
C
C      IFLAG = FL$RET + FL$USE
C      CALL DELET$(CCHN,CREC,CNAME,CARR,IFLAG,$9000,0,0,0,0)
C      WRITE(1,30)CNAME
C 30      FORMAT(1X,10A2,' DELETED')
C      CALL DMORE(MORE)
C      IF (MORE) GOTO 10
C      RETURN
C
C 8000      IF (CERR .NE. 7) GOTO 8005
C      WRITE(1,8004) CNAME
C 8004      FORMAT(10A2,' COMMAND NOT FOUND')
C      RETURN
C
C 8005      WRITE(1,8010)IERR,CNAME
C 8010      FORMAT(1X,'MIDAS ERROR ',I3,' IN FINDING ',10A2)
C      RETURN
C
C 9000      IERR = CERR
C      WRITE(1,9010)IERR,CNAME
C 9010      FORMAT(1X,'MIDAS ERROR ',I2,' IN DELETING ',10A2)
C      RETURN
C      END

```

C DLTP: DELETE A PLAN

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0 JUNE 1,1980                          *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C *
C *****
C *CALLED BY: PLAN
C *FUNCTION: 1. GET PART#,INSPECTION NAME FROM OPERATOR      *
C *          2. CREATE THE PLAN KEYWORD                      *
C *          3. GET THE RECORD FROM DATA BASE              *
C *          4. DELETE SECONDARY AND PRIMARY KEYWORDS AND RECORD *
C ARG 1: IERR  INTEGER
C      SUBROUTINE DLTP(IERR)
C $INSERT PCOM
C $INSERT SYSCOM>PARM.K
C $INSERT SYSCOM>A$KEYS
C $INSERT SYSCOM>KEYS.F
C      INTEGER IBUF(20),DPART(10),DIIN(10),DKEY(20),BIAS,PTEMP(PSZW)
C      LOGICAL ZCM
C      EQUIVALENCE (IBUF,DPART)
C      EQUIVALENCE (IBUF(11),DIIN)
C      CALL ZFIL(IBUF,40,0) /* ZERO THE PEDIT BUFFER
C      CALL VOPEN$('DEMO.SCREEN.FT',14,1,ICH,IERR) /* OPEN PEDIT CHANNEL
C      IF (IERR .NE. 0) GOTO 9000
C SET UP PEDIT PARAMETERS
C      IERR = 0
C      IS = 51
C      IE = 52
C      NEWSCR = 2
C GET PART # AND INSPECTION NAME FROM PEDIT
C      CALL PEDIT(ICH,IBUF,IS,IE,NEWSCR)
C CLOSE PEDIT RECORD
C      CALL CLOS%A(ICH)
C MAKE IBUF INTO KEY
C      CALL CRPKEY(DPART,20,DIIN,20,0,0,0,0,DKEY,40)
C GET THE RECORD
C      CALL NEXT$(PCHN,PREC,DKEY,PLARR,FL$RET,$9050,0,0,0,0)
C      BIAS = 22
C 10 IF (PREC(BIAS) .EQ. 0) GOTO 20
C DELETE SECONDARY KEYS
C FIND SECONDARY KEYS
C      IFLAG = FL$RET
C 30 CALL NEXT$(PCHN,PTEMP,PREC(BIAS),PLARR,IFLAG,$20,1,0,0,0)
C COMPARE PREC AND PTEMP
C      IFLAG = FL$RET + FL$USE
C      ISIZE = PSZB
C      IF (.NOT. ZCM(PTEMP,ISIZE,PREC,ISIZE,ICODE)) GOTO 30
C DELETE THE SECONDARY KEY

```

```

C      CALL DELET$(PCHN,PREC,PREC(BIAS),PLARR,IFLAG,$20,1,0,0,0)
C      BIAS = BIAS + PNWT
C      GOTO 10
C
C  DELETE PRIMARY KEY AND RECORD
C
20      IFLAG = FL$RET
      CALL DELET$(PCHN,PREC,DKEY,PLARR,IFLAG,$9100,0,0,0,0)
      WRITE(1,80)DKEY
      FORMAT(1X,'DELETED ',20A2)
      RETURN
9000     WRITE(1,9010)IERR
9010     FORMAT(1X,'ERROR ',I2,' IN OPENING DEMO.SCREEN.FT')
      RETURN
C
9050     IF (PLERR .NE. 7) GOTO 9065
      WRITE(1,9060)
      FORMAT(1X,'RECORD NOT FOUND')
      RETURN
9065     WRITE(1,9070)PLERR,PREC(BIAS)
9070     FORMAT(1X,'MIDAS ERROR ',I2,' IN FINDING KEY ',20A2)
      IERR = PLERR
      RETURN
C
9100     WRITE(1,9110) PLERR,DKEY
9110     FORMAT(1X,'MIDAS ERROR ',I2,' IN DELETING ',20A2)
      IERR = PLERR
      RETURN
      END

```

C DLTR: DELETE RESULTS

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM
C *      VERSION 1.0      JUNE 1,1980
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY
C *
C *****
C
C *CALLED FROM: PLAN
C *FUNCTION: 1. GET PART#,INSPECTION NAME FROM OPERATOR
C *          2. FIND A MATCH IN DATA BASE
C *          3. IF CORRECT RESULT, DELETE IT AND THE RESULT TESTS *
C
C      SUBROUTINE DLTR(IER)
C
C      WRITE (1,1)
1      FORMAT('** DELETE RESULTS IS A FORBIDDEN TRANSACTION **')
      IER = 0
      RETURN
      END

```

C DLTT: DELETE TEST PLAN

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0      JUNE 1,1980                      *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C *
C *****
C *CALLED BY: PLAN
C *FUNCTION: 1. GET TEST NAME FROM OPERATOR
C *           2. FIND RECORD IN TEST DATA BASE
C *           3. DELETE THE TEST RECORD
C
C ARG 1: IERR      INTEGER
C
C      SUBROUTINE DLTT(IERR)
C
C $INSERT PTCON
C $INSERT SYSCOM>PARAM.K
C
C      INTEGER PTNAME(10)
C      LOGICAL MORE
C
C GET TEST NAME
C
10      WRITE(1,20)
20      FORMAT(1X,'TEST NAME?')
      READ(1,25,ERR = 10) PTNAME
25      FORMAT(10A2)
C
C FIND THE RECORD
C
      IFLAG = FL$RET
      CALL NEXT$(PTCHN,PTREC,PTNAME,PTARR,IFLAG,$8000,0,0,0,0)
C
C DELETE THE RECORD
C
      IFLAG = FL$RET + FL$USE
      CALL DELET$(PTCHN,PTREC,PTNAME,PTARR,IFLAG,$9000,0,0,0,0)
30      WRITE(1,30)PTNAME
      FORMAT(1X,10A2,' DELETED')
      CALL DMORE(MORE)
      IF (MORE) GOTO 10
      RETURN
C
8000      IF (PTERR .NE. 7) GOTO 8005
      WRITE(1,8004) PTNAME
8004      FORMAT(10A2,' TEST PROCEDURE NOT FOUND')
      RETURN
C
8005      WRITE(1,8010)IERR,PTNAME
8010      FORMAT(1X,'MIDAS ERROR ',I3,' IN FINDING ',10A2)
      RETURN
C
9000      IERR = PTERR
      WRITE(1,9010)IERR,PTNAME
9010      FORMAT(1X,'MIDAS ERROR ',I2,' IN DELETING ',10A2)
      RETURN
      END

```

C DMORE: DO YOU WANT TO DO MORE?

PAGE 0001

C DMORE: DO YOU WANT TO DO MORE?

```
C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0      JUNE 1,1980                      *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C * *****
C *
C *      *GENERAL PURPOSE SUBROUTINE TO SEE IF OPERATOR NEEDS TO DO *
C *      * A FUNCTION AGAIN *
C
C      SUBROUTINE DMORE(MORE)
C
C $INSERT SYSCOM>A$KEYS
C
C      LOGICAL MORE,YSNO$A
C
C      MORE = YSNO$A('More',4,A$DNO)
C      RETURN
C      END
```

C DSPLD: DISPLAY OR PRINT PLAN OR RESULT D PAGE 0001

C DSPLD: DISPLAY OR PRINT PLAN OR RESULT DATA.

```
C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0      JUNE 1,1980                      *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C *
C *****
C *CALLED FROM: PLAN
C *FUNCTION: 1. GET DISPLAY FUNCTION FROM OPERATOR (GDFNCT) *
C *          2. CALL ONE OF FOLLOWING SUBROUTINES           *
C *              DSPP - DISPLAY A PLAN                     *
C *              DSPPT - DISPLAY A TEST                     *
C *              DSPR - DISPLAY MAIN RESULTS                 *
C *              DSPRT - DISPLAY TEST RESULTS                *
C *              DSPT - DISPLAY TAPE RECORD                 *
C *              LISTIX- LIST RECORD BY INDEX NO.           *
C *          3. IF MAILSTOP GIVEN, SPOOL INFO TO LINE PRINTER *
C ARG 1: IERR      INTEGER
C
C      SUBROUTINE DSPLD(IERR)
C
C          INTEGER SPFNAM(7), SPCHN
C
C      GET DISPLAY FUNCTION
C10      CALL GDFNCT(MENU, SPCHN, SPFNAM)
C
C          IF (MENU .EQ. 0) RETURN
C          GOTO (100,200,300,400,500,600,700,800), MENU
C
C      DISPLAY MAIN PLAN
C100      CALL DSPP(SPCHN)
C          GOTO 8000
C
C      DISPLAY PLAN TEST
C200      CALL DSPPT(SPCHN)
C          GOTO 8000
C
C      DISPLAY MAIN RESULTS
C300      CALL DSPR(SPCHN)
C          GOTO 8000
C
C      DISPLAY TEST RESULTS
C400      CALL DSPRT(SPCHN)
C          GOTO 8000
C
C      DISPLAY TAPE RECORD
C500      CALL DSPT(SPCHN)
C          GOTO 8000
C
C      LIST INDEX
C600      CALL LISTIX(SPCHN)
C          GOTO 8000
C
C      SOME OTHER FUNCTION
C700      GOTO 8000
C
C      SOME OTHER FUNCTION
C800      GOTO 8000
C
C      PRINT RESULTS?
```

C DSPLD: DISPLAY OR PRINT PLAN OR RESULT D PAGE 0002

```
8000  IF (SPCHN .EQ. 0) GOTO 10
      CALL SPOLIT(SPCHN,SPFNAM)
      GOTO 10
      END
```

C DSPP: DISPLAY MAIN PLAN

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0      JUNE 1,1980                      *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C *
C *****
C *CALLED BY: DSPLD *
C *FUNCTION: 1. GET PART#,INSPECTION NAME FROM OPERATOR (PEDIT) *
C *          2. FIND RECORD IN DATA BASE                    *
C *          3. PRINT/DISPLAY THE RECORD                     *
C *          4. IF OPERATOR WANTS TO SEE A TEST:            *
C *              A. FIND THE RECORD                          *
C *              B. DISPLAY IT                                *
C
C      SUBROUTINE DSPP(REPCHN)
C
C      ARG 1: REPCHN      INTEGER, REPORT CHANNEL
C
C      $INSERT PCOM
C      $INSERT PTCOM
C      $INSERT SYSCOM>PARM,K
C      $INSERT SYSCOM>A$KEYS
C
C      INTEGER REPCHN,PKEY(20)
C
C      OPEN SCREEN TEMPLATE
C
C      CALL VOPEN('DEMO.SCREEN.FT',14,1,IPCHN,IER)
C      IF (IER.NE. 0) GOTO 9000
C
C      GET PART NUMBER, INSPECTION NAME
C
C      N = -1
C      IS = 49
C      IE = 52
C      NEWSCR = 2      /* DO NOT ERASE SCREEN TO START
C      CALL ZFIL(PREC,PSZB,0)
C      CALL PEDIT(IPCHN,PREC,IS,IE,NEWSCR)
C
C      CALL CRPKEY(PFN,20,FIN,20,0,0,0,0,PKEY,40)      /* CREATE KEYWORD
C      NCHAR = LSIZE(PKEY,40) /* GET LENGTH OF KEYWORD
C      IFLAG = FL$RET + FL$BIT
C
C      FIND THE RECORD
C
C      10      CALL NEXT$(PCHN,PREC,PKEY,PLARR,IFLAG,$9100,0,0,0,NCHAR)
C
C      PRINT/DISPLAY THE RECORD
C
C      IS = 49
C      IE = 64
C      NEWSCR = 2
C      WRITE (1,11)
C      FORMAT(/)
C      11      CALL RPTGEN(REPCHN,IPCHN,PREC,LINES,IS,IE,NEWSCR)
C      20      IF (REPCHN.NE. 0) GOTO 9900
C      IFLAG = FL$RET + FL$USE + FL$BIT
C
C      CALL READN(
C      - '1 = QUIT, 0 = MORE MATCHES, N = SEE NTH TEST: ',47,N)
C      IF (N) 9900,10,25
C
C      FIND A TEST RECORD
C
C      25      IBIAS = 22 + (N-1)*PNWT
C      IFLAG = FL$RET
C      30      CALL NEXT$(PTCHN,PTREC,PREC(IBIAS),PTARR,IFLAG,$9200,0,0,0,0)
C
C      PRINT/DISPLAY THE TEST PLAN
C
C      IS = 74

```

```
      IE = 99
      WRITE (1,11)
      CALL RPTGEN(REPCHN,IPCHN,PTREC,LINES,IS,IE,NEWSR)
      GOTO 20
C
C ERRORS
C
9000  WRITE(1,9010)IER
9010  FORMAT(1X,'ERROR ',I3,' IN OPENING DEMO.SCREEN.FT')
      RETURN
C
9100  IF (PLERR .NE. 7) GOTO 9120
      IF (N.EQ. 0) GOTO 9900
      WRITE(1,9110)PKEY
9110  FORMAT(/,1X,20A2,' NOT FOUND')
      GOTO 9900
9120  IF (PLERR .EQ. 22 .OR. PLERR .EQ. 24) GOTO 9140
      WRITE(1,9130)PLERR,PKEY
9130  FORMAT(1X,'MIDAS ERROR ',I3,' IN FINDING ',20A2)
      GOTO 9900
9140  CALL RECYCL
      GOTO 10
C
9200  IF (PTERR .NE. 7) GOTO 9220
      CALL TNOUA(PREC(IBIAS),20)
      CALL TNOU(' NOT FOUND',10)
      GOTO 9900
9220  IF (PTERR .EQ. 22 .OR. PTERR .EQ. 24) GOTO 9240
      WRITE(1,9230)PTERR
9230  FORMAT(1X,'MIDAS ERROR ',I3,' IN FINDING ')
      CALL TNOU(PREC(IBIAS),50)
      GOTO 9900
9240  CALL RECYCL
      GOTO 30
C
C EXIT
C
9900  CALL CLOS$A(IPCHN)
      RETURN
      END
```

C DSPPT: DISPLAY ONE TEST PLAN

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0      JUNE 1,1980                      *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C *
C *****

```

```

C *CALLED BY: DSFLD
C *FUNCTION : 1. GET TEST NAME FROM OPERATOR (PEDIT)      *
C *          2. FIND THE RECORD IN TEST DATA BASE        *
C *          3. PRINT/DISPLAY THE RECORD                  *
C

```

SUBROUTINE DSPPT(SPCHN)

C ARG 1: SPOOL CHANNEL SPCHN INTEGER

```

C $INSERT PTCOM
C $INSERT SYSCOM>PARM.K
C $INSERT SYSCOM>A$KEYS
C

```

C OPEN SCREEN TEMPLATE

```

C CALL VOPEN$('DEMO.SCREEN.FT',14,1,IPCHN,IERR)
C IF (IERR. NE. 0) GOTO 9000
C

```

C GET THE TEST NAME

```

C IS = 74
C IE = 75
C NEWSR = 2 /* DO NOT ERASE SCREEN TO START
C CALL ZFIL(PTREC,PTSZB,0)
C CALL PEDIT(IPCHN,PTREC,IS,IE,NEWSR)
C WRITE (1,1)
C FORMAT(/)
C

```

```

C NCHAR = LSIZE(PTREC,20) /* GET LENGTH OF KEYWORD
C IFLAG = FL$RET + FL$BIT
C

```

C FIND THE TEST RECORD

```

C 10 CALL NEXT$(PTCHN,PTREC,PTREC,PTARR,IFLAG,$9100,0,0,0,NCHAR)
C

```

C PRINT/DISPLAY THE RECORD

```

C IS = 73
C IE = 99
C WRITE (1,11)
C 11 FORMAT(/)
C CALL RPTGEN(SPCHN,IPCHN,PTREC,LINES,IS,IE,2)
C IFLAG = FL$BIT + FL$RET + FL$USE
C

```

C ANY MORE MATCHES?

```

C J = 1
C IF (SPCHN .EQ. 0) CALL PAUS(J)
C IF (J .EQ. 0) GOTO 9900
C GOTO 10
C

```

C ERRORS

```

C 9000 WRITE(1,9010)IERR
C 9010 FORMAT(1X,'ERROR ',I3,' IN OPENING DEMO.SCREEN.FT')
C RETURN
C

```

```

C 9100 IF (PTERR .EQ. 7) GOTO 9900
C IF (PTERR .NE. 24 .AND. PTERR .NE. 22) GOTO 9150
C CALL RECYCL
C GOTO 10
C

```

```

C 9150 WRITE(1,9160)PTERR,PTNM
C 9160 FORMAT(1X,'MIDAS ERROR ',I3,' IN FINDING ',20A2)
C

```

C DSPPT: DISPLAY ONE TEST PLAN

PAGE 0002

C
9900 CALL CLOS* A(IPCHN)
RETURN
END

C DSPR: DISPLAY RESULTS

```

*****
*                                     *
*      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
*      VERSION 1.0   JUNE 1,1980                        *
*      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
*                                     *
*****

*CALLED BY: DSPLO AND RETR
*FUNCTION : 1. GET PART#,INSP NAME,SERIAL#,INSP ID FROM OPERATOR
*          2. CREATE THE RESULT KEYWORD
*          3. FIND A MATCH IN DATA BASE
*          4. PRINT/DISPLAY THE RECORD
*          5. IF TEST RESULT DESIRED:
*             A. FIND THE RECORD
*             B. DISPLAY IT

ARG 1: REPCHN    INTEGER

SUBROUTINE DSPR(REPCHN)
C
$INSERT RCOM
$INSERT RTCOM
$INSERT SYSCOM>PARM.K
$INSERT SYSCOM>A$KEYS
C
    INTEGER RMPRI          /* RESULTS PRIMARY KEY LENGTH WORDS
    PARAMETER RMPRI = 20
    INTEGER RKEY(RMPRI)
    INTEGER REPCHN,PN(10),IN(10),SNO(10),IID(5),IBUF(35)
    EQUIVALENCE (IBUF,IID) /*INSPECTORS ID#
    EQUIVALENCE (IBUF(6),PN) /*PART NUMBER
    EQUIVALENCE (IBUF(16),IN) /*INSPECTION NAME
    EQUIVALENCE (IBUF(26),SNO) /*SERIAL NUMBER

C OPEN SCREEN TEMPLATE
C
    CALL VOPEN('DEMO.SCREEN.FT',14,1,IPCHN,IERR)
    IF (IERR.NE. 0) GOTO 9000

C GET PART#,INSP NAME,SERIAL#,INSP ID
C
    IS = 0
    IE = 5
    NEWSCR = 2 /* DO NOT ERASE SCREEN TO START
    CALL ZFIL(IBUF,70,0)
    CALL PEDIT(IPCHN,IBUF,IS,IE,NEWSCR)

C
    NBYTES = RMPRI*2
    CALL CRPKEY(PN,20,IN,20,SNO,20,0,0,RKEY,NBYTES) /* CREATE KEYWORD
    NCHAR = LSIZE(RKEY,40) /* GET LENGTH OF KEYWORD
    IFLAG = FL$RET + FL$BIT

C FIND A MATCH
C
10    CALL NEXT$(RLCHN,RREC,RKEY,RLARR,IFLAG,$9100,0,0,0,NCHAR)
C PRINT/DISPLAY THE RECORD
C
    IS = 225
    IE = 244
    NEWSCR = 2
    WRITE (1,11)
    FORMAT(/)
11    CALL RPTGEN(REPCHN,IPCHN,RREC,LINES,IS,IE,NEWSCR)
20    IF (REPCHN.NE. 0) GOTO 9900
    IFLAG = FL$RET + FL$USE + FL$BIT

C
    WRITE (1,11)
    CALL READN
    + ('-1 = QUIT, 0 = MORE MATCHES, N = SEE NTH TEST: ',47,N)
    IF (N) 9900,10,25

```

C FIND A TEST RECORD

```
C
25      IBIAS = 43 + (N-1)*RNWT
        IFLAG = FL$RET
30      CALL NEXT$(RTCHN,RTREC,RREC(IBIAS),RTARR,IFLAG,$9200,0,0,0,0)
```

C PRINT/DISPLAY THE TEST RESULT

```
C
        IS = 250
        IE = 273
        WRITE (1,11)
        CALL RPTGEN(REPCHN,IPCHN,RTREC,LINES,IS,IE,NEWSCK)
        GOTO 20
```

C ERRORS

```
C
9000    WRITE(1,9010)IERR
9010    FORMAT(1X,'ERROR ',I3,' IN OPENING DEMO.SCREEN.FT')
        RETURN
```

```
C
9100    IF (RLERR .NE. 7) GOTO 9120
        WRITE(1,9110)RKEY
9110    FORMAT(1X,20A2,' NOT FOUND')
        GOTO 9900
9120    IF (RLERR .EQ. 22 .OR. RLERR .EQ. 24) GOTO 9140
        WRITE(1,9130)RLERR,RKEY
9130    FORMAT(1X,'MIDAS ERROR ',I3,' IN FINDING ',20A2)
        GOTO 9900
9140    CALL RECYCL
        GOTO 10
```

```
C
9200    IF (RTERR .NE. 7) GOTO 9220
        WRITE (1,9201) (RREC(IBIAS+I),I=0,24)
9201    FORMAT(25A2,' NOT FOUND')
        GOTO 9900
9220    IF (RTERR .EQ. 22 .OR. RTERR .EQ. 24) GOTO 9240
        WRITE(1,9230)RTERR,(RREC(IBIAS+I),I=0,24)
9230    FORMAT(1X,'MIDAS ERROR ',I3,' IN FINDING ',25A2)
        GOTO 9900
9240    CALL RECYCL
        GOTO 30
```

C EXIT

```
C
9900    CALL CLOS$(IPCHN)
        RETURN
        END
```

C DSPRT: DISPLAY RESULTS OF ONE TEST

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0      JUNE 1,1980                      *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C *
C *****
C *CALLED BY: DSPLD AND RETR                                *
C *FUNCTION: 1. GET PARTN,INSP NAME,SERIALN,INSP ID FROM OPERATOR *
C *          2. CREATE PLAN TEST KEYWORD                    *
C *          3. GET THE RECORD                              *
C *          4. DISPLAY THE RECORD                          *

```

C ARG 1: SPOOL CHANNEL SPCHN INTEGER

C SUBROUTINE DSPRT(SPCHN)

```

C $INSERT RTCON
C $INSERT SYSCOM>PARM.K
C $INSERT SYSCOM>A$KEYS

```

```

C      INTEGER RTPRI          /* NUMBER OF WORDS IN TEST NAME
C      PARAMETER RTPRI = 20
C      INTEGER RKEY(RTPRI)
C      INTEGER REPCHN,PN(10),IN(10),SNO(10),IID(5),IBUF(45)
C      INTEGER TN(10)
C      EQUIVALENCE (IBUF,IID)          /*INSPECTORS ID#
C      EQUIVALENCE (IBUF(6),PN)        /*PART NUMBER
C      EQUIVALENCE (IBUF(16),IN)       /*INSPECTION NAME
C      EQUIVALENCE (IBUF(26),SNO)      /*SERIAL NUMBER
C      EQUIVALENCE (IBUF(36),TN)      /*TEST NAME

```

C OPEN SCREEN TEMPLATE

```

C      CALL VOPEN$('DEMO.SCREEN.FT',14,1,IPCHN,IERR)
C      IF (IERR. NE. 0) GOTO 9000

```

C GET PARTN,INSP NAME,SERIALN,INSP ID

```

C      IS = 0
C      IE = 6
C      NEWSCR = 2          /* DO NOT ERASE SCREEN TO START
C      CALL ZFIL(IBUF,90,0)
C      CALL PEDIT(IPCHN,IBUF,IS,IE,NEWSCR)

```

```

C      NBYT = RTPRI*2
C      CALL CRPKEY(PN,20,IN,20,SNO,20,TN,20,RKEY,NBYT) /* CREATE KEYWORD
C      NCHAR = LSIZE(RKEY,40) /* GET LENGTH OF KEYWORD
C      IFLAG = FL$RET + FL$BIT

```

C FIND THE TEST RECORD

```

C      CALL NEXT$(RTCHN,RTREC,RKEY,RTARR,IFLAG,$9100,0,0,0,NCHAR)

```

C PRINT/DISPLAY THE RECORD

```

C      IS = 250
C      IE = 273
C      WRITE (1,11)
C      FORMAT (/)
C      CALL RPTGEN(SPCHN,IPCHN,RTREC,LINES,IS,IE,2)
C      IFLAG = FL$BIT + FL$RET + FL$USE

```

C ANY MORE MATCHES?

```

C      CALL PAUS(J)
C      IF (J.EQ. 0) GOTO 9900
C      GOTO 10

```

C ERRORS

```

C      9000 WRITE(1,9010)IERR

```

```
9010  FORMAT(1X,'ERROR ',I3,' IN OPENING DEMO.SCREEN.FT')  
      RETURN  
C  
9100  IF (RTERR .EQ. 7) GOTO 9900  
      IF (RTERR .NE. 24 .AND. RTERR .NE. 22) GOTO 9150  
      CALL RECYCL  
      GOTO 10  
C  
9150  WRITE(1,9160)RTERR,RKEY  
9160  FORMAT(1X,'MIDAS ERROR ',I3,' IN FINDING ',20A2)  
C  
9900  CALL CLOS4A(IPCHN)  
      RETURN  
      END
```

C DSPT: DISPLAY TAPE DATA

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM
C *      VERSION 1.0      JUNE 1,1980
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY
C *
C *****
C *
C *      *CALLED BY: DSPLD AND RETR
C *      *FUNCTION : 1. GET TAPE NAME OR IMAGE NAME FROM OPERATOR (PEDIT)
C *                  2. FIND THE RECORD
C *                  3. PRINT/DISPLAY THE RECORD (RPTGEN)
C *
C ARG 1: SPCHN  SPOOL FILE CHANNEL  INTEGER
C      SUBROUTINE DSPT(SPCHN)
C
C $INSERT TCOM
C $INSERT SYSCOM>PARM.K
C $INSERT SYSCOM>A$KEYS
C
C      INTEGER IBUF(26),KEY(TNMT)
C      LOGICAL MORE
C      EQUIVALENCE (IBUF,INDEX)      /* MIDAS INDEX
C      EQUIVALENCE (IBUF(2),KEY)     /* MIDAS KEYWORD
C
C OPEN SCREEN FILE
C
C      CALL VOPEN('DEMO.SCREEN.FT',14,1,ICH,IERR)
C      IF (IERR.NE. 0) GOTO 9000
C
C 10      IS = 100
C          IE = 105
C          NEWSR = 2      /*DO NOT ERASE SCREEN TO START
C
C GET INDEX AND PRIMARY OR SECONDARY KEYWORD
C PRIMARY KEY=TNAM,  SECONDARY KEYS=IMAGE NAMES
C
C      CALL PEDIT(ICH,IBUF,IS,IE,NEWSR)
C
C      NBYTES = TNMT*2
C      LEN = LSIZE(KEY,NBYTES) /*FIND SIZE OF KEYWORD FOR MIDAS
C
C FIND THE RECORD
C
C      IFLAG = FL$RET + FL$BIT
C      INDEX = INDEX /* ***CHECKOUT
C      CALL NEXT$(TCHN,TREC,KEY,TPARR,IFLAG,$9100,INDEX,0,0,LEN)
C
C DISPLAY THE RECORD
C
C      IS = 106
C      IE = 160
C      WRITE (1,11)
C 11      FORMAT(/)
C          CALL RPTGEN(SPCHN,ICH,TREC,LINES,IS,IE,2)
C          CALL DMORE(MORE)
C          IF (MORE) GOTO 10
C
C CLOSE THE SCREEN FILE
C
C 100      CALL CLOS$(ICH)
C          RETURN
C
C 9000      WRITE(1,9010)IERR
C 9010      FORMAT(1X,'ERROR ',13,' IN OPENING DEMO.SCREEN.FT')
C          RETURN
C
C 9100      IF (TPERR.NE. 7) GOTO 9110
C          WRITE(1,9101) KEY
C 9101      FORMAT('NOT FOUND: ',25A2)
C          GOTO 100

```

```

9110 WRITE(1,9111)TPERR,KEY
9111 FORMAT(1X,'MIDAS ERROR ',I3,' IN FINDING ',25A2)
      GOTO 100
      END

```

C FREECH: RETURNS A NUMBER OF AVAILABLE PAGE 0001

C FREECH: RETURNS A NUMBER OF AVAILABLE PRIMOS CHANNELS

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0      JUNE 1,1980                      *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C *
C *****
C
C      SUBROUTINE FREECH(NUM,ICHN)
C
C      GENERAL PURPOSE SUBROUTINE TO FIND ONE OR MORE AVAILABLE CHANNELS
C
C      NUM   = NUMBER OF FREE CHANNELS REQUESTED
C      ICHN  = AN ARRAY SIZE NUM FOR CHANNELS AVAILABLE
C
C      $INSERT SYSCOM>KEYS.F
C      $INSERT SYSCOM>ERRD.F
C
C      INTEGER ICHN(1) /* DUMMY DIMENSION
C      LOGICAL UNIT$A
C
C      I = 1
C
C      DO 100 ICH=1,63
C        IF (UNIT$A(ICH)) GOTO 100
C        ICHN(I) = ICH
C        I = I + 1
C        IF (I .GT. NUM) RETURN
100    CONTINUE
C
C      ALL UNITS IN USE: ERROR :41
C      IER = E$FUIU
C      RETURN
C
C      END

```

C GDFNCT: GET DISPLAY FUNCTION

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0      JUNE 1,1980                      *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C *
C *****
C *CALLED BY: DSPLD                                         *
C *FUNCTION: 1. GET FUNCTION FROM OPERATOR                  *
C *          2. GET MAILSTOP FROM OPERATOR IF PRINTOUT DESIRED *
C *          3. IF PRINTOUT DESIRED, OPEN A SPOOL CHANNEL    *
C ARG 1: MENU      INTEGER
C ARG 2: SPCHN     SPOOL CHANNEL NUMBER  INTEGER
C ARG 3: SPFNAM    SPOOL FILE NAME      STRING, INTEGER ARRAY(7)
C
C      SUBROUTINE GDFNCT(MENU,SPCHN,SPFNAM)
C
C $INSERT SYSCOM>A$KEYS
C
C      INTEGER SPCHN,SPFNAM(7)
C      SPCHN = 0
C
C      CALL ZFIL(SPFNAM,14,0)
C
C OPEN A CHANNEL TO SCREEN TEMPLATE
C
C      CALL VOPEN$('DEMO.SCREEN.FT',14,1,ICH,IERR)
C      IF (IERR .NE. 0) GOTO 9000
C
C GET MENU AND MAIL STOP FROM SCREEN
C
C      IS = 175
C      IE = 184
C      NEWSCR = 2      /* DO NOT ERASE SCREEN TO START
10      CALL PEDIT(ICH,SPFNAM,IS,IE,NEWSCR)
C      IF (SPFNAM(1) .GE. 0 .AND. SPFNAM(1) .LE. 8) GOTO 30
C      WRITE(1,20)
20      FORMAT(1X,'MENU NOT IN RANGE')
C      GOTO 10
C
C CLOSE THE SCREEN FILE
C
C30      CALL CLOS$A(ICH)
C
C      MENU = SPFNAM(1)
C
C      IF (SPFNAM(5) .EQ. 0) RETURN
C
C      CALL ZMVD('MAILSTOP.',SPFNAM,9)
C
C      CALL PACK(SPFNAM,14)
C
C OPEN A CHANNEL TO SPOOL FILE FOR WRITING
C
C      CALL VOPEN$(SPFNAM,14,2,SPCHN,IERR)
C      IF (IERR .NE. 0) GOTO 9100
C      RETURN
C
C ERROR MESSAGES
C
C9000      WRITE(1,9010)IERR
C9010      FORMAT(1X,'ERROR ',13,' IN OPENING DEMO.SCREEN.FT')
C      RETURN
C9100      WRITE(1,9110)IERR,SPFNAM
C9110      FORMAT(1X,'ERROR ',13,' IN OPENING ',7A2)
C      RETURN
C      END

```

C GFUNCT: GET FUNCTION, MAIN MENU

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0      JUNE 1,1980                      *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C * *****
C
C * CALLED BY: AI *
C * FUNCTION: GET MENU FUNCTION - INSPECT,PLAN,QUERY RESULTS,EXIT *
C
C      SUBROUTINE GFUNCT(MENU)
C
10      WRITE(1,20)
20      FORMAT(//,'MAIN MENU: '//
E1X,'0      EXIT'//
11X,'1      PERFORM AN INSPECTION'//
21X,'2      PLANNING FUNCTIONS'//
31X,'3      QUERY INSPECTION RESULTS')
      READ(1,30,ERR=40)MENU
30      FORMAT(I1)
      IF (MENU .LT. 0 .OR. MENU .GT. 3) GOTO 10
      RETURN
40      MENU = 0
      RETURN
      END

```

C GIIN: GET INSPECTOR INPUT: INSPECTOR ID, PAGE 0001

C GIIN: GET INSPECTOR INPUT: INSPECTOR ID, PART#, TESTNAME, SERIAL#

```
C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0      JUNE 1,1980                      *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C *
C *****
```

```
C *CALLED BY: INSP *
C *FUNCTION: GET INSPECTION INFORMATION WITH SCREEN EDITOR *
```

C SUBROUTINE GIIN(IID,PART,TESTID,SERIAL,IERR)

C DIMENSION IBUF(36)

C MOVE DATA TO EDIT BLOCK

```
C CALL MSUB$A(IID,10,1,10,IBUF,70,1,10)
C CALL MSUB$A(PART,20,1,20,IBUF,70,11,30)
C CALL MSUB$A(TESTID,20,1,20,IBUF,70,31,50)
C CALL MSUB$A(SERIAL,20,1,20,IBUF,70,51,70)
```

C OPEN SCREEN FILE

```
C-----
C CALL VOPEN$('DEMO.SCREEN.FT',14,1,ICHN,IERR)
C IF (IERR .EQ. 0) GOTO 20
C WRITE(1,10)IERR
10  FORMAT(IX,'ERROR ',I2,' IN OPENING DEMO.SCREEN.FT')
C RETURN
```

C SET UP PEDIT PARAMETERS

```
C-----
20  IS = 0
C IE = 4
C NEWSCR = 2 /* DO NOT ERASE SCREEN TO START
```

C GET OPERATOR'S INPUT

```
C-----
25  CALL PEDIT(ICHN,IBUF,IS,IE,NEWSCR)
C MAKE SURE SERIAL# NE 0
```

```
C-----
C IF (IBUF(26) .NE. 0) GOTO 40
C WRITE(1,30)
30  FORMAT(IX,'SERIAL NO. MANDATORY!')
C GOTO 25
```

C LOAD INPUT DATA INTO PARAMETER VARIABLES

```
C-----
40  CALL MSUB$A(IBUF,70,1,10,IID,10,1,10)
C CALL MSUB$A(IBUF,70,11,30,PART,20,1,20)
C CALL MSUB$A(IBUF,70,31,50,TESTID,20,1,20)
C CALL MSUB$A(IBUF,70,51,70,SERIAL,20,1,20)
```

C CLOSE SCREEN FILE

```
C-----
C CALL CLOS$A(ICHN)
C RETURN
C END
```

C GPFNCT: GET PLAN FUNCTION

```

*****
*                                     *
*      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
*      VERSION 1.0      JUNE 1,1980                      *
*      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
*                                     *
*****

```

```

*CALLED BY: PLAN                                     *
*FUNCTION:  GET PLANNING FUNCTION FROM OPERATOR        *

```

SUBROUTINE GPFNCT(MENU)

WRITE (1,10)

FORMAT(/,'MENU: ',/)

```

1  ' 1  CREATE NEW PLAN',/
2  ' 2  CREATE NEW TEST PROCEDURE',/
3  ' 3  CREATE NEW IMAGE COMMAND',/
4  ' 4  MAKE NEW PLAN FROM OLD',/
5  ' 5  MAKE NEW TEST FROM OLD',/
6  ' 6  MAKE NEW IMAGE COMMAND FROM OLD',/
7  ' 7  MODIFY PLAN',/
8  ' 8  MODIFY TEST PROCEDURE',/
9  ' 9  MODIFY IMAGE COMMAND',/
.  '10  DELETE PLAN',/
1  '11  DELETE TEST PROCEDURE',/
2  '12  DELETE TEST RESULTS',/
3  '13  DELETE IMAGE COMMAND',/
4  '14  DISPLAY DATA',/

```

CALL READN('?',2,MENU)

IF (MENU .LT. 1) RETURN

IF (MENU .GT. 14) GOTO 5

RETURN

END

C GPLNM: GET PLAN RECORD

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM
C *      VERSION 1.0      JUNE 1,1980
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY
C *
C *****
C *CALLED BY: INSP
C *FUNCTION: 1.  CREATE THE PLAN KEYWORD
C *           2.  FIND THE PLAN RECORD
C *           3.  IF PLAN RECORD NON-EXISTANT, MAY USE DEFAULT PLAN*
C
C      SUBROUTINE GPLNM(PART,TESTID,DEFAULT,IERR)
C
C      $INSERT FCOM
C      $INSERT SYSCOM>PARM.K
C      $INSERT SYSCOM>A$KEYS
C
C      INTEGER KEY(15),DPLAN(15)
C      LOGICAL DEFAULT,YES,YSNO$A
C
C      CALL CRPKEY(PART,20,TESTID,20,0,0,0,0,KEY,30)/*CREATE KEYWORD
C      WRITE(1,1000)KEY
C      1000  FORMAT(1X,'KEY: ',15A2)
C      DEFAULT = .FALSE.
C      10    CALL ZFIL(PREC,PSZB,0) /*INITIALIZE PREC
C      IFLAG = FL$RET
C      WRITE(1,1000)KEY
C      CALL TDUMP(KEY,30)
C      CALL NEXT$(PCHN,PREC,KEY,PLARR,IFLAG,$9000,0,0,0,0)
C      WRITE(1,1001)
C      1001  FORMAT(1X,'NO ERROR FROM NEXT$')
C      IERR = 0
C      RETURN
C
C      9000  IERR = PLERR
C      WRITE(1,1002)IERR
C      1002  FORMAT(1X,'ERROR ',12)
C      IF(PLERR.EQ.7) GOTO 9100
C      IF(PLERR.NE.22 .OR. PLERR .NE. 24) GOTO 9900
C      CALL RECYCL
C      GO TO 10 /* TRY AGAIN
C
C      Record does not exist
C      9100  YES = YSNO$A('NOT FOUND. USE DEFAULT PLAN',27,A$DN0)
C      IF (YES) GOTO 9200
C      IERR = -1
C      RETURN
C      9200  CALL ZMVD('DEFAULT PLAN/DEFAULT ',KEY,30)
C      DEFAULT = .TRUE.
C      GOTO 10
C      9900  WRITE(1,9910)PLERR,KEY
C      9910  FORMAT(1X,'MIDAS ERROR ',13,' IN LOCATING ',15A2)
C      RETURN
C      END

```

C GPTST: GET A TEST PLAN

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0    JUNE 1,1980                        *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C *
C *****

```

```

C *CALLED BY: DINSF                                         *
C *FUNCTION: 1.  GET TEST PLAN RECORD                       *
C *            2.  DISPLAY TEST DESCRIPTION                 *

```

C SUBROUTINE GPTST(ITEST,IERR)

C ITEST = TEST SEQUENCE NUMBER

```

C $INSERT PCOM
C $INSERT PTCOM
C $INSERT SYSCOM>PARM.K

```

C EQUIVALENCE (PDA,LPDA), (PDB,LPDB), (PDC,LPDC), (PDD,LPDD)

C IBIAS = 22 + (ITEST-1)*PNWT

C IFLAG = FL\$RET

10 CALL NEXT\$(PTCHN,PTREC,PREC(IBIAS),PTARR,IFLAG,\$9000,0,0,0,0)

C DISPLAY DESCRIPTION TEXT

```

C -----
15  WRITE (1,15)
    FORMAT(/)
    IF (LPDA .EQ. 0) GOTO 30
    WRITE (1,20) PDA
    IF (LPDA .EQ. 0) GOTO 30
    WRITE (1,20) PDB
    IF (LPDC .EQ. 0) GOTO 30
    WRITE (1,20) PDC
    IF (LPDD .EQ. 0) GOTO 30
    WRITE (1,20) PDD
20  FORMAT(39A2)
30  WRITE (1,40)
40  FORMAT(/)
    RETURN

```

C

```

C 9000  IERR = PTERR
      IF (PTERR.NE.22 .OR. PTERR.NE.24) GOTO 9900
      WRITE(1,1002) /* ***CHECKOUT***

```

1002 FORMAT(1X,'CALLING RECYCLE')

CALL RECYCL

GO TO 10 /* TRY AGAIN

9900 IF (PTERR.NE.7) GOTO 9905

WRITE(1,9901)PTNM,PTNAM

9901 FORMAT(1X,'TEST ',10A2,' NOT FOUND IN DATA BASE ',16A2)

RETURN

9905 WRITE(1,9910)PTERR,PTNM,PTNAM

9910 FORMAT(1X,'MIDAS ERROR ',13,' IN TEST ',16A2/1X,

C 'FROM DATA BASE ',16A2)

RETURN

END

C GRFNCT: GET RETRIEVAL FUNCTION

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0   JUNE 1,1980                        *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C * *****
C
C *CALLED BY: RETR
C *FUNCTION:  GET RETRIEVAL FUNCTION FROM OPERATOR
C
C ARG 1: MENU   INTEGER
C
C      SUBROUTINE GRFNCT(MENU)
C
C      WRITE(1,10)
C      FORMAT(1X,/, 'HISTORY MENU'/,
C      1 ' 0  EXIT',/,
C      2 ' 1  RETRIEVE AN IMAGE FOR DISPLAY',/,
C      3 ' 2  RECREATE ONE TEST RESULT',/,
C      4 ' 3  RECREATE AN INSPECTION',/,
C      5 ' 4  DISPLAY MAIN RESULTS',/,
C      6 ' 5  DISPLAY TEST RESULTS',/,
C      7 ' 6  DISPLAY TAPE RECORD',/,
C      8 ' 7  LIST INDECIES')
C      READ(1,20,ERR = 30)MENU
C      FORMAT(I1)
C      IF (MENU .LT. 0 .OR. MENU .GT. 7) GOTO 5
C      RETURN
C      END

```

C INITIAL: SET UP TERMINAL CODES AND OPEN D PAGE 0001

C INITIAL: SET UP TERMINAL CODES AND OPEN DATA BASE FILES

```
C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0   JUNE 1,1980                        *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY    *
C *****
C * CALLED BY: AI                                         *
C * FUNCTION: 1.  SET UP DATA BASE PARAMETERS (DBPRM)   *
C *             2.  OPEN DATA BASE FILES                 *
C
```

SUBROUTINE INITIAL(IERR)

```
C
C $INSERT PCOM
C $INSERT RCOM
C $INSERT PTCOM
C $INSERT RTCOM
C $INSERT TCOM
C $INSERT CCOM
C $INSERT SYSCOM>KEYS.F
C
```

INTEGER TYPE

C SET UP TERMINAL CODES

```
C-----
C      CALL SETERM(TYPE)
C      IF (TYPE .GT. 0 .AND. TYPE .LT. 4) GOTO 10
C      IERR = 17
C      IF (TYPE .EQ. 4) GOTO 4
C      RETURN
4      CALL TONL
C      CALL TNOU ('This program will not run on your terminal.',43)
C      RETURN
C
```

C GET DATA BASE PARAMETERS

C-----
10 CALL DBPRM
C

C OPEN PLAN DATA BASE FILES

```
C-----
C      CALL VOPEN$(PLNAM,PLLEN,1,PCHN,IERR)
C      IF (IERR .EQ. 0) GOTO 30
C      WRITE(1,20)IERR,PLNAM
20      FORMAT(1X,'ERROR ',I2,' IN OPENING ', 16A2)
C      RETURN
C
```

```
30      CALL VOPEN$(PTNAM,PTLEN,1,PTCHN,IERR)
C      IF (IERR .EQ. 0) GOTO 40
C      WRITE(1,20)IERR,PTNAM
C      CALL CLOALL
C      RETURN
C
```

C OPEN RESULTS DATA BASE FILES

```
C-----
40      CALL VOPEN$(RLNAM,RLEN,1,RLCHN,IERR)
C      IF (IERR .EQ. 0) GOTO 50
C      WRITE(1,20)IERR,RLNAM
C      CALL CLOALL
C      RETURN
C
```

```
50      CALL VOPEN$(RTNAM,RTLEN,1,RTCHN,IERR)
C      IF (IERR .EQ. 0) GOTO 60
C      WRITE(1,20)IERR,RTNAM
C      CALL CLOALL
C      RETURN
C
```

C OPEN TAPE DATA BASE FILE

```
C-----
60      CALL VOPEN$(TPNAM,TLEN,1,TCHN,IERR)
C      IF (IERR .EQ. 0) GOTO 70
C      WRITE(1,20)IERR,TPNAM
C      CALL CLOALL
C
```

C INITIAL: SET UP TERMINAL CODES AND OPEN D PAGE 0002

```
      RETURN
C
C OPEN COMMAND DATA BASE FILE
C-----
70    CALL VOPEN$(CNNAM,CLEN,1,CCHN,IERR)
      IF (IERR.EQ. 0) GOTO 80
      WRITE(1,20)IERR,CNNAM
80    CALL CLOALL
      RETURN
      END
```

C INSP: PERFORM AN INSPECTION

PAGE 0001

C INSP: PERFORM AN INSPECTION

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM
C *      VERSION 1.0      JUNE 1,1980
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY
C *
C *****
C *CALLED BY: AI
C *FUNCTION: 1. GET INSPECTOR ID,PART#,TEST,SERIAL# (GIIN)
C *          2. GET THE PLAN (GPLNM)
C *          3. DO THE INSPECTION (DINSP)
C
C      SUBROUTINE INSP(EQUIP,IERR)
C
C      INTEGER IID(5),PART(10),TESTID(10),SERIAL(10)
C      LOGICAL DEFALT,EQUIP,MORE
C
C GET INSPECTOR INPUT: INSPECTOR ID, PART#, TESTNAME, SERIAL#
C-----
10    CALL GIIN(IID,PART,TESTID,SERIAL,IERR)
      IF (IERR.NE. 0) RETURN
C
C GET PLAN RECORD
C-----
      CALL GPLNM(PART,TESTID,DEFALT,IERR)
      IF (IERR) 20,30,200
C
C      IERR = 0
C      GOTO 100
C
C DO THE INSPECTION
C-----
30    CALL DINSP(IID,SERIAL,EQUIP,DEFALT,IERR)
      IF (IERR.NE. 0) RETURN
C
C DO MORE?
C-----
100   CALL DMORE(MORE)
      IF (MORE) GOTO 10
200   RETURN
      END
C
```

C INTAP: INITIALIZE TAPE

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM
C *      VERSION 1.0      JUNE 1,1980
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY
C *
C *****
C *CALLED BY: DINSP
C *FUNCTION: 1. ASKS IF MAG TAPE NEEDED
C *          2. ASSURES THAT CORRECT TAPE IS ON-LINE
C *          3. ASSURES THAT TAPE IS AT CORRECT POSITION
C *          4. GETS TAPE RECORD
C
C      SUBROUTINE INTAP(EQUIP,NOTAPE,IERR)
C
C      EQUIP = EQUIP INITIALIZED FLAG
C      NOTAPE = TAPE INUSE FLAG, RETURNED
C      IERR = ERROR RETURNED
C
C $INSERT TCOM
C $INSERT SYSCOM>PARM.K
C $INSERT SYSCOM>A$KEYS
C
C      LOGICAL YSN0$A,NEW,CSTR$A,NOTAPE,YES,EQUIP,ITAPE,NTAP
C      INTEGER STATUS(3),NAME(10)
C
C      IF (.NOT. EQUIP) GOTO 10          /* IS EQUIPMENT INITIALIZED?
C      YES = YSN0$A('DOES THIS INSPECTION REQUIRE MAG TAPE',37,A$DNO)
C      IF (YES) NOTAPE = .FALSE.
C
C      IF (.NOT. YES .AND. NOTAPE) RETURN /* PAST PRESENT
C      IF (.NOT. NOTAPE .AND. YES) GOTO 5 /* NOTAPE NOTAPE
C      IF (NOTAPE .AND. YES) GOTO 2 /* TAPE TAPE
C      IF (.NOT. NOTAPE .AND. .NOT. YES) GOTO 12 /* NOTAPE TAPE
C
C      NTAP = YSN0$A('IS CORRECT TAPE MOUNTED AND READY',33,A$DNO)
C      IF (.NOT. NTAP) GOTO 1
C      IF (TNAM(1) .EQ. 0) GOTO 15
C      GOTO 5
C
C      IF (TNFL + 1 .EQ. TFCNT) RETURN /* IS TAPE AT CORRECT POSITION?
C      WRITE(1,6)TNAM
C      FORMAT(1X,'CURRENT TAPE ON LINE: ',10A2)
C      ITAPE = YSN0$A('IS THIS CORRECT',15,A$DNO)
C      IF (ITAPE) GOTO 8
C
C      WRITE(1,7)
C      FORMAT(1X,'PLEASE MOUNT NEW TAPE: HIT ANY KEY WHEN READY')
C      CALL PAUS(J)
C      GOTO 14
C
C      N = TNFL - TFCNT + 1
C      GOTO 70
C
C      CALL ZFIL(TREC,TSZB,0)
C      CALL TONL
C      YES = YSN0$A('DOES THIS INSPECTION REQUIRE MAG TAPE',37,A$DNO)
C      IF (.NOT. YES) NOTAPE = .TRUE.
C      IF (.NOT. YES) RETURN
C      NOTAPE = .FALSE.
C
C GET TAPE NAME
C
C      CALL ZFIL(TREC,TSZB,0)
C      CALL TONL
C      CALL TNOUA('TAPE NAME (ENTER "NEW" IF A NEW TAPE) ',38)
C      CALL READL(TNAM,NCHAR,20)
C      NEW = CSTR$A(TNAM,NCHAR,'NEW',3)
C      IF (.NOT. NEW) GOTO 30
C
C NEW TAPE, ADD IT TO THE DATA-BASE

```

```

C-----
25  CALL TNOUA('ENTER NEW TAPE NAME: ',21)
    CALL READL(TNAM,NCHAR,20)
    CALL ADD1$(TCHN,TREC,TNAM,TPARR,FL$RET,$9000,0,0,0,0)
    GOTO 50
C
C EXISTING TAPE - FIND IN DATA BASE
C-----
30  CALL NEXT$(TCHN,TREC,TNAM,TPARR,FL$RET,$9100,0,0,0,0)
C
C CHECK THE TAPE FOR ONLINE
C-----
50  TFCNT = 1          /* PRESENT FILE NO.
    CALL T$MT(0,LOC(TREC),0,:100000,STATUS)
    IF (AND(:300,STATUS(2)) .NE. :300) GOTO 9200
C
C REWIND TAPE
C-----
60  CALL T$MT(0,LOC(TREC),0,:40,STATUS)
    IF(AND(STATUS(2),1) .EQ. 0) GOTO 60
C
C MOVE TO CORRECT PLACE ON TAPE
C-----
    N = TNFL
70  IF (N .EQ. 0) RETURN
    CALL T$MT(0,LOC(TREC),0,:22200,STATUS) /*MOVE TAPE 1 FILE
    IF (STATUS(1) .EQ. 0) GOTO 90
80  CALL T$MT(0,LOC(TREC),0,:100000,STATUS)
    IF (STATUS(1) .NE. 0) GOTO 80
90  N = N - 1
    TFCNT = TFCNT + 1
    GOTO 70
C
9000 IF (TPERR .NE. 12) GOTO 9010
    WRITE (1,9001) TNAM
9001 FORMAT(/,10A2,' ALREADY EXISTS.')
    GOTO 10
9010 IERR = TPERR
    WRITE (1,9011) IERR,TNAM
9011 FORMAT('MIDAS ERROR = ',I3,', KEY = ',10A2)
    RETURN
C
9100 IERR = TPERR
    IF(TPERR.NE.7) GOTO 9150
C
C RECORD NOT FOUND
    WRITE(1,9160)
9160 FORMAT(1X,'RECORD NOT FOUND')
    IERR = 0
    GOTO 11 /*GIVE OPERATOR ANOTHER CHANCE
9150 IF(TPERR.NE.22 .OR. TPERR .NE. 24) GOTO 9190
    CALL RECYCL
    GOTO 30 /* TRY AGAIN
9190 WRITE (1,9191)IERR,TNAM
9191 FORMAT('MIDAS ERROR: ',I3,', TAPE N ',10A2)
    RETURN
C
9200 WRITE(1,9210)
9210 FORMAT(1X,'PLEASE MOUNT TAPE AND PUT ONLINE')
    DO 9211 I = 1,3
    CALL T00CT(STATUS(I))
9211 CALL TNOUA(' ',1)
    CALL PAUS(J)
    GOTO 50
END

```


C INTMOT: INITIALIZE MOTORS

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0      JUNE 1,1980                      *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C *
C *****

```

C SUBROUTINE INTMOT(IER)

C \$INSERT SYSCOM>A\$KEYS

```

C LOGICAL YSN0$A
C PARAMETER NMOT = 5
C INTEGER MOTOR(NMOT), STATUS

```

```

C MOTOR 1 POSITIONING IMPLEMENTED IF MOTOR(1) = 1
C *** NOTE: MOTOR NUMBERS START AT 0! ***
C DATA MOTOR /0,0,1,0,0/

```

C RESET THE ABORT IF SET

```

10 CALL MOTION(:140000,0,0,IER)
   IF (IER-1) 20,15,900
15 CALL TNOUA('PUT MOTOR CONTROL IN REMOTE, HIT ANY KEY ',41)
   CALL PAUS(160)
   GO TO 10

```

C ASK OPERATOR IF STANDARDIZATION IS NECESSARY

```

20 IF (.NOT.YSN0$A(
   1 'HAS COUNTER BEEN FIXED SINCE LAST POWER-UP',42,A$DNO))
   2 GO TO 80

```

C OPERATOR THINKS EVERYTHING IS OK.

```

40 CALL POSIT(2,IVAL,STATUS)
   IF (STATUS-1) 50,45,900
45 CALL TNOUA('PUT MOTOR CONTROL IN REMOTE, HIT ANY KEY ',41)
   CALL PAUS(160)
   GO TO 40

```

```

50 WRITE(1,51)IVAL
51 FORMAT(1X,'COUNTER SAYS YOU ARE AT ',I6)
   IF (YSN0$A('IS THIS CORRECT',15,A$DNO)) RETURN

```

```

C
C 80 DO 100 N=1,NMOT
   IF (MOTOR(N) .EQ. 0) GO TO 100
   MOT = N - 1
   CALL POSINT(MOT,IER)
   IF (IER .NE. 0) RETURN
100 CONTINUE
200 RETURN
C

```

END

C LISTIX: LIST KEYNAMES BY INDEXN

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM
C *      VERSION 1.0      JUNE 1,1980
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY
C *
C *****

```

```

C *CALLED BY: DSPLO AND RETR
C *FUNCTION : 1. GET WHAT DATA OPERATOR WANTS TO VIEW AND INDEXN *
C *           2. SET LENGTH OF KEYWORD AND DATA BASE CHANNEL # *
C *           3. GET REQUESTED DATA *
C *           4. DISPLAY/PRINT THE DATA *

```

C ARG 1: SPOOL CHAN N

C SUBROUTINE LISTIX(SFCHN)

```

C $INSERT PCOM
C $INSERT PTCOM
C $INSERT RCOM
C $INSERT RTCOM
C $INSERT TCOM
C $INSERT SYSCOM>PARM.K

```

```

C      INTEGER RMPRI      /* RESULTS MAIN PRIMARY KEYWORD SIZE
C      PARAMETER RMPRI = 20
C      INTEGER TMREC(TSZW),TMARRY(14),TMERR,TMKEY(TNWT),TACHR,TMWD
C      EQUIVALENCE(TMARRY,TMERR)

```

C GET MENUH

```

C
C 10      WRITE(1,20)
C 20      FORMAT(1X,/, 'LIST INDICIES MENU'/
C      . 1X,'00  EXIT'/
C      . 1X,'10  PLANS BY NAME'/
C      . 1X,'11  PLANS BY TEST NAME'/
C      . 1X,'20  TEST NAMES'/
C      . 1X,'30  RESULTS BY NAME'/
C      . 1X,'31  RESULTS BY TEST NAME'/
C      . 1X,'40  TEST RESULTS'/
C      . 1X,'50  TAPE NAMES'/
C      . 1X,'51  IMAGE NAME'/)
C      READ(1,30,ERR=210)N,I
C 30      FORMAT(2I1)
C      WRITE (1,33)
C 33      FORMAT(/)

```

C CHECK FOR VALIDITY OF MENU SELECTION

```

C      IF (N .GT. 0 .AND. N .LE. 5) GOTO 40
C      IF (N .LE. 0) RETURN
C      GOTO 10
C 40      IF (I .GE. 0 .AND. I .LE. 1) GOTO 50
C      GOTO 10

```

C 50 IFLAG = FL\$RET + FL\$UKY + FL\$FST

C SET MIDAS CHANNEL,NCHAR IN KEYWORD

C GOTO (110,120,130,140,150),N

C PLAN

```

C 110      MCHN = PCHN
C      IF (I .EQ. 0) TACHR = 30
C      IF (I .EQ. 1) TACHR = PNWT*2
C      GOTO 160

```

C PLAN TEST

```

C 120      MCHN = PTCMN
C      TACHR = 20

```

```

      GOTO 160
C
C RESULTS
130   MCHN = RLCHN
      IF (I .EQ. 0) TMCHR = RMPRI*2
      IF (I .EQ. 1) TMCHR = RNWT*2
      GOTO 160
C
C RESULTS TEST
140   MCHN = RTCHN
      TMCHR = RNWT*2
      GOTO 160
C TAPE
150   MCHN = TCHN
      IF (I .EQ. 0) TMCHR = 20
      IF (I .EQ. 1) TMCHR = TNWT*2
C
C CALCULATE NWORDS IN KEYWORD
160   TMWDS = TMCHR/2
C
C SPOOL LOOP
170   DO 200 J = 1,22
C
C GET DATA
180   CALL NEXT$(MCHN, TMREC, TMKEY, TMARRY, IFLAG, $9000, 1, 0, 0, 0)
      IFLAG = FL$USE + FL$RET + FL$PLW + FL$UKY
      IF (SPCHN .NE. 0) GOTO 190
C
C OUTPUT LINE TO TERMINAL
      CALL TNOU(TMKEY, TMCHR)
      GOTO 200
C
C OUTPUT LINE TO SPOOL FILE
190   CALL WTLIN$(SPCHN, TMKEY, TMWDS, IERR)
      IF (IERR .NE. 0) GOTO 9100
200   CONTINUE
C
C END OF SPOOL LOOP - GO BACK AND GET MORE
      IF (SPCHN .NE. 0) GOTO 170
      CALL PAUS(J)
      IF (J .NE. 0) GOTO 170
210   RETURN
C
C ERRORS
9000  IF (TMERR .NE. 7) GOTO 9005
      WRITE (1,33)
      J = 1
      IF (SPCHN .EQ. 0) CALL PAUS(J)
      IF (J .EQ. 0) RETURN
      GOTO 10
9005  IF (TMERR .NE. 22 .AND. TMERR .NE. 24) GOTO 9010
      CALL RECYCL
      GOTO 180
9010  WRITE(1,9020) TMERR, TMKEY
9020  FORMAT(1X, 'MIDAS ERROR ', I3, ' IN FINDING ', 25A2)
      RETURN
C
9100  WRITE(1,9110) IERR, SPCHN, TMKEY
9110  FORMAT(1X, 'ERROR ', I3, ' IN WRITING TO SPCHN ', I3, 'KEY= ', 25A2)
      RETURN
C
      END

```

C MDFYC: MODIFY IMAGE PROCESSOR COMMAND

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0      JUNE 1,1980                      *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C * *****
C
C *CALLED FROM: PLAN *
C *FUNCTION: 1. GET COMMAND NAME FROM OPERATOR *
C *          2. GET COMMAND RECORD FROM DATA BASE *
C *          3. EDIT THE RECORD *
C *          4. STORE THE NEW RECORD IN DATA BASE *
C
C      SUBROUTINE MDFYC(IER)
C
C $INSERT CCOM
C $INSERT SYSCOM>A$KEYS
C $INSERT SYSCOM>PARM.K
C
C      INTEGER NAME(16)
C
C      DATA NAME /'DEMO.SCREEN.FT'
C
C      Open a channel to the screen template file
C-----
C      CALL VOPEN$(NAME,32,1,IFCH,IER)      /* OPEN FOR READ
C      IF (IER.NE. 0) GOTO 9000
C
C      IS = 318
C      IE = 319
C      NEWSCR = 2      /* DO NOT ERASE THE SCREEN TO START
C
C      Clear record buffer
C-----
C10      CALL ZFIL(CREC,CSZB,0)
C
C      Get the test name.
C-----
C20      CALL PEDIT(IFCH,CREC,IS,IE,NEWSCR)
C
C      Get the record for editing.
C-----
C30      CALL LOCK$(CCHN,CREC,CNAM,CARR,FL$RET,$9100,0,0,0,0)
C
C      Edit the file
C-----
C      IS = 318
C      IE = 330
C      CALL PEDIT(IFCH,CREC,IS,IE,NEWSCR)
C
C      Store the record
C-----
C      CALL UPDAT$(CCHN,CREC,CNAM,CARR,FL$USE,$9200,0,0,0,0)
C
C      MORE?
C-----
C300      IF (YSNO$( 'More',4,A$DN0)) GOTO 20
C
C      CLOSE THE SCREEN CHANNEL NO.
C-----
C400      CALL CLOS$(IFCH)
C      RETURN
C
C9000      WRITE (1,9001) IER,NAME
C9001      FORMAT('ERROR ',13,' OPENING FILE ',16A2)
C      RETURN
C
C9100      IF (CERR.NE. 7) GOTO 9110
C      WRITE (1,9101) CNAM
C9101      FORMAT(10A2,' NOT FOUND')

```

C MDFYC: MODIFY IMAGE PROCESSOR COMMAND PAGE 0002

```

9110 GOTO 300
      IF (CERR.EQ. 22) GOTO 30
      IF (CERR.EQ. 24) GOTO 30
      WRITE (1,9111) PTERR,PTNAM
9111  FORMAT('MIDAS LOCK ERROR =',I3,' KEY = ',I5A2)
      GOTO 400
C
C
9200 WRITE (1,9101) CERR,CNNAM
9201  FORMAT('MIDAS UPDAT ERROR =',I3,' KEY = ',I5A2)
      GOTO 400
C
C
      END

```

C MDFYP: MODIFY PLAN PAGE 0001

C MDFYP: MODIFY PLAN

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM
C *      VERSION 1.0      JUNE 1,1980
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY
C *
C *****
C
C      SUBROUTINE MDFYP(IER)
C
C      *CALLED FROM: PLAN
C      *FUNCTION: 1. ASK OPERATOR FOR PARTN, INSPECTION NAME
C      *           2. GET THE PLAN RECORD
C      *           3. ALLOW OPERATOR TO MODIFY THE PLAN RECORD
C      *           4. DELETE OLD AND INSERT NEW SECONDARY KEYS
C
C      IER = 0
C      CALL INOU ('NOT IMPLEMENTED!',16)
C      RETURN
C      END

```

C MDFYT: MODIFY TEST PROCEDURE

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM
C *      VERSION 1.0      JUNE 1,1980
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY
C *
C *****
C
C *CALLED FROM: PLAN
C *FUNCTION: 1. GET TEST NAME FROM OPERATOR
C *          2. GET TEST RECORD FROM DATA BASE
C *          3. EDIT THE RECORD
C *          4. STORE THE NEW RECORD IN DATA BASE
C
C      SUBROUTINE MDFYT(IER)
C
C      $INSERT PTCON
C      $INSERT SYSCOM>A$KEYS
C      $INSERT SYSCOM>PARM.K
C
C      INTEGER NAME(16)
C      INTEGER KEYO(10)      /* PRIMARY KEY
C
C      DATA NAME /'DEMO.SCREEN.FT'
C
C      Open a channel to the screen template file
C      -----
C      CALL VOPEN$(NAME,32,1,IFCH,IER)      /* OPEN FOR READ
C      IF (IER .NE. 0) GOTO 9000
C
C      IS = 197
C      IE = 199
C      NEWSCR = 2      /* DO NOT ERASE THE SCREEN TO START
C
C      Clear record buffer
C      -----
C      10 CALL ZFIL(PTREC,PTSZB,0)
C
C      Get the test name.
C      -----
C      20 CALL PEDIT(IFCH,PTREC,IS,IE,NEWSCR)
C
C      Get the record for editing.
C      -----
C      30 CALL LOCK$(PTCHN,PTREC,PTNM,PTARR,FL$RET,$9100,0,0,0,0)
C
C      Edit the file
C      -----
C      IS = 275
C      IE = 297
C      CALL PEDIT(IFCH,PTREC,IS,IE,NEWSCR)
C
C      Store the record
C      -----
C      CALL UPDAT$(PTCHN,PTREC,PTNM,PTARR,FL$USE,$9200,0,0,0,0)
C
C      MORE?
C      -----
C      300 IF (YSN0$( 'More',4,A$DNO)) GOTO 20
C
C      CLOSE THE SCREEN CHANNEL NO.
C      -----
C      400 CALL CLOS$(IFCH)
C      RETURN
C
C      9000 WRITE (1,9001) IER,NAME
C      9001 FORMAT('ERROR ',I3,' OPENING FILE ',16A2)
C      RETURN
C
C      9100 IF (PTERR .NE. 7) GOTO 9110
C      WRITE (1,9101) PTNM

```

PAGE 0002

CC
99
CC

[illegible]

```

*****
*
*   REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM
*   VERSION 1.0      JUNE 1,1980
*   BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY
*
*****

```

```
*CALLED FROM: PLAN
*FUNCTION: 1. ASK OPERATOR FOR COMMAND, INSPECTION NAME
*          2. GET THE COMM RECORD
*          3. ALLOW OPERATOR TO MODIFY THE COMM RECORD
*          4. WRITE NEW RECORD
```

```
IER = 0
CALL TNOU ('NOT IMPLEMENTED!',16)
RETURN
END
```

C MKFMP: MAKE NEW PLAN FROM OLD.

C MKFMP: MAKE NEW PLAN FROM OLD.

```
C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0    JUNE 1,1980                        *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C * *****
C
C      SUBROUTINE MKFMP(IER)
C
C      *CALLED FROM: PLAN                                     *
C      *FUNCTION: 1. GET OLD PLAN NAME FROM OPERATOR         *
C      *          2. GET NEW PLAN NAME FROM OPERATOR         *
C      *          3. CHECK FOR EXISTANCE OF NEW PLAN NAME    *
C      *          4. EDIT OLD PLAN                           *
C      *          5. INSERT NEW PLAN INTO DATA BASE         *
C
C      IER = 0
C      CALL TNOU ('NOT IMPLEMENTED! ',16)
C      RETURN
C      END
```

C MKFMT: MAKE A NEW TEST PLAN FROM OLD.

C MKFMT: MAKE A NEW TEST PLAN FROM OLD.

```
C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0    JUNE 1,1980                        *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C * *****
C
C      SUBROUTINE MKFMT(IER)
C
C      *CALLED BY: PLAN                                     *
C      *FUNCTION: 1. GET OLD AND NEW TEST NAMES FROM OPERATOR *
C      *          2. CHECK FOR EXISTANCE OF NEW TEST NAME    *
C      *          3. EDIT THE RECORD                         *
C      *          4. INSERT NEW TEST INTO DATA BASE         *
C
C      IER = 0
C      CALL TNOU ('NOT IMPLEMENTED! ',16)
C      RETURN
C      END
```

C MOTOR: CONTROLS MOTION OF A MOTOR

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0 JUNE 1,1980                          *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C *
C *****

```

C SUBROUTINE MOTOR(MOT,SPEED,POS,IER)

```

C
C   PARAMETER MSEC=1000
C   INTEGER MOT,SPEED,POS,STATUS,DIR
C   INTEGER SLWDN,STRBRK,TOL,DIF,VALUE,VALUE1
$INSERT SYSCOM>CRTCTRL

```

C ARGUMENTS:

```

C   MOT      =MOTORN, FROM 0 TO 7
C   SPEED    =THE RELATIVE SPEED OF THE MOTOR, FROM 0 TO 127
C   POS      =THE TARGET POSITION THAT IS DESIRED
C   IER      =ERROR FLAG  0 FOR GOOD
C               1 FOR CONTROL IN MANUAL
C               2 GPIB ERROR

```

C PARAMETERS:

```

C   STRBRK   =THE NUMBER OF COUNTS AWAY FROM TARGET POSITION THAT THE
C               BRAKES ARE APPLIED
C   SLWDN    =THE NUMBER OF COUNTS AWAY FROM TARGET POSITION THAT THE
C               MOTOR IS SLOWED DOWN
C   TOL      =THE TOLERANCE THAT IS ALLOWED AWAY FROM TARGET POSITION

```

C SET UP THE PARAMETERS

```

C   PARAMETER (STRBRK=:3,SLWDN=:100,TOL=:2)
C   IER=0

```

C IF SPEED IS ZERO THEN BYPASS

```

C   IF(SPEED .EQ. 0)GO TO 999

```

C FIND THE CURRENT POSITION

```

C   20   CALL POSIT(MOT,VALUE,STATUS)
C       IF (STATUS-1) 30,950,990

```

C COMPUTE THE DIRECTION TO MOVE

```

C   30   IF(VALUE .GT. POS)DIR=1
C       IF(VALUE .LE. POS)DIR=0
C       IF(VALUE .LT. 0)DIR=0
C       IF(POS .GT. VALUE)DIF=POS-VALUE
C       IF(POS .LE. VALUE)DIF=VALUE-POS
C       IF(DIF .LT. SLWDN)GO TO 100

```

C SEE IF THERE YET

```

C   CALL MOTION(MOT,DIR,SPEED,STATUS)
C   40   CALL POSIT(MOT,VALUE,STATUS)
C       IF (STATUS-1) 45,950,990
C   45   IF(POS .GT. VALUE)DIF=POS-VALUE
C       IF(POS .LE. VALUE)DIF=VALUE-POS
C       IF(DIF .LT. SLWDN)GO TO 100
C       GO TO 40

```

C ALMOST THERE! SLOW DOWN.

```

C   100  IF(DIF .LT. STRBRK)GO TO 200
C       SDIF=DIF
C       CALL MOTION(MOT,DIR,20,STATUS)

```

AD-A098 658

BOEING AEROSPACE CO SEATTLE WA
LOW COST HIGH VOLUME RADIOGRAPHIC INSPECTION. (U)

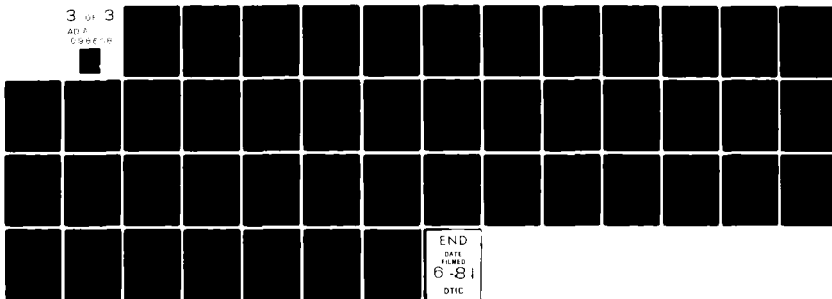
F/6 13/8

JAN 81 N M LOWRY, H J ABPLANALP, J M TANKE DAAK40-78-C-0197
D180-26159-1 DRSMI/RS-CR-81-1 NL

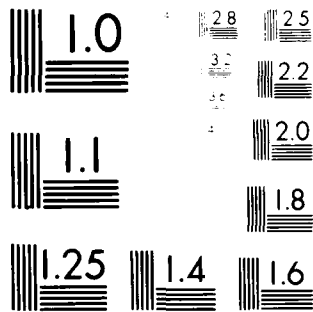
UNCLASSIFIED

3 of 3

AD-A
C988658



END
DATE
FILMED
6-81
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
110 CALL POSIT(MOT,VALUE,STATUS)
    IF (STATUS-1) 115,950,990
115 IF(POS .GT. VALUE)DIF=POS-VALUE
    IF(DIF .GT. SDIF)GO TO 20
    IF(POS .LE. VALUE)DIF=VALUE-POS
    IF(DIF .LT. STRBRK)GO TO 200
    GO TO 110
C
C PUT ON THE BRAKES
C
200 CALL MOTION(MOT,DIR,0,STATUS)
210 CALL POSIT(MOT,VALUE,STATUS)
    IF (STATUS-1) 215,950,990
215 VALUE1=VALUE
C
C PAUSE FOR 1 SEC.
    CALL SLEEP$(MSEC)
C
    CALL POSIT(MOT,VALUE,STATUS)
    IF (STATUS-1) 225,950,990
225 IF(VALUE .NE. VALUE1)GO TO 210
    IF(POS .GT. VALUE)DIF=POS-VALUE
    IF(POS .LE. VALUE)DIF=VALUE-POS
    IF(DIF .LE. TOL)GO TO 999
    GO TO 30
C
C SOMEONE PROBABLY SWITCHED FROM REMOTE TO LOCAL IN THE
C MIDDLE OF MOVING TO THE TARGET LOCATION. SHAME ON THEM!
C
950 CONTINUE
    IER=1
    GO TO 999
C
C ERROR ON GPIB
C
990 CONTINUE
    IER=2
C
999 RETURN
    END
```

C MOVTAPE: VERIFY AND MOVE TAPE

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM
C *      VERSION 1.0      JUNE 1,1980
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY
C *
C *****
C *CALLED BY: RVING AND RVINSP
C *FUNCTION:  1. IF A NEW TAPE, MAKE SURE ON-LINE, AND REWIND
C *           2. MOVE THE TAPE TO THE CORRECT POSITION
C *

```

```

C SUBROUTINE MOVTAPE(NEW,NFIL,IERR)
C

```

```

C NEW = FLAG FOR NEW MOUNT OR ALREADY INSTALLED
C NFIL = FILE NUMBER TO POSITION TAPE
C IERR = ERROR RETURNED
C

```

```

C $INSERT TCOM
C $INSERT SYSCON>A$KEYS
C

```

```

C LOGICAL NEW
C INTEGER STATUS(3),STAT1,STAT2,UPDN,STATR(3),STATR1
C EQUIVALENCE (STATUS(1),STAT1), (STATUS(2),STAT2)
C EQUIVALENCE (STATR,STATR1)
C

```

```

C TRACE N,NFIL,UPDN,TFCNT /* ***CHECKOUT***
C NFIL = NFIL
C

```

```

C UPDN = 1
C MOVE = :22200 /* MOVE FORWARD ONE FILE
C IF (.NOT. NEW) GOTO 40
C

```

```

C NEW TAPE, VERIFY THAT TAPE UNIT IS SET UP
C

```

```

C CHECK THE TAPE FOR ONLINE
C

```

```

C 20 TFCNT = 1 /* PRESENT FILE NO.
C CALL T$MT(0,LOC(TREC),0,:100000,STATUS)
C CALL MSDUMP('SELECT',6,STATUS) /****CHECKOUT
C IF (AND(:300,STAT2) .NE. :300) GOTO 9200
C

```

```

C REWIND TAPE
C

```

```

C 30 CALL T$MT(0,LOC(TREC),0,:40,STATR)
C CALL MSDUMP('REWIND',6,STATR) /****CHECKOUT
C 32 IF (AND(STATR1,1) .EQ. 0) GOTO 35
C CALL T$MT(0,LOC(TREC),0,:100000,STATUS)
C GOTO 32
C 35 N = NFIL - TFCNT
C GO TO 70
C

```

```

C TAPE ALREADY INSTALLED POSITION IT
C

```

```

C 40 N = NFIL - TFCNT
C IF (N .GE. 1) GOTO 70
C UPDN = -1
C MOVE = :20100 /* BACKWARD ONE FILE
C N = -N
C

```

```

C MOVE TO CORRECT PLACE ON TAPE
C

```

```

C 70 IF (N .EQ. 0) RETURN
C CALL T$MT(0,LOC(TREC),0,MOVE,STATUS) /*MOVE TAPE 1 FILE
C IF (STAT1 .EQ. 0) GOTO 90
C 80 CALL T$MT(0,LOC(TREC),0,:100000,STATUS)
C IF (STAT1 .NE. 0) GOTO 80
C 90 N = N - 1
C TFCNT = TFCNT + UPDN
C GOTO 70
C

```

C
C RECORD NOT FOUND
C

9200 WRITE(1,9210)
9210 FORMAT(1X,'PLEASE MOUNT TAPE AND PUT ONLINE')
DO 9211 I = 1,3
CALL T00CT(STATUS(I))
9211 CALL TNOUA(' ',1)
CALL PAUS(J)
GOTO 20
END

C PING:PROCESS THE IMAGE

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM
C *      VERSION 1.0      JUNE 1,1980
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY
C *
C *****
C
C *CALLED BY: DINSF
C *FUNCTION:  1. INITIALIZE IMAGE DIRECTORY (CLEAN)
C *           2. SET IMAGE PROCESSOR TO FREERUN
C *           3. ALLOW OPERATOR TO ADJUST MOTORS,CAMERA
C *           4. DO EACH IMAGE PROCESSING REQUEST (CMDM70)
C *           5. IF CODE = 0: DO NEXT PROCESS IMMEDIATELY
C *                1: WAIT FOR OPERATOR
C *                2: WRITE IMAGE TO TAPE
C *                3: LET OPERATOR POSITION MOTOR,
C *                   THEN TELL HIM MOTOR POSITION
C
C      SUBROUTINE PING(ITEST,NOTAPE,IERR)
C
C      ITEST = TEST SEQUENCE NO
C      NOTAPE = FLAG FOR NOTAPE
C      IERR  = ERROR CODE RETURNED
C
C      INTEGER S500N(8),ISTAT(2),DIGIT(40),COMND,DIRCK(40)
C      LOGICAL MORE
C
C      $INSERT PTCON
C      $INSERT RTCON
C      $INSERT TCON
C      $INSERT CCON
C      $INSERT SYSCON>PARM.K
C
C      LOGICAL NOTAPE
C      DATA DIGIT/'>DIGITIZE>$A; ',33*'  '/'
C      DATA DIRCK/'>DIRCK; ',36*'  '/'
C
C      CLEAN THE IMAGE DIRECTORY
C      -----
C      CALL CLEAN
C
C      SET IMAGE PROCESSOR TO FREERUN AND GIVE OPERATOR A CHANCE TO FINE
C      TUNE THE MOTORS AND CAMERA
C      -----
C      CALL CMDM70(DIGIT,IERR)
C      WRITE(1,5)
C      5  FORMAT(1X,'ADJUST MOTORS AND CAMERA AS NECESSARY - HIT A KEY WHEN
C      1  READY',/)
C      CALL PAUS(J)
C
C      DO EACH IMAGE REQUEST
C      -----
C      IBIAS = 177
C      DO 100 I = 1,PNIMG
C      COMND = PTREC(IBIAS)
C      IF (COMND.EQ. 0) RETURN
C      IF (COMND.EQ. ' ') GOTO 8
C      IFLAG=FL$RET
C      6  CALL NEXT$(CCHN,CREC,PTREC(IBIAS),CARR,IFLAG,$9000,0.0,0.0)
C      DO 7 J = 1,CNPROC
C      JBIAS = 11 + (J-1)*CNWI
C      ICMD = CREC(JBIAS)
C      IF (ICMD.EQ. 0) GOTO 7
C      IF (ICMD.EQ. ' ') GOTO 7
C      CALL CMDM70(CREC(JBIAS),IERR)
C      7  CONTINUE
C      8  NEXT = IBIAS + 40
C      IBIAS = IBIAS + PNWI
C      MOVE = PTREC(NEXT)
C      IF (MOVE.LE. 0) GOTO 100
C      IF (MOVE.GT. 4) GOTO 98

```

```

C          GOTO (10,20,30,40), MOVE
C
10      CALL PAUS(J)
        GOTO 100
C
C SAVE IMAGE ON TAPE
20      IF (NOTAPE) GOTO 100
        CALL ZMVD(RTINA,S500N,16)
        CALL SVINT(S500N)
C
C CREATE THE TAPE ID RECORD
        CALL SUTAPR(IERR)
        GOTO 100
C
C ALLOW OPERATOR TO POSITION MOTOR, THEN TELL HIM WHERE HE'S AT
30      CALL PAUS(J)
        CALL POSIT(2,IVAL,ISTAT)
        WRITE(1,1)IVAL
1       FORMAT(1X,'CURRENT POSITION IS ',15)
        CALL DMORE(MORE)
        IF (MORE) GOTO 30
        GOTO 100
C
C DIRCK
C
40      CALL CMDM70(DIRCK,IERR)
        CALL TNOUA(PTREC(IBIAS),40)
        CALL PAUS(J)
        GOTO 100
C
C ERROR OCCURRED
C
98      WRITE (1,99) MOVE,I
99      FORMAT('**ERROR** PLANNED NEXT MOVE = ',13,' TEST',13)
C
100     CONTINUE
C
C SET IMAGE PROCESSOR TO FREERUN
        CALL CMDM70(DIGIT,IERR)
C
        RETURN
C
C MIDAS ERROR IN FINDING COMMAND FILE
C
9000    IERR = CERR
        IF (CERR.NE.22 .OR. CERR .NE. 24) GOTO 9900
        CALL RECYCL
        GO TO 6 /* TRY AGAIN
9900    IF (CERR .NE. 7) GOTO 9905
        JBIAS=IBIAS+9
        WRITE(1,9901)(PTREC(I),I=IBIAS,JBIAS),CCNAM
9901    FORMAT(1X,'TEST ',10A2,' NOT FOUND IN DATA BASE ',16A2)
        CALL TDUMP(PTREC(IBIAS),20)
        RETURN
9905    JBIAS = IBIAS + 9
        WRITE(1,9910)CERR,(PTREC(I), I=IBIAS,JBIAS),CCNAM
9910    FORMAT(1X,'MIDAS ERROR ',13,' IN TEST ',16A2/1X,
C'FROM DATA BASE ',16A2)
        RETURN
        END

```

C PLAN: MAIN PLANNING ROUTINE

```

*****
*          REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM          *
*          VERSION 1.0      JUNE 1,1980                          *
*          BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY          *
*****

```

```

*CALLED FROM: AI
*FUNCTION: 1. GET PLANNING FUNCTION DESIRED (GPFNCT)
*          2. CALL ONE OF FOLLOWING SUBROUTINES:
*          CNEWP - CREATE A NEW PLAN
*          CNEWT - CREATE A NEW TEST PLAN
*          MKFMP - MAKE A NEW PLANFROM AN OLD ONE
*          MKFMT - MAKE A NEW TEST PLAN FROM AN OLD ONE
*          MDFYP - MODIFY A PLAN
*          MDFYT - MODIFY A TEST PLAN
*          DLTP - DELETE A PLAN
*          DLTT - DELETE A TEST PLAN
*          DLTR - DELETE A RESULT
*          DSPLD - DISPLAY DATA
*          CNEWC - CREATE A NEW COMMAND FILE
*          MDFYC - MODIFY A COMMAND FILE
*          DLTC - DELETE A COMMAND FILE

```

C SUBROUTINE PLAN(IER)

```

IER = 0
CALL GPFNCT(MENU)
IF (MENU .LE. 0) RETURN

```

```

GOTO (
1 100,200,300,400,500,600,700,800,900,
2 1000,1100,1200,1300,1400
3 ), MENU
GOTO 5

```

```

C CREATE NEW PLAN
100 CALL CNEWP(IER)
IF (IER) 9000,5,9000

```

```

C CREATE NEW TEST PROC.
200 CALL CNEWT(IER)
IF (IER) 9000,5,9000

```

```

C ADD A COMMAND FILE
300 CALL CNEWC(IER)
IF (IER) 9000,5,9000

```

```

C MAKE NEW PLAN FROM OLD
400 CALL MKFMP(IER)
IF (IER) 9000,5,9000

```

```

C MAKE NEW TEST PROC FROM OLD
500 CALL MKFMT(IER)
IF (IER) 9000,5,9000

```

```

C MAKE A NEW COMMAND FROM OLD
600 CALL MKFMC(IER)
IF (IER) 9000,5,9000

```

```

C MODIFY PLAN
700 CALL MDFYP(IER)
IF (IER) 9000,5,9000

```

```

C MODIFY TEST PROC
800 CALL MDFYT(IER)
IF (IER) 9000,5,9000

```

```

C MODIFY A COMMAND FILE
900 CALL MDFYC(IER)
IF (IER) 9000,5,9000

```

```
C  DELETE A PLAN
1000  CALL DLTP(IER)
      IF (IER) 9000,5,9000
C
C  DELETE A TEST PROC.
1100  CALL DLTT(IER)
      IF (IER) 9000,5,9000
C
C  DELETE RESULTS
1200  CALL DLTR(IER)
      IF (IER) 9000,5,9000
C
C  DELETE A COMMAND FILE
1300  CALL DLTC(IER)
      IF (IER) 9000,5,9000
C
C  DISPLAY FUNCTIONS
1400  CALL DSPLO(IER)
      IF (IER) 9000,5,9000
C
C
9000  RETURN
      END
```

C PMOT: POSITION MOTORS

```

*****
*                                     *
*      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
*      VERSION 1.0      JUNE 1,1980                      *
*      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
*                                     *
*****
*CALLED BY: DINSP
*FUNCTION: 1. POSITION A MOTOR (MOTOR)
*          2. IF MOTOR CONTROL PUT TO MANUAL DURING
*          POSITIONING, PAUSES UNTIL IN REMOTE
SUBROUTINE PMOT
$INSERT PTCOM
C TURN ON MOTORS WHICH ARE TO RUN CONTINUOUSLY DURING POSITIONING
C
  IPOS = 167
  NDIR = 1
  DO 5 I = 1,PNMOT
    NSPEED = PTREC(IPOS+1)
    NPOS = PTREC(IPOS)
    IF (NPOS .NE. -1) GOTO 7
    M = I - 1
    CALL MOTION(M,NDIR,NSPEED,IERR)
    IF (IERR .NE. 0) GOTO 30
    IPOS = IPOS + PNWM
  CONTINUE
7
5
C POSITION MOTORS WHICH HAVE ENCODERS
C
  IPOS = 167
  DO 20 I = 1,PNMOT
    NSPEED = PTREC(IPOS+1)
    IF (NSPEED .EQ. 0) GOTO 10
    NPOS = PTREC(IPOS)
    IF (NPOS .LE. 0) GOTO 10
    M = I - 1
    CALL MOTOR(M,NSPEED,NPOS,IERR)
    IF (IERR .NE. 0) GOTO 30
    IPOS = IPOS + PNWM
  CONTINUE
8
10
20
C TURN OFF MOTORS WHICH WERE RUNNING CONTINUOUSLY DURING POSITIONING
C
  IPOS = 167
  DO 25 I = 1,PNMOT
    NSPEED = PTREC(IPOS+1)
    NPOS = PTREC(IPOS)
    IF (NPOS .NE. -1) GOTO 27
    M = I - 1
    CALL MOTION(M,NDIR,0,IERR)
    IF (IERR .NE. 0) GOTO 30
    IPOS = IPOS + PNWM
  CONTINUE
25
  RETURN
C
30
  IF (IERR .NE. 1) GOTO 60
  CALL TNOUA('PUT MOTOR CONTROL IN REMOTE, HIT ANY KEY ',41)
  CALL PAUS(J)
  GOTO 8
C
60
  WRITE(1,61)M,NSPEED,IERR
  FORMAT(1X,'MOTOR # ',I3,' SPEED ',I3,' ERROR ',I3)
  RETURN
61
  END

```

C POSINT: POSITION MOTOR & INITIALIZE ENC PAGE 0001

C POSINT: POSITION MOTOR & INITIALIZE ENCODER

```
C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0   JUNE 1,1980                        *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C *
C *****
C *CALLED BY: INTMOT                                         *
C *FUNCTION: 1. POSITION THE MOTOR TO '0' POSITION            *
C *          2. SET THE ENCODER COUNT TO '0'                *
C
C      SUBROUTINE POSINT(ENCDR,IER)
C          ENCDR = THE ENCODER NUMBER TO INITIALIZE
C          IER   = THE ERROR FLAG (0 IF GOOD, 1 IF GPIOB ERROR)
C
C      NUMBER OF MILLISECONDS PAUSE TO WAIT FOR GEAR BACKLASH
C      INTEGER*4 MSEC
C      PARAMETER MSEC=1000
C
C      INTEGER ENCDR,ENC
C      INTEGER DIR(10),SPEED,VALUE,VALUE1
C      DATA DIR / 0,0,1,0,0,0,0,0,0,0 /
C      IER=0
C
C      MOVE THE MOTOR IN THE DIRECTION SPECIFIED
C
C      10      CALL MOTION(ENCDR,DIR(ENCDR+1),127,IER)
C          IF (IER-1) 20,15,200
C      15      CALL TNOUA('PUT MOTOR CONTROL IN REMOTE, HIT ANY KEY ',41)
C          CALL PAUS(IG0)
C          GOTO 10
C
C      CHECK TO SEE IF THE ENCODER IS STILL CHANGING
C
C      20      CALL POSIT(ENCDR,VALUE1,IER)
C          IF (IER-1) 30,25,200
C      25      CALL TNOUA('PUT MOTOR CONTROL IN REMOTE, HIT ANY KEY ',41)
C          CALL PAUS(IG0)
C          GOTO 20
C      30      CALL SLEEP$(MSEC)
C          CALL POSIT(ENCDR,VALUE,IER)
C          IF(VALUE1-VALUE .EQ. 0)GO TO 50
C          GO TO 20
C
C      STOP THE MOTOR
C
C      50      CALL MOTION(ENCDR,DIR(ENCDR),0,IER)
C
C      CLEAR THE ENCDR
C
C          ENC=OR(ENCDR,:100000)
C          CALL POSIT(ENC,VALUE,IER)
C          CALL POSIT(ENCDR,VALUE,IER)
C
C      SET ERROR FLAG (FOR ENCODER NOT CLEARING)
C
C          IF (VALUE .NE. -1) IER = 1
C
C      200      RETURN
C          END
```

C RETR: RETRIEVE HISTORICAL RECORD

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM
C *      VERSION 1.0      JUNE 1,1980
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY
C *
C *****
C *CALLED BY: AI
C *FUNCTION : 1. GET RETRIEVAL FUNCTION
C *           2. INITIALIZE THE IMAGE PROCESSOR (INTIP)
C *           3. CALL ONE OF THE FOLLOWING SUBROUTINES:
C *              RVING - GET ONE IMAGE
C *              RVIST - RECREATE ONE TEST
C *              RVINSP- RECREATE AN INSPECTION
C *              DSPR - DISPLAY MAIN RESULTS
C *              DSPRT - DISPLAY TEST RESULTS
C *              DSPT - DISPLAY TAPE RECORD
C *              LISTIX- LIST DATA BASE BY INDEXN
C
C ARG 1: IERR  INTEGER
C
C      SUBROUTINE RETR(EQUIP,IERR)
C
C      INTEGER SPCHN,DIRCK(40)
C      LOGICAL EQUIP
C      DATA DIRCK/'>DIRCK ;',36*'  '/
C
C      SET SPOOL CHAN = 0 FOR DISPLAY ONLY
C
C      SPCHN = 0
C
C      GET RETRIEVAL FUNCTION
C
C 10      CALL GRFNCT(MENU)
C          IF (MENU .EQ. 0) RETURN
C          IF (MENU .GT. 3) GOTO 20
C          IF (EQUIP) GOTO 20
C
C      INITIALIZE THE IMAGE PROCESSOR
C
C          CALL INTIP(IERR)
C          IF (IERR .EQ. 0) GOTO 20
C          WRITE(1,15)IERR
C 15      FORMAT(1X,'ERROR ' 13,' IN INITIALIZING IMAGE PROCESSOR')
C          RETURN
C
C 20      GOTO (100,200,300,400,500,600,700), MENU
C
C      RETRIEVE IMAGE FOR DISPLAY
C
C 100      CALL RVING
C           GOTO 10
C
C      RECREATE ONE TEST RESULT
C
C 200      CALL RVIST
C           GOTO 10
C
C      RECREATE AN INSPECTION
C
C 300      CALL RVINSP
C           GOTO 10
C
C      DISPLAY MAIN RESULTS
C
C 400      CALL DSPR(SPCHN)
C           GOTO 10
C
C      DISPLAY TEST RESULTS
C
C 500      CALL DSPRT(SPCHN)
C           GOTO 10

```

```

C
C DISPLAY TAPE RECORD
C
600 CALL DSPT(SPCHN)
    GOTO 10
C
C LIST INDEX
C
700 CALL LISTIX(SPCHN)
    GOTO 10
C
    END

```

C RMNAB: REMOVE NON-ALPHABETICAL CHARAC PAGE 0001

```

C RMNAB: REMOVE NON-ALPHABETICAL CHARACTERS
C
C *****
C *
C * REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM *
C * VERSION 1.0 JUNE 1,1980 *
C * BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY *
C * *****
C
C *GENERAL PURPOSE SUBROUTINE TO REMOVE NON-ALPHABETICAL *
C *CHARACTERS FROM A STRING *
C
C SUBROUTINE RMNAB(IBUF,NCHAR)
C
C IBUF = TEXT STRING
C NCHAR = NO. CHARS IN TEXT, MAX = 80 CHARS.
C
$INSERT SYSCOM>A$KEYS
$INSERT SYSCOM>KEYS.F
C
INTEGER JBUF(40)
C
C INITIALIZE PARAMETERS
C
K = 0
N = NCHAR
CALL ZFIL(JBUF,80,' ')
IF (N .GT. 80) N = 80
C
C PACK INTO TEMP BUFFER
C
DO 100 I=1,N
IC = RS(GCHR$(IBUF,I),8)
IF (IC .LT. :260 .OR. IC .GT. :332) GOTO 100
IF (IC .GT. :271 .AND. IC .LT. :301) GOTO 100
K = K + 1
CALL MCHR$(JBUF,K,IC,2)
100 CONTINUE
C
C NOW REPLACE THE ORIGINAL STRING WITH THE PACKED STRING
CALL MSTR$(JBUF,N,IBUF,N)
RETURN
END

```

C RVING: RETRIEVE AN IMAGE

```

C *****
C *
C *          REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM
C *          VERSION 1.0      JUNE 1,1980
C *          BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY
C *
C *****

```

```

C *CALLED BY: RETR
C *FUNCTION :1. GET TAPE NAME, IMAGE NAME, FILE FROM OPERATOR
C *          2. FIND THE RECORD IN THE TAPE DATA BASE
C *          3. IS CORRECT TAPE MOUNTED?
C *          4. POSITION TAPE TO CORRECT FILE (MOVTAP)
C *          5. WRITE IMAGE TO IMAGE PROCESSOR (CMDM7D)
C

```

SUBROUTINE RVING

```

C $INSERT TCOM
C $INSERT SYSCOM>A$KEYS
C $INSERT SYSCOM>PARAM.K
C

```

```

C LOGICAL ZCM,NEW,MORE
C INTEGER IBUF(36),TAPE(10),IMAGE(25),FILE,ENTER(40),SELECT(40)
C

```

```

C EQUIVALENCE (IBUF,TAPE)          /*TAPE NAME FROM PEDIT
C EQUIVALENCE (IBUF(11),IMAGE)     /*IMAGE NAME FROM PEDIT
C EQUIVALENCE (IBUF(36),FILE)      /*FILE NO FROM PEDIT
C

```

```

C DATA ENTER/'>ENTER>$A:',35*' '/'
C DATA SELECT/'$A>SELECT:',35*' '/'
C

```

C OPEN SCREEN TEMPLATE

```

C CALL VOPEN('DEMO.SCREEN.FT',14,1,ICH,IERR)
C IF (IERR.NE. 0) GOTO 9000
C

```

C GET TAPE NAME, IMAGE NAME, OR FILE NO

```

C CALL ZFIL(IBUF,72,0)
C IS = 200
C IE = 204
C NEWSR = 2 /* DO NOT ERASE SCREEN TO START
C CALL PEDIT(ICH,IBUF,IS,IE,NEWSR)
C FILE = FILE /***CHECKOUT
C

```

C GET RECORD BASED ON PEDIT INFO

```

C IF (TAPE(1).EQ. 0) GOTO 120
C NEW = .NOT.ZCM(TAPE,20,TNAM,20,ICODE)
C IF (.NOT. NEW) GOTO 120
C

```

C FIND TAPENAME REQUESTED IN MIDAS

```

C IFLAG = FL$RET
C CALL NEXT$(TCHN,TREC,TAPE,TPARR,IFLAG,$100,0,0,0,0)
C

```

C NO FILE NUMBER SPECIFIED: FIND CORRECT IMAGE BY IMAGE NAME

```

C IF (FILE) 130,5,50
C DO 10 N = 1,TNFL
C IBIAS = 12 + (N-1)*TNWT
C IF (ZCM(IMAGE,50,TREC(IBIAS),50,ICODE)) GOTO 50
C

```

10 CONTINUE

C CORRECT IMAGE NOT FOUND

```

C CALL ZFIL(TREC,TSZB,0)
C WRITE(1,30)IMAGE
C 30 FORMAT(1X,'IMAGE ',25A2,' NOT FOUND.')
C GOTO 500
C

```

C FILE NUMBER SPECIFIED: FIND CORRECT IMAGE BY FILE NUMBER

C FILE NUMBER FOUND: READ TAPE TO IMAGE DIRECTORY AND IMAGE PROCESSOR

50 WRITE(1,60)TNAM

```

60    FORMAT('MOUNT TAPE: ' ,10A2,'& HIT ANY KEY WHEN READY')
      CALL PAUS(J)
70    CALL MOVTAPE(NEW,FILE,IERR)
      IF (IERR .NE. 0) GOTO 9100
      CALL CLEAN
      CALL DIRCK /****CHECKOUT
      CALL CMDN70(ENTER,IERR)
      CALL CMDN70(SELECT,IERR)
      CALL DIRCK /****CHECKOUT
      TFCNT = TFCNT + 1
      GOTO 500
C
C TAPE RECORD NOT FOUND
C
100   WRITE(1,110)TAPE
110   FORMAT(1X,'TAPE RECORD ',10A2,' NOT FOUND')
      GOTO 500
C
C TAPE IS CURRENT ONE: WHAT IS FILE#?
C
120   IF (FILE) 130,150,70
C
C FILE# LESS THAN ZERO - GIVE 'EM ANOTHER CHANCE
C
130   CALL TNOU('PLEASE GIVE MORE SPECIFIC DATA',29)
135   CALL PAUS(J)
      GOTO 3
C FILE# NOT SPECIFIED: FIND IMAGE BY IMAGE NAME
C
150   DO 160 N = 1,TNFL
      IBIAS = 12 + (N-1)*TNWT
      IF (ZCH(IMAGE,50,TREC(IBIAS),50,ICODE)) GOTO 70
160   CONTINUE
      GOTO 20
C
C MORE?
C
500   CALL DMORE(MORE)
      IF (MORE) GOTO 3
C
C CLOSE SCREEN TEMPLATE
C
600   CALL CLOS$A(ICH)
      RETURN
C
C
9000  WRITE(1,9010)IERR
9010  FORMAT(1X,'ERROR ',I3,' IN OPENING DEMO.SCREEN.FT')
      RETURN
C
9100  WRITE (1,9101) FILE,IERR
9101  FORMAT('ERROR IN LOCATING FILE',I3,' ERROR =' ,I3)
      GOTO 600
C
      END

```

PAGE 0001

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1

```

*****
*
* REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM
* VERSION 1.0 JUNE 1,1980
* BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY
*
*****

*CALLED BY: RETR
*FUNCTION: 1. GET PART#,INSP NAME, SERIAL# FROM OPERATOR
*          2. FIND CORRECT MATCH IN RESULTS DATA BASE
*          3. OPERATOR: MOUNT CORRECT TAPE
*          4. POSITION TAPE TO FIRST INSPECTION
*          5. OUTPUT IMAGE, PROCESS IMAGE AS WAS DONE ORIGINALLY
*          6. REPEAT FOR EACH TEST

```

SUBROUTINE RVINSP

```
WRITE(1,1)
FORMAT('*** NOT IMPLEMENTED ***',/, 'DO ONE TEST AT A TIME.')
RETURN
END
```

C RVST: RECREATE ONE TEST RESULT

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0   JUNE 1,1980                        *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C *****
C *CALLED BY: RETR                                          *
C *FUNCTION : 1. GET PART#, INSP NAME, SERIAL#, TESTNAME (PEDIT) *
C *           2. CREATE KEYWORD (CRPKEY)                   *
C *           3. FIND TEST RESULT RECORD IN DATA BASE      *
C *           4. CORRECT TAPE MOUNTED?                     *
C *           5. POSITION TAPE (MOVTAP)                      *
C *           6. DISPLAY TEST RESULT RECORD ON SCREEN (RPTGEN) *
C *           7. OUTPUT IMAGE TO IMAGE PROCESSOR            *
C *           8. PROCESS IMAGE AS WAS DONE ORIGINALLY       *
C
C      SUBROUTINE RVST
C
C $INSERT RTCOM
C $INSERT TCOM
C $INSERT CCOM
C $INSERT SYSCOM>PARM.K
C $INSERT SYSCOM>A$KEYS
C
C      INTEGER IID(5),PART(10),IN(10),SNO(10),TN(10),IBUF(45),RKEY(20)
C      INTEGER ENTER(40),SELECT(40)
C      LOGICAL NEW,MORE,YSNO$A,ZCM
C      EQUIVALENCE (IBUF,IID)
C      EQUIVALENCE (IBUF(6),PART)
C      EQUIVALENCE (IBUF(16),IN)
C      EQUIVALENCE (IBUF(26),SNO)
C      EQUIVALENCE (IBUF(36),TN)
C
C      DATA ENTER/'>ENTER>$A;',35* ' '/
C      DATA SELECT/'$A>SELECT;',35* ' '/
C
C OPEN SCREEN TEMPLATE
C
C      CALL VOPEN$('DEMO.SCREEN.FT',14,1,IPCH,IERR)
C      IF (IERR .NE. 0) GOTO 9000
C
C GET PART#,INSP NAME,SERIAL#,TEST NAME
C
C      CALL ZFIL(IBUF,90,0)
C      IS = 0
C      IE = 6
C      NEWSCR = 2      /*DO NOT ERASE SCREEN TO START
C      CALL PEDIT(IPCH,IBUF,IS,IE,NEWSCR)
C
C CREATE RTCOM KEYNAME,FIND ITS LENGTH
C
C      CALL CRPKEY(PART,20,IN,20,SNO,20,TN,20,RKEY,40)
C      NCHAR = LSIZE(RKEY,40)
C      IFLAG = FL$RET + FL$BIT
C      CALL TNOU(RKEY,40)
C      CALL TDUMP(RKEY,40)
C
C FIND THE RECORD FOR RTCOM
C
C      CALL ZFIL(RTREC,RTSZB,0)
10    CALL NEXT$(RTCHN,RTREC,RKEY,RTARR,IFLAG,$9100,0,0,0,NCHAR)
C      IFLAG = FL$BIT + FL$RET + FL$USE
C      IF (RTTN(1) .EQ. 0) GOTO 10
C
C CHANGE PASS/FAIL CODE TO ASCII
C
C      RTPF = RTPF + 1260
C
C IS CORRECT TAPE MOUNTED?
C
C      NEW = .FALSE.

```

```

      IF (TNAM(1) .EQ. 0) GOTO 15
      IF (ZCH(RTTNM,20,TNAM,20,ICODE)) GOTO 40
15    NEW = .TRUE.
      CALL ZFIL(TREC,TSZB,0)
      IFLAG = FL$RET
      CALL NEXT$(TCHN,TREC,RTTNM,TPARR,IFLAG,$9200,0,0,0,0)
      WRITE(1,20)RTTNM
20    FORMAT('PLEASE MOUNT TAPE: ',10A2,'% HIT ANY KEY WHEN READY')
      CALL PAUS(J)
C
40    CALL MOVTAP(NEW,RTFNM,IERR)
      IF (IERR .NE. 0) GOTO 9300
C
C READ IMAGE TO IMAGE PROCESSOR
C
      CALL CLEAN
      CALL CDM70(ENTER,IERR)
      CALL CDM70(SELECT,IERR)
      TFCNT = TFCNT + 1
C
C DISPLAY TEST NAME, DESCRIPTION,PASS/FAIL,COMMENTS,TAPENAME,FILE#,IMAGENAME
C
      IS = 304
      IE = 318
      NEWSCR = 2          /* DO NOT ERASE SCREEN TO START
      CALL RPTGEN(0,IPCH,RTREC,LINES,IS,IE,NEWSCR)
C
C FIND IMAGE COMMAND THAT CAUSES IMAGE TO BE WRITTEN TO TAPE
C
      NBIAS = 217
C
      DO 50 I = 1,8
        IMG = I
        NEXT = RTREC(NBIAS)
        IF (NEXT .EQ. 2) GOTO 60
        NBIAS = NBIAS + RNWI
50    CONTINUE
C
C RECREATE THE IMAGE PROCESSING
60    IBIAS = NBIAS + 1
C
      DO 70 I = IMG,8
        NBIAS = NBIAS + RNWI
        ICMD = RTREC(IBIAS)
        IF (ICMD .EQ. ' ') GOTO 66
        IF (ICMD .EQ. 0) GOTO 80
        IFLAG=FL$RET
62    CALL NEXT$(CCHN,CREC,RTREC(IBIAS),CARR,IFLAG,$9500,0,0,0,0)
        DO 64 J = 1,CNPROC
          JBIAS = 11 + (J-1)*CNWI
          PROC = CREC(JBIAS)
          IF (PROC .EQ. 0) GOTO 64
          IF (PROC .EQ. ' ') GOTO 64
          CALL CDM70(CREC(JBIAS),IERR)
          IF (IERR .NE. 0) GOTO 9400
64    CONTINUE
          IF (RTREC(NBIAS) .EQ. 0) GOTO 66
          CALL PAUS(J)
66    IBIAS = NBIAS + 1
70    CONTINUE
C
C ANY MORE MATCHES?
C
80    IFLAG = FL$BIT + FL$RET + FL$USE
      MORE = YSN0$A('LOOK FOR MORE MATCHES',21,A$DNO)
      IF (MORE) GOTO 10
      GOTO 9900
C
C ERRORS
C
9000  WRITE(1,9010)IERR
9010  FORMAT(1X,'ERROR ',13,' IN OPENING DEMO.SCREEN.FT')
      RETURN
C
9100  IF (RTERR .EQ. 7) GOTO 9900

```

```

      IF (RTERR .NE. 22 .AND. RTERR .NE. 24) GOTO 9110
      CALL RECYCL
      GOTO 10
9110  WRITE(1,9120)RTERR,RKEY
9120  FORMAT(1X,'MIDAS ERROR ',I3,' IN FINDING ',20A2)
      GOTO 9900
C
9200  IF (TPERR .NE. 7) GOTO 9208
      WRITE(1,9201) RTTNM
9201  FORMAT('DATA BASE ERROR: TAPE ',10A2,' NOT FOUND.')
      GOTO 9900
9208  WRITE(1,9210)TPERR,RTTNM
9210  FORMAT(1X,'MIDAS ERROR ',I3,' IN FINDING TAPE ',10A2)
      CALL TDUMP(RTTNM,20)
      GOTO 9900
C
9300  WRITE(1,9310)IERR,RTTNM,RTFNM
9310  FORMAT(1X,'MOVTAPE ERROR ',I3,' IN MOVING TAPE ',10A2,' TO FILE '
C      ,I3)
      GOTO 9900
C
9400  WRITE(1,9410)IERR,IBIAS
9410  FORMAT(1X,'CMDM70 ERROR ',I3,' IN IMAGE PROCESS ',I3)
      GOTO 9900
C
9500  IF (CERR .NE. 7) GOTO 9510
      JBIAS=IBIAS+9
      WRITE(1,9501)(RTREC(I),I=IBIAS,JBIAS),CCNAM
9501  FORMAT(1X,'TEST ',10A2,' NOT FOUND IN DATA BASE ',16A2)
CX    CALL TDUMP(RTREC(IBIAS),20)
      GOTO 9900
9510  IF (CERR .NE. 22 .OR. CERR .NE. 24) GOTO 9520
      CALL RECYCL
      GO TO 62          /* TRY AGAIN
9520  JBIAS = IBIAS + 9
      WRITE(1,9521)CERR,(RTREC(I),I=IBIAS,JBIAS),CCNAM
9521  FORMAT(1X,'MIDAS ERROR ',I3,' IN TEST ',16A2/1X,
C      'FROM DATA BASE ',16A2)
      GOTO 9900
C
C CLOSE SCREEN TEMPLATE
C
9900  CALL CLOS$A(IPCH)
      RETURN
      END

```

C SPOLIT: SPOOL RETRIEVAL INFORMATION

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0      JUNE 1,1980                      *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY     *
C *
C *****

```

```

C *CALLED BY: DSPLOD
C *FUNCTION : 1. CLOSE THE INTERMEDIATE SPOOL FILE
C *           2. SPOOL THE FILE TO THE LINE PRINTER (SPOOL)
C *           3. DELETE THE INTERMEDIATE SPOOL FILE
C *

```

```

C ARG 1: SPCHN  INTEGER
C ARG 2: SPFNAM STRING
C

```

```

C      SUBROUTINE SPOLIT(SPCHN,SPFNAM)
C

```

```

C $INSERT SYSCOM>A$KEYS
C $INSERT SYSCOM>KEYS.F
C $INSERT SYSCOM>PARM.K
C

```

```

C      INTEGER ICHN(2),BSIZE,NSIZE
C      PARAMETER BSIZE = 2000, NSIZE = 7
C      PARAMETER NBTS = NSIZE*2
C      INTEGER BUFFER(BSIZE),INFO(12),SPCHN,SPFNAM(NSIZE)
C      LOGICAL DEL
C

```

```

C      GET 2 FREE CHANNELS FOR THE SPOOLER & LOAD INFO
C

```

```

C      CALL FREECH(2,ICHN)
C      INFO(1) = ICHN(1)
C      INFO(2) = ICHN(2)
C      INFO(3) = 0
C      INFO(4) = ' '
C      INFO(5) = ' '
C      INFO(6) = ' '
C      INFO(7) = 0
C      INFO(11) = 0
C

```

```

C      CLOSE SPOOLFILENAME FILE
C

```

```

C      CALL CLOS$(SPCHN)
C      KEY = 1 /* COPY FILE INTO SPOOL QUEUE
C

```

```

C      /*****CHECKOUT
C      IERR = IERR      /****CHECKOUT
C      JSIZ=BSIZE
C      WRITE(1,9000) SPCHN,KEY,SPFNAM,INFO,JSIZ,IERR
9000  FORMAT('SPOOL CHAN =',I3,/
C      - 'KEY = ',I3,/
C      - 'FILE NAME= ',7A2,/
C      - 'INFO = ',3I4,3A2,I4,3A2,I4,I4,/
C      - 'BUFFER ',I6,/
C      - 'ERROR = ',I3)
C

```

```

C      SPOOL OUT THE HISTORY DATA
C

```

```

C      CALL SPOOL$(KEY,SPFNAM,NBTS,INFO,BUFFER,BSIZE,IERR)
C      IF (IERR .EQ. 0) GOTO 100
C      WRITE(1,10)IERR,SPFNAM
10    FORMAT(1X,'ERROR ',I3,' IN SPOOLING ',7A2)
C

```

```

C      DELETE SPFNAM
C

```

```

100   DEL = DELE$(SPFNAM,14)
C      IF(DEL) RETURN
C      WRITE(1,150)IERR,SPFNAM
150   FORMAT(1X,'ERROR ',I3,' IN DELETING ',SPFNAM,7A2)
C      RETURN
C      END

```

C SVIND: SAVE IMAGE ON DISK

C SVIND: SAVE IMAGE ON DISK

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0      JUNE 1,1980                      *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C *
C *****
C *CALLED BY: SURES                                         *
C *FUNCTION:  SAVE THE IMAGE ON DISK                        *
C
C      SUBROUTINE SVIND(NAME)
C      CALL TNOUA('SAVE IMAGE: ',12)
C      CALL TNOUA(NAME,32)
C      CALL TNOU (' NOT IMPLEMENTED!',17)
C      RETURN
C      END

```

C SVINT: SAVE IMAGE ON TAPE

C SVINT: SAVE IMAGE ON TAPE

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0      JUNE 1,1980                      *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C *
C *****
C *CALLED BY: PIMG                                         *
C *FUNCTION:  1. ADD THIS IMAGE'S NAME TO THE TAPE RECORD  *
C *           2. WRITE THE IMAGE TO TAPE (CMD70)           *
C
C      SUBROUTINE SVINT(S500N)
C      S500N = SYSTEM 500 IMAGE NAME
C      $INSERT RTCOM
C      $INSERT TCOM
C      $INSERT SYSCOM>A$KEYS
C      INTEGER LINE(40)
C
C      MOVE IMAGE NAME TO TREC
C      IBIAS = 12 + TNFL*TNWT
C      NBYT = TNWT*2
C      CALL ZMVD(RTIMN,TREC(IBIAS),NBYT)
C      TNFL = TNFL + 1
C      TFCNT = TFCNT + 1
C      FORMAT A COMMAND LINE TO WRITE IMAGE TO TAPE
C      CALL ZHV('$A>XFER;',8,LINE,80)
C      CALL ZMVD(S500N,LINE,16)
C      CALL PACK(LINE,80)
C      WRITE TAPE USING THE S500 COMMAND LINE
C      CALL CMD70(LINE,IERR)
C      RETURN
C      END

```

C SVMRES: SAVE RESULTS - MAIN RECORD

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM
C *      VERSION 1.0      JUNE 1, 1980
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY
C *
C *****
C *
C *      *CALLED BY: DINSF
C *      *FUNCTION: 1. CREATE MAIN RESULTS KEYWORD
C *      *              2. ADD MAIN RESULTS RECORD & TESTNAMES TO DATA BASE
C *
C      SUBROUTINE SVMRES(IEERR)
C $INSERT RCOM
C $INSERT PCOM
C $INSERT SYSCOM>PARM.K
C      INTEGER RKEY(20),BIAS
C
C      CALL ZFIL(RKEY,40,' ')
C      CALL CRPKEY(RPN,20,RIN,20,RSNO,20,0,0,RKEY,40) /*CR RES KEY
C      CALL APDAT(RKEY,40) /*APPEND DATE
C
C      ADD MAIN RESULT FILE TO DATA BASE
C      CALL ADD1$(RLCHN,RREC,RKEY,RLARR,FL$RET,$9000,0,0,0,0)
C
C      ADD TEST NAMES TO DATA BASE AS SECONDARD KEYNAMES
C      DO 100 I = 1,PNT
C          BIAS = 43 + (I-1)*RNWT /* FIND TEST NAME
C          CALL ADD1$(RLCHN,RKEY,RREC(BIAS),RLARR,FL$USE,$9050,1,0,0,0)
100      CONTINUE
C          IEERR = 0
C          RETURN
C
C      ERROR IN INSERTING MAIN RECORD
C
C      9000 IEERR = RLERR
C          WRITE(1,9010)IEERR,RKEY
C      9010 FORMAT(1X,'MIDAS ERROR ',I2,' IN ADDING ',20A2,' TO RESULT DB')
C          RETURN
C
C      ERROR IN INSERTING A SECONDARY KEY
C
C      9050 WRITE(1,9060)IEERR,RREC(BIAS),I
C      9060 FORMAT(1X,'MIDAS ERROR ',I2,' IN ADDING ',25A2,' INDEX: ',I2)
C          RETURN
C          END

```

C SVRES: SAVE RESULTS OF INSPECTION

```

C *****
C *
C *      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
C *      VERSION 1.0      JUNE 1,1980                      *
C *      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
C *
C *****
C
C *CALLED BY: DINSF
C *FUNCTION:  1. GET PASS/FAIL CODE,COMMENTS ON TEST
C *           2. OUTPUT TAPE NAME,FILE NUMBER, IMAGE NAME
C *           3. SAVE THE TEST RESULTS (SVTSTR)
C *           4. SET IMAGE PROCESSOR TO FREERUN
C

```

SUBROUTINE SVRES(DEFAULT,NOTAPE,ILOOP,RTKEY,IERR)

```

C $INSERT RTCOM
C $INSERT RCOM
C $INSERT SYSCOM>A$KEYS
C

```

```

      LOGICAL DEFAULT,NOTAPE
      INTEGER DIGIT(40)
      DATA DIGIT/'>DIGITIZE>$A; ',33*' '/'

```

IF (.NOT.DEFAULT) GOTO 40

C DEFAULT PLAN - SAVE RESULTS? TAPE OR DISK?

```

C -----
C 5      WRITE(1,10)
C 10     FORMAT(/,'DEFAULT PLAN USED'/
C        C1X,'0  DO NOT SAVE RESULTS'/
C        11X,'1  SAVE RESULTS ON DISK'/
C        21X,'2  SAVE RESULTS ON TAPE')
C        READ(1,20,ERR=5)ISAVE
C 20     FORMAT(I1)
C        IF (ISAVE.EQ. 0) GOTO 100
C        GOTO (30,40),ISAVE
C        GOTO 5

```

C DEFAULT PLAN - SAVE RESULTS ON DISC

```

C -----
C 30     CALL SVIND(RTIMN)
C        GOTO 100

```

C SAVE RESULTS ON TAPE

```

C -----
C OPEN SCREEN TEMPLATE
C 40     CALL VOPEN$('DEMO.SCREEN.FT',14,1,ICHN,IERR)
C        IF (IERR.EQ. 0) GOTO 60
C        WRITE(1,50)IERR
C 50     FORMAT(1X,'ERROR ',12,' IN OPENING DEMO.SCREEN.FT')
C        RETURN

```

C SET UP PEDIT PARAMETERS

```

C -----
C 60     IS = 25
C        IE = 36
C        NEWSR = 2      /* DO NOT ERASE SCREEN TO START
C        CALL PEDIT(ICHN,RTREC,IS,IE,NEWSR)

```

C CLOSE SCREEN FILE

```

C -----
C      CALL CLOS$A(ICHN)

```

C SAVE TEST RESULTS

```

C -----
C      IF (DEFAULT) GOTO 100
C      CALL SVTSTR(RTKEY,IERR)

```

C SET IMAGE PROCESSOR TO FREERUN

```

C -----
C 100    CALL CLEAN
C        CALL CNDW70(DIGIT,IERR)

```

C SVRES: SAVE RESULTS OF INSPECTION

PAGE 0002

RETURN
END

C SVTSTR: SAVE TEST RESULTS

PAGE 0001

C SVTSTR: SAVE TEST RESULTS

```
C *****  
C *  
C *          REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM          *  
C *          VERSION 1.0      JUNE 1,1980                          *  
C *          BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY          *  
C *          *****  
C *  
C *CALLED BY: SVRES *  
C *FUNCTION: SAVE TEST RESULTS TO DATA BASE *  
C  
C      SUBROUTINE SVTSTR (KEY,IERR)  
C  
C      $INSERT RCOM  
C      $INSERT RTCOM  
C      $INSERT SYSCOM>PARM.K  
C  
C      INTEGER KEY(25)  
C  
C      IF (RTPF .EQ. 0) GOTO 10  
C      RPF = RTPF  
C      INSERT TEST RESULTS INTO DATA BASE  
10      IFLAG = FL$RET  
C      CALL ADD1$(RTCHN,RTREC,KEY,RTARR,FL$RET,$9000,0,0,0,0)  
C      RETURN  
9000      IERR = RTERR  
C      WRITE(1,9010)IERR,KEY  
9010      FORMAT(1X,'MIDAS ERROR ',12,',', KEYWORD = ',25A2)  
C      RETURN  
C      END
```

C SVTAPR: SAVE THE TAPE RECORD (MIDAS)

```

*****
*          REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM          *
*          VERSION 1.0      JUNE 1,1980                          *
*          BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY          *
*****
*CALLED BY: PING
*FUNCTION:  ADD THE IMAGE NAME TO THE RECORD OF THIS TAPE
SUBROUTINE SVTAPR(IERR)
C
$INSERT TCOM
$INSERT SYSCOM>PARM.K
      INTEGER TMPREC(TSZW)
C LOCK TAPE DATA BASE FOR EDITING
      IFLAGS = FL$RET
10    CALL LOCK$(TCHN,TMPREC,TNAM,TPARR,IFLAGS,$9000,0,0,0,0)
C UPDATE TAPE DATA BASE RECORD
      IFLAGS = FL$USE
      CALL UPDAT$(TCHN,TREC,TNAM,TPARR,IFLAGS,$9050,0,0,0,0)
      IBIAS = 12 + (TNFL - 1)*TNWT
      CALL ADD1$(TCHN,TREC,TREC(IBIAS),TPARR,IFLAGS,$9200,1,0,0,0)
      IERR = 0
      RETURN
C ERROR FROM LOCK:IF BUSY, RECYCLE
9000  IF (TPERR.NE. 22 .AND .TPERR.NE. 24) GOTO 9100
      CALL RECYCL
      GOTO 10          /*BUSY - DIAL AGAIN
C WRONG NUMBER? - ERROR FROM LOCK$
9100  WRITE(1,9110)TPERR,TNAM
9110  FORMAT(1X,'MIDAS ERROR ',I2,' FROM LOCK$, TNAM= ',10A2)
      IERR = TPERR
      RETURN
C ERROR IN UPDATING
9050  WRITE(1,9060)TPERR,TNAM
9060  FORMAT(1X,'MIDAS ERROR ',I2,' FROM LOCK$,TNAM = ',10A2)
      IERR = TPERR
      RETURN
9200  WRITE(1,9210)TPERR
9210  FORMAT(1X,'MIDAS ERROR ',I3,' IN ADDING IMAGE NAME')
      IERR = TPERR
      RETURN
      END

```

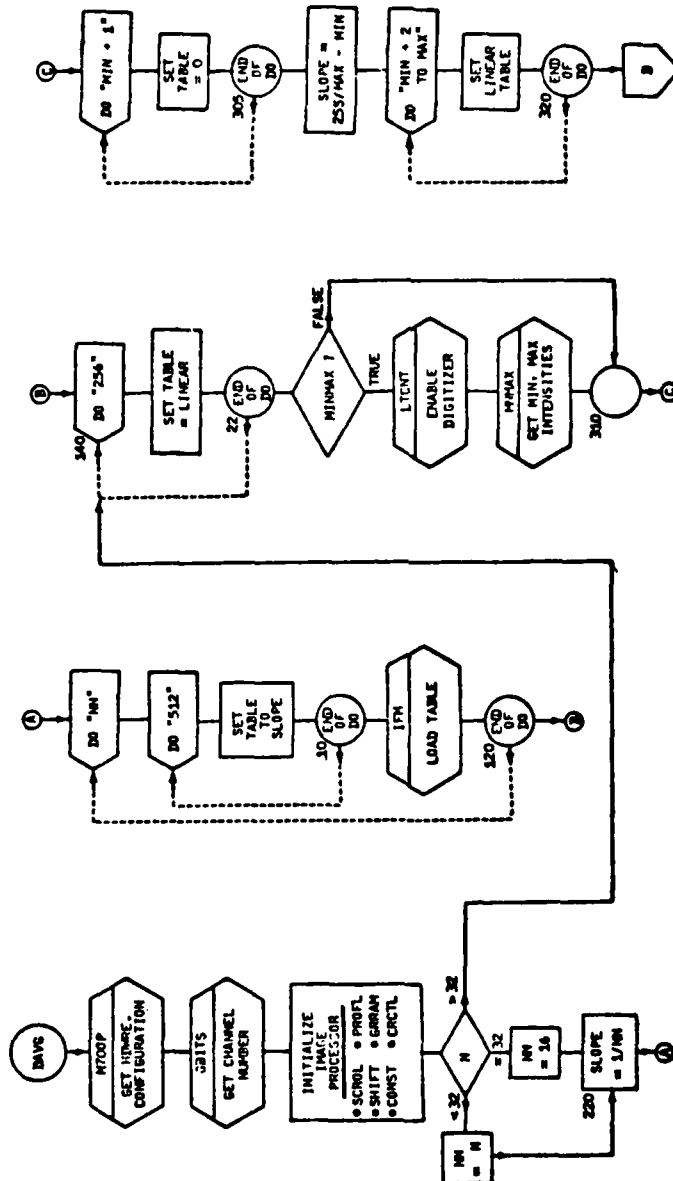
APPENDIX 6

STC SYSTEM 500 SOFTWARE ADDITIONS, LOGIC AND CODE

DAVG - PRIMITIVE FOR AVERAGING VIDEO FRAMES

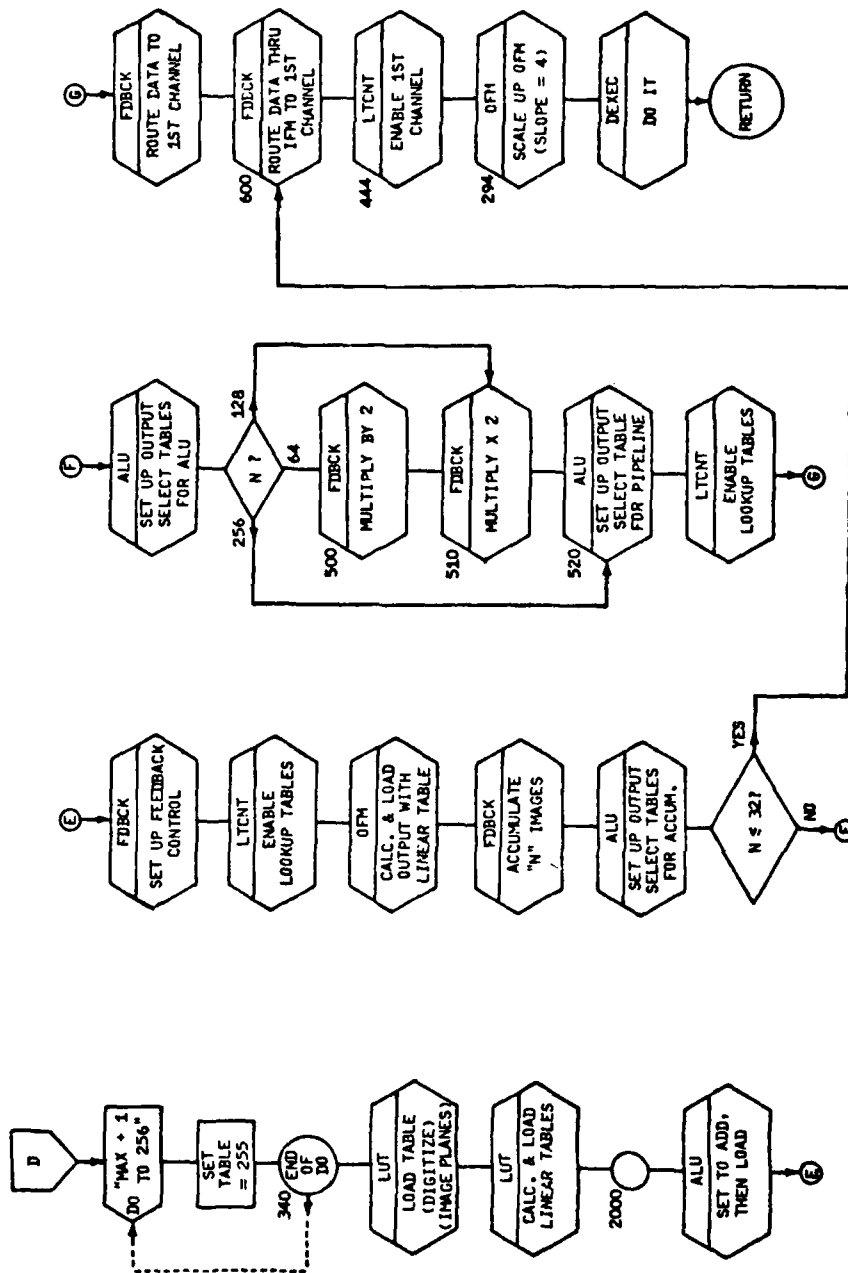
ARGUMENTS: 8

- INTEGER: FUNCTION CONTROL BLOCK
- INTEGER: BUFFER
- INTEGER: NO. OF FRAMES TO AVERAGE
- INTEGER: TIME TO DISPLAY IMAGE
- INTEGER: 1ST BREAKPOINT (MIN)
- INTEGER: 2ND BREAKPOINT (MAX)
- LOGICAL: KEYWORD (MINMAX)
- LOGICAL: BATCH



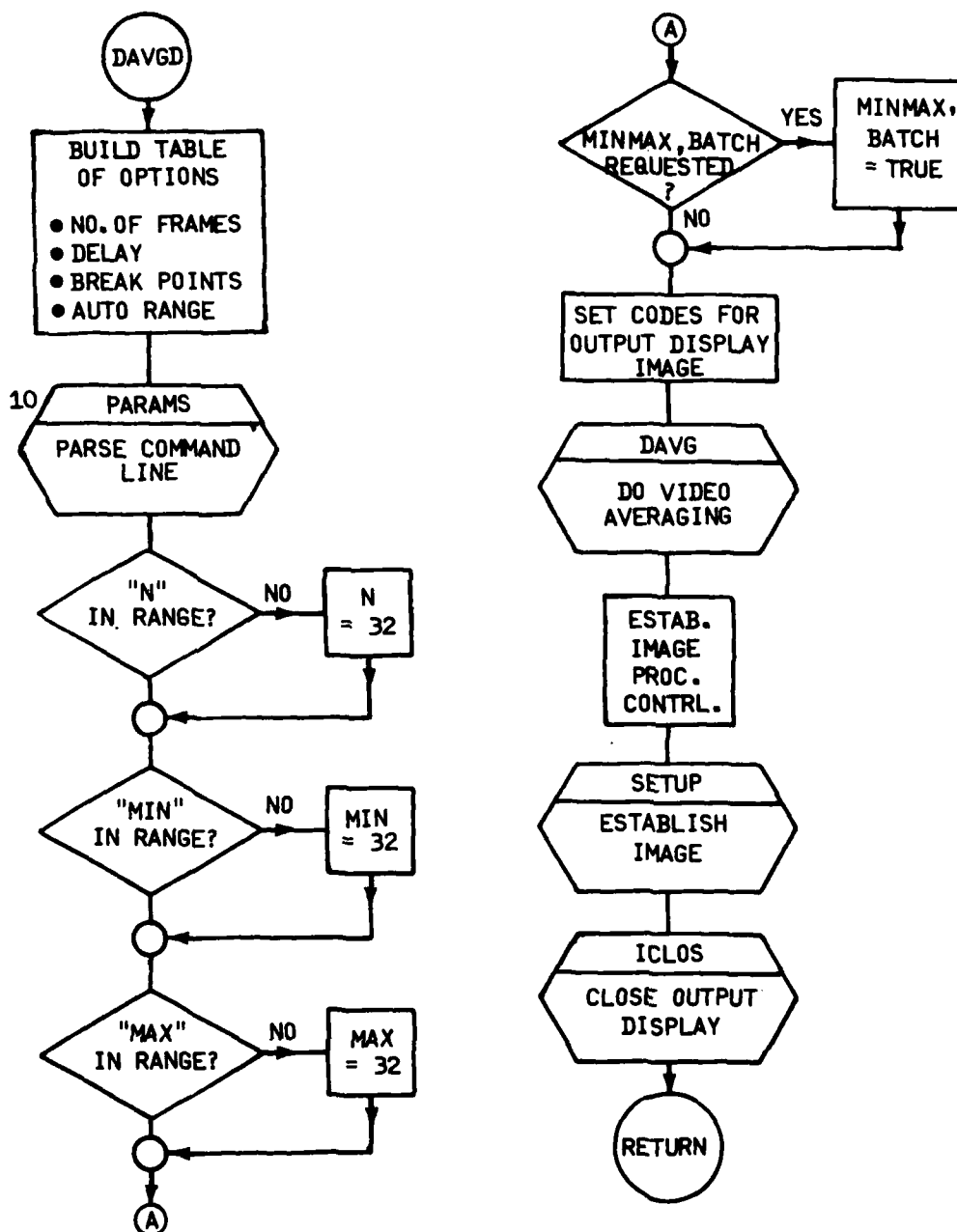
CONTINUED ON
NEXT PAGE

DAVG - PRIMITIVE FOR AVERAGING VIDEO FRAMES (CONT'D)



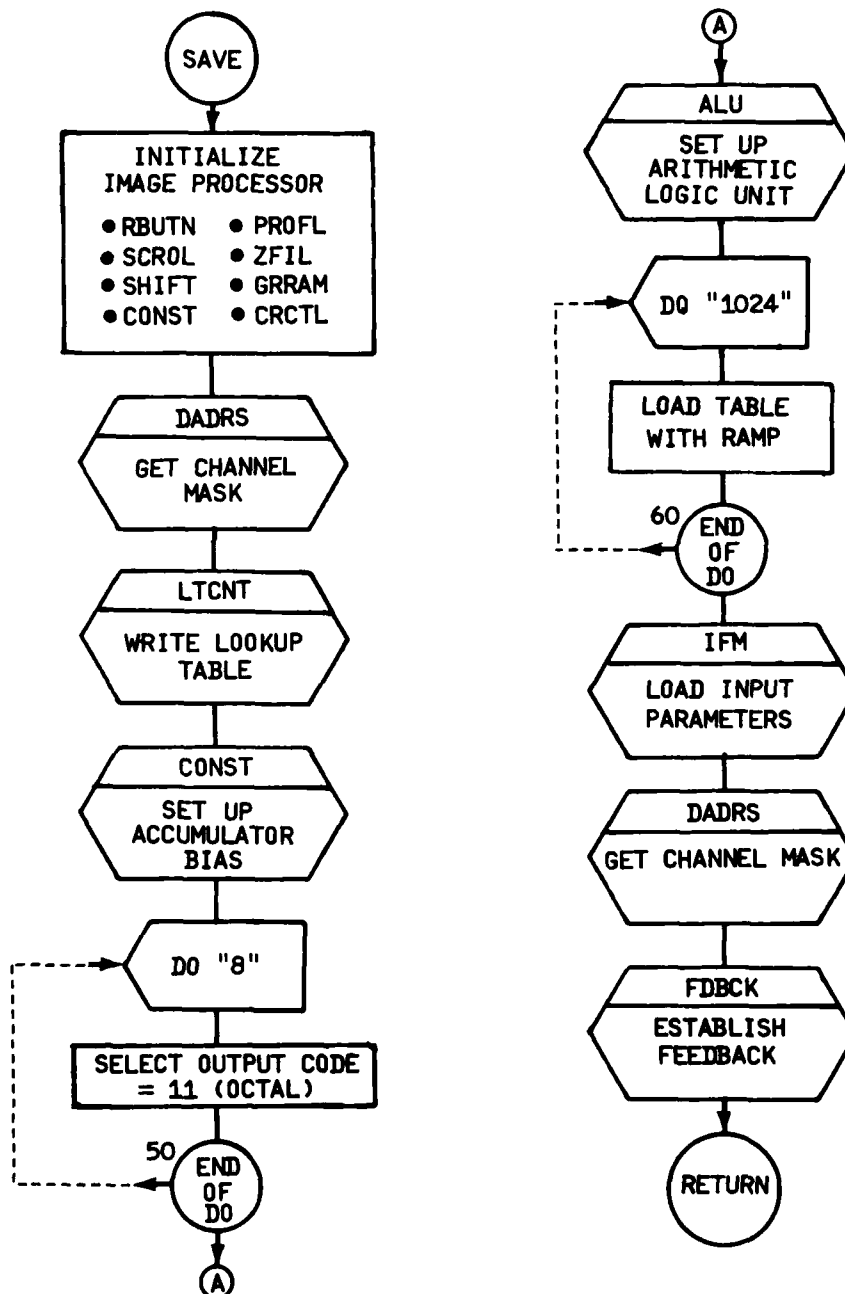
DAVGD - INTERFACE TO PRIMITIVE FOR AVERAGING VIDEO FRAMES

ARGUMENTS: 5
 • INTEGER: FCB
 • INTEGER: ZERO
 • INTEGER: NIDS
 • INTEGER: NODS
 • INTEGER: BUFFER



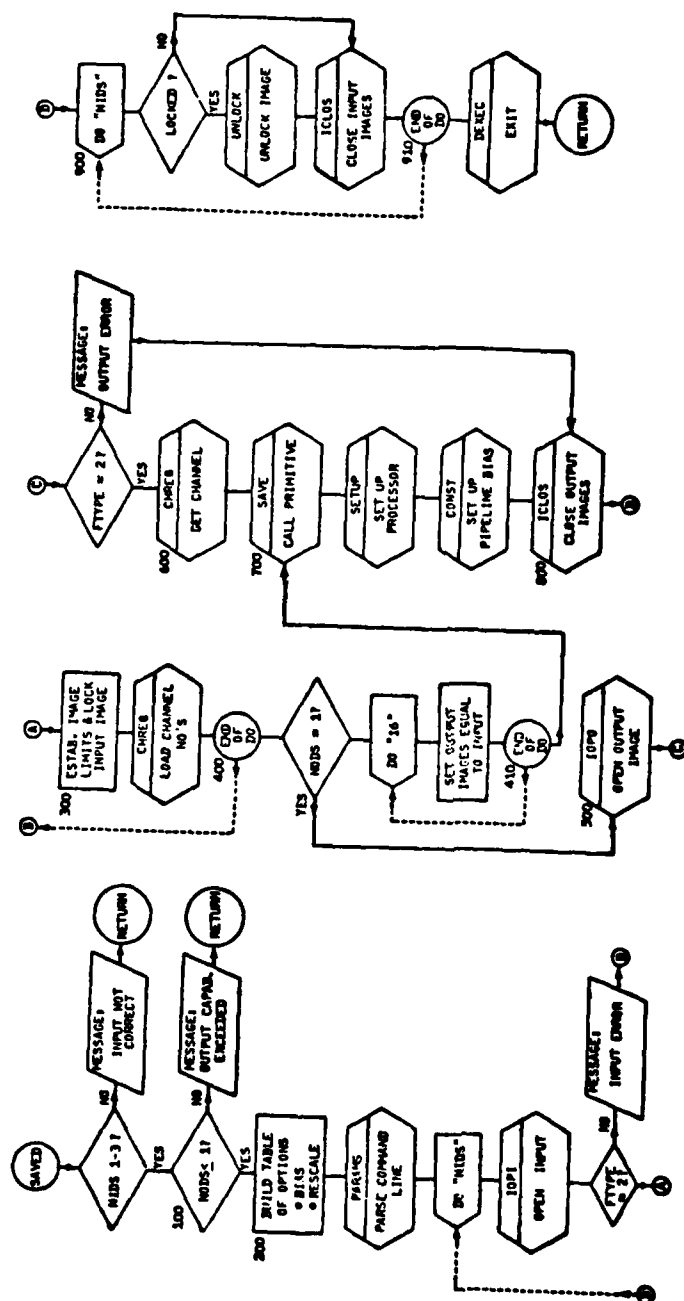
SAVE - PRIMITIVE TO SAVE OUTPUT OF PIPELINE

ARGUMENTS: 7
 • INTEGER: FCB
 • INTEGER: BUFFER
 • INTEGER: ICHNNO
 • INTEGER: DCHNNO
 • INTEGER: NBANDS
 • INTEGER: BIAS
 • REAL: RSCALE



SAVED - SAVE THE OUTPUT OF THE PIPELINE

ARGUMENTS: 5
 • INTEGER: FCB
 • INTEGER: ZERO
 • INTEGER: NIDS
 • INTEGER: NODS
 • INTEGER: Ibuff



C DAVG: PRIMITIVE FOR AVERAGING VIDEO FRA PAGE 0001

C DAVG: PRIMITIVE FOR AVERAGING VIDEO FRAMES

```
*****
*                                     *
*      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
*      VERSION 1.0   JUNE 1,1980                        *
*      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
*                                     *
*****
```

MODIFIED TO RUN FROM AUTOMATIC INSPECTION SYSTEM

SUBROUTINE DAVG (FCB, BUFFER, N, WAIT, MIN, MAX, MINMAX, BATCH)

DESCRIPT - PRIMITIVE TO AVERAGE 'N' FRAMES IN THE ACCUMULATOR,
AND DISPLAY THE RESULT. CYCLE REPEATS UNTIL STOPPED
OR SAVED BY BUTTON PUSH. (DELETED)

PASSED IN - BUFFER WORK SPACE (1024+)
N NUMBER OF FRAMES TO AVERAGE / CYCLE
WAIT TIME IN 1/10TH OF SEC. TO DISPLAY IMAGE
MIN 1ST BREAKPOINT ON IFM PIECEWISE MAPPING
MAX 2ND BREAKPOINT ON IFM PIECEWISE MAPPING
MINMAX KEYWORD FOR AUTOMATIC RANGE FINDING

RETURNED - NONE

SUBS - M700P GETS VERSION INFORMATION
GBITS FILLS ARRAY WITH BIT PATTERN IN WORD
PBITS FILLS WORD WITH BIT PATTERN IN ARRAY
DMASK COMPUTES CHANNEL MASK FROM CHANNEL NUMBER
RBUTN CLEAR THE BUTTON WORD (DELETED)
SCROL CLEAR THE SCROL REGISTER
SHIFT CLEAR THE OFM SELECT SHIFT
CONST CLEAR CONSTANT REGISTER
LTCNT WRITES LUT MASK TO ENABLE CHANNELS
MINMAX FIND THE MINIMUM AND MAXIMUM OF IMAGE
LUT LOADS THE LUT WITH SPECIFIED MAP
OFM LOADS THE OFM WITH SPECIFIED MAP
ALU DEFINES ALU OPERATION
FDBCK WRITES THE FEEDBACK LOOP CONTROL WORD
DEXEC FLUSHES THE M70 COMMAND BUFFER
DWAIT PAUSE FOR AWHILE
IFM LOADS THE IFM WITH SPECIFIED MAP

INTEGER FCB(1), BUFFER(1)
INTEGER N, WAIT, MIN, MAX
LOGICAL MINMAX

INTEGER VRSION, DIGCHN, DIGMSK
INTEGER X, Y
INTEGER OUTSEL(8)
REAL SLOPE

INTEGER PBITS, DMASK /* EXTERNAL FUNCTIONS

GET VERSION INFO

1 CALL M700P (FCB, VRSION, NCHAN, LEVELS, IRMEM, I, I, I,
IFDBCK, I, I, I, I, I, I, I, I, I, I)

GET DIGITIZER CHANNEL NUMBER

CALL GBITS (IRMEN, BUFFER, 16)
DIGCHN = PBITS (BUFFER(9), 4)
DIGMSK = DMASK (DIGCHN)

INITIALIZE M70

CALL SCROL (FCB, 0, 0, 17, 0, 0)
CALL SHIFT (FCB, 0, 0, 0, 0, 0)
CALL CONST (FCB, 0, 0, 0, 0, 0)

C DAVG: PRIMITIVE FOR AVERAGING VIDEO FRA PAGE 0002

```

CALL PROFL (FCB, 0, 0, 0, 0, 0)
CALL ZFIL (BUFFER, 512, 0)
CALL GRRAM (FCB, BUFFER, 0, 0)
CALL CRCTL (FCB, 0, 0, 0, 0, 0, 0, 0, 0)
C IF N > 32 THEN DON'T NEED TO LOAD IFM
C
  IF(N .GT. 32)GO TO 140
  NN=N
  IF(N .EQ. 32)NN=16
  SLOPE=1./FLOAT(NN)
C
  DO 120 J=1,NN
    DO 110 I=1,512
      ISTART=(J-1)*512
      IDX=ISTART+I-1
      BUFFER(I)=IFIX(IDX*SLOPE)
110    CONTINUE
    CALL IFM(FCB,BUFFER,ISTART,512,0,0,0)
C
C      FCB
C      MAP      =      IFM CONTENTS
C      START    =      ZERO RELATIVE POSITION OF STARTING POINT
C      COUNT    =      THE NUMBER OF IFM ELEMENTS TO TRANSFER
C      PACK     =      1 IMPLIES PACKED MODE
C      VRTRTC   =
C      READ     =      0 IMPLIES WRITE, 1 IMPLIES READ
120  CONTINUE
C
C DEFINE AND LOAD LUT MAP FOR DIGITIZING CHANNEL FOR SLOPE OF 1
C
140  DO 22 I=1,256
      BUFFER(I)=I-1
22    CONTINUE
C
      CALL LUT(FCB,BUFFER,DIGMSK,7,0,0)
C
C CHECK IF WANT DIFFERENT SCALING
C
      IF( .NOT. MINMAX)GO TO 310
C
C ENABLE THE DIGITIZING LUT, GET MIN AND MAX, RELOAD LUT
C
      CALL LTCNT(FCB,DIGMSK,7,0,0)
      CALL MNMAX(FCB,MIN,MAX,I,I,I,I)
310  MINP1=MIN+1
C
      DO 305 I=1,MINP1
        BUFFER(I)=0
305  CONTINUE
C
      SLOPE=255./(MAX-MIN)
      MINP2=MINP1+1
C
      DO 320 I=MINP2,MAX
        BUFFER(I)=IFIX((I-MINP1)*SLOPE+.5)
320  CONTINUE
C
      MAXP1=MAX+1
C
      DO 340 I=MAXP1,256
        BUFFER(I)=255
340  CONTINUE
C
      CALL LUT(FCB,BUFFER,7,DIGMSK,0,0)
C
C      FCB
C      MAP      =      MAP FUNCTION FOR LUT
C      COLOR    =      BIT MASK FOR WHICH LUTS TO WRITE
C      VRTRTC   =
C      READ     =      0 IMPLIES WRITE, 1 IMPLIES READ

```

C DAVG: PRIMITIVE FOR AVERAGING VIDEO FRA PAGE 0003

C LOAD THE OTHER LUTS

C DO 40 I=1,256
C BUFFER(I)=I-1
40 CONTINUE

C CALL LUT(FCB,BUFFER,7,7,0,0)

C FCB = MAP FUNCTION FOR LUT
C MAP = BIT MASK FOR WHICH LUT'S TO WRITE
C COLOR = CHANNEL CODE BIT MAP
C CHANL =
C VRTRTC =
C READ = 0 IMPLIES WRITE, 1 IMPLIES READ

C SET UP THE ALU TO ACCOMPLISH AN 'A + B'

C DO 70 I=1,8
C OUTSEL(I)=:12 /*ALU OUTPUT
70 CONTINUE

C CALL ALU(FCB,0,:11,:11,BUFFER,OUTSEL,1,0,0,0,I,I,0)

C MODE = 0 IMPLIES ARITHMETIC, 1 IMPLIES LOGICAL
C BFUNC = BLOTCH FUNCTION (:11 IMPLIES A + B)
C NFUNC = NORMAL FUNCTION (:11 IMPLIES A + B)
C CONST = 8 WORD INTEGER CONSTANT ARRAY
C OUTSEL = SELECTION (:12 IMPLIES ALU OUTPUT)
C EFOM = 1 IMPLIES EXTEND SIGN BIT
C ESHIFT = RIGHT SHIFT OUTPUT WITH SIGN EXTENSION
C CARYIN = OVERFLOW FLAG SET ONLY DURING READ
C CARRY = OVERFLOW FLAG SET ONLY DURING READ
C EQUAL = FRAME EQUALITY FLAG SET DURING READ
C READ = 0 IMPLIES WRITE, 1 IMPLIES READ

C CLEAR THE ACCUMULATOR

C CALL FDBCK(FCB,2,3,-1,1,0,0,1,1,0)

C FCB = BIT MASK SELECTING COLOR
C COLOR = BIT MAP SELECTING CHANNEL
C CHANL = BIT MAP SELECTING BIT PLANES
C BITF = 0 IMPLIES USE IFM, 1 IMPLIES BYPASS IFM
C BYPIFM = PIXEL OFFSET
C PIXOFF = 1 IMPLIES EXTERNAL INPUT
C EXTERN = 0 IS NORMAL, 1 IMPLIES FEEDBACK ALL 0'S
C ZERO = 1 IMPLIES ACCUMULATOR MODE
C ACCUM = 1 IMPLIES ADDITIVE WRITE
C ADWRT =

C ENABLE THE LUTS FOR THE DIGITIZER TO GET A BLACK & WHITE PICTURE

C CALL LTCNT(FCB,DIGMSK,7,0,0)

C FCB = BIT MASK FOR WHICH LUTS TO ENABLE
C MASK = BIT MASK FOR WHICH COLORS TO ENABLE
C COLOR =
C VRTRTC =
C READ = 0 IMPLIES WRITE, 1 IMPLIES READ

C RELOAD OFMS WITH POSITIVE UNITY TRANSFORM

C DO 198 I=1,512
C BUFFER(I)=I-1
198 CONTINUE

C CALL OFM(FCB,BUFFER,7,0,0)

C FEEDBACK N TIMES

C DO 80 I=1,N
C CALL FDBCK(FCB,2,3,-1,1,0,0,0,1,0)

C DAVG: PRIMITIVE FOR AVERAGING VIDEO FRA PAGE 0004

```

C
C
C      FCB
C      COLOR      =      4,2,1 FOR RED, GREEN, BLUE (ORDER?)
C      CHANL      =      BIT MASK SELECTING DESTINATION CHANNEL
C      BITP      =      BIT MASK SELECTING BIT PLANES
C      BYPIFM     =      0 IMPLIES USE IFM, 1 IMPLIES BYPASS IFM
C      PIXOFF     =
C      EXTERN
C      ZERO
C      ACCUM
C      ADWRT
C
C      80      CONTINUE
C      SET UP ALU OST FOR ACCUMULATOR
C
C      DO 50 I=1,8
C          OUTSEL(I)=:10
C      50      CONTINUE
C      CALL ALU(FCB,0,:14,:14,BUFFER,OUTSEL,0,0,0,0,0,0,0)
C
C      FCB
C      MODE      =      0 IS ARITHMETIC, 1 IMPLIES LOGICAL
C      BFUNC     =      SET FOR A+A (USED WHEN N>32)
C      NFUNC     =      SET FOR A+A (USED WHEN N>32)
C      CONST     =
C      OUTSEL=    OUTPUT SELECTION ARRAY
C      EOFM      =
C      ESHIFT    =
C      SHIFT     =
C      CARYIN    =
C      CARRY     =
C      EQUAL     =
C      READ      =
C
C      IF(N .LE. 32)GO TO 600
C      OTHERWISE SHIFT DATA UP TO THE MSB
C      SET ALU FOR ALU OUTPUT
C
C      DO 88 I=1,8
C          OUTSEL(I)=:12
C      88      CONTINUE
C      CALL ALU(FCB,0,:14,:14,BUFFER,OUTSEL,0,0,0,0,0,0,0)
C
C      FCB
C      MODE      =      0 IMPLIES ARITHMETIC
C      BFUNC     =      ALU FUNCTION IN ROI (:14 IMPLIES A+A)
C      NFUNC     =      ALU FUNCTION OUTSIDE ROI (:14 IMPLIES A+A)
C      CONST     =      8 WORD CONSTANT ARRAY
C      OUTSEL    =      OUTPUT SELECTION ARRAY
C      EOFM      =      SIGN EXTEND OFM
C      ESHIFT    =      SIGN EXTEND RIGHT SHIFT
C      SHIFT     =      RIGHT SHIFT
C      CARYIN    =      SETS THE CARRY IN CONDITION
C      CARRY     =
C      EQUAL     =
C      READ      =      0 IMPLIES WRITE, 1 IMPLIES READ
C
C      IF(N .EQ. 64)GO TO 500
C      IF(N .EQ. 128)GO TO 510
C      IF(N .EQ. 256)GO TO 520
C      GO TO 600
C
C      MULTIPLY BY TWO BY FEEDING BACK AND DOING A+A
C      500      CALL FDBCK(FCB,0,3,-1,1,0,0,0,1,0)
C      510      CONTINUE

```

C DAUG: PRIMITIVE FOR AVERAGING VIDEO FRA PAGE 0005

```
C
C MULTIPLY BY TWO BY FEEDING BACK AND DOING A+A
C   CALL FDBCK(FCB,0,3,-1,1,0,0,0,1,0)
C
C SET ALU OST FOR OFM AND THEN FEED BACK THE 2ND CHANNEL TO THE 1ST
C
520 DO 99 I=1,8
    OUTSEL(I)=:11
99  CONTINUE
    CALL ALU(FCB,0,:14,:14,BUFFER,OUTSEL,0,0,0,0,0,0)
    CALL LTCNT(FCB,2,7,0,0)
    CALL FDBCK(FCB,2,1,-1,1,0,0,0,0,0)
    GO TO 444
C
C FEEDBACK 1 TIME THRU THE IFM TO ACCOMPLISH A DIVIDE BY N
C AND PUT IN 1ST MEMORY CHANNEL
C
600 CALL FDBCK(FCB,2,1,-1,0,0,0,0,0,0)
C
C ENABLE THE LUTS FOR THE CHANNEL THAT THE OUTPUT WAS FED BACK TO
C
444 CALL LTCNT(FCB,1,7,0,0)
C
C SCALE UP BY RELOADING OFMS WITH SLOPE OF 4
C
    DO 294 I=1,512
        BUFFER(I)=I+I+I+I-4
        BUFFER(512+I)=512+I+I+I+I-4
294 CONTINUE
C
    CALL OFM(FCB,BUFFER,7,0,0)
C
C TRANSFER COMMAND BUFFER TO MODEL 70
C
    CALL DEXEC (FCB)
C
    RETURN
    END
```

C DAVGD: INTERFACE FOR VIDEO AVERAGEING. PAGE 0001

C DAVGD: INTERFACE FOR VIDEO AVERAGEING.

```
*****
*                                     *
*      REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM      *
*      VERSION 1.0      JUNE 1, 1980                     *
*      BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY      *
*                                     *
*****
```

MODIFIED TO RUN FROM AUTOMATIC INSPECTION SYSTEM

SUBROUTINE DAVGD (FCB, ZERO, NIDS, NODS, BUFFER)

DESCRIPT - DIGITIZES A VIDEO IMAGE AND SUMS FRAMES IN THE ACCUMULATOR
TO DISPLAY A FRAME AVERAGE IN SEMI-REAL TIME.
SETS UP PARAMETER SYSTEM FOR CALL TO DAVG.
WRITTEN BY DAN KINNEY, BOEING QAT, 3/17/80

INTEGER ZERO(1), FCB(1), BUFFER(1)
INTEGER NIDS, NODS

INTEGER OCHNNO(16)
INTEGER ODSRN, DTYPE, FTYPE, NS, NL, NBANDS
INTEGER N, WAIT, MIN, MAX
LOGICAL MINMAX, BATCH
INTEGER INAMES(48), CODES(6), SIZES(6), COUNTS(6), NPARMS
INTEGER*4 ADDR1, ADDR2, ADDR3(6)

BUILD PARAMETER PROMPTS

SET N = 16 DEFAULT
CALL ZMVD ('N', INAMES(1), 16)
CODES(1) = 0
SIZES(1) = 1
ADDRS(1) = ADDR1 (N)
COUNTS(1) = 0
N = 16

SET DEFAULT WAIT TO 10
CALL ZMVD ('WAIT', INAMES(9), 16)
CODES(2) = 0
SIZES(2) = 1
ADDRS(2) = ADDR1 (WAIT)
COUNTS(2) = 0

WAIT = 10

SET MIN = 0, DEFAULT
CALL ZMVD ('MIN', INAMES(17), 16)
CODES(3) = 0
SIZES(3) = 1
ADDRS(3) = ADDR1 (MIN)
COUNTS(3) = 0
MIN = 0

SET MAX = 255 (DEFAULT)
CALL ZMVD ('MAX', INAMES(25), 16)
CODES(4) = 0
SIZES(4) = 1
ADDRS(4) = ADDR1 (MAX)
COUNTS(4) = 0
MAX = 255

CALL ZMVD ('MINMAX', INAMES(33), 16)
CODES(5) = 3
COUNTS(5) = 0

CALL ZMVD ('BATCH', INAMES(41), 16)

```

      CODES(6) = 3
      COUNTS(6) = 0

      NPARMS = 6
C
      CALL PARAMS (FCB, INAMES, CODES, SIZES, ADDRS, COUNTS, NPARMS)
      IF (N .LT. 1) N = 2
      IF (N .GT. 256) N = 256
      N = 2**IFIX(ALOG(FLOAT(N))/ALOG(2.))
      IF (MIN .LT. 0) MIN = 0
      IF (MIN .GE. MAX) MIN = 0
      IF (MAX .LE. MIN) MAX = 255
      IF (MAX .GT. 255) MAX = 255
C
      IF (COUNTS(5).EQ.1) MINMAX = .TRUE.
      IF (COUNTS(6).EQ.1) BATCH = .TRUE.
C
      OPEN OUTPUT DISPLAY IMAGE
      ODSRN = 1
      DTYPE = 1
      FTYPE = 2
      NS = 512
      NL = 512
      NBANDS = 1
C
C
C DO ACTUAL DIGITIZING AND SUMMATION
      CALL DAUG (FCB, BUFFER, N, WAIT, MIN, MAX, MINMAX, BATCH)
      NLEVS = 256
      OCHNNO(1) = 0
      CALL SETUP (FCB, BUFFER, NBANDS, NLEVS, OCHNNO, 0, 1)
C
C CLEAN UP
      CALL ICLOS (FCB, ODSRN)
      RETURN
      END

```

C SAVE: PRIMITIVE TO SAVE OUTPUT OF PIPE PAGE 0001

C SAVE: PRIMITIVE TO SAVE OUTPUT OF PIPELINE

* REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM *
* VERSION 1.0 JUNE 1,1980 *
* BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY *
* *****

SUBROUTINE SAVE (FCB, BUFFER, ICHNNO, OCHNNO, NBANDS,
BIAS, RSCALE)

WRITTEN BY DAN KINNEY, BOEING QAT R&D, 4/21/80.

DESCRIPT - PRIMITIVE TO SAVE THE PIPELINE OUTPUT OF A DISPLAY
IMAGE IN REFRESH MEMORY

PASSED IN - BUFFER INTEGER WORK SPACE (1024+)
ICHNNO ARRAY OF INPUT CHANNEL NUMBERS
OCHNNO ARRAY OF OUTPUT CHANNEL NUMBERS
NBANDS NUMBER OF BANDS IN INPUT IMAGE
BIAS CONSTANT VALUE TO BE ADDED TO IMAGE
RSCALE RESCALE FACTOR

RETURNED - NONE

SUBS - DADRS
LTCNT
CONST LOADS CONSTANT REGISTERS
ALU DEFINES THE ALU OPERATION
IFM
FDBCK WRITES THE FEEDBACK LOOP CONTROL WORD

CHANGES - COMPUTATION OF IFM RAMP CHANGED FROM 'I*RESCALE' TO
'(I-1)*RESCALE' JON 6/9/80.

INTEGER FCB(1), BUFFER(1)
INTEGER ICHNNO(16), OCHNNO(16), NBANDS
INTEGER BIAS
REAL RSCALE
INTEGER CHMASK(16), CHCODE
INTEGER OUTSEL(8)
INTEGER BUTTON, X, Y

INITIALIZE M70
CALL RBUTN (FCB, BUTTON, X, Y)
CALL SCROL (FCB, 0, 0, 17, 0, 0)
CALL SHIFT (FCB, 0, 0, 0, 0, 0)
CALL CONST (FCB, 0, 0, 0, 0, 0)
CALL PROFL (FCB, 0, 0, 0, 0, 0)
CALL ZFIL (BUFFER, 512, 0)
CALL GRRAM (FCB, BUFFER, 0, 0)
CALL CRCTL (FCB, 0, 0, 0, 0, 0, 0, 0, 0, 0)

COMPUTE INPUT CHANNEL MASK FROM CHANNEL CODE ARRAY
CALL DADRS (CHMASK, ICHNNO, CHCODE, NBANDS)
WRITE LUT MASK TO SELECT INPUT IMAGE OR IMAGES
CALL LTCNT (FCB, CHCODE, 7, 0, 0)
/* MASK = CHCODE MASK FOR WHICH CHANNELS TO ENABLE
/* COLOR = 7 MASK FOR WHICH COLORS TO ENABLE

SET UP CONSTANT REGISTERS FOR BIASING IMAGE AFTER ADDER
CALL CONST (FCB, BIAS, BIAS, BIAS, 0, 0)

SET UP ALU
DO 50 I=1,8
OUTSEL(I) = :11
50 CONTINUE

DEFINE ALU OPERATION

C — SAVE: PRIMITIVE TO SAVE OUTPUT OF PIPE PAGE 0002

```

C      CALL ALU (FCB, 0, 0, 0, BUFFER, OUTSEL, 0, 0, 0, 0, 0, 0)
C      /* OUTSEL = :11      PIPELINE OUTPUT
C
C      DEFINE IFM RESCALE MAP
C      THE .25 = 1/(2**2) WHICH SCALES 10 BIT TV DOWN TO 8 BIT MEMORY
C      DO 60 I=1,1024
C      BUFFER(I) = (I-1) * RSCALE * .25
60    CONTINUE
C
C      CALL IFM (FCB, BUFFER, 0, 1024, 0, 0, 0)
C      /* MAP = BUFFER      RESCALE RAMP FUNCTION
C      /* START = 0          FIRST POSITION TO LOAD IN IFM
C      /* COUNT = 1024       NUMBER OF ELEMENTS TO TRANSFER
C      /* PACK = 0           1 IMPLIES PACKED MODE TRANSFER
C      /* VRTRC = 0          WAIT FOR VERTICAL RETRACE INTERVAL
C      /* READ = 0           0 IMPLIES READ, 1 IMPLIES WRITE
C
C      COMPUTE OUTPUT CHANNEL MASK FROM CHANNEL CODE ARRAY
C      CALL DAORS (CHMASK, OCHNMO, CHCODE, NBANDS)
C      WRITE THE FEEDBACK LOOP CONTROL WORD
C      CALL FDBCK (FCB, 1, CHMASK, -1, 0, 0, 0, 0, 0, 0)
C      /* COLOR = 1          4, 2, 1, FOR RED, GREEN, AND BLUE
C      /* CHANL = CHMASK     BIT MASK SELECTING DESTINATION CHANNEL
C      /* BITP = -1          BIT MASK SELECTING THE BIT PLANES
C      /* BYPIFM = 0         0 IMPLIES USE IFM, 1 IMPLIES BYPASS IFM
C
C      RETURN
C      END

```

C SAVED: SAVE THE OUTPUT OF THE PIPELINE PAGE 0001

C SAVED: SAVE THE OUTPUT OF THE PIPELINE

```
C *****
C *
C * REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM *
C * VERSION 1.0 JUNE 1,1980 *
C * BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY *
C * *****
C
C SUBROUTINE SAVED (FCB, ZERO, NIDS, NODS, IBUFF)
C
C WRITTEN BY DAN KINNEY, BOEING QAT R&D, 4/21/80.
C
C DESCRIPTION: SAVED calls the primitive SAVE to save the pipeline
C output of a display image in refresh memory.
C
C specify passed variables
C INTEGER ZERO(1), FCB(1), IBUFF(1)
C INTEGER NIDS, NODS
C
C specify input and output image variables
C INTEGER ICHNNO(16), OCHNNO(16)
C INTEGER IDSRN, ODSRN, DTYPE, FTYPE, NS, NL, NBANDS
C INTEGER IDX, NLEVS
C
C specify parameter system variables
C INTEGER INAMES(16), CODES(2), SIZES(2), COUNTS(2), NPARMS
C INTEGER*4 ADDR1, ADDR, ADDR2(2)
C INTEGER BIAS
C REAL RSCALE
C
C LOGICAL LOCKED(16)
C INTEGER ICBPT, CHAN1, ILOCK
C
C check value of NIDS
C IF (1.LE.NIDS .AND. NIDS.LE.3) GOTO 100
C CALL TNOUA ('ERROR: ',8)
C CALL TNOU ('NO INPUT, OR MORE THEN THREE INPUTS SPECIFIED',45)
C RETURN
C 100 CONTINUE
C check value of NODS
C IF (NODS.LE.1) GOTO 200
C CALL TNOUA ('ERROR: ',8)
C CALL TNOU ('MORE THEN ONE OUTPUT SPECIFIED',30)
C RETURN
C 200 CONTINUE
C build parameter system prompts
C CALL ZMVD ('BIAS ',INAMES(1), 16)
C CODES(1) = 0
C SIZES(1) = 1
C ADDR2(1) = ADDR1 (BIAS)
C COUNTS(1) = 0
C BIAS = 0
C
C CALL ZMVD ('RESCALE ',INAMES(9), 16)
C CODES(2) = 1
C SIZES(2) = 1
C ADDR2(2) = ADDR (RSCALE)
C COUNTS(2) = 0
C RSCALE = 1.0
C
C NPARMS = 2
C
C call parameter system to check/prompt options
C CALL PARAMS (FCB, INAMES, CODES, SIZES, ADDR2, COUNTS, NPARMS)
C
C open input image
C IOX = 1
C DO 400 IDSRN = 1,NIDS
C CALL IOPI (FCB, IBUFF, IDSRN, DTYPE, FTYPE, NS, NL, NBANDS)
```

C SAVED: SAVE THE OUTPUT OF THE PIPELINE PAGE 0002

```

      IF (FTYPE.EQ.2) GOTO 300
      CALL TNOUA ('ERROR: ',8)
      CALL TNOU ('INPUT MUST BE A DISPLAY IMAGE',29)
      GOTO 900
300  CONTINUE
C      prevent memory management from overwriting image
      ICBPT = IDSRN * 256
      CHAN1 = FCB(ICBPT+87)
      ILOCK = AND (ZERO(906+CHAN1), :17)
      LOCKED(IDSRN) = .FALSE.
      IF (ILOCK.EQ.2) LOCKED(IDSRN) = .TRUE.
      IF (.NOT.LOCKED(IDSRN)) CALL LOCK (FCB, IDSRN)
C
C      load array with channel number corresponding to each band
      CALL CHREQ (FCB, IDSRN, ICHNNO(IDX), NLEVS)
      IDX = IDX + NBANDS
400  CONTINUE
      NBANDS = IDX - 1
C
C      open output image
      IF (NODS.EQ.1) GOTO 500
      IF NO OUTPUT IS SPECIFIED, OVERWRITE THE IMAGE
      DO 410 I=1,16
      OCHNNO(I) = ICHNNO(I)
410  CONTINUE
      GOTO 700
C
500  CONTINUE
      ODSRN = NIDS + 1
      CALL IOPO (FCB, IBUFF, ODSRN, DTYPE, FTYPE, NS, NL, 1)
      IF (FTYPE.EQ.2) GOTO 600
      CALL TNOUA ('ERROR: ',8)
      CALL TNOU ('OUTPUT MUST BE A DISPLAY IMAGE',30)
      GOTO 800
600  CONTINUE
      CALL CHREQ (FCB, ODSRN, OCHNNO, NLEVS)
C
700  CONTINUE
C      Call subroutine or primitive to do the actual operation
      CALL SAVE (FCB, IBUFF, ICHNNO, OCHNNO, NBANDS, BIAS, RSCALE)
      CALL SETUP (FCB, IBUFF, NBANDS, NLEVS, OCHNNO, 0, 1)
      CALL CONST (FCB, BIAS, BIAS, BIAS, 0, 1)
C
800  CONTINUE
C      close all output images
      CALL ICLOS (FCB, ODSRN)
C
900  CONTINUE
C      close all input images
      DO 910 IDSRN = 1,NIDS
      IF (.NOT.LOCKED(IDSRN)) CALL UNLOCK (FCB, IDSRN)
      CALL ICLOS (FCB, IDSRN)
910  CONTINUE
C
C      exit
      CALL DEXEC (FCB)
      RETURN
      END
```

APPENDIX 7

SUPPORTING SOFTWARE DESCRIPTIONS

APPENDIX 7
SUPPORTING SOFTWARE DESCRIPTIONS

A BRIEF DESCRIPTION OF EACH OF THE ADDITIONAL SUBROUTINES USED IN THE COMPUTER-AIDED INSPECTION SOFTWARE IS LISTED BELOW FOR CLARITY. ALL ARGUMENTS ARE 16 BIT INTEGERS (INTEGER*2) UNLESS OTHERWISE NOTED.

ADD1\$ (CHAN,BUFFER,KEY,ARRAY,FLAGS,ALTRTN,INDEX,FNO,BUFLEN,KEYLEN)
ADD1\$ IS A MIDAS (MULTIPLE INDEX DATA ACCESS SYSTEM) SUBROUTINE USED TO ADD A RECORD AND PRIMARY KEY OR A SECONDARY KEY TO A DATA-BASE. THE ARGUMENTS ARE:
CHAN = SYSTEM CHANNEL NUMBER TO THE DATA FILE.
BUFFER = AN ARRAY IN MEMORY CONTAINING THE RECORD TO BE ADDED, OR THE PRIMARY KEYWORD IF A SECONDARY KEY IS ADDED.
KEY = KEYWORD TO BE ADDED TO THE INDEX.
ARRAY = A 14 WORD ARRAY CONTAINING INDEX AND RECORD POINTERS AND ERROR CODE.
FLAGS = A 16 BIT WORD USED TO PASS OPTION CONDITIONS TO MIDAS.
ALTRTN = RETURN TAKEN IF ANY ERROR OCCURES.
INDEX = INDEX LEVEL, 0 = PRIMARY, 1-19 FOR SECONDARIES.
FNO = FILE NUMBER, ALWAYS = 0
BUFLEN = LENGTH OF DATA BUFFER, 0 IF WHOLE RECORD.
KEYLEN = LENGTH OF KEYWORD, 0 IF WHOLE KEYWORD.

CLEAN
CLEAN IS A SUBROUTINE FROM THE STC SYSTEM 500 SOFTWARE USED TO INITIALIZE THE MODEL 70 HARDWARE AND CLEAN THE IMAGE DIRECTORY. THIS SUBROUTINE WAS MODIFIED TO BE COMPATIBLE WITH AUTOMATIC OPERATION.

CLOS\$A (ICH)
PRIME SUPPLIED SUBROUTINE USED TO CLOSE A DISK FILE. ICH IS CHANNEL TO BE CLOSED.

CMDS70 (CMD,ERR)
CMDS70 PASSES A COMMAND TO THE STC SYSTEM 500. THIS SUBROUTINE WAS WRITTEN TO BE COMPATIBLE WITH AUTOMATIC PROCESSES. THE ARGUMENTS ARE:
CMD = COMMAND LINE IN SYSTEM 500 FORMAT.
ERR = ERROR CODE, 0 = NO ERROR.

CMHDR
CMHDR IS A SYSTEM 500 SUBROUTINE USED TO LOAD DATA INTO COMMON. IT WAS MODIFIED TO BE COMPATIBLE WITH AUTOMATIC OPERATION.

CNVB\$A (KEY,VALUE,BUFFER,BUFLEN)
CNVB\$A IS A PRIME SUPPLIED SUBROUTINE TO CONVERT A DOUBLE PRECISION INTEGER TO ASCII. THE ARGUMENTS ARE:
KEY = KEY FOR CONVERSION TO OCTAL, DECIMAL, OR HEXIDEcimal.
VALUE = 4 BYTE INTEGER TO BE CONVERTED.
BUFFER = ARRAY TO RECEIVE ASCII CONVERSION.
BUFLEN = LENGTH OF ARRAY IN BYTES.

CSTR\$A (TEXTA,LENA,TEXTB,LENB)
CSTR\$A IS A LOGICAL FUNCTION SUPPLIED BY PRIME USED TO COMPARE TWO TEXT STRINGS. ITS VALUE IS TRUE IF THE STRINGS COMPARE. THE ARGUMENTS ARE:
TEXTA = FIRST STRING FOR REFERENCE.
LENA = LENGTH OF TEXTA IN BYTES.
TEXTB = SECOND STRING TO COMPARE.
LENB = LENGTH OF TEXTB.

DATE\$A (BUFFER)

DATE\$A IS A DOUBLE PRECISION REAL FUNCTION SUPPLIED BY PRIME, AND IS USED TO GET THE CURRENT DATE. THE VALUE OF THE FUNCTION IS ASCII IN THE FORM OF 'MM/DD/YY'. THE ARGUMENT IS:
BUFFER = DATE IN THE FORM 'DAY, MON DD YEAR', AND MUST BE AT LEAST 16 BYTES LONG.

DELE\$A (FILE,LEN)

DELE\$A IS A LOGICAL FUNCTION SUPPLIED BY PRIME, USED TO DELETE A FILE. THE FUNCTION IS RETURNED TRUE IF SUCCESSFUL. THE ARGUMENTS ARE:
FILE = NAME OF FILE TO BE DELETED.
LEN = LENGTH OF FILE NAME IN BYTES.

DELET\$ (CHAN,BUFFER,KEY,ARRAY,FLAGS,ALTRTN,INDEX,FNO,BUFLEN,KEYLEN)

DELET\$ IS A MIDAS (MULTIPLE INDEX DATA ACCESS SYSTEM) SUBROUTINE USED TO DELETE A RECORD AND PRIMARY KEY OR A SECONDARY KEY FROM A DATA-BASE. THE ARGUMENTS ARE:
CHAN = SYSTEM CHANNEL NUMBER TO THE DATA FILE.
BUFFER = AN ARRAY IN MEMORY CONTAINING THE RECORD TO BE DELETED.
KEY = KEYWORD TO BE DELETED FROM THE INDEX.
ARRAY = A 14 WORD ARRAY CONTAINING INDEX AND RECORD POINTERS AND ERROR CODE.
FLAGS = A 16 BIT WORD USED TO PASS OPTION CONDITIONS TO MIDAS.
ALTRTN = RETURN TAKEN IF ANY ERROR OCCURES.
INDEX = INDEX LEVEL, 0 = PRIMARY, 1-19 FOR SECONDARIES.
FNO = FILE NUMBER, ALWAYS = 0
BUFLEN = LENGTH OF DATA BUFFER, 0 IF WHOLE RECORD.
KEYLEN = LENGTH OF KEYWORD, MUST BE 0

EXIT

EXIT IS A PRIME SUBROUTINE USED TO RETURN TO THE OPERATING SYSTEM.

FILL\$A (BUFFER,BUFLEN,'CHAR')

FILL\$A IS A PRIME SUPPLIED SUBROUTINE TO FILL AN ARRAY WITH BYTES OR CHARACTERS. THE ARGUMENTS ARE:
BUFFER = ARRAY TO BE FILLED.
BUFLEN = NUMBER OF BYTES IN BUFFER.
'CHAR' = THE CHARACTER TO USED TO FILL THE ARRAY. THE CHARACTER MUST BE LEFT JUSTIFIED.

GCHR\$A (TEXT,POS)

GCHR\$A IS AN INTEGER FUNCTION SUPPLIED BY PRIME USED TO GET A CHARACTER FROM A TEXT STRING. THE VALUE OF THE FUNCTION IS A LEFT JUSTIFIED, BLANK PADDED 2 BYTE INTEGER. THE ARGUMENTS ARE:
TEXT = TEXT STRING SUPPLYING THE CHARACTER.
POS = POSITION IN TEXT OF CHARACTER WANTED.

LOCK\$ (CHAN,BUFFER,KEY,ARRAY,FLAGS,ALTRTN,INDEX,FNO,BUFLEN,KEYLEN)

LOCK\$ IS A MIDAS (MULTIPLE INDEX DATA ACCESS SYSTEM) SUBROUTINE USED TO LOCATE AND LOCK A RECORD FOR EDITING AND UPDATING. THE ARGUMENTS ARE:
CHAN = SYSTEM CHANNEL NUMBER TO THE DATA FILE.
BUFFER = AN ARRAY IN MEMORY CONTAINING THE RECORD TO BE LOCKED.
KEY = KEYWORD TO LOCATE THE RECORD.
ARRAY = A 14 WORD ARRAY CONTAINING INDEX AND RECORD POINTERS AND ERROR CODE.
FLAGS = A 16 BIT WORD USED TO PASS OPTION CONDITIONS TO MIDAS.
ALTRTN = RETURN TAKEN IF ANY ERROR OCCURES.
INDEX = INDEX LEVEL, 0 = PRIMARY, 1-19 FOR SECONDARIES.
FNO = FILE NUMBER, ALWAYS = 0
BUFLEN = LENGTH OF DATA BUFFER, MUST BE 0
KEYLEN = LENGTH OF KEYWORD, MUST BE 0

LSIZE (TEXT,LEN)
LSIZE IS AN INTEGER FUNCTION USED TO DETERMINE THE LAST NON BLANK CHARACTER IN A TEXT STRING. THE VALUE OF THE FUNCTION RETURNED IS THE BYTE COUNT OF THE LAST NON-BLANK CHARACTER. THE ARGUMENTS ARE:
TEXT = TEXT STRING TO BE EXAMINED.
LEN = LENGTH OF TEXT.

MCHR\$A (DTEXT,DPOS,STEXT,SPOS)
MCHR\$A IS A PRIME SUPPLIED SUBROUTINE USED TO MOVE ONE CHARACTER FROM ONE STRING TO ANOTHER. THE ARGUMENTS ARE:
DTEXT = DESTINATION TEXT STRING.
DPOS = POSITION IN DESTINATION STRING.
STEXT = SOURCE TEXT STRING.
SPOS = POSITION IN SOURCE STRING.

MSTR\$A (STEXT,LENS,DTEXT,LEND)
MSTR\$A IS A PRIME SUPPLIED SUBROUTINE USED TO MOVE A STRING FROM ONE ARRAY TO ANOTHER. THE ARGUMENTS ARE:
STEXT = SOURCE TEXT.
LENS = LENGTH OF SOURCE STRING.
DTEXT = DESTINATION TEXT.
LEND = LENGTH OF DESTINATION ARRAY (BLANK PADDING OR TRUNCATION IS PERFORMED)

MSUB\$A (STEXT,LENS,STARTS,ENDS,DTEXT,LEND,STARTD,ENDD)
MSUB\$A IS A PRIME SUPPLIED SUBROUTINE USED TO MOVE A TEXT STRING FROM ONE ARRAY TO ANOTHER. THE ARGUMENTS ARE:
STEXT = SOURCE TEXT.
LENS = LENGTH OF SOURCE TEXT.
STARTS = START POSITION IN SOURCE TEXT.
ENDS = END POSITION IN SOURCE TEXT.
DTEXT = DESTINATION TEXT.
LEND = LENGTH OF DESTINATION TEXT.
STARTD = POSITION IN DESTINATION TEXT TO DEPOSIT TEXT.
ENDD = END POSITION IN DESTINATION TEXT FOR DEPOSITED TEXT.

NEXT\$ (CHAN,BUFFER,KEY,ARRAY,FLAGS,ALTRN,INDEX,FNO,BUFLEN,KEYLEN)
NEXT\$ IS A MIDAS (MULTIPLE INDEX DATA ACCESS SYSTEM) SUBROUTINE USED TO LOCATE A SPECIFIC RECORD BY KEYWORD OR THE NEXT RECORD IN THE INDEX. THE ARGUMENTS ARE:
CHAN = SYSTEM CHANNEL NUMBER TO THE DATA FILE.
BUFFER = AN ARRAY IN MEMORY TO RECEIVE THE RECORD.
KEY = KEYWORD TO LOCATE THE RECORD.
ARRAY = A 14 WORD ARRAY CONTAINING INDEX AND RECORD POINTERS AND ERROR CODE.
FLAGS = A 16 BIT WORD USED TO PASS OPTION CONDITIONS TO MIDAS.
ALTRN = RETURN TAKEN IF ANY ERROR OCCURES.
INDEX = INDEX LEVEL, 0 = PRIMARY, 1-19 FOR SECONDARIES.
FNO = FILE NUMBER, ALWAYS = 0.
BUFLEN = LENGTH OF DATA BUFFER, 0 FOR FULL RECORD.
KEYLEN = LENGTH OF KEYWORD, 0 FOR FULL KEY, >0 FOR PARTIAL KEY.

PACK (TEXT,LEN)
PACK IS A SUBROUTINE USED TO REMOVE ILLEGAL FILE NAME CHARACTERS FROM A TEXT STRING. THE ARGUMENTS ARE:
TEXT = TEXT STRING.
LEN = LENGTH OF STRING.

PAUS (IGO)
PAUS IS A SUBROUTINE USED TO HOLD THE PROGRAM FLOW UNTIL THE OPERATOR PASSES A TERMINAL KEY. THE ARGUMENT IS:
IGO = RETURNED 0 IF 'Q' OR 'ESCAPE' KEY IS PRESSED, OR 1 FOR ANY OTHER KEY.

PEDIT (CHAN,BUFFER,START,END,ERASE)
PEDIT IS A SUBROUTINE USED TO PERFORM THE SCREEN EDITING FUNCTION.
THE ARGUMENTS ARE:
CHAN = CHANNEL TO A FORMATTED SCREEN TEMPLATE FILE.
BUFFER = ARRAY HOLDING ALL THE DATA INPUT OR OUTPUT VIA THE
TEMPLATE.
START = FIRST LINE NUMBER IN TEMPLATE FILE TO BE USED BY PEDIT.
END = LAST LINE NUMBER IN TEMPLATE FILE TO BE USED BY PEDIT.
ERASE = FLAG TO ERASE SCREEN BEFORE TEMPLATE DISPLAY.

READL (BUFFER,NCHAR,LENB)
READL IS A SUBROUTINE USED TO GET A LINE OF TEXT FROM THE
OPERATOR. THE ARGUMENTS ARE:
BUFFER = ARRAY TO ACCEPT TEXT.
NCHAR = NUMBER OF CHARACTERS ENTERED BY OPERATOR.
LENB = LENGTH OF BUFFER.

READN (TEXT,LENG,NUM)
READN IS A SUBROUTINE USED TO DISPLAY A LINE OF TEXT AND ACCEPT
A DECIMAL INTEGER (IN ASCII) FROM THE TERMINAL OPERATOR.
THE ARGUMENTS ARE:
TEXT = A LINE OF TEXT TO BE DISPLAYED AS A PROMPT.
LENG = THE LENGTH OF THE TEXT IN BYTES.
NUM = RETURNED INTEGER ENTERED AT TERMINAL.

RECYCL
RECYCL IS A PRIME SUPPLIED SUBROUTINE USED TO ALLOW OTHER USERS
OF THE SYSTEM TO USE YOUR TIME SLICE.

RPTGEN (SPCHN,CHAN,BUFFER,LINES,START,END,FORM)
RPTGEN IS A SUBROUTINE FROM THE SCREEN EDITING SYSTEM USED TO
PRODUCE REPORTS. THE ARGUMENTS ARE:
SPCHN = CHANNEL TO A LINE PRINTER SPOOL FILE, IF 0 THEN TO
THE OPERATOR'S TERMINAL.
CHAN = CHANNEL TO THE FORMATTED SCREEN TEMPLATE FILE.
BUFFER = ARRAY CONTAINING ALL THE DATA FOR THE REPORT.
LINES = RETURNED NUMBER OF LINES SENT TO SPOOL FILE.
START = FIRST LINE IN TEMPLATE FILE USED IN REPORT.
END = LAST LINE IN TEMPLATE FILE USED IN REPORT.
FORM = FLAG FOR FORMFEED AT START OF REPORT.

RS (ARG,SHIFT)
RS IS A PRIME SUPPLIED FUNCTION USED TO SHIFT A SINGLE PRECISION
INTEGER (16 BITS) TO THE RIGHT (TOWARD LOWER SIGNIFICANCE).
THE ARGUMENTS ARE:
ARG = INTEGER TO BE SHIFTED.
SHIFT = NUMBER OF BITS TO SHIFT.

SETERM (TYPE)
SETERM IS A SUBROUTINE USED TO GET THE OPERATOR'S TERMINAL TYPE
AND LOAD A COMMON ARRAY WITH CONTROL COMMANDS. THE ARGUMENT IS:
TYPE = RETURNED TYPE CODE FOR TERMINAL.

SPOOL\$ (KEY,FILE,FLEN,INFO,BUFFER,BLEN,ERROR)
SPOOL\$ IS A SUBROUTINE SUPPLIED BY PRIME TO PASS A FILE TO THE
LINE PRINTER SPOOLER. THE ARGUMENTS ARE:
KEY = FLAG FOR SPOOLER MODE, 1 = PRINT.
FILE = FILE NAME TO BE PRINTED.
FLEN = LENGTH OF FILE NAME.
INFO = 12 WORD ARRAY USED TO PASS INFORMATION TO SPOOLER.
BUFFER = WORKING ARRAY FOR SPOOLER.
BLEN = LENGTH OF BUFFER IN WORDS.
ERROR = RETURNED ERROR CODE.

T\$MT (UNIT,ADDR,COUNT,COMND,STATUS)
T\$MT IS A PRIME SUPPLIED SUBROUTINE USED TO CONTROL THE MAGNETIC TAPE. THE ARGUMENTS ARE:
UNIT = MAGNETIC TAPE DRIVE NUMBER, 0 RELATIVE.
ADDR = MEMORY ADDRESS OF BUFFER FOR RECORD I/O, FOUR BYTES IF IN VIRTUAL MODE.
COUNT = NUMBER OF WORDS TO TRANSFER, BETWEEN 0 AND 6000.
COMND = COMMAND FLAG FOR ACTION REQUESTED.
ERROR = RETURNED ERROR CODE.

T10U (CHAR)
T10U IS A PRIME SUPPLIED SUBROUTINE USED TO OUTPUT ONE CHARACTER TO THE OPERATOR'S TERMINAL. THE ARGUMENT IS:
CHAR = THE CHARACTER TO BE OUTPUT, LEFT JUSTIFIED.

TIME\$A (BUFFER)
TIME\$A IS A DOUBLE PRECISION REAL FUNCTION SUPPLIED BY PRIME. IT IS USED TO GET THE CURRENT TIME OF DAY. THE VALUE OF THE FUNCTION RETURNED IS EQUAL TO HOURS SINCE MIDNIGHT. THE ARGUMENT IS:
BUFFER = RETURNED ASCII VALUE OF TIME IN FORM 'HR:MN:SC', AND MUST BE AT LEAST 8 BYTES LONG.

TNOU (TEXT,NCHAR)
TNOU IS A PRIME SUPPLIED SUBROUTINE USED TO OUTPUT A LINE OF TEXT TO THE OPERATOR'S TERMINAL, FOLLOWED BY A CARRIAGE RETURN AND LINE FEED. THE ARGUMENTS ARE:
TEXT = TEXT OF OUTPUT MESSAGE.
NCHAR = NUMBER OF CHARACTERS IN TEXT.

TNOUA (TEXT,NCHAR)
TNOUA IS A PRIME SUPPLIED SUBROUTINE USED TO OUTPUT A LINE OF TEXT TO THE OPERATOR'S TERMINAL WITHOUT CARRIAGE RETURN AND LINE FEED. THE ARGUMENTS ARE:
TEXT = TEXT OF OUTPUT MESSAGE.
NCHAR = NUMBER OF CHARACTERS IN TEXT.

TONL
TONL IS A PRIME SUPPLIED SUBROUTINE TO OUTPUT A CARRIAGE RETURN AND LINE FEED TO THE OPERATOR'S TERMINAL.

TOOCT (NUM)
TOOCT IS A PRIME SUPPLIED SUBROUTINE USED TO CONVERT A NUMBER TO OCTAL AND DISPLAY AT THE OPERATOR'S TERMINAL.

UNIT\$A (CHAN)
UNIT\$A IS A LOGICAL FUNCTION SUPPLIED BY PRIME, USED TO CHECK FOR THE USE OF A CHANNEL NUMBER. THE FUNCTION IS RETURNED TRUE IF THE CHANNEL IS IN USE. THE ARGUMENT IS:
CHAN = CHANNEL NUMBER TO BE CHECKED.

UPDAT\$ (CHAN,BUFFER,KEY,ARRAY,FLAGS,ALTRTN,INDEX,FNO,BUFLEN,KEYLEN)
UPDAT\$ IS A MIDAS (MULTIPLE INDEX DATA ACCESS SYSTEM) SUBROUTINE USED TO REWRITE A RECORD PREVIOUSLY FOUND BY LOCK\$.
 THE ARGUMENTS ARE:
CHAN = SYSTEM CHANNEL NUMBER TO THE DATA FILE.
BUFFER = AN ARRAY IN MEMORY CONTAINING THE RECORD TO BE UPDATED.
KEY = KEYWORD
ARRAY = A 14 WORD ARRAY CONTAINING INDEX AND RECORD POINTERS AND ERROR CODE.
FLAGS = A 16 BIT WORD USED TO PASS OPTION CONDITIONS TO MIDAS.
ALTRTN = RETURN TAKEN IF ANY ERROR OCCURES.
INDEX = INDEX LEVEL, 0 = PRIMARY, 1-19 FOR SECONDARIES.
FNO = FILE NUMBER, ALWAYS = 0
BUFLEN = LENGTH OF DATA BUFFER, MUST BE 0
KEYLEN = LENGTH OF KEYWORD, MUST BE 0

VOPEN\$ (FILE,FLEN,MODE,CHAN,ERROR)

VOPEN\$ IS A SUBROUTINE USED TO OPEN A FILE ON AN AVAILABLE CHANNEL (SUPPLIED BY THE SYSTEM). THE ARGUMENTS ARE:

FILE = FILE NAME TO BE OPENED.
FLEN = LENGTH OF FILE NAME.
MODE = READ/WRITE FLAG.
CHAN = RETURNED CHANNEL FOR OPENED FILE.
ERROR = RETURNED ERROR CODE.

WTLIN\$ (CHAN,BUFFER,COUNT,ERROR)

WTLIN\$ IS A PRIME SUPPLIED SUBROUTINE USED TO WRITE AN ASCII STRING TO A FILE. THE ARGUMENTS ARE:

CHAN = FILE CHANNEL NUMBER.
BUFFER = ARRAY CONTAINING THE STRING TO BE WRITTEN.
COUNT = NUMBER OF 16 BIT WORDS IN BUFFER.
ERROR = RETURNED ERROR CODE.

YSNO\$A (TEXT,LEN,KEY)

YSNO\$A IS A LOGICAL FUNCTION SUPPLIED BY PRIME USED TO DISPLAY A MESSAGE AND RECEIVE A YES OR NO FROM THE OPERATOR. THE FUNCTION IS RETURNED TRUE FOR A YES RESPONSE. THE ARGUMENTS ARE:

TEXT = MESSAGE USED AS A PROMPT, A '?' IS APPENDED.
LEN = LENGTH OF MESSAGE IN BYTES.
KEY = DEFAULT KEY

ZFIL (BUFFER,LENB,CHAR)

ZFIL IS A PRIME SUPPLIED SUBROUTINE USED TO FILL AN ARRAY WITH A SPECIFIC CHARACTER. ZFIL RUNS IN VIRTUAL MODE ONLY.

THE ARGUMENTS ARE:
BUFFER = ARRAY TO BE FILLED.
LENB = LENGTH OF BUFFER IN BYTES.
CHAR = CHARACTER FOR FILLING, LEFT JUSTIFIED.

ZMV (TEXTS,LENS,TEXTD,LEND)

ZMV IS A PRIME SUPPLIED SUBROUTINE USED TO MOVE A TEXT STRING FROM ONE ARRAY TO ANOTHER, TRUNCATING OR BLANK PADDING. ZMV RUNS IN VIRTUAL MODE ONLY. THE ARGUMENTS ARE:

TEXTS = SOURCE TEXT.
LENS = LENGTH OF SOURCE STRING.
TEXTD = DESTINATION TEXT.
LEND = LENGTH OF DESTINATION STRING.

ZMVD (TEXTS,TEXTD,LEN)

ZMVD IS A PRIME SUPPLIED SUBROUTINE USED TO MOVE TEXT FROM ONE ARRAY TO ANOTHER. THE ARRAYS ARE ASSUMED TO BE OF EQUAL SIZE. ZMVD RUNS IN VIRTUAL MODE ONLY. THE ARGUMENTS ARE:

TEXTS = SOURCE TEXT STRING.
TEXTD = DESTINATION ARRAY.
LEN = LENGTH OF STRING IN BYTES.

END

DATE
FILMED

6 - 8 - 1

DTIC