TECHNICAL REPORT TR-RS-CR-81-1

LOW COST HIGH VOLUME RADIOPHGRAPHIC 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
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 ENDORSEMENT OR APPROVAL OF THE USE OF SUCH COMMERCIAL HARDWARE OR SOFTWARE.
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

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOREWORD</td>
<td>iv</td>
</tr>
<tr>
<td>SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>1.0 INTRODUCTION</td>
<td>3</td>
</tr>
<tr>
<td>2.0 SCOPE OF WORK</td>
<td>5</td>
</tr>
<tr>
<td>3.0 EQUIPMENT DESCRIPTION</td>
<td>7</td>
</tr>
<tr>
<td>3.1 MECHANICAL EQUIPMENT</td>
<td>7</td>
</tr>
<tr>
<td>3.2 REAL TIME RADIOGRAPHIC EQUIPMENT</td>
<td>11</td>
</tr>
<tr>
<td>3.3 IMAGE PROCESSOR</td>
<td>11</td>
</tr>
<tr>
<td>3.4 COMPUTER SYSTEM</td>
<td>12</td>
</tr>
<tr>
<td>3.5 EQUIPMENT INTERFACING</td>
<td>16</td>
</tr>
<tr>
<td>4.0 RADIOGRAPHY</td>
<td>17</td>
</tr>
<tr>
<td>4.1 REAL TIME RADIOGRAPHY</td>
<td>17</td>
</tr>
<tr>
<td>4.1.1 THE ADVANTAGE OF MOTION</td>
<td>17</td>
</tr>
<tr>
<td>4.1.2 METHODS OF REDUCING SCATTERED RADIATION</td>
<td>18</td>
</tr>
<tr>
<td>4.1.3 VIEWING OF THE X-RAY IMAGE</td>
<td>18</td>
</tr>
<tr>
<td>4.2 IMAGE PROCESSING</td>
<td>20</td>
</tr>
<tr>
<td>4.2.1 NOISE REDUCTION</td>
<td>20</td>
</tr>
<tr>
<td>4.2.2 EDGE ENHANCEMENT</td>
<td>21</td>
</tr>
<tr>
<td>4.2.3 CONTRAST MANIPULATION</td>
<td>26</td>
</tr>
<tr>
<td>4.2.4 COLOR REPRESENTATION</td>
<td>26</td>
</tr>
<tr>
<td>4.2.5 SOFTWARE CONSIDERATIONS</td>
<td>27</td>
</tr>
<tr>
<td>4.3 RADIOGRAPHIC QUALITY</td>
<td>28</td>
</tr>
<tr>
<td>4.3.1 PARAMETERS AFFECTING RESOLUTION</td>
<td>29</td>
</tr>
<tr>
<td>4.3.2 MEASUREMENT OF RESOLUTION</td>
<td>31</td>
</tr>
<tr>
<td>4.4 RADIOGRAPHIC RECORDS</td>
<td>32</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS (Continued)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.0 COMPUTER AIDED INSPECTION</td>
<td>34</td>
</tr>
<tr>
<td>5.1 CONCEPT</td>
<td>34</td>
</tr>
<tr>
<td>5.2 SOFTWARE DESIGN</td>
<td>34</td>
</tr>
<tr>
<td>5.2.1 INSPECTION</td>
<td>37</td>
</tr>
<tr>
<td>5.2.2 PLANNING</td>
<td>37</td>
</tr>
<tr>
<td>5.2.3 RETRIEVAL</td>
<td>40</td>
</tr>
<tr>
<td>5.3 APPLICATION TO ROLAND FINAL INSPECTION</td>
<td>40</td>
</tr>
<tr>
<td>6.0 COST-BENEFIT</td>
<td>42</td>
</tr>
<tr>
<td>7.0 CONCLUSIONS AND RECOMMENDATIONS</td>
<td>46</td>
</tr>
<tr>
<td>7.1 REAL TIME X-RAY</td>
<td>46</td>
</tr>
<tr>
<td>7.1.1 EQUIPMENT</td>
<td>46</td>
</tr>
<tr>
<td>7.1.2 APPLICATIONS</td>
<td>46</td>
</tr>
<tr>
<td>7.2 IMAGE PROCESSING</td>
<td>46</td>
</tr>
<tr>
<td>7.3 COMPUTER-AIDED INSPECTION</td>
<td>47</td>
</tr>
<tr>
<td>7.4 ACCOMPLISHMENTS AND RECOMMENDATIONS FOR FUTURE WORK</td>
<td>47</td>
</tr>
<tr>
<td>8.0 BIBLIOGRAPHY</td>
<td>48</td>
</tr>
</tbody>
</table>

## APPENDICES

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. STC M-70 I/O SCHEMATICS</td>
<td>A1-1</td>
</tr>
<tr>
<td>2. IMAGE PROCESSOR SOFTWARE, LOGIC AND CODE</td>
<td>A2-1</td>
</tr>
<tr>
<td>3. MOTOR CONTROL SCHEMATICS</td>
<td>A3-1</td>
</tr>
<tr>
<td>4. MOTOR CONTROL SOFTWARE, LOGIC AND CODE</td>
<td>A4-1</td>
</tr>
<tr>
<td>5. COMPUTER AIDED INSPECTION SOFTWARE, LOGIC AND CODE</td>
<td>A5-1</td>
</tr>
<tr>
<td>6. STC SYSTEM 500 SOFTWARE ADDITIONS, LOGIC AND CODE</td>
<td>A6-1</td>
</tr>
<tr>
<td>7. SUPPORTING SOFTWARE DESCRIPTION</td>
<td>A7-1</td>
</tr>
<tr>
<td>Figure</td>
<td>Title</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>1.0-1</td>
<td>Real Time Radiography - Concept</td>
</tr>
<tr>
<td>2.0-1</td>
<td>Major Program Milestones</td>
</tr>
<tr>
<td>3.0-1</td>
<td>High Volume Radiographic Inspection System</td>
</tr>
<tr>
<td>3.1-1</td>
<td>Part Manipulation Table</td>
</tr>
<tr>
<td>3.1-2</td>
<td>Lead Shielding</td>
</tr>
<tr>
<td>3.1-3</td>
<td>X-Ray Source and Image Intensifier Configuration</td>
</tr>
<tr>
<td>3.3-1</td>
<td>Image Processing System Comparisons</td>
</tr>
<tr>
<td>3.4-1</td>
<td>Computer Comparisons</td>
</tr>
<tr>
<td>4.2.2-1</td>
<td>Difference Edge Enhancement</td>
</tr>
<tr>
<td>4.2.2-2</td>
<td>Gradient Edge Enhancement</td>
</tr>
<tr>
<td>4.2.2-3</td>
<td>Laplacian Edge Enhancement</td>
</tr>
<tr>
<td>4.2.3-1</td>
<td>Linear Mapping Image Enhancement</td>
</tr>
<tr>
<td>4.2.3-2</td>
<td>Piecewise Linear Mapping Image Enhancement</td>
</tr>
<tr>
<td>4.2.3-3</td>
<td>Color Level Slicing</td>
</tr>
<tr>
<td>4.2.4-1</td>
<td>Pseudocolor Image Enhancement</td>
</tr>
<tr>
<td>4.3.1-1</td>
<td>Measured Focal Spot Size for Phillips Electronic Instruments, Inc. Model MCN 321</td>
</tr>
<tr>
<td>4.3.1-2</td>
<td>Resolution Limitations of Prototype System Due to X-Ray Focal Spot, Image Intensifier, and Video Display</td>
</tr>
<tr>
<td>4.3.2-1</td>
<td>Penetrameter Sensitivity</td>
</tr>
<tr>
<td>5.2-1</td>
<td>Computer-Aided Inspection Data Base</td>
</tr>
<tr>
<td>5.2-2</td>
<td>Computer-Aided Inspection System</td>
</tr>
<tr>
<td>5.2.1-1</td>
<td>Inspection Subsystem</td>
</tr>
<tr>
<td>5.2.2-1</td>
<td>Planning Screen Formats</td>
</tr>
<tr>
<td>5.3-1</td>
<td>Roland Image Processing Sequence</td>
</tr>
<tr>
<td>6.0-1</td>
<td>Real-Time X-Ray System Capital Costs</td>
</tr>
<tr>
<td>6.0-2</td>
<td>Investment Analysis - Real Time X-Ray</td>
</tr>
<tr>
<td>6.0-3</td>
<td>Non-film Radiography with Computer-Aided Inspection</td>
</tr>
</tbody>
</table>
FOREWORD

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 invaluably to the success of the project.
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.
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.
5. Image Processor, Model 70/E, Stanford Technology Corp.
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.

![Diagram of X-ray source, part manipulation table, and image intensifier configuration](image.png)
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 an 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, pseudo 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.
<table>
<thead>
<tr>
<th>VIDEO SYSTEMS</th>
<th>INPUT</th>
<th>HEIGHT</th>
<th>PROCESSING (MEASUREMENT)</th>
<th>PROCESSING (OUTPUT)</th>
<th>CPU INTERFACE</th>
<th>MEMORY</th>
<th>I/O CAPABILITY</th>
<th>INTERFACE</th>
<th>DISPLAY</th>
<th>TRANSFER</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUARTER CORP DAY-30</td>
<td>512 LINES</td>
<td>512</td>
<td>1024</td>
<td>16 BIT</td>
<td>16 BIT</td>
<td>YES</td>
<td>BIT-BIT</td>
<td>YES</td>
<td>16 BIT</td>
<td>YES</td>
</tr>
<tr>
<td>RANTES CORP</td>
<td>512 LINES</td>
<td>512</td>
<td>1024</td>
<td>16 BIT</td>
<td>16 BIT</td>
<td>YES</td>
<td>BIT-BIT</td>
<td>YES</td>
<td>16 BIT</td>
<td>YES</td>
</tr>
<tr>
<td>CRANFIELD SYSTEMS</td>
<td>512 LINES</td>
<td>512</td>
<td>1024</td>
<td>16 BIT</td>
<td>16 BIT</td>
<td>YES</td>
<td>BIT-BIT</td>
<td>YES</td>
<td>16 BIT</td>
<td>YES</td>
</tr>
<tr>
<td>CERNUS CORP</td>
<td>512 LINES</td>
<td>512</td>
<td>1024</td>
<td>16 BIT</td>
<td>16 BIT</td>
<td>YES</td>
<td>BIT-BIT</td>
<td>YES</td>
<td>16 BIT</td>
<td>YES</td>
</tr>
<tr>
<td>CENTRAL CORP VISION GNE</td>
<td>512 LINES</td>
<td>512</td>
<td>1024</td>
<td>16 BIT</td>
<td>16 BIT</td>
<td>YES</td>
<td>BIT-BIT</td>
<td>YES</td>
<td>16 BIT</td>
<td>YES</td>
</tr>
<tr>
<td>HAMAMATSU TV CO. LTD</td>
<td>512 LINES</td>
<td>512</td>
<td>1024</td>
<td>16 BIT</td>
<td>16 BIT</td>
<td>YES</td>
<td>BIT-BIT</td>
<td>YES</td>
<td>16 BIT</td>
<td>YES</td>
</tr>
<tr>
<td>COLORADO VISION INC.</td>
<td>512 LINES</td>
<td>512</td>
<td>1024</td>
<td>16 BIT</td>
<td>16 BIT</td>
<td>YES</td>
<td>BIT-BIT</td>
<td>YES</td>
<td>16 BIT</td>
<td>YES</td>
</tr>
<tr>
<td>STANFORD TECHNOLOGY COMP</td>
<td>512 LINES</td>
<td>512</td>
<td>1024</td>
<td>16 BIT</td>
<td>16 BIT</td>
<td>YES</td>
<td>BIT-BIT</td>
<td>YES</td>
<td>16 BIT</td>
<td>YES</td>
</tr>
<tr>
<td>AVIR CONTROLS</td>
<td>512 LINES</td>
<td>512</td>
<td>1024</td>
<td>16 BIT</td>
<td>16 BIT</td>
<td>YES</td>
<td>BIT-BIT</td>
<td>YES</td>
<td>16 BIT</td>
<td>YES</td>
</tr>
<tr>
<td>GRAPHIC DISPLAY GNE</td>
<td>512 LINES</td>
<td>512</td>
<td>1024</td>
<td>16 BIT</td>
<td>16 BIT</td>
<td>YES</td>
<td>BIT-BIT</td>
<td>YES</td>
<td>16 BIT</td>
<td>YES</td>
</tr>
<tr>
<td>B.M.A. SYSTEMS INC. LF 5000</td>
<td>512 LINES</td>
<td>512</td>
<td>1024</td>
<td>16 BIT</td>
<td>16 BIT</td>
<td>YES</td>
<td>BIT-BIT</td>
<td>YES</td>
<td>16 BIT</td>
<td>YES</td>
</tr>
<tr>
<td>CEP INC.</td>
<td>512 LINES</td>
<td>512</td>
<td>1024</td>
<td>16 BIT</td>
<td>16 BIT</td>
<td>YES</td>
<td>BIT-BIT</td>
<td>YES</td>
<td>16 BIT</td>
<td>YES</td>
</tr>
<tr>
<td>LEXDATA CORP.</td>
<td>512 LINES</td>
<td>512</td>
<td>1024</td>
<td>16 BIT</td>
<td>16 BIT</td>
<td>YES</td>
<td>BIT-BIT</td>
<td>YES</td>
<td>16 BIT</td>
<td>YES</td>
</tr>
<tr>
<td>SPATIAL DATA SYSTEMS INC.</td>
<td>512 LINES</td>
<td>512</td>
<td>1024</td>
<td>16 BIT</td>
<td>16 BIT</td>
<td>YES</td>
<td>BIT-BIT</td>
<td>YES</td>
<td>16 BIT</td>
<td>YES</td>
</tr>
<tr>
<td>PRINCETON ELECTRONICS</td>
<td>512 LINES</td>
<td>512</td>
<td>1024</td>
<td>16 BIT</td>
<td>16 BIT</td>
<td>YES</td>
<td>BIT-BIT</td>
<td>YES</td>
<td>16 BIT</td>
<td>YES</td>
</tr>
</tbody>
</table>

**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.
<table>
<thead>
<tr>
<th>Model</th>
<th>Size</th>
<th>Tape</th>
<th>Prob.</th>
<th>Data</th>
<th>Software</th>
<th>Speed</th>
<th>Cost</th>
<th>Software</th>
<th>Notable Features</th>
<th>Comments</th>
<th>Reasons for Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQUIREMENTS</td>
<td>50MB</td>
<td>800/1600</td>
<td>HP1</td>
<td>128MB</td>
<td>HP1</td>
<td>MULTIR</td>
<td>USERS</td>
<td>WIFLE</td>
<td>EXTENSIONS</td>
<td>DATA BASE</td>
<td>HIGH</td>
</tr>
<tr>
<td>PRIME 550</td>
<td>64KB</td>
<td>0</td>
<td>32KB</td>
<td>(INCL.</td>
<td>DATA)</td>
<td>0.1</td>
<td>64</td>
<td>MAX</td>
<td>0</td>
<td>0</td>
<td>VERY</td>
</tr>
<tr>
<td>HEMLENT-PACKARD 2000</td>
<td>50MB</td>
<td>800/1600</td>
<td>HP1</td>
<td>54KB</td>
<td>1.6MB</td>
<td>64</td>
<td>MAX</td>
<td>03</td>
<td>0</td>
<td>FAST</td>
<td>80.66%</td>
</tr>
<tr>
<td>DATA GENERAL 5/130</td>
<td>50MB</td>
<td>0</td>
<td>64KB</td>
<td>INCL.</td>
<td>DATA</td>
<td>0</td>
<td>2</td>
<td>MAX</td>
<td>0</td>
<td>ADDITIONAL</td>
<td>COST</td>
</tr>
<tr>
<td>DIBITAL EQUIP. CORP. PDP 11/34</td>
<td>150h</td>
<td>800</td>
<td>HP1</td>
<td>54KB</td>
<td>INCL.</td>
<td>DATA</td>
<td>MULTIPLE</td>
<td>0</td>
<td>R. B.</td>
<td>R. B.</td>
<td>101K + PERIP.</td>
</tr>
<tr>
<td>DIBITAL EQUIP. CORP. VAX 11-780</td>
<td>67MB</td>
<td>0</td>
<td>32KB</td>
<td>INCL.</td>
<td>DATA</td>
<td>0</td>
<td>96</td>
<td>MAX</td>
<td>0</td>
<td>0</td>
<td>R. B.</td>
</tr>
</tbody>
</table>
| PRIME 400    | 80MB | 0 | 32KB | INCL. | DATA | 0 | 65 | MAX | 0 | VERY  | FAST | 140K | PERIP. | 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 (GPIB) and installed in the computer. Two cables connect the STC image processor to the GPIB. 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:

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

\[
\begin{bmatrix}
1 & -8 & 1 \\
1 & 1 & 1 \\
\end{bmatrix}
\]

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

\[ U = F \times \frac{t}{D}, \]

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, \( t = 14 \text{ cm} \) (5.5 in.) and \( D = 213 \text{ cm} \) (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.

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

**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.

<table>
<thead>
<tr>
<th>RESOLUTION FACTOR</th>
<th>LIMITATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geometric Unsharpness</td>
<td>0.13 mm</td>
</tr>
<tr>
<td></td>
<td>(small spot)</td>
</tr>
<tr>
<td></td>
<td>(large spot)</td>
</tr>
<tr>
<td>Image Intensifier</td>
<td>0.18 mm</td>
</tr>
<tr>
<td></td>
<td>(high mag.)</td>
</tr>
<tr>
<td></td>
<td>(normal)</td>
</tr>
<tr>
<td>Pixel Separation</td>
<td>0.16 mm</td>
</tr>
<tr>
<td></td>
<td>(high mag.)</td>
</tr>
<tr>
<td></td>
<td>(normal)</td>
</tr>
</tbody>
</table>

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 penetrater 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 penetrater. The penetrater thickness used for 2.54 cm (1.0") material is 0.0508 cm (0.02"). The quality level is expressed as the penetrater thickness and the smallest hole size visible, e.g., 2%-2T = .04" dia. hole in .02" thick penetrater 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 penetrameter holes.

<table>
<thead>
<tr>
<th>ALUMINUM THICKNESS</th>
<th>QUALITY LEVEL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>REAL TIME RADIOGRAPHY</td>
</tr>
<tr>
<td>3.81 cm (1.5)</td>
<td>4% - 2T</td>
</tr>
<tr>
<td>2.54 cm (1.0)</td>
<td>4% - 2T</td>
</tr>
<tr>
<td>0.92 cm (.375)</td>
<td>4% - 2T</td>
</tr>
</tbody>
</table>

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.
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.
5.2.1 INSPECTION

The inspection function of CAI is designed around a of 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.
OPERATOR INPUT
- INSPECTION NAME
- PART NUMBER
- SERIAL NUMBER
- INSPECTOR I.D.

LOCATE PLAN
GENERATE INSPECTION POINTS

TEST PLAN FILE

POSITION TEST OBJECT

MOTOR CONTROL

PROCESS IMAGE

IMAGE PROCESSOR

OPERATOR INPUT
PASS/FAIL OBSERVATIONS

IMAGE

TEST RESULTS FILE

FIGURE 5.2.1-1 INSPECTION SUBSYSTEM
<table>
<thead>
<tr>
<th>TEST NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOTOR MASK: X POSITION</td>
<td>X SPEED</td>
</tr>
<tr>
<td>MOTOR TABLE: X POSITION</td>
<td>X SPEED</td>
</tr>
<tr>
<td>MOTOR ROTATION POSITION</td>
<td>SPEED</td>
</tr>
<tr>
<td>IMAGE PROCESS 1</td>
<td></td>
</tr>
<tr>
<td>IMAGE PROCESS 2</td>
<td></td>
</tr>
<tr>
<td>IMAGE PROCESS 3</td>
<td></td>
</tr>
<tr>
<td>IMAGE PROCESS 4</td>
<td></td>
</tr>
<tr>
<td>IMAGE PROCESS 5</td>
<td></td>
</tr>
<tr>
<td>IMAGE PROCESS 6</td>
<td></td>
</tr>
<tr>
<td>IMAGE PROCESS 7</td>
<td></td>
</tr>
<tr>
<td>IMAGE PROCESS 8</td>
<td></td>
</tr>
</tbody>
</table>

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

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.
### Real-Time-X-Ray

1. X-Ray to Visible Light Conversion and Camera 44
2. Monitor 5
3. Room Monitor 1
4. Part Manipulator 20

**SUBTOTAL** 70

### Image Processing and Computer Aided Inspection

1. Prime 550 Computer with 150
   - 1 Mbyte Memory
   - 64 Mbyte Disk
   - Dual Density Tape Drive
2. Computer Interfaces for Image Processor and Part Manipulator 10
3. Image Processor 70

**GRAND TOTAL** 230

---

**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)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Cost</th>
<th>Life</th>
<th>Salvage Value</th>
<th>Interest Rate</th>
<th>Sales Tax</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Real Time X-Ray System</td>
<td>$70,000</td>
<td>8 Years</td>
<td>0</td>
<td>7,000</td>
<td>3,780</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total: $70,000

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>3,780</td>
<td>-3,780</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-3,780</td>
<td>-1,966</td>
<td>0</td>
<td>70,000</td>
<td>-71,966</td>
<td>-71,966</td>
</tr>
<tr>
<td>1</td>
<td>51,429</td>
<td>6,429</td>
<td>45,000</td>
<td>7,778</td>
<td>62,222</td>
<td>9,644</td>
<td>5,787</td>
<td>43,009</td>
<td>29,365</td>
<td>0</td>
<td>37,142</td>
<td>-34,823</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>51,429</td>
<td>6,429</td>
<td>45,000</td>
<td>14,583</td>
<td>97,639</td>
<td>7,384</td>
<td>4,430</td>
<td>34,847</td>
<td>18,120</td>
<td>0</td>
<td>32,704</td>
<td>-2,119</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>51,429</td>
<td>6,429</td>
<td>45,000</td>
<td>12,639</td>
<td>35,000</td>
<td>5,425</td>
<td>3,526</td>
<td>35,887</td>
<td>18,661</td>
<td>0</td>
<td>31,300</td>
<td>29,181</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>51,429</td>
<td>6,429</td>
<td>45,000</td>
<td>10,694</td>
<td>24,306</td>
<td>3,767</td>
<td>2,449</td>
<td>36,754</td>
<td>19,112</td>
<td>0</td>
<td>29,800</td>
<td>58,988</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>51,429</td>
<td>6,429</td>
<td>45,000</td>
<td>8,750</td>
<td>15,556</td>
<td>2,411</td>
<td>1,567</td>
<td>37,817</td>
<td>19,665</td>
<td>0</td>
<td>28,415</td>
<td>87,403</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>51,429</td>
<td>6,429</td>
<td>45,000</td>
<td>6,806</td>
<td>8,750</td>
<td>1,356</td>
<td>882</td>
<td>39,076</td>
<td>20,320</td>
<td>0</td>
<td>27,125</td>
<td>114,528</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>51,429</td>
<td>6,429</td>
<td>45,000</td>
<td>4,861</td>
<td>3,889</td>
<td>603</td>
<td>392</td>
<td>40,531</td>
<td>21,076</td>
<td>0</td>
<td>25,937</td>
<td>140,665</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>51,429</td>
<td>6,429</td>
<td>45,000</td>
<td>2,917</td>
<td>972</td>
<td>151</td>
<td>98</td>
<td>42,181</td>
<td>21,934</td>
<td>0</td>
<td>24,851</td>
<td>165,316</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>51,429</td>
<td>6,429</td>
<td>45,000</td>
<td>972</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>44,028</td>
<td>22,894</td>
<td>0</td>
<td>23,867</td>
<td>189,182</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>51,429</td>
<td>6,429</td>
<td>45,000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>70,000</td>
<td>189,182</td>
<td>0</td>
<td>189,182</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total: 462,861

PAYBACK PERIOD: 2 YEARS (0 MONTHS)

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

Activation Date: 1981

Item 1. $300,000 Life 8 Years Salvage Value of Existing Eqpt 0

Item 2.

Item 3. $16,200

Sales Tax 0

Other Procurement/Installation Expense 0

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>220,571 16,200 -16,200 0 0 0 -16,200</td>
<td>8,424 0 300,000 1308,424</td>
<td>308,424</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>220,571 27,571 193,000 33,333 266,667 41,333 24,800 184,667 125,923 0 159,256 149,168</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>220,571 27,571 193,000 62,500 204,167 31,646 14,947 119,247 77,733 0 140,233 -8,934</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>220,571 27,571 193,000 54,167 150,000 23,250 15,112 133,946 80,052 0 134,218 125,284</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>220,571 27,571 193,000 45,833 104,167 16,146 10,955 157,661 81,953 0 127,617 253,101</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>220,571 27,571 193,000 37,500 66,667 10,333 6,717 162,217 64,353 0 121,853 374,954</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>220,571 27,571 193,000 29,167 37,500 5,812 3,778 167,611 87,158 0 116,325 491,279</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>220,571 27,571 193,000 20,833 16,667 2,583 1,679 173,816 90,000 0 111,123 602,512</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>220,571 27,571 193,000 12,500 4,167 6,486 4,209 150,129 71,250 0 100,378 709,090</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>220,571 27,571 193,000 4,167 0 0 0 131,750 81,989 502,788 211,450 0 300,000 811,550</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Total 1,985,139 266,339 1,720,800 300,000 131,750 81,989 502,788 211,450 0 300,000 811,550</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PAYBACK PERIOD 2 YEARS 0 MONTHS

RATE OF RETURN ON INVESTMENT 64.68% (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 cannot 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. "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

MODEL 70 CONTROL SYSTEM

STC MODEL 70

DATA

STATUS

GENERAL PURPOSE INTERFACE BOARD

(DP13)

DCMD

DFLG

MC

PRIME

A1-2
GENERAL PURPOSE INTERFACE BOARD (GP1B)
APPENDIX 2

IMAGE PROCESSOR SOFTWARE, LOGIC AND CODE
M7010 - COMPUTER-PROCESSOR COMMUNICATION

Diagram of computer-processor communication process.
M7010: MODEL 70 10 DRIVER

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

M7010  11FEB1980  11:00
CALL M7010(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 INTEGER 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 M7010,M70ECB
EXT SP28EM
SEG
D64V
M7010 EQU *
ARGT *

*INITIALIZE THE CONTROLLER
OCF '1760 CLEAR ALL THE REGISTERS ON THE BOARD

A2-3
* OCP '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
LDD =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
BED 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 2'S COMPLEMENT
STA WDCTM OTA '360 OUTPUT TO WC REGISTER
BCNE ERR
LDX =0 LOAD INDEX REGISTER
STX XSAV
WLOOP LDA BUFFER,*1 LOAD NEXT WORD IN BUFFER
INH OTA '1060 OUTPUT DATA
BCNE ERR
WAIT SP2SEM SUSPEND
SKS '1060 SKIP IF DFGL IS HIGH
BCNE INTERRU
LDX XSAV
LDA WDCTM
AIA ADD 1 TO WDCTM
STA WDCTM OTA '360 OUTPUT TO WORD COUNT REGISTER
INC
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

A2-4
BCNE ERR           SET INDEX = 0
LDX =0
STX XSAV
RLOOP
  EQU *
  INH
  OC1 '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
  LOX XSAV
  STA BUFFER,*1 STORE IT IN BUFFER
  LOA WDCTM
  AIA WDCTM ADD 1 TO WDCTM
  I1X
  STX XSAV
  BLT RLOOP BRANCH IF WDCTM < 0
  JMP DONE

*MASTER CLEAR OPERATION
* MC
  EQU *
  OC1 '1160 OUTPUT MASTER CLEAR TO M70
  JMP DONE
* DONE ROUTINE
* DONE
  EQU *
  OC1 '460 DISABLE INTERRUPTS
  LOX =1
  INA '560 INPUT STATUS REGISTER
  BCNE ERR
  STA IOCB,*1 IOCB(2)
  I1X
  INA '1560 INPUT VECTOR ADDRESS
  BCNE ERR
  STA IOCB,*1 IOCB(3)
  I1X
  INA '360 INPUT WORD COUNT
  BCNE ERR
  TCA 2'S COMPLEMENT
  STA IOCB,*1 IOCB(4)
  I1X
  INA '160 INPUT DMT ADDRESS
  BCNE ERR
  STA IOCB,*1 IOCB(5)
  I1X
  INA '1060 INPUT INPUT DATA
  BCNE ERR
  STA IOCB,*1 IOCB(6)
*
PRTN
  EQU *
  DOFF DEVICE NOT ON!!!!!!
  LDX =0
  LOA =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
  LOA =2 SET FLAG TO 2
  STA IOCB,*1 PUT IT IN IOCB(1)
  PRTN
* INTERRUPT ERROR ROUTINE
* INTERR
  EQU *
  AN INTERRUPT HAS OCCURED
A2-5
LDA =8          SET FLAG TO 8
STA I0CB, #1    PUT IT IN I0CB(1)
JMP DONE

*DATA DEFINITION
*WDCTM 0        TWOS COMPLEMENT OF COUNT
CSREG 0
FMTN 0
INCT 0
XSAV 0
DYNM I0CB(3),FUNC(3),BUFFER(3),COUNT(3)
*LINK
M70ECB ECB M70IO,,I0CB,4
END
APPENDIX 3

MOTOR CONTROL SCHEMATICS
MOTION CONTROL SYSTEM

MOTORS

ENCODER

MOTION TABLE CONTROL

GENERAL PURPOSE INTERFACE BOARD

COMPUTER

A3-2
MOTION CONTROL INTERFACE BOARD

Diagram of a motion control interface board showing the flow of data between various registers and buses.
MOTOR CONTROL SYSTEM

COMPUTER INPUTS/OUTPUTS

TO COMPUTER

MANUAL INPUT

MOTOR CONTROL PANEL & LOGIC

CONTROL ROOM

VAULT

AC MOTOR CONTROLLERS

A.C. POWER "ON" RELAY

D.C. MOTORS

A.C. MOTORS

SHAFT ENCODER
MOTOR CONTROL LOGIC BOARD

- 16 CONDUCTOR CABLES (2)
- CONTROL PANEL LOGIC/COMPUTER INTERFACE
- CONTROL BUS
- DATA BUS
- 16 CONDUCTOR CABLES (2)

A.C. MOTOR LOGIC X TABLE "5"
A.C. MOTOR LOGIC Y TABLE "4"
A.C. MOTOR LOGIC ROLL "3"
D.C. MOTOR DRIVE X MASK "7"
D.C. MOTOR DRIVE Y MASK "6"
D.C. MOTOR DRIVE X-RAY VOLTAGE "0"
D.C. MOTOR DRIVE X-RAY CURRENT "1"

A3-7
APPENDIX 4

MOTOR CONTROL SOFTWARE,
LOGIC AND CODE
MOTION - MOTOR(S) CONTROL

MOTION

SET STATUS = 0
LOAD MOTOR NO.

BIT 1 SET?

BIT 2 SET?

IN REMOTE MODE?

OUTPUT ABORT

RETURN

SET STATUS = 1

NO

NO

NO

STORE IN STATUS

RETURN

NO

LOAD DIRECTION AND SPEED, CONFIGURE WORD

OUTPUT WORD TO REGISTER

OUTPUT STROBE PULSE

RETURN

INPUT SLOT ID

YES

YES

RETURN

A4-2
POSIT - READ ENCODER POSITION

1. STORE DATA INTO RELATION
2. INPUT DATA
3. STORE IN VALUE
4. RETURN

- SET STATUS = 0
- ENC. ENCODER NUMBER

- BIT 0 SET
  - BIT 1 SET
  - OUTPUT ADDRESS
  - BIT 10 SET
  - OUTPUT ADDRESS
  - BIT 10 SET
  - STORE STATUS

- YES
- NO

- INPUT SET 10
- RETURN

- OUTPUT ADDRESS
- SEND CLEAR PULSE
- RETURN
CALL MOTION(MOTOR#, DIRECTION, SPEED, STATUS)

MOTOR#= 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
* POSIT: ENTRY FOR ENCODER READOUT PAGE 0001

* POSIT: ENTRY FOR ENCODER READOUT

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

CALL POSIT(ENCODER#, VALUE, STATUS) 29MAY1980

ENCODER# 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 ROTOR *
*FUNCTION: READ POSITION OF ENCODER *
*WRITTEN IN ASSEMBLY LANGUAGE *

DYNT POSIT
SEG
END
A1 - AUTOMATED INSPECTION (TOP LEVEL)

START

INSERT
- PCOM
- RTCOM
- PTCOM
- TCOM
- RCOM
- SYS.COM
- CRTCTRL

INITIAL
OPEN FILES

ERROR? YES → EXIT

GFUNCT
GET FUNCTION FROM OPERATOR

0? YES

20 INSPECTION
PERFORM INSPECTION

30 PLAN
PLANNING FUNCTIONS

40 RETR
RETRIEVE HISTORY

NO ERROR? YES

950 CLOSEM
CLOSE FILES

1000 EXIT

A5-2
APDAT - APPEND DATE TO KEYWORD

ARGUMENTS: 2
- STRING: KEY
- INTEGER: LENGTH

A5-3
CLOALL - CLOSE ALL OPEN CHANNELS
ARGUMENTS: NONE

CLOALL

INSERT
*SYS.COM> A*KEYS

DO 62

CHANNEL OPEN?

CLO$A
CLOSE CHANNEL

END OF DO

RETURN
CNEWC - CREATE NEW IMAGE PROCESSOR COMMAND

ARGUMENTS: 1
- INTEGER: ERROR
CNEWP - CREATE NEW PLAN

ARGUMENTS: 1

- INTEGER: ERROR

[Diagram of flowchart for CNEWP - CREATE NEW PLAN]
CNEWT - CREATE NEW TEST ENTRY

ARGUMENTS: 1
  INTEGER: ERROR

---

A5-7
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

A5-9
CRTRES - CREATE A TEST RESULTS RECORD

ARGUMENTS: 2
  • INTEGER: TEST SEQUENCE NO.
  • STRING: RESULTS KEYWORD

CRTRES

INSERT
  • PTCOM • RTCOM
  • RCOM • TCOM

ZFIL
ZERO THE RECORD

ZMVD
MOVE TEST PLAN TO RESULTS

ZMVD
STORE IMAGE INFO IN RESULTS RECORD

ZMVD
STORE IMAGE INFO IN KEYWORD

ZMVD
STORE TAPE NAME AND NUMBER

RETURN

A5-10
DBPRM - INSERT DATA BASE PARAMETERS
ARGUMENTS: NONE

1. DBPRM
2. INSERT
   - PCOM
   - RCOM
   - PTCOM
   - RTCOM
   - TCOM
3. ZMV
4. INSERT "PLAN"
5. ZMV
6. INSERT "PLANTEST"
7. ZMV
8. INSERT "INSPECTION RESULTS"
9. ZMV
10. INSERT "INSPECTION TEST RESULTS"
11. ZMV
12. INSERT "TAPE"
13. ZMV
14. INSERT "COMMANDS"
15. RETURN
A5-11
DINSP - DO THE INSPECTION PER PLAN
ARGUMENTS: 5
- STRING: INSPECTOR I.D.
- STRING: PART NUMBER
- STRING: TEST I.D.
- STRING: SERIAL NUMBER
- INTEGER: ERROR

A5-12
DLTC - DELETE IMAGE PROCESSOR COMMAND

ARGUMENTS:
1
- INTEGER: ERROR

PROGRAMME:

1. INSERT
   - CCOM
   - SYS.COM > PARM.K

10. MESSAGE:
    WHAT IS COMMAND NAME?

NEXTS
FIND THE RECORD

8000
CERR = 7?
YES
8004
MESSAGE:
COMMAND NOT FOUND
RETURN

8040
MESSAGE:
MIDAS ERROR
RETURN

DELETES
DELETE RECORD

9010
MESSAGE:
MIDAS ERROR
RETURN

30. MESSAGE:
DELETED

IMORE
MORE?

A5-13
DLTP - DELETE A PLAN

ARGUMENTS:
1. INTEGER: ERROR
DLTR - DELETE RESULTS

ARGUMENTS: 1
• INTEGER: ERROR

DLTR

MESSAGE:
"DELETE RESULTS FORBIDDEN"

RETURN

DMORE - DO YOU WANT TO DO MORE?

ARGUMENTS: 1
• LOGICAL: MORE

DMORE

INSERT
• SYSCOM>A$KEYS

YSNO$A

MORE?

RETURN

A5-15
DLTT - DELETE TEST PLAN
ARGUMENTS: 1
- INTEGER: ERROR

DLTT

INSERT
- PTCOM
- SYS.COM>PARM.K

MESSAGE:
TEST NAME?
ACCEPT NAME

NEXT$ FIND RECORD
ALT RTN

8000 PTERR = 7?
YES
8004 MESSAGE:
TEST PROC. NOT FOUND
RETURN
NO
8005 MESSAGE:
MIDAS ERROR
RETURN

DELET$ DELETE RECORD
ALT RTN

9000 MESSAGE:
MIDAS ERROR
RETURN

30 MESSAGE:
DELETED

DMORE

DO MORE?

YES RESPONSE
NO RETURN

A5-16
DSPLD - DISPLAY OR PRINT PLAN AND RESULT DATA

ARGUMENTS: 1
- INTEGER: ERROR

DSPLD

10
GDFNCT
GET DISPLAY FUNCTION

MENU = 0?
YES
RETURN

100
DSPP
DISPLAY MAIN PLAN

200
DSPPT
DISPLAY PLAN TEST

300
DSPR
DISPLAY MAIN RESULTS

400
DSPRT
DISPLAY TEST RESULTS

500
DSPT
DISPLAY TAPE

600
LISTIX
LIST INDEX

8000
SPCHN = 0?

YES
SPLIT
PRINT THE RESULTS

A5-17
DSPP - DISPLAY MAIN PLAN

ARGUMENTS: 1
  • INTEGER: REPORT CHANNEL

---

A5-18
DSPPT - DISPLAY ONE TEST PLAN

ARGUMENTS: 1
- INTEGER: SPOOL CHANNEL

A5-19
DSPR - DISPLAY RESULTS

ARGUMENTS: 1
- INTEGER: REPORT CHANNEL

A5-20
DSPPT - DISPLAY ONE TEST PLAN

ARGUMENTS:  1
  INTEGER:  SPOOL CHANNEL

A5-19
DSPRT - DISPLAY RESULTS OF ONE TEST

ARGUMENTS: 1
- INTEGER: SPOOL CHANNEL

A5-21
DSPT - DISPLAY TAPE DATA

ARGUMENTS: 1
1. INTEGER: SPool FILE CHANNEL

A5-22
FREECH - RETURN THE NUMBER OF AVAILABLE PRIMOS CHANNELS

ARGUMENTS: 2
- INTEGER: FREE CHANNELS REQUESTED
- INTEGER: ARRAY SIZE

```
FREECH

INSERT
- SYS.COM.KEYS.F
- SYS.COM.ERRD.F

DO "ICH"

UNITSA IN USE?

1 > NUM. AVAIL.?

ERROR = 41

RETURN
```

A5-23
GDFNCT - GET DISPLAY FUNCTION

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

Flowchart Diagram:

A5-24
GFunct - Get Function, Main Menu

GPFNCT - Get Plan Function

GRFNCT - Get Retrieval Function

Arguments: 1
- Integer: MENU

Diagram:

GFunct
GPFNCT
GRFNCT

PRINT/DISPLAY MENU

ACCEPT OPERATORS CHOICE

NO

WITHIN RANGE?

YES

RETURN
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

GPLNM

INSERT
- PCOM
- SYS.COM> PARM.K
- SYS.COM> A$KEYS

CRPKEY
CREATE KEYWORD

ZFIL
ZERO THE RECORD

9200
ZMVD
DEFAULT PLAN
Y
YES

RESPONSE

RETURN

10

NEXT$?
FIND THE RECORD

9100
YSNO$A
NOT FOUND; USE
DEFAULT PLAN

PLERR = 7?

RETURN

PLERR = 22 OR 24?

9900
MESSAGE:
MIDAS ERROR

RECYCL
TRY AGAIN

A5-27
GPTST - GET A TEST PLAN

ARGUMENTS: 2
- INTEGER: TEST NO.
- INTEGER: ERROR

FLOW DIAGRAM:

GPTST → INSERT
  - PCOM
  - PTCOM
  - SYSCOM>PARM.K

1. NEXT$ ≥ 10
2. FIND TEST RECORD
3. ALT. = 22 OR 24?
   - YES → PTERR = 1000
     → CALLING RECYCLE
   - NO → PRINT/DISPLAY TEXT

4. RETURN

5. PTERR = 9900
6. PTERR = 7?
   - YES → 9901
     → MESSAGE: NOT IN DATA BASE
   - NO → MESSAGE: MIDAS ERROR

7. RETURN

9901 → TRY AGAIN

A5-28
INITIAL - SET UP TERMINAL CODES AND OPEN DATA BASE FILES

ARGUMENTS:
- INTEGER: ERROR

FLOWCHART DESCRIPTION:
- Process starts with initial steps involving input and setting values.
- Follows with error handling and file opening operations.
- Continues with conditional checks and loops as per the flowchart.
- Ends with return statements.

A5-29
INSP - PERFORM AN INSPECTION

ARGUMENTS: 2
LOGICAL: EQUIPMENT
INTEGER: ERROR

INSP

10 GET INSPI INPUT

ERROR? YES RETURN

GIIN

GET PLAN RECORD

1000 ERROR? YES RETURN

DINSN

DO THE INSPECTION

1001 ERROR? YES RETURN

DMORE

MORE?

YES RESPONSE NO RETURN

A5-30
INTAP - INITIALIZE TAPE

ARGUMENTS: 3
- INTEGER: EQUIPMENT INIT. FLAG
- INTEGER: EQUIP. IN USE FLAG
- INTEGER: ERROR
INTAP - INITIALIZE TAPE (CONT'D)
INTIP - INITIALIZE THE IMAGE PROCESSOR

ARGUMENTS:  1
   INTEGER: ERROR

INTIP

VERIFY IMAGE PROCESSOR IS ON

CMDIR
LOAD SYSTEM 500 COMMON

ICLEAN
INITIAL CLEAN OF IMAGE DIRECTORY

CMDM70
TURN ON THE CAMERA

RETURN
INTMOT - INITIALIZE MOTORS

ARGUMENTS:
1
INTEGER: ERROR

A5-34
MDFYC - MODIFY IMAGE PROCESSOR COMMAND

ARGUMENTS: 1

INTEGER: ERROR

A5-36
MDFYP - MODIFY PLAN
MKFMC - MAKE IMAGE COMMAND FROM OLD
MKFMP - MAKE FROM PLAN
MKFMT - MAKE FROM TEST
ARGUMENTS: 1
  • INTEGER: ERROR

RVINSPI - RECREATE AN INSPECTION
ARGUMENTS: NONE

SVIMD - SAVE IMAGE ON DISK
ARGUMENTS: 1
  • STRING: NAME

MESSAGE: NOT IMPLEMENTED

RETURN
MDFYT - MODIFY TEST PROCEDURE

ARGUMENTS:
- 1
  - INTEGER: ERROR

A5-38
MOTOR - CONTROL MOTION OF A MOTOR

ARGUMENTS: 4
- INTEGER: MOTOR NUMBER
- INTEGER: MOTOR SPEED
- INTEGER: POSITION
- INTEGER: ERROR

A5-39
MOV TAP - VERIFY AND MOVE TAPE

ARGUMENTS:
- INTEGER: FILE NO. TO POSITION TAPE
- INTEGER: ERROR
- LOGICAL: NEW TAPE

Diagram with flowchart and algorithm details.
PIMG - PROCESS THE IMAGE

ARGUMENTS:
- INTEGER: TEST SEQUENCE NO.
- LOGICAL: NO TAPE FLAG
- INTEGER: ERROR

Diagram of the process flow.
PLAN - MAIN PLANNING ROUTINE

ARGUMENTS: 1

INTEGER: ERROR
PMOT - POSITION MOTORS
ARGUMENTS: NONE

PMOT

INSERT
- PTCOM

IPOS = 167

DO "PMOT"

= 0 MOTOR SPEED ≠ 0

5 MOTOR TURN MOTOR ON

ERROR?

YES 30 IERR = 1?

YES TNOUA PUT MTR CNTRL IN REMOTE

NO 61 MESSAGE: MOTOR SPEED ERROR

10 IPOS = IPOS + PNWM

20 END OF DO

RETURN

RETURN
POSINT - POSITION MOTOR & INITIALIZE ENCODER

ARGUMENTS: 2
- INTEGER: ENCODER
- INTEGER: ERROR
RETR - RETRIEVE HISTORICAL RECORD

ARGUMENTS: 2
INTEGER: ERRORk
LOGICAL: EQUIPMENT

RETR

10
GRFNCT
GET MENU

>3
MENU = ?
0
RETURN
1 THRU 3

INTIP
INITIALIZE IMAGE PROCESSOR

ERROR?
YES
MESSAGE: ERROR IN INITIALIZING IMAGE PROC.
RETURN

15

100
RVIMG
RETRIEVE IMAGE

200
RVIST
RECREATE ONE TEST RESULT

300
RVINSP
RECREATE AN INSPECTION

400
DSPR
DISPLAY MAIN RESULTS

500
DSPRT
DISPLAY TEST RESULTS

600
DSPT
DISPLAY TAPE RECORD

700
LISTIX
LIST INDEX

A5-45
RMNAB - REMOVE NON-ALPHAHBETICAL CHARACTERS

ARGUMENTS: 2
- STRING: TEXT
- INTEGER: NO. OF CHARACTERS
RVIMG - RETRIEVE AN IMAGE
ARGUMENTS: NONE
RVTST - RECREATE ONE TEST RESULT

ARGUMENTS: NONE

CONTINUED ON NEXT PAGE
SPOLIT - SPOOL RETRIEVAL INFORMATION

ARGUMENTS: 2
- INTEGER: SPOOL CHANNEL
- STRING: SPOOL FILE NAME

A5-50
SVIMT - SAVE IMAGE ON TAPE

ARGUMENTS: 1
- STRING: SYSTEM IMAGE NAME

SVIMT

INSERT
- RTCOM
- TCOM
- SYS.COM>A$KEYS

ZMVD
MOVE IMAGE NAME

TNFL= TNFL+1
TFCNT= TFCNT+1

ZMV
MOVE COMMAND LINE

ZMVD
MOVE SYSTEM NAME

PACK
REMOVE ILLEGAL CHARACTERS

CMDM70
SAVE IMAGE ON TAPE

RETURN

A5-51
SVMRES - SAVE RESULTS - MAIN RECORD

ARGUMENTS: 1
- INTEGER: ERROR

A5-52
SVRES - SAVE RESULTS OF INSPECTION

ARGUMENTS:
- LOGICAL: DEFAULT
- LOGICAL: NO TAPE
- INTEGER: LOOP
- INTEGER: RTKEY
- INTEGER: ERROR

```
SVSTR
```

CLOSE SCREEN FILE

CLOSEDSCREEN FILES

GET SAVE OPTION

SAVE = 0

SAVE = 1

SAVE SPACE ON DISK

FILE NAME ERROR CODE

ERROR

GET RESULTS

A5-53
SVTAPR - SAVE THE TAPE RECORD (MIDAS)

ARGUMENTS: 1
- INTEGER: ERROR
SVTSTR - SAVE TEST RESULTS

ARGUMENTS: 2
- INTEGER: KEY
- INTEGER: ERROR

SVTSTR

INSERT
- RCOM
- TCOM
- SYSCOMP-PARM.K

RTPF

IFLAG = FL$RET

ADD1S
- PUT TEST RESULTS IN DATA BASE

9010 MESSAGE: MIDAS ERROR

RETURN

RETURN
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**

C

C

C

C

C

**REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM**

**VERSION 1.0**

**JUNE 1, 1980**

**BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY**

C INTEGER PSZW, PSZB, FNWT
PARAMETER PNWT = 10
/* No. OF WORDS IN TEST NAME
PARAMETER PSZU = 10*PNWT + 21
/* PLAN RECORD SIZE IN WORDS
PARAMETER F'SZB = F'SZW*2
/* PLAN RECORD SIZE IN BYTES

C INTEGER PREC(FSZW) /* PLAN RECORD
C 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

C PRIMARY KEY: CONDENSATION OF PART-NUMBER AND INSPECTION NAME
WITH / SEPARATING THEM. 30 CHARACTERS LONG.
SECONDARY KEY: TEST NAMES.
C EQUIVALENCE (PREC(1), PPN) /* PLAN PART No.
EQUIVALENCE (PREC(11), PIN) /* " INSPECTION NAME
EQUIVALENCE (PREC(22), PTNO1) /* TEST 01 NAME
EQUIVALENCE (PREC(32), PTNO2) /* TEST 02 NAME
EQUIVALENCE (PREC(42), PTNO3) /* TEST 03 NAME
EQUIVALENCE (PREC(52), PTNO4) /* TEST 04 NAME
EQUIVALENCE (PREC(62), PTNO5) /* TEST 05 NAME
EQUIVALENCE (PREC(72), PTNO6) /* TEST 06 NAME
EQUIVALENCE (PREC(82), PTNO7) /* TEST 07 NAME
EQUIVALENCE (PREC(92), PTNO8) /* TEST 08 NAME
EQUIVALENCE (PREC(102), PTNO9) /* TEST 09 NAME
EQUIVALENCE (PREC(112), PTNO10) /* TEST 10 NAME

C 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
C EQUIVALENCE (PLARR, PLERR)
COMMON /PCOM/ PREC, PLNAM, PLEN, PCHN, PLARR

A5-56
C PICOM: PLAN TEST COMMON FILE

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

INTEGER PTSZW, PTSZB, PNWR, F'NWI, F'NMOT, PNIMG
PARAMETER PNMOT=5 /* No. of Motor Commands
PARAMETER PNWM=10 /* No. of Words per Motor Command
PARAMETER PNWI=41 /* No. of Words per Image Command
PARAMETER PTSZW=PNMOT*PNWM+PNIMG*PNWI /* No. of Bytes in PLAN TEST Record

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 PIPC(40) /* Image Process C
INTEGER PIPD(40) /* Image Process D
INTEGER PIPC(40) /* Image Process E
INTEGER PIPF(40) /* Image Process F
INTEGER PIPG(40) /* Image Process G
INTEGER PIPH(40) /* Image Process H
INTEGER PNXH /* Code for Next Operation

PRIMARY KEY = PTNM (1st. Element, 20 Char.)
SECONDARY KEY = NONE.

EQUIVALENCE (PTREC(1), PTNM) /* Test Name
EQUIVALENCE (PTREC(11), PDA) /* Description Line A
EQUIVALENCE (PTREC(12), PDB) /* Description Line B
EQUIVALENCE (PTREC(13), PDC) /* Description Line C
EQUIVALENCE (PTREC(15), PDD) /* Description Line D
EQUIVALENCE (PTREC(16), PKXP) /* Mask X Position
EQUIVALENCE (PTREC(16), PKXS) /* Mask X Speed
EQUIVALENCE (PTREC(16), PKYP) /* Mask Y Position
EQUIVALENCE (PTREC(16), PKYS) /* Mask Y Speed
EQUIVALENCE (PTREC(17), PMXP) /* Table X Position
EQUIVALENCE (PTREC(17), PMXS) /* Table X Speed
EQUIVALENCE (PTREC(17), PMYP) /* Table Y Position
EQUIVALENCE (PTREC(17), PMYS) /* Table Y Speed
EQUIVALENCE (PTREC(18), PMRP) /* Rotate Position
EQUIVALENCE (PTREC(18), PMRS) /* Rotate Speed
EQUIVALENCE (PTREC(19), PIPA) /* Image Process A
EQUIVALENCE (PTREC(21), PNXA) /* Code for Next Operation
EQUIVALENCE (PTREC(21), PIPB) /* Image Process B
EQUIVALENCE (PTREC(23), PNXH) /* 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(424),PNXG)  /* CODE FOR NEXT OPERATION
EQUIVALENCE (PTREC(463),PIPH)  /* IMAGE PROCESS H
EQUIVALENCE (PTREC(464),PNXH)  /* CODE FOR NEXT OPERATION

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

EQUIVALENCE (PTARR,PTERR)
COMMON /PTCOM/ PTREC,PTNAM,PTLEN,PTCHN,PTARR
REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM

VERSION 1.0  JUNE 1, 1980

BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY

INTEGER CNPROC,CWNI,CSZW,CSZB
PARAMETER CNPROC=10  /* NO. OF PROCESSES FOR THIS COMMAND
PARAMETER CWNI=40  /* NO. OF WORDS PER PROCESS LINE
PARAMETER CSZW=CNPROC*CWNI+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(CWNI)  /* 1ST PROCESS COMMAND LINE
INTEGER PROC2(CWNI)  /* 2ND PROCESS COMMAND LINE
INTEGER PROC3(CWNI)  /* 3RD PROCESS COMMAND LINE
INTEGER PROC4(CWNI)  /* 4TH PROCESS COMMAND LINE
INTEGER PROC5(CWNI)  /* 5TH PROCESS COMMAND LINE
INTEGER PROC6(CWNI)  /* 6TH PROCESS COMMAND LINE
INTEGER PROC7(CWNI)  /* 7TH PROCESS COMMAND LINE
INTEGER PROC8(CWNI)  /* 8TH PROCESS COMMAND LINE
INTEGER PROC9(CWNI)  /* 9TH PROCESS COMMAND LINE
INTEGER PROCB(CWNI)  /* 10TH PROCESS COMMAND LINE

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

EQUIVALENCE (CREC(1), CNAM)  /* COMMAND NAME
EQUIVALENCE (CREC(11), PROC1)  /* 1ST 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), PROCB)  /* 10TH PROCESS COMMAND LINE

DATA BASE NAME AND MIDAS INDEX POINTER ARRAY
INTEGER CNAM(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,CNAM,CLEN,CCHN,CARR

A5-59
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, RSZ, RNT, RNWT
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 RSNO(10)
/* SERIAL NO.
INTEGER RIID(5)
/* INSPECTOR'S ID
INTEGER RDATI(6)
/* DATE-TIME
INTEGER RPF
/* PASS-FAIL CODE
INTEGER RTNO1(RNWT)
/* TEST 01 NAME
INTEGER RTNO2(RNWT)
/* TEST 02 NAME
INTEGER RTNO3(RNWT)
/* TEST 03 NAME
INTEGER RTNO4(RNWT)
/* TEST 04 NAME
INTEGER RTNO5(RNWT)
/* TEST 05 NAME
INTEGER RTNO6(RNWT)
/* TEST 06 NAME
INTEGER RTNO7(RNWT)
/* TEST 07 NAME
INTEGER RTNO8(RNWT)
/* TEST 08 NAME
INTEGER RTNO9(RNWT)
/* TEST 09 NAME
INTEGER RTNO10(RNWT)
/* TEST 10 NAME

C PRIMARY KEY: CONDENSATION OF PART-NUMBER, INSPE. NAME, & SERIAL NO.
SEPARATED BY "/". 40 CHARACTERS LONG.

C SECONDARY KEY: TEST NAMES

EQUIVALENCE (RREC(1), RPN)
/* RESULTS PART NO.
EQUIVALENCE (RREC(11), RIN)
/* INSPECTION NAME
EQUIVALENCE (RREC(21), RNWT)
/* SERIAL NO.
EQUIVALENCE (RREC(31), RIID)
/* INSPECTOR'S ID
EQUIVALENCE (RREC(36), RDATI)
/* DATE-TIME
EQUIVALENCE (RREC(42), RPF)
/* PASS-FAIL CODE
EQUIVALENCE (RREC(43), RTNO1)
/* TEST 01 NAME
EQUIVALENCE (RREC(48), RTNO2)
/* TEST 02 NAME
EQUIVALENCE (RREC(93), RTNO3)
/* TEST 03 NAME
EQUIVALENCE (RREC(118), RTNO4)
/* TEST 04 NAME
EQUIVALENCE (RREC(143), RTNO5)
/* TEST 05 NAME
EQUIVALENCE (RREC(168), RTNO6)
/* TEST 06 NAME
EQUIVALENCE (RREC(193), RTNO7)
/* TEST 07 NAME
EQUIVALENCE (RREC(218), RTNO8)
/* TEST 08 NAME
EQUIVALENCE (RREC(243), RTNO9)
/* TEST 09 NAME
EQUIVALENCE (RREC(268), RTNO10)
/* TEST 10 NAME

C 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

A5-60
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, RNIMG
PARAMETER RNMOT = 5  
   /* No. of Motor Commands
PARAMETER RNIMG = 10  
   /* No. of Image Commands
PARAMETER RNWI = 2  
   /* No. of Words per Motor Command
PARAMETER RTSZW = RNWI*RNWM*RNIMG*RNMOT  
   /* No. of Bytes in Plan Test Record
PARAMETER RTSZB = RTSZW*2

INTEGER RTREC(RTSZW)

INTEGER RTNM(10)  
   /* Test Name
INTEGER RDA(39)  
   /* Description Line A
INTEGER RDC(39)  
   /* Description Line B
INTEGER RDD(39)  
   /* Description Line C
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 RIBP(40)  
   /* Image Process B
INTEGER RIMB(40)  
   /* Image Process C
INTEGER RIPB(40)  
   /* Image Process D
INTEGER RIPC(40)  
   /* Image Process E
INTEGER RIPF(40)  
   /* Image Process F
INTEGER RNXF  
   /* Code for Next Operation
INTEGER RNXG  
   /* Code for Next Operation
INTEGER RNXH  
   /* Code for Next Operation
INTEGER RNXI  
   /* Code for Next Operation
INTEGER RNXJ  
   /* Code for Next Operation
INTEGER RNXK  
   /* Code for Next Operation
INTEGER RNXL  
   /* Code for Next Operation
INTEGER RNXM  
   /* Code for Next Operation
INTEGER RNXN  
   /* Code for Next Operation
INTEGER RNXO  
   /* Code for Next Operation
INTEGER RNXP  
   /* Code for Next Operation
INTEGER RNAP  
   /* Code for Next Operation
INTEGER RNBD  
   /* Code for Next Operation
INTEGER RNID  
   /* Code for Next Operation
INTEGER RIND  
   /* Code for Next Operation
INTEGER RINM  
   /* Code for Next Operation
INTEGER RINP  
   /* Code for Next Operation

INTEGER RTPF  
   /* PASS-FAIL Code
INTEGER RTIMN(25)  
   /* Tape Name
INTEGER RTFNM  
   /* File No.
INTEGER RTNM(10)  
   /* Image Name

C PRIMARY KEY = SAME AS TEST NAME FROM RRREC, AND RTIMN
C 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), RMYS)  
   /* Table Y Speed

A5-61
EQUIVALENCE (RTREC(175), RNRP)  /* ROTATE POSITION */
EQUIVALENCE (RTREC(176), RNRPS) /* ROTATE SPEED */
EQUIVALENCE (RTREC(177), RIPA)  /* IMAGE PROCESS A */
EQUIVALENCE (RTREC(217), RNXA)  /* CODE FOR NEXT OPERATION */
EQUIVALENCE (RTREC(218), RIPR)  /* 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), RIPC)  /* 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(564), RTCM2) /* COMMENT 2 */
EQUIVALENCE (RTREC(606), RTNM)  /* TAPE NAME */
EQUIVALENCE (RTREC(616), RTFNM) /* FILE # */
EQUIVALENCE (RTREC(617), RTIMN) /* IMAGE NAME */

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
INTEGER TSZW, TSZB, TNWT
PARAMETER TNWT=25 /* No. of words in TEST-RESULTS NAME
PARAMETER TSZW=TNWT*52+11 /* TAPE RECORD SIZE IN WORDS
PARAMETER TSZB=TSZW*2 /* TAPE RECORD SIZE IN BYTES

INTEGER TREC(TSZW) /* TAPE RECORD
INTEGER TNAM(10) /* TAPE NAME
INTEGER TNFL /* No. FILES RECORDED ON THIS TAPE
INTEGER TIMNA(TNWT) /* IMAGE A NAME

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

EQUIVALENCE (TREC(1), TNAM) /* TAPE NAME
EQUIVALENCE (TREC(11), TNFL) /* No. OF ENTRIES THIS FILE
EQUIVALENCE (TREC(12), TIMNA) /* IMAGE NAME

DATA-BASE NAME AND MIDAS INDEX POINTER ARRAY
INTEGER TPNAM(16) /* TAPE DATA FILE NAME
INTEGER TPLEN /* NAME LENGTH
INTEGER TCHN /* CHAN. TO DATA BASE FILE
INTEGER TPARR(14) /* MIDAS INFO ARRAY
INTEGER TPERR /* MIDAS ERR. CODE
INTEGER TFCNT /* TAPE FILE COUNT (PRESENT)
/* TFCNT = 1 FOR 1st FILE.

EQUIVALENCE (TPARR, TPERR)
COMMON /TCOM/, TREC, TPNAM, TPLEN, TCHN, TPARR, TFCNT
C AI: AUTOMATED INSPECTION - TOP LEVEL
**********REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM**********
*VERSION 1.0 JUNE 1,1980*
*BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY*
*THIS IS THE MAIN PROGRAM FOR THE AUTOMATED INSPECTION SYSTEM*
*FUNCTION: 1. OPEN DATA BASE FILES
*2. INITIATE, INSPECTION, PLANNING, OR RETRIEVAL
*3. OR CLOSE DATA BASE FILES AND EXIT*

$INSERT PCOM
$INSERT PCON
$INSERT RCOM
$INSERT RCON
$INSERT TCOM
$INSERT SYSCOM>CRTCTRL

COMMON/OREZ/ZERO(8191)

LOGICAL EQUIP
EQUIP = .FALSE.

SET UP TERMINAL CODES, OPEN DATA BASE FILES
----------
call INITAL(IERR)
If (IERR .NE. 0) call exit

GET FUNCTION: INSPECT, PLAN, RETRIEVE HISTORY, CLOSE FILES
----------
call GFUNCT(MENU)
If (MENU .EQ. 0) goto 950
Goto (20,30,40), MENU
Goto 950

PERFORM INSPECTION
----------
call INSPEQ(EQUIP,IERR)
If (IERR) 950,10,950

PLAN
----------
call PLAN(IERR)
If (IERR) 950,10,950

RETRIEVE HISTORY
----------
call RETR(EQUIP,IERR)
If (IERR) 950,10,950

CLOSE DATA BASE FILES
----------
call CLOALL

EXIT
----------
call EXIT
end

A5-64
SUBROUTINE APDAT(KEY, LEN)

REAL*8 DD, DT, DUMMY(2)
DIMENSION IDT(8)
INTEGER BIAS, IE
EQUIVALENCE (IDT(1), DD)
EQUIVALENCE (IDT(5), DT)

DD = DATE*A(DUMMY)  /GET DATE
TT = TIME*A(DT)  /GET TIME
CALL RANAB(IDT, 16)  /REMOVE NON-ALPHABETICAL CHARACTERS
IE = LEN
IE = IE - 11
IF (IE .LE. 0) RETURN
CALL MSUBA(IDT, 12, 1, 12, KEY, LEN, IE)  /* APPEND THE DATE
RETURN
END

SUBROUTINE CLOALL

DO 100 ICH = 1, 62
   IF (.NOT. UNITA(ICH)) GO TO 100
   IF (.NOT. CLOS$A(ICH)) GO TO 100
100 CONTINUE
RETURN
END

A5-65
C CREATE NEW IMAGE PROCESSOR COMMAND

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

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

SUBROUTINE CNEWC(IER)

*INSERT CCOM
*INSERT SYSCOM>PARAM.K
*INSERT SYSCOM>A$KEYS

INTEGER TRPR(CSZW),NAME(16)
DATA NAME /'DEMO.SCREEN.FT'/
CALL TONL

Open a channel to the screen template file
---------------------------------------------
CALL VOPEN$(NAME,32,1,IFCH,IER)  /* OPEN FOR READ  
IF (IER .NE. 0) GOTO 9200

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

Clear record buffer
-------------------
10 CALL ZFIL(TRPR,CSZB,0)

20 CALL PEDIT(IFCH,TRPR,IS,IE,NEWSCR)
CALL ZMVD(TRPR,CREC,CSZB)

Check to see if record already exists.
--------------------------------------
IFLAG = FL*RET
CALL NEXT$(CCHN,CREC,CNAM,CARR,IFLAG,$200,$0,$0)
CALL TONL
CALL TMOU ('COMMAND ALREADY EXISTS',24)
CALL TONL
GO TO 300

Store the record
------------------
200 CALL ADD$(CCHN,CREC,CNAM,CARR,IFLAG,$9000,$0,$0,$0)

MORE?
-----
IER = 0
300 IF (YNOSA('More',4,A$DN0)) GOTO 20

CLOSE THE SCREEN CHANNEL NO.
-----------------------------
400 CALL CLOS$(IFCH)
RETURN

IER = CERR
WRITE (1,9001) CERR,CNAM
9001 FORMAT('MIDAS ERROR =',13,' KEY = ',15A2)
GOTO 400

IER = IER
WRITE (1,9201) IER,NAME
9201 FORMAT('ERROR ',13,' OPENING FILE ',16A2)
RETURN

A5-66
C NEWC: CREATE NEW IMAGE PROCESSOR CO PAGE 0002

END
SUBROUTINE CNEWP(IER)

INTEGER TRPR(PSZW),NARE(16)
INTEGER KEYO(15) /* PRIMARY KEY

DATA NARE /'DENO.SCREEN.FT'/

CALL TONL

Open a channel to the screen template file

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

IS = 50
IE = 66
NEWSCR = 2 /* DO NOT ERASE THE SCREEN TO START

Clear record buffer

CALL ZFIL(TRPR,PSZW,0)

CALL PEDIT(IFCH,TRPR,IS,IE,NEWSCR)
CALL ZAVD(TRPR,PREC,PSZW)

Check for internal consistency

IF (PMT .LE. 10) GOTO 27
WRITE (1,26)
26 FORMAT(/,'10 IS MAXIMUM NUMBER OF TESTS!',/)
GOTO 20

IF (TRPR(II) .NE. 0) 60TO 28
WRITE (1,31)
31 FORMAT('INSPECTION NAME MANDATORY!')
GOTO 20

DO 30 I=1,PMT
  IBIAS = 22 + (I-1)*PMT
  IF (TRPR(II) .NE. 0) GOTO 30
  WRITE (1,29)
29 FORMAT(/,'NUMBER OF TESTS NOT CONSISTENT!',/)
GOTO 20
30 CONTINUE

Create a primary keyword.

CALL CRPKEY(PMN,20,PIN,20,0,0,0,0,KEY0,30)
LEN = LSIZE(KEY0,30)

Check to see if record already exists.
C CNEWP: CREATE NEW PLAN ENTRY

IFLAG = FL$RET
CALL NEXT$(PCHN,PREC,KEYO,PLARR,IFLAG,$200,0,0,0,0)
CALL T0NL
CALL TNQU ('PLAN NUMBER ALREADY EXISTS',26)
CALL T0NL
GO TO 300

C Store the record

200 CALL ADDI$(PCHN,PREC,KEYO,PLARR,IFLAG,$9000,0,0,0,0)

C Insert the secondary keys

IBIAS = 22
IFLAG = FL$USE
DO 220 I=1,PNT
CALL ADDI$(PCHN,KEYO,PREC(IBIAS),PLARR,IFLAG,$9100,1,0,0,0)
IBIAS = IBIAS + PNWT
220 CONTINUE

C MORE?

300 IF (YSNOSA('More',4,A$DNO)) GOTO 20

C CLOSE THE SCREEN CHANNEL NO.

400 CALL CLOS$A(IFCH)
RETURN

C IER = PLERR
WRITE (1,9001) PLERR,KEYO
9001 FORMAT('MIDAS ERROR = ',13,' KEY = ',15A2)
GOTO 400

C IER = PLERR
JBIAS = IBIAS + PNWT - 1
WRITE (1,9101) PLERR,(PREC(I),I=IBIAS,JBIAS)
9101 FORMAT('MIDAS ERROR = ',13,' KEY = ',15A2)
GOTO 400

C WRITE (1,9201) IER,NAMEx
9201 FORMAT('ERROR ',13,' OPENING FILE ',16A2)
RETURN

C FORMAT('ERROR = ',13)
END

A5-69
SUBROUTINE CNEWT(IER)

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

INTEGER TRPR(PTSZW),NAME(16)    
INTEGER KEYO(10)    

DATA NARE /'DERO.SCREEN.FT'    

CALL TONI    

C
Open a channel to the screen template file    
C----------------------------------------------------------    
CALL VOPEN$(NAME,32,1,IFCH,IER)    
* OPEN FOR READ    
IF (IER .NE. 0) GOTO 9200

I = 75
IE = 98
NEWSCR = 2    /* DO NOT ERASE THE SCREEN TO START    

C Clear record buffer    
C---------------------    
10 CALL ZFIL(TRPR,PTSZB,0)    

C
20 CALL PEDIT(IFCH,TRPR,IS,IE,NEWSCR)    
CALL ZRVD(TRPR,PTREC,PTSZB)    
WRITE(1,1098)PTNM    /* ***CHECKOUT***
1098 FORMAT(1X,'PTNM FROM CNEWT=',10A2)    

Check to see if record already exists.    
C----------------------------------------------------------
IFLAG = FLSRET
CALL NEXT$(PTCHN,PTREC,PTNM,PTARR,IFLAG,$200,0,0,0,0)
CALL TONL
CALL TNOU('PLAN TEST ALREADY EXISTS',24)
CALL TONL
GO TO 300

C Store the record    
C-------------------    
200 CALL ADDI$(PTCHN,PTREC,PTNM,PTARR,IFLAG,$9000,0,0,0,0)
CALL TDUMP(PTREC,PTSZB)

MORE?
C-------    
300 IF (YSN04A('More',4,A*DNO)) GOTO 20

C CLOSE THE SCREEN CHANNEL NO.
C---------------------------    
400 CALL CLOS$(IFCH)    
RETURN

C
9000 IER = PTERR
WRITE (1,9001) PTERR,PTNM
9001 FORMAT('MIDAS ERROR = ',I3,' KEY = ',15A2)    
A5-70
C CNEWT: CREATE NEW TEST ENTRY  PAGE 0002

GOTO 400

C
9200 WRITE (1,9201) IER,NAMEN
9201 FORMAT('ERROR ',13,', OPENING FILE ',16A2)
RETURN
C
9900 FORMAT('ERROR = ',13)
END

C CRPKEY: CREATE PLAN KEYWORD  PAGE 0001

C CRPKEY: CREATE PLAN KEYWORD

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

*GENERAL PURPOSE SUBROUTINE TO CREATE A KEYWORD
*FUNCTION: 1. CONCATENATE TEXT STRINGS WITH A '/ ' IN BETWEEN
*2. APPENDS DATE AND TIME TO END

SUBROUTINE CRPKEY(TEXT1,MAX1,TEXT2,MAX2,TEXT3,MAX3,TEXT4,MAX4,
C KEY,MAXKEY)

$INSERT SYSCOM>ASKYS
CALL ZFIL(KEY,MAXKEY,' ')
LEN1 = LSIZE(TEXT1,MAX1)      /*FIND LENGTH OF FIRST STRING
CALL ZMV(TEXT1,LEN1,KEY,MAXKEY) /*MOVE FIRST STRING INTO KEY
IF (TEXT2 .EQ. 0) GOTO 1000
IPOS = LEN1 + 1
CALL MCHR$A(KEY,IPOS,'/',1)  /*INSERT "/ " INTO KEY
IPOS = IPOS + 1
LEN2 = LSIZE(TEXT2,MAX2)      /*FIND LENGTH OF SECOND STRING
CALL MSUB$(A(TEXT2,MAX2,1,LEN2,KEY,MAXKEY,IPOS,MAXKEY)
IF (MAX3 .LE. 0) GOTO 1000
IF (TEXT3 .EQ. 0) GOTO 1000
C
IPOS = LSIZE(KEY,MAXKEY) + 1
CALL MCHR$A(KEY,IPOS,'/',1)  /*INSERT "/ " INTO KEY
IPOS = IPOS + 1
CALL MSUB$(A(TEXT3,MAX3,1,MAX3,KEY,MAXKEY,IPOS,MAXKEY)
IF (MAX4 .LE. 0) GOTO 1000
C
IPOS = LSIZE(KEY,MAXKEY) + 1
IF (TEXT4 .EQ. 0) GOTO 1000
CALL MCHR$A(KEY,IPOS,'/',1)  /*INSERT "/ " INTO KEY
IPOS = IPOS + 1
CALL MSUB$(A(TEXT4,MAX4,1,MAX4,KEY,MAXKEY,IPOS,MAXKEY)
1000 RETURN
END

A5-71
SUBROUTINE CRRES(SERIAL, IID, IERR)

INTEGER BIAS, BI, IBUF(10), IKEY(RNWT), BIAS2, B12, TBIAS

NSIZ = RNWT * 2

TBIAS = 22  /* TEST NAME BIAS

DO 10 I = 1, PN7  /* CREATE TESTNAMES FOR RCOM
    CALL NEXT$(PTCHNYPTRECPREC(TBIAS), PTARR, FL$RET, $9000, 0, 0, 0)
    TBIAS = TBIAS + PNWT
    BIAS = 2 * (22 + (I-1)*PNWT) - 1
    BIAS2 = BIAS + 2*PNWT - 1
    BI = 2 * (43 + (I-1)*RNWT) - 1
    B12 = BI + 2*RNWT - 1

    DO 10 I = 1, PN7  /* CREATE TESTNAMES FOR RCOM

    CALL NEXT$(PTCHNYPTRECPREC(TBIAS), PTARR, FL$RET, $9000, 0, 0, 0)
    TBIAS = TBIAS + PNWT
    BIAS = 2 * (22 + (I-1)*PNWT) - 1
    BIAS2 = BIAS + 2*PNWT - 1
    BI = 2 * (43 + (I-1)*RNWT) - 1
    B12 = BI + 2*RNWT - 1

    CALL PUT PREC(BIAS) INTO IBUF

    CALL MSUB$A(PREC, PSZB, BIAS, BIAS2, IBUF, 20, 1, 20)

    CALL CREATE KEY INTO IKEY

    CALL CRPKEY(PPN, PIN, 20, SERIAL, 20, IBUF, 20, IKEY, NSIZ)
    CALL APDAT(IKEY, NSIZ)

    CALL MOVE IKEY INTO RREC(BI)

    CALL MSUB$A(IKEY, NSIZ, 1, NSIZ, RREC, RSZB, BI, B12)

10 CONTINUE

CALL ZMVD(PPN, RPW, 20)
CALL ZMVD(PIN, RIN, 20)
CALL ZMVD(IID, RIID, 10)
CALL ZMVD(SERIAL, RSNO, 20)
RETURN

9000 IERR = PERR
WRITE(1, 9010) IERR, TBIAS
9010 FORMAT(1X, 'ERROR ', 'I2', ' BIAS = ', 'I3', ' IN CRRES CHECK PLANTEST')
RETURN

END
CRTRES: CREATE A TEST RESULTS RECORD.

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

*CALLED BY:  DINSHP
*FUNCTION:  1. MOVE TEST PLAN RECORD TO TEST RESULTS RECORD
*         2. MOVE IMAGE NAME TO TEST RESULTS RECORD
*         3. MOVE TAPE NAME TO TEST RESULTS RECORD

SUBROUTINE CRTRES(I,RTKEY)
I = TEST SEQUENCE NO.

*INSERT PTCOM
*INSERT RCOM
*INSERT RTCOM
*INSERT TCOM

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

ZERO THE RECORD
CALL ZFIL(RTREC,RTSZB,0)

MOVE A COPY OF TEST PLAN TO RESULTS RECORD
CALL ZMVD(FTREC,RTREC,RTSZB)

STORE THE IMAGE NAME
CALL ZMVD(RREC(IBIAS),RTIMN,IMGSIZ)
CALL ZMVD(RTIMN,RTKEY,IMGSIZ)

PLUS THE TAPE NAME & NUMBER
CALL ZMVD(TMAM,RTTIMN,20)
RFNM = TFCNT

RETURN
END

A5-73
C DBPRM: INSERT DATA BASE PARAMETERS

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

* CALLED BY: INITIAL
* FUNCTION: LOAD DATA BASE NAMES, NAMENLENGTHS INTO COMMON FILES *

SUBROUTINE DBPRM

$INSERT PCOM
$INSERT PCON
$INSERT RCOM
$INSERT RTCOM
$INSERT TCOM
$INSERT CCOM

INSERT NAMES

PLLEN = 8
CALL ZMV('PLAN.DB ',PLLEN,PLNAM,32)

PTLEN = 11
CALL ZMV('PLNTEST.DB ',PTLEN,PTNAM,32)

RLLEN = 10
CALL ZMV('INSRES.DB ',RLLEN,RLNAM,32)

RTLEN = 12
CALL ZMV('INSTESTR.DB ',RTLEN,RTNAM,32)

TPLEN = 8
CALL ZMV('TAPE.DB ',TPLEN,TPNAM,32)

CLEN = 11
CALL ZMV('COMMAND.DB ',CLEN,CNNAM,32)

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

**CALLED BY: INS**

**FUNCTION:**
1. INITIALIZE IMAGE PROCESSOR, IF REQUIRED (INTIP)
2. INITIALIZE MOTORS, IF REQUIRED (INTMOT)
3. INITIALIZE MAG TAPE, IF REQUIRED (INTAP)
4. CREATE A MAIN RESULT RECORD (CRRES)
5. DO THE FOLLOWING FOR EACH TEST:
   A. GET TEST PLAN (GPTST)
   B. CREATE TEST RESULT RECORD (CRTRES)
   C. POSITION MOTORS (PMOT)
   D. PROCESS THE IMAGE (PIMG)
   E. SAVE THE TEST RESULTS (SVRES)
6. SAVE THE MAIN TEST RESULTS (SVARES)

**SUBROUTINE DINSP(IID, SERIAL, EQUIP, DEFAUL, IERR)**

```fortran
C @INSERT PCOM
@INSERT RCOM
LOGICAL EQUIP INTEGER RTKEY(RNWT)
IF (EQUIP) GOTO 10
CALL INTIP(IERR)
IF (IERR .NE. 0) RETURN
CALL INTMOT(IERR)
IF (IERR .NE. 0) RETURN
CALL INTAP(EQUIP, NOTAPE, IERR)
IF (IERR .NE. 0) RETURN
EQUIP = .TRUE.
RPF = 0
CALL CRRES(SERIAL, IID, IERR)
IF (IERR .NE. 0) RETURN
DO 30 1 = 1, PNT
CALL GPTST(I, IERR)
IF (IERR .NE. 0) RETURN
CALL CRTRES(I, RTKEY)
CALL PMOT
CALL PIMG(I, NOTAPE, IERR)
CALL SVRES(DEFALT, NOTAPE, I, RTKEY, IERR)
IF (IERR .NE. 0) RETURN
30 CONTINUE
```

AS-75
C DINSP: DO THE INSPECTION

C SAVE THE RESULTS MAIN RECORD
C------------------
CALL SVARES(IER)
RETURN
END

A5-76
C DLTC: DELETE IMAGE PROCESSOR COMMAND

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

*CALLED BY: PLAN  
*FUNCTION: 1. GET COMMAND NAME FROM OPERATOR  
*      2. FIND RECORD IN COMMAND DATA BASE  
*      3. DELETE THE COMMAND RECORD  

ARG 1: IERR   INTEGER

SUBROUTINE DLTC(IERR)
C
$INSERT CCOM
$INSERT SYSCON>PARM.K
C
INTEGER CNAME(10)
LOGICAL MORE
C
GET TEST NAME
C
10    WRITE(1,20)
20    FORMAT(1X,'COMMAND NAME?')
    READ(1,25,ERR = 10) CNAME
25    FORMAT(10A2)
C
FIND THE RECORD
C
IFLAG = FL*RET
    CALL NEXT$(CCHN,CREC,CNAME,CARR,IFLAG,$8000,0,0,0,0)
C
DELETE THE RECORD
C
IFLAG = FL*RET + FL$USE
    CALL DELET$(CCHN,CREC,CNAME,CARR,IFLAG,$9000,0,0,0,0)
    WRITE(1,10A2,CNAME
30    FORMAT(1X,10A2,' DELETED')
    CALL DMORE(MORE)
    IF (MORE) GOTO 10
    RETURN
C
8000  IF (CERR .NE. 7) GOTO 8005
    WRITE(1,8004) CNAME
8004  FORMAT(10A2,' COMMAND NOT FOUND')
    RETURN
C
8005  WRITE(1,8010) IERR,CNAME
8010  FORMAT(1X,'MIDAS ERROR ',13,' IN FINDING ',10A2)
    RETURN
C
9000  IERR = CERR
    WRITE(1,9010) IERR,CNAME
9010  FORMAT(1X,'MIDAS ERROR ',12,' IN DELETING ',10A2)
    RETURN
END

A5-77
SUBROUTINE DLTP(IERR)

INTEGER IBUF(20), DPART(10), DIIN(10), DKEY(20), BIAS, PTENP(PSZW)
LOGICAL ZCM

EQUIVALENCE (IBUF, DPART)
EQUIVALENCE (IBUF(11), DIIN)

CALL ZFIL(IBUF, 40, 0)  ; zero the pedit buffer
CALL VOPEN('DENO.SCREEN.FT', 14, ICH, IERR) ; open pedit channel
IF (IERR .NE. 0) GOTO 9000

SET UP PEDIT PARAMETERS
IERR = 0
IS = 51
IE = 52
NEWSCR = 2

GET PART # AND INSPECTION NAME FROM PEDIT
CALL PEDIT(ICH, IBUF, IS, IE, NEWSCR)

CLOSE PEDIT RECORD
CALL CLOS$A(ICH)

MAKE IBUF INTO KEY
CALL CRPKEY(DPART, 20, DIIN, 20, 0, 0, 0, 0, DKEY, 40)

GET THE RECORD
CALL NEXT$(PCHN, PREC, DKEY, PLARR, FL$RET, $9050, 0, 0, 0)
BIAS = 22
IF (PREC(BIAS) .EQ. 0) GOTO 20

DELETE SECONDARY KEYS
IFLAG = FL$RET
CALL NEXT$(PCHN, PTENP, PREC(BIAS), PLARR, IFLAG, $20, 1, 0, 0, 0)

COMPARE PREC AND PTENP
IFLAG = FL$RET + FL$USE
ISIZE = PSZW
IF (.NOT. ZCM(PTENP, ISIZE, PREC, ISIZE, ICODE)) GOTO 30

DELETE THE SECONDARY KEY
A5-78
CALL DELET$(PCHN,PREC,PREC(BIAS),PLARR,IFLAG,#20,1,0,0,0)

CALL DELET$(PCHN,PREC,DKEY,PLARR,IFLAG,$9100,0,0,0)

 FORMAT(1X,'DELETED ',20A2)

WRITE(1,80)DKEY
FORMAT(IX,'DELETED ',20A2)
RETURN

IFLAG = FL$RET
CALL DELET$(PCHN,PREC,DKEY,PLARR,IFLAG,$9100,0,0,0)
FORMAT(1X,'ERROR ',12,' IN OPENING DEMO.SCREEN.FT')
RETURN

IF (PLERR .NE. 7) GOTO 9065
WRITE(1,9060)
FORMAT(IX,'RECORD NOT FOUND')
RETURN

WRITE(1,9110)PLERR,DKEY
FORMAT(IX,'NIDAS ERROR ',12,' IN DELETEING ',20A2)
IERR = PLERR
RETURN

**DELETE RESULTS**

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

**Called from: PLAN
**Function: 1. Get part#,inspection name from operator
** 2. Find a match in data base
** 3. If correct result, delete it and the result tests

SUBROUTINE DLTR(IER)

WRITE (1,1)
FORMAT('** DELETE RESULTS IS A FORBIDDEN TRANSACTION **')
IER = 0
RETURN
END

A5-79
C DLTT: DELETE TEST PLAN  

******************************************************************************************  
# REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM  
# VERSION 1.0       JUNE 1,1980  
# BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY  
#  
#***************************************************************************  
#CALLED BY: PLAN  
#FUNCTION: 1. GET TEST NAME FROM OPERATOR  
#          2. FIND RECORD IN TEST DATA BASE  
#          3. DELETE THE TEST RECORD  
#  
#ARG 1: IERR   INTEGER  
C SUBROUTINE DLTT(IERR)  
C $INSERT PTCOM  
$INSERT SYS.COM>PARM.K  
C INTEGER PTNAME(10)  
LOGICAL MORE  
C GET TEST NAME  
10 WRITE(1,20)  
20 FORMAT(IX,'TEST NAME?')  
READ(1,25,ERR = 10) PTNAME  
25 FORMAT(10A2)  
C FIND THE RECORD  
C IFLAG = FL$RET  
CALL NEXT$(PTCHN,PTREC,PTNAME,PTARR,IFLAG,$8000,0,0,0)  
C DELETE THE RECORD  
C IFLAG = FL$RET + FL$USE  
CALL DELET$(PTCHN,PTREC,PTNAME,PTARR,IFLAG,$9000,0,0,0)  
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  
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  

A5-80
C DMORE: DO YOU WANT TO DO MORE?

C OMORE: DO YOU WANT TO DO MORE?

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

*GENERAL PURPOSE SUBROUTINE TO SEE IF OPERATOR NEEDS TO DO *
* A FUNCTION AGAIN *

SUBROUTINE DMORE(MORE)

$INSERT SYS.COM>A$KEYS

C LOGICAL MORE, YSN0$A

MORE = YSN0$A('More',4,A$DNO)
RETURN
END

AS-81
C DSPLD: DISPLAY OR PRINT PLAN OR RESULT DATA.

*******************************************************************************

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

CALLED FROM: PLAN
FUNCTION:  1. GET DISPLAY FUNCTION FROM OPERATOR (GDFNCT)
2. CALL ONE OF FOLLOWING SUBROUTINES
   DSPP - DISPLAY A PLAN
   DSPPT - DISPLAY A TEST
   DSPR - DISPLAY MAIN RESULTS
   DSPRT - DISPLAY TEST RESULTS
   DSPT - DISPLAY TAPE RECORD
   LISTIX- LIST RECORD BY INDEX NO.
3. IF MAILSTOP GIVEN, SPOOL INFO TO LINE PRINTER

ARG 1: IERR    INTEGER
SUBROUTINE DSPLD(IERR)

INTEGER SPFNAR(7), SPCHN

GET DISPLAY FUNCTION
10   CALL GDFNCT(MENU,SPCHN,SPFNAR)
   IF (MENU .EQ. 0) RETURN
   GOTO (100,200,300,400,500,600,700,800), MENU

C   DISPLAY MAIN PLAN
100   CALL DSPP(SPCHN)
   GOTO 8000

C   DISPLAY PLAN TEST
200   CALL DSPPT(SPCHN)
   GOTO 8000

C   DISPLAY MAIN RESULTS
300   CALL DSPR(SPCHN)
   GOTO 8000

C   DISPLAY TEST RESULTS
400   CALL DSPRT(SPCHN)
   GOTO 8000

C   DISPLAY TAPE RECORD
500   CALL DSPT(SPCHN)
   GOTO 8000

C   LIST INDEX
600   CALL LISTIX(SPCHN)
   GOTO 8000

C   SOME OTHER FUNCTION
700   GOTO 8000

C   SOME OTHER FUNCTION
800   GOTO 8000

C   PRINT RESULTS?    A5-82
8000 IF (SPCHN .EQ. 0) GOTO 10
     CALL SPOL1T(SPCHN,SPFNAM)
     GOTO 10
END
C DSPP: DISPLAY MAIN PLAN

******************************************************************************
* REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM *
* VERSION 1.0  JUNE 1,1980 *
* BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY *
******************************************************************************
*CALLED BY: DSPLD *
*FUNCTION: 1. GET INSPECTION NAME FROM OPERATOR (PEDIT) *
* 2. FIND RECORD IN DATA BASE *
* 3. PRINT/DISPLAY THE RECORD *
* 4. IF OPERATOR WANTS TO SEE A TEST: *
*    A. FIND THE RECORD *
*    B. DISPLAY IT *
*
SUBROUTINE DSPP(REPCHN)
C
ARG.1: REPCHN INTEGER, REPORT CHANNEL

$INSERT PCOM
$INSERT PTCOM
$INSERT SYSCOM>PARM,K
$INSERT SYSCOM>ANKEYS

INTEGER REPCHN,PKEY(20)

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

GET PART NUMBER, INSPECTION NAME
N = -1
IS = 49
IE = 52
NEWSCR = 2  */ DO NOT ERASE SCREEN TO START
CALL ZFIL(PREC,PSZB,0)
CALL PEDIT(IPCHN,PREC,IS,IE,NEWSCR)

CALL CRPKEY(PPN,20,PIN,20,0,0,0,0,PKEY,40)  /* CREATE KEYWORD
NCHAR = LSIZE(PKEY,40)  /* GET LENGTH OF KEYWORD
IFLAG = FL*RET + FL*BIT

FIND THE RECORD
10 CALL NEXT$(PCHN,PREC,PKEY,PLARR,IFLAG,$9100,0,0,0,NCHAR)

PRINT/DISPLAY THE RECORD
C
IS = 49
IE = 64
NEWSCR = 2
WRITE (1,11)
FORMAT(/)
11 CALL RPTGEN(REPCHN,IPCHN,PREC,LINES,IS,IE,NEWSCR)
20 IF (REPCHN .NE. 0) GOTO 9900
IFLAG = FL*RET + FL*USE + FL*BIT

CALL READN(
- '*' = QUIT, 0 = MORE MATCHES, N = SEE NTH TEST: ',47,N)
IF (N) 9900,10,25

FIND A TEST RECORD
25 IBIAS = 22 + (N-1)*PNWT
IFLAG = FL*RET
30 CALL NEXT$(PCHN,PREC,PREC(IBIAS),PTARR,IFLAG,$9200,0,0,0,0)

PRINT/DISPLAY THE TEST PLAN
C
IS = 74
A5-84
IE = 99
WRITE (1,11)
CALL RPTGEN(REPCHN,IPCHN,PTREC,LINES,IS,IE,NEWSCR)
GOTO 20

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)PERR,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 TNOU(' NOT FOUND',10)
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(' NOT FOUND',10)
GOTO 9900
9240 CALL RECYCL
GOTO 30

C
C EXIT
C
9900 CALL CLOS(A(IPCHN))
RETURN
END
SUBROUTINE DSPPT(SPCHN)

$INSERT PTCOM
$INSERT SYSCOM>PARM,K
$INSERT SYSCOM>A$KEYS

OPEN SCREEN TEMPLATE
CALL VOPENS('DERO.SCREEN.FT',14,1,IPCHN,IERR)
IF (IERR.EQ.0) GOTO 9000

GET THE TEST NAME
IS = 74
IE = 75
NEWSCR = 2  /* DO NOT ERASE SCREEN TO START
CALL ZFIL(PTREC,PTSZB,0)
CALL PEDIT(IPCHN,PTREC,IS,IE,NEWSCR)
WRITE (1,1)
FORMAT(/)

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

FIND THE TEST RECORD
CALL NEXT$(PTCHN,PTREC,PTREC,PTARR,IFLAG,$9100,0,0,NEWSCR)
PRINT/DISPLAY THE RECORD
IS = 73
IE = 99
WRITE (1,11)
FORMAT(/)
CALL RPTGEN(SPCHN,IPCHN,PTREC,LINES,IS,IE,2)
IFLAG = FL$BIT + FL$RET + FL$USE

ANY MORE MATCHES?
J = 1
IF (SPCHN.EQ.0) CALL PAUS(J)
IF (J.EQ.0) GOTO 9900
GOTO 10

ERRORS
WRITE(1,9010)IERR
9010 FORMAT(IX,'ERROR',I3,' IN OPENING DEMO.SCREEN.FT')
RETURN

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

WRITE(1,9160)PTERR,PTNR
9160 FORMAT(1X,'NIOAS ERROR',I3,' IN FINDING ',20A2)
A5-86
C DSPPT: DISPLAY ONE TEST PLAN

C 9900 CALL CLOSEA(IPCHN)
RETURN
END
C DSPRI DISPLAY RESULTS

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

*CALLED BY: DSPLD AND RETR
*FUNCTION: 1. GET PART#, INSP NAME, SERIAL#, INSIP 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 RCON
C INSERT RTCON
C INSERT SYSCON>PARM.K
C INSERT SYSCON>ASKEYS

INTEGER RAPRI /* RESULTS PRIMARY KEY LENGTH WORDS
PARAMETER RAPRI = 20
INTEGER RKEY(RAPRI)
INTEGER REPCHN,PN(10),IN(10),SNO(10),IID(5),IBUF(35)
EQUIVALENCE (IBUF,110) /* INSPERS ID
EQUIVALENCE (IBUF(6),PN) /* PART NUMBER
EQUIVALENCE (IBUF(16),IN) /* INSP 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#, INSIP ID

IS = 0
IE = 5
NEWSCR = 2 /* DO NOT ERASE SCREEN TO START
CALL ZFIL(IBUF,70,0)
CALL PEDIT(IPCHN,IBUF,IS,IE,NEWSCR)

NBYTS = RAPRI*2
CALL CRPKEY(PN,20,IN,20,SNO,20,0,0,RKEY,NBYTS) /* CREATE KEYWORD
NCHAR = LSIZE(RKEY,40) /* GET LENGTH OF KEYWORD
IFLAG = FLRET + FLBIT

C FIND A MATCH
C
CALL NEXTS(RLCHN,RREC,RKEY,RLARR,IFLAG,9100,0,0,0,NCHAR)

C PRINT/DISPLAY THE RECORD

IS = 225
IE = 244
NEWSCR = 2
WRITE (1,11)
11 FORMAT(//)
CALL RPTGEN(REPCHN,IPCHN,RREC,LINES,IS,IE,NEWSCR)
20 IF (REPCHN .NE. 0) GOTO 9000
IFLAG = FLRET + FLUSE + FLBIT
WRITE (1,11)
CALL READN('-1 = QUIT, 0 = MORE MATCHES, N = SEE NTH TEST',47,N)
IF (N) 9900,10,25

C A5-88
C DSPR: DISPLAY RESULTS

C FIND A TEST RECORD
C
25   IBIAS = 43 + (N-1)*RNWT
     IFLAG = FL$RET
30   CALL NEXT$(RTCHN,RTREC,RREC(IBIAS),RTARR,IFLAG,$9200,0,0,0)
C
C PRINT/DISPLAY THE TEST RESULT
C
IS = 250
IE = 273
WRITE (1,11)
CALL RPTGEN(REPCHN,IPCHN,RTREC, LINES, IS, IE, NEWSCR)
GOTO 20
C
C ERRORS
C
9000   WRITE(1,9010)IERR
9010   FORMAT(IX,'ERROR ','13,' IN OPENING DEMO.SCREEN.FT')
       RETURN
C
9100   IF (RLERR .NE. 7) GOTO 9120
9110   WRITE(1,9110)RKEY
9120   IF (RLERR .EQ. 22 .OR. RLERR .EQ. 24) GOTO 9140
9130   FORMAT(IX,'MIDAS ERROR ','13,' IN FINDING ',20A2)
9140   CALL RECYCL
       GOTO 10
C
9200   IF (RTERR .NE. 7) GOTO 9220
9201   FORMAT(25A2,' NOT FOUND')
9220   IF (RTERR .EQ. 22 .OR. RTERR .EQ. 24) GOTO 9240
9230   FORMAT(IX,'MIDAS ERROR ',13,' IN FINDING ',25A2)
9240   CALL RECYCL
       GOTO 30
C
C EXIT
C
9900   CALL CLOS$A(IPCHN)
       RETURN
END

A5-89
C DSPRT: DISPLAY RESULTS OF ONE TEST

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

* CALLED BY: DSPLD AND RETR
* FUNCTION: 1. GET PART#, INSPECTION NAME, SERIAL#, INSPECTION ID FROM OPERATOR
* 2. CREATE PLAN TEST KEYWORD
* 3. GET THE RECORD
* 4. DISPLAY THE RECORD

ARG 1: SPOOL CHANNEL SPCHN INTEGER

SUBROUTINE DSPRT(SPCHN)

$INSERT RTCOM
$INSERT SYS.COM>PARM.K
$INSERT SYS.COM>AK.EYS

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

OPEN SCREEN TEMPLATE

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

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

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

FIND THE TEST RECORD

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

PRINT/DISPLAY THE RECORD

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

ANY MORE MATCHES?

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

ERRORS

9000 WRITE(1,9010)IERR

A5-90
C OSPRT: DISPLAY RESULTS OF ONE TEST

9010 FORMAT(IX,'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(9,9160)RTERR,KRTERR
  9160 FORMAT(IX,'MIDAS ERROR ',I3,' IN FINDING ',20A2)

C 9900 CALL CLOST(IPCHN)
  RETURN
END
SUBROUTINE DSPT(SPCHN)

INTEGER IBUF(26),KEY(TNWT)
LOGICAL MORE

EQUIVALENCE (IBUFINDEX) /\ AIDAS INDEX \ /
EQUIVALENCE (IBUF(2),KEY) /\ MIDAS KEYWORD \ /

OPEN SCREEN FILE

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

IS = 100
IE = 105
NEWSCR = 2       /*DO NOT ERASE SCREEN TO START

GET INDEX AND PRIMARY OR SECONDARY KEYWORD
PRIMARY KEY=TNAM, SECONDARY KEYS=IMAGE NAMES

CALL PEDIT(ICHIBUF,ISIENEVSCR)

NBYTS = TNWT*2
LEN = LSIZE(KEY,NBYTS) /*FIND SIZE OF KEYWORD FOR MIDAS

FIND THE RECORD

IFLAG = FL$RET + FL$BIT
INDEX = INDEX     /*CHECKOUT
CALL NEXT$(TCHN,TREC,KEY,TPARRIFLAG,$9100INDEXD,D,LEN)

DISPLAY THE RECORD

IS = 106
IE = 160
WRITE (1,11)
FORMAT(/)
CALL RPTGEN(SPCHN,ICH,TREC,LINES,IS,IE,2)
CALL DMORE(MORE)
IF (MORE) GOTO 10

CLOSE THE SCREEN FILE

CALL CLOS#4(ICH)
RETURN

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

IF (TPERR .NE. 7) GOTO 9110
WRITE(1,9101) KEY
9101 FORMAT('NOT FOUND: ',25A2)
GOTO 100

A5-92
FREECH: RETURNS A NUMBER OF AVAILABLE PRIMOS CHANNELS

CDSPT: DISPLAY TAPE DATA

C
FREECH: RETURNS A NUMBER OF AVAILABLE PRIMOS CHANNELS

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

SUBROUTINE FREECH(NUM,ICHN)

C
GENERAL PURPOSE SUBROUTINE TO FIND ONE OR MORE AVAILABLE CHANNELS

NUM = NUMBER OF FREE CHANNELS REQUESTED
ICHN = AN ARRAY SIZE NUM FOR CHANNELS AVAILABLE

$INSERT SYSCOM>KEYS.F
$INSERT SYSCOM>ERRD.F

INTEGER ICHN(1) /* DUMMY DIMENSION
LOGICAL UNITSA

I = 1

DO 100 ICH=1,63
   IF (UNITSA(I)) GOTO 100
   ICHN(I) = ICH
   I = I + 1
   IF (I .GT. NUM) RETURN
100 CONTINUE

C ALL UNITS IN USE: ERROR 41
IER = E*FUIU
RETURN

END
C GDFNCT: GET DISPLAY FUNCTION

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

CALLED BY: DSPLD
FUNCTION: 1. GET FUNCTION FROM OPERATOR
# 2. GET MAILSTOP FROM OPERATOR IF PRINTOUT DESIRED
# 3. IF PRINTOUT DESIRED, OPEN A SPOOL CHANNEL

ARG 1: MENU INTEGER
ARG 2: SPCHN SPOOL CHANNEL NUMBER INTEGER
ARG 3: SPFNAM SPOOL FILE NAME STRING, INTEGER ARRAY(7)

SUBROUTINE GDFNCT(MENU,SPCHN,SPFNAM)

$INSERT SYSCOM>A$KEYS

INTEGER SPCHN,SPFNAM(7)
SPCHN = 0

CALL ZFIL(SPFNAM,14,0)

OPEN A CHANNEL TO SCREEN TEMPLATE

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

GET MENU AND MAIL STOP FROM SCREEN

IS = 175
IE = 184
NEWSCR = 2  /* DO NOT ERASE SCREEN TO START

10 CALL PEDIT(ICH,SPFNAM,IS,IE,NEWSCR)
IF (SPFNAM(1) .GE. 0 .AND. SPFNAM(1) .LE. 8) GOTO 30
WRITE(1,20)
20 FORMAT(IX,'MENU NOT IN RANGE')
   GOTO 10

CLOSE THE SCREEN FILE

30 CALL CLOS$A(ICH)

MENU = SPFNAM(1)
 IF (SPFNAM(5) .EQ. 0) RETURN

CALL ZMVD('MAILSTOP.',SPFNAM,9)

CALL PACK(SPFNAM,14)

OPEN A CHANNEL TO SPOOL FILE FOR WRITING

CALL VOPEN(SPFNAM,14,2,SPCHN,IERR)
IF (IERR .NE. 0) GOTO 9100
RETURN

ERROR MESSAGES

9000 WRITE(1,9010)IERR
9010 FORMAT(IX,'ERROR ',I3,' IN OPENING DEMO.SCREEN.FT')
   RETURN
9100 WRITE(1,9110)IERR,SPFNAM
9110 FORMAT(IX,'ERROR ',I3,' IN OPENING ',A2)
   RETURN
END

A5-94
C GFUNCT: GET FUNCTION, MAIN MENU

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

* CALLED BY: AI
* FUNCTION: GET MENU FUNCTION - INSPECT, PLAN, QUERY RESULTS, EXIT

SUBROUTINE GFUNCT(MENU)

WRITE(1,20)
20 FORMAT(//, 'MAIN MENU: ',
       '0 EXIT',
       '1 PERFORM AN INSPECTION',
       '2 PLANNING FUNCTIONS',
       '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

******************************************************************************
* REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM
* VERSION 1.0       JUNE 1, 1990
* BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY
******************************************************************************

*CALLED BY:   INS
*FUNCTION: GET INSPECTION INFORMATION WITH SCREEN EDITOR

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

DIMENSION IBUF(36)

MOVE DATA TO EDIT BLOCK

CALL MSUB4A(IID,10,1,1,IBUF,70,1,10)
CALL MSUB4A(PART,20,1,20,IBUF,70,11,30)
CALL MSUB4A(TESTID,20,1,20,IBUF,70,31,50)
CALL MSUB4A(SERIAL,20,1,20,IBUF,70,51,70)

OPEN SCREEN FILE

CALL VOPEN$("DEMO.SCREEN.FT",14,ICHN,IERR)
IF (IERR .EQ. 0) GOTO 20
WRITE(1,10)IERR
RETURN

10 FORMAT(IX,'ERROR ',I2,' IN OPENING DEMO.SCREEN.FT')
RETURN

C SET UP PEDIT PARAMETERS

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

C GET OPERATOR'S INPUT

C MAKE SURE SERIAL# NE 0

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

C LOAD INPUT DATA INTO PARAMETER VARIABLES

40 CALL MSUB4A(IBUF,70,1,10,IBUF,70,1,10)
CALL MSUB4A(IBUF,70,11,30,IBUF,70,1,20)
CALL MSUB4A(IBUF,70,31,50,TESTID,20,1,20)
CALL MSUB4A(IBUF,70,51,70,SERIAL,20,1,20)

C CLOSE SCREEN FILE

CALL CLOS$A(ICHN)
RETURN
END

A5-96
SUBROUTINE GPFNCT(MENU)

WRITE (1,10)
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 ' 10 DELETE PLAN',/
 11 ' 11 DELETE TEST PROCEDURE',/
 12 ' 12 DELETE TEST RESULTS',/
 13 ' 13 DELETE IMAGE COMMAND',/
 14 ' 14 DISPLAY DATA',/
CALL READN(?' ',2,MENU)
IF (MENU .LT. 1) RETURN
IF (MENU .GT. 14) GOTO 5
RETURN
END
SUBROUTINE GPLNR(PART,TESTID,DEFALT,IER)

INTEGER KEY(15),DPLAN(15)
LOGICAL DEFALT,YES,YSNO

CALL CRPKEY(PART,20,TESTID,20,0,0,0,0,KEY,30) /*CREATE KEYWORD
WRITE(1,1000)KEY
1000 FORMAT(1X,'KEY: ',15A2)
DEFALT = .FALSE.
10 CALL ZFIL(PREC,PSZB,0) /*INITIALIZE PREC
IFLAG = .FALSE.
WRITE(1,1000)KEY
C
CALL TOUTP(KEY,30)
CALL NEXT*(CHNPRECKEYPLARRIFLAG,$9000,O,O,,O)
WRITE(1,1001)
1001 FORMAT(1X,'NO ERROR FROM NEXT*')
IERR = 0
RETURN
C
9000 IERR = PLERR
WRITE(1,1002)IERR
1002 FORMAT(1X,'ERROR ',I2)
IF(PLERR.EQ.7) GOTO 9100
IF(PLERR.NE.22 .OR. PLERR .NE. 24) GOTO 9900
CALL RECYCL
GOTO 10
C
Record does not exist
9100 YES = YSN0
ENTRY('NOT FOUND. USE DEFAULT PLAN',27.A&DNO)
IF (YES) GOTO 9200
IERR = -1
RETURN
9200 CALL ZNVD('DEFAULT PLAN/DEFAULT ' ,KEY,30)
DEFALT = .TRUE.
GOTO 10
9900 WRITE(1,9910)PLERR,KEY
9910 FORMAT(1X,'MIDAS ERROR ',13,' IN LOCATING ',15A2)
RETURN
END
REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM
VERSION 1.0 JUNE 1, 1980
BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY

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

SUBROUTINE GPTST(ITEST, IERR)

$INSERT PCOM
$INSERT PTCON
$INSERT SYSCOM>PARM.K

EQUIVALENCE (PDALPDA), (PDB,LPDB), (PDC,LPDC), (PDD,LPDD)

IBIAS = 22 + (ITEST-1)*PNWT
IFLAG = FLTRET
CALL NEXT$(PTCHN,PTREC,PREC(IBIAS),PTARR,IFLAG,$9000,0,0,0,0)

DISPLAY DESCRIPTION TEXT

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

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)PINM,PINAM
9901 FORMAT(1X,'TEST ',10A2,' NOT FOUND IN DATA BASE ',16A2)
   RETURN
9905 WRITE(1,9910)PTERR,PINM,PINAM
9910 FORMAT(1X,'MIDAS ERROR ',13,' IN TEST ',16A2/1X,
   C'FROM DATA BASE ',16A2)
   RETURN
END

A5-99
C GRFNCT: GET RETRIEVAL FUNCTION

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

* CALLED BY: RETR
* FUNCTION: GET RETRIEVAL FUNCTION FROM OPERATOR

ARG 1: MENU INTEGER

SUBROUTINE GRFNCT(MENU)

WRITE(1,10)
10 FORMAT(1X,'HISTORY MENU',/,
      '0 EXIT',/,
      '1 RETRIEVE AN IMAGE FOR DISPLAY',/,
      '2 RECREATE ONE TEST RESULT',/,
      '3 RECREATE AN INSPECTION',/,
      '4 DISPLAY MAIN RESULTS',/,
      '5 DISPLAY TEST RESULTS',/,
      '6 DISPLAY TAPE RECORD',/,
      '7 LIST INDECIES')
READ(1,20,ERR = 30)MENU
20 FORMAT(1I)
    IF (MENU .LT. 0 .OR. MENU .GT. 7) GOTO 5
30 RETURN
END

A5-100
CINITIAL: SET UP TERMINAL CODES AND OPEN DATA BASE FILES

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

* CALLED BY: AI
* FUNCTION: 1. SET UP DATA BASE PARAMETERS (DBPRM)
*          2. OPEN DATA BASE FILES

SUBROUTINE INITAL(IERR)

INTEGER TYPE

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

C GET DATA BASE PARAMETERS
CALL DBPRM

C OPEN PLAN DATA BASE FILES
CALL VOPENS(PLNAM,PLLEN,I,PCHNIERR)
IF (IERR .EQ. 0) GOTO 30
WRITE(1,20)IERR,PLNAM
20 FORMAT(IX,'ERROR ',12,' IN OPENING ', 16A2)
RETURN
30 CALL VOPENS(PTNAM,PTLEN,I,PTCHNIERR)
IF (IERR .EQ. 0) GOTO 40
WRITE(1,20)IERR,PTNAM
CALL CLOALL
RETURN

C OPEN RESULTS DATA BASE FILES
CALL VOPENS(RLNAM,RLLEN,IRLCHNIERR)
IF (IERR .EQ. 0) GOTO 50
WRITE(1,20)IERR,RLNAM
CALL CLOALL
RETURN

C OPEN TAPE DATA BASE FILE
CALL VOPENS(TPNAM,TPLEN,ITCHNIERR)
IF (IERR .EQ. 0) GOTO 70
WRITE(1,20)IERR,TPNAM
CALL CLOALL

A5-101
C INITIAL: SET UP TERMINAL CODES AND OPEN D PAGE 0002

RETURN

C OPEN COMMAND DATA BASE FILE

70 CALL UOPEN$(CNNAM,CLEN,1,CCHN,IERR)
    IF (IERR .EQ. 0) GOTO 80
    WRITE(1,20)IERR,CNNAM
    CALL CLOALL
80 RETURN
END

C INSP: PERFORM AN INSPECTION

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

*CALLED BY: AI
*FUNCTION: 1. GET INSPECTOR ID, PART#, TEST, SERIAL# (GIIN)
*         2. GET THE PLAN (GPLNM)
*         3. DO THE INSPECTION (DINSP)

SUBROUTINE INSP(EQUIP,IERR)

INTEGER IID(5), PART(10), TESTID(10), SERIAL(10)
LOGICAL DEFAULT, EQUIP, MORE

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

10 CALL GIIN(IID,PART,TESTID,SERIAL,IERR)
    IF (IERR .NE. 0) RETURN

GET PLAN RECORD

20 CALL GPLNM(PART,TESTID,DEFAULT,IERR)
    IF (IERR .NE. 0) RETURN

DO THE INSPECTION

30 CALL DINSP(IID,SERIAL,EQUIP,DEFAULT,IERR)
    IF (IERR .NE. 0) RETURN

DO MORE?

100 CALL DMORE(MORE)
    IF (MORE) GOTO 10
200 RETURN
END

A5-102
**C INTAP: INITIALIZE TAPE**

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

* * *

**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** EQUIP = EQUIP INITIALIZED FLAG
**C** NOTAPE = TAPE INUSE FLAG, RETURNED
**C** IERR = ERROR RETURNED

**C** LOGICAL YSNO$A,NEW,CSTR$A,NOTAPE,YES,EQUIP,ITAPE,NTAP
**C** INTEGER STATUS(3),NAME(10)

**C** IF (.NOT. EQUIP) GOTO 10 /* IS EQUIPMENT INITIALIZED?
**C** YES = YSNO$A(‘DOES THIS INSPECTION REQUIRE MAG TAPE’,37,A$DNO)
**C** IF (YES) NOTAPE = .FALSE.

**C** IF (.NOT. YES .AND. NOTAPE) RETURN /* NOTAPE NOTAPE
**C** IF (.NOT. NOTAPE .AND. YES) GOTO 5 /* TAPE TAPE
**C** IF (NOTAPE .AND. YES) GOTO 2 /* NOTAPE TAPE
**C** IF (.NOT.NOTAPE .AND. .NOT. YES) GOTO 12 /*TAPE NOTAPE

**C** NTAP = YSNO$A(‘IS CORRECT TAPE MOUNTED AND READY’,33,A$DNO)
**C** IF (.NOT. NTAP) GOTO 1
**C** IF (TNAM(1).EQ.0) GOTO 15
**C** GOTO 5

**C** IF (TNFL - TFCNT) RETURN /* IS TAPE AT CORRECT POSITION?
**C** WRITE(1,6)TNAM
**C** 6 FORMAT(IX,’CURRENT TAPE ON LINE: ’,10A2)
**C** ITAPE = YSNO$A(‘IS THIS CORRECT’ ,15,A$DNO)
**C** IF (ITAPE) GOTO 8

**C** WRITE(1,7)
**C** 7 FORMAT(IX,’PLEASE MOUNT NEW TAPE: HIT ANY KEY WHEN READY’)
**C** CALL PAUS(J)
**C** GOTO 14

**C** N = TNFL - TFCNT + 1
**C** GOTO 70

**C** CALL ZFIL(TREC,TSZB,0)
**C** CALL TONL

**C** YES = YSNO$A(‘DOES THIS INSPECTION REQUIRE MAG TAPE’,37,A$DNO)

**C** IF (.NOT. YES) NOTAPE = .TRUE.
**C** IF (.NOT. YES) RETURN
**C** NOTAPE = .FALSE.

**C** GET TAPE NAME
**C** ---------------
**C** CALL ZFIL(TREC,TSZB,0)
**C** CALL TONL
**C** YES = YSNO$A(‘TAPE NAME (ENTER “NEW” IF A NEW TAPE) ’,38)
**C** CALL READL(TNAM,NCHAR,20)
**C** NEW = CSTR$A(TNAM,NCHAR,’NEW’,3)
**C** IF (.NOT.NEW) GOTO 30

**C** NEW TAPE, ADD IT TO THE DATA-BASE

A5-103
C INTAP: INITIALIZE TAPE  PAGE 0002

C-----------------------------------
C EXISTING TAPE - FIND IN DATA BASE
C-----------------------------------
C CHECK THE TAPE FOR ONLINE
C-----------------------------------
C REWIND TAPE
C-----------------------------------
C MOVE TO CORRECT PLACE ON TAPE
C-----------------------------------
C RECORD NOT FOUND
C-----------------------------------
C
SUBROUTINE INTIP(IERR)

INTEGER ZERO
INTEGER DIGIT(40)
COMMON /OREZ/ ZERO(B191)
DATA DIGIT/"DIGITIZE">',33*"/

IS THE IMAGE PROCESSOR ON?
CALL TONL
CALL TNOUA('POWER ON THE MODEL 70 & HIT A KEY ',34)
CALL PAUS(J)
ZERO(3029)=50
ZERO(3053)=0
ZERO(3054)=0

LOAD COMMON
CALL CALDR

CLEAN THE DIRECTORY AND INITIALIZE M70
CALL ICLEAN

TURN ON THE CAMERA
CALL CMDM70(DIGIT, IERR)

RETURN
END
SUBROUTINE INTMOT(IER)

LOGICAL YSNOSA
PARAMETER NMOT = 5
INTEGER MOTOR(NMOT), STATUS

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

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(IGO)
   GO TO 10

ASK OPERATOR IF STANDARDIZATION IS NECESSARY
20 IF (.NOT.YSNOSA(1
   'HAS COUNTER BEEN FIXED SINCE LAST POWER-UP',42,A&DNO))
     2 GOTO 80

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(IGO)
   GO TO 40

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

DO 100 N=1,NMOT
   IF (MOTOR(N) .EQ. 0) GOTO 100
   NMOT = N - 1
   CALL POSINT(MOTOR,IER)
   IF (IER .NE. 0) RETURN
   CONTINUE
100 RETURN

END
C LISTIX: LIST KEYNAMES BY INDEX

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

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

**ARG 1: SPOOL CHANNEL

SUBROUTINE LISTIX(SPCHN)

$INSERT PCOM
$INSERT PTCOM
$INSERT RCOM
$INSERT RTCON
$INSERT SYSCOM>PARAM.K

INTEGER RAPRI  /* RESULTS MAIN PRIMARY KEYWORD SIZE
PARAMETER RAPRI = 20
INTEGER TMREC(TSZW),TMARRY(14),TMERR,TMKEY(TNWT),TMCHR,TMDS
EQUIVALENCE(TMARRY,TMERR)

C GET MENU
C
10 WRITE(1,20)
20 FORMAT(1X,'LIST INDICIES MENU'/
  ,1X,'00 EXIT'/
  ,1X,'10 PLANS BY NAME'/
  ,1X,'11 PLANS BY TEST NAME'/
  ,1X,'20 TEST NAMES'/
  ,1X,'30 RESULTS BY NAME'/
  ,1X,'31 RESULTS BY TEST NAME'/
  ,1X,'40 TEST RESULTS'/
  ,1X,'50 TAPE NAMES'/
  ,1X,'51 IMAGE NAME'/)
READ(1,30,ERR=210)N,I
30 FORMAT(211)
33 WRITE (1,33)
3 FORMAT(/)
C CHECK FOR VALIDITY OF MENU SELECTION
C
IF (N .GT. 0 .AND. N .LE. 5) GOTO 40
IF (N .LE. 0) RETURN
GOTO 10
40 IF (I .GE. 0 .AND. I .LE. 1) GOTO 50
GOTO 10
C
50 IFLAG = FL*RET + FL*UKY + FL*FST
C SET MIDAS CHANNEL,NCHAR IN KEYWORD
GOTO (110,120,130,140,150),N
C PLAN
C
110 MCHN = PCHN
IF (I .EQ. 0) TMCHR = 30
IF (I .EQ. 1) TMCHR = PNWT#2
GOTO 160
C PLAN TEST
120 MCHN = PICHN
TMCHR = 20
A5-107
C LISTIX: LIST KEYNAMES BY INDEX

GOTO 160

C RESULTS
130 MCHN = RLCHN
IF (I .EQ. 0) TMCHR = RAMPR#2
IF (I .EQ. 1) TMCHR = RNWT#2
GOTO 160

C RESULTS TEST
140 MCHN = RICHN
TMCHR = RNWT#2
GOTO 160

C TAPE
150 MCHN = TCHN
IF (I .EQ. 0) TMCHR = 20
IF (I .EQ. 1) TMCHR = TNWT#2

C CALCULATE #WORDS IN KEYWORD
160 TMWDS = TMCHR/2

C SPOOL LOOP
170 DO 200 J = 1,22

C GET DATA
180 CALL NEXTS(MCHN,TMREC,TMKEY,TMARRAY,IFLAG,$9000,1,0,D,0)
IFLAG = FL$USE + FL$RET + FL$PLW + FL$UKY
IF (SPCHN .NE. 0) GOTO 190

C OUTPUT LINE TO TERMINAL
CALL TOUT(TIME,Y,TIMCHR)
GOTO 200

C OUTPUT LINE TO SPOOL FILE
190 CALL WTLINS(SPCHN,TMKEY,TMWDS,IERR)
IF (IERR .NE. 0) GOTO 9100

200 CONTINUE

C END OF SPOOL LOOP - GO BACK AND GET MORE

C IF (SPCHN .NE. 0) GOTO 170
CALL PAUS(J)
IF (J .NE. 0) GOTO 170

210 RETURN

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(IX,'MIDAS ERROR',13,' IN FINDING ',25A2)
RETURN

9100 WRITE (1,9110)IERR,SPCHN,TMKEY
9110 FORMAT(IX,'ERROR',13,' IN WRITING TO SPCHN ',13,' KEY=',25A2)
RETURN

C END
C MODFYC: MODIFY IMAGE PROCESSOR COMMAND

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

*CALLED FROM: PLAN *                                               
*FUNCTION: 1. GET COMMAND NAME FROM OPERATOR *                   
*  2. GET COMMAND RECORD FROM DATA BASE *                       
*  3. EDIT THE RECORD *                                         
*  4. STORE THE NEW RECORD IN DATA BASE *                       

SUBROUTINE MODFYC(IER)

INTEGER NAME(16)
DATA NAME /'DEMO.SCREEN.FT'/

CALL VOPEN$(NAME,32,1,IFCH,IER)    /* OPEN FOR READ 
IF (IER .NE. 0) GOTO 9000

IS = 318
IE = 319
NEWSCR = 2  /* DO NOT ERASE THE SCREEN TO START

CALL ZFIL(CRECCSZB,0)

CALL PEDIT(IFCHCRECIS,IENEWSCR)

CALL LOCK$(CCHN,CREC,CNAMCARRFL$RET,$9100,0,0,0,0)

CALL PEDIT(IFCHCRECIS,IENEWSCR)

CALL UPDAT$(CCHNCRECCNAMCARRFL$USE,$9200,,0,0,0,0)

IF (YSNO$A('More',4,ASDNO)) GOTO 20

CALL CLOSSA(IFCH)
RETURN

WRITE (1,9001) IER,NAME
9001 FORMAT('ERROR ',13,' OPENING FILE ',16A2)
RETURN

IF (CERR .NE. 7) GOTO 9110
WRITE (1,9101) CNAM
9101 FORMAT(1DA2,' NOT FOUND')
C MDFYC: MODIFY IMAGE PROCESSOR COMMAND  PAGE 0002

9110  GOTO 300
9111  IF (CERR .EQ. 22) GOTO 30
9112  IF (CERR .EQ. 24) GOTO 30
9113  WRITE (1,9111) PTERR,PTNM
9114  FORMAT('AIDAS LOCK ERROR =',13,' KEY = ',15A2)
9115  GOTO 400
C
C
9200  WRITE (1,9101) CERR,CNNAM
9201  FORMAT('AIDAS UPDAT ERROR =',13,' KEY = ',15A2)
9202  GOTO 400
C
END

C MDFYP: MODIFY PLAN  PAGE 0001

MDFYP: MODIFY PLAN

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

SUBROUTINE MDFYP(IER)

*CALLED FROM: PLAN *
*FUNCTION: 1. ASK OPERATOR FOR PARTN, INSPECTION NAME *
*          2. GET THE PLAN RECORD *
*          3. ALLOW OPERATOR TO MODIFY THE PLAN RECORD *
*          4. DELETE OLD AND INSERT NEW SECONDARY KEYS *

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

A5-110
C MDFYT: MODIFY TEST PROCEDURE

C MDFYT: MODIFY TEST PROCEDURE

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

#CALLED FROM: PLAN
#FUNCTION: 1. GET TEST NAME FROM OPERATOR
#  2. GET TEST RECORD FROM DATA BASE
#  3. EDIT THE RECORD
#  4. STORE THE NEW RECORD IN DATA BASE

SUBROUTINE MDFYT(IER)

$INSERT PTCOA
$INSERT SYSCOM>A$KEYS-
$INSERT SYSCOM>PARM.K

INTEGER NAME(16)
INTEGER KEYO(10)  /* PRIMARY KEY

DATA NAME /'DEMO.SCREEN.FT /

Open a channel to the screen template file
--------------------------------------------
CALL VOPENS(NAME,32,1,IFCH,IER)  /* OPEN FOR READ
IF (IER .NE. 0) GOTO 9000

IS = 197
IE = 199
NEWSCR = 2  /* DO NOT ERASE THE SCREEN TO START

Clear record buffer
----------------------
10 CALL ZFIL(PTREC,PTSZB,0)

Get the test name.
------------------
20 CALL PEDIT(IFCH,PTREC,IS,IE,NEWSCR)

Get the record for editing.
---------------------------
30 CALL LOC$(PTCHN,PTREC,PTNM,PTARRFLSRET,$9100,0,0,0)

Edit the file
-------------
IS = 275
IE = 297
CALL PEDIT(IFCH,PTREC,IS,IE,NEWSCR)

Store the record
----------------
CALL UPDAT$(PTCHN,PTREC,PTNM,PTARR,FL$RET,$9200,0,0,0)

MORE?
-----
300 IF (YSNO$A('More',4,A$DNO)) GOTO 20

CLOSE THE SCREEN CHANNEL NO.
---------------------------
400 CALL CLOS$A(IFCH)
RETURN

9000 WRITE (1,9001) IER,NAME
9001 FORMAT('ERROR ',I3,': OPENING FILE ',16A2)
RETURN

9100 IF (PTERR .NE. 7) GOTO 9110
WRITE (1,9101) PTNM
A5-111
**C MODFY: MODIFY TEST PROCEDURE**

```fortran
9101  FORMAT(10A2, ' NOT FOUND')
     GOTO 300
9110  IF (PTERR .EQ. 22) GOTO 30
     IF (PTERR .EQ. 24) GOTO 30
     WRITE (1,9111) PTERR,PTNM
9111  FORMAT('MIDAS LOCK ERROR =',13,' KEY = ',15A2)
     GOTO 400

C C
9200  WRITE (1,9101) PTERR,PTNM
9201  FORMAT('MIDAS UPDAT ERROR =',13,' KEY = ',15A2)
     GOTO 400
C C
END
```

**C MKFMC: MAKE IMAGE COMMAND FROM OLD**

```fortran
C MKFMC: MAKE IMAGE COMMAND FROM OLD

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

SUBROUTINE MKFMC(IER)

*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 THOU ('NOT IMPLEMENTED!',16)
RETURN
END
```

A5-112
C MKFMP: MAKE NEW PLAN FROM OLD.

C MKFMP: MAKE NEW PLAN FROM OLD.

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

SUBROUTINE MKFMP(IER)

*CALLED FROM: PLAN
*FUNCTION: 1. GET OLD PLAN NAME FROM OPERATOR
* 2. GET NEW PLAN NAME FROM OPERATOR
* 3. CHECK FOR EXISTANCE OF NEW PLAN NAME
* 4. EDIT OLD PLAN
* 5. INSERT NEW PLAN INTO DATA BASE

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

C MKFRMT: MAKE A NEW TEST PLAN FROM OLD.

C MKFRMT: MAKE A NEW TEST PLAN FROM OLD.

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

SUBROUTINE MKFRMT(IER)

*CALLED BY: PLAN
*FUNCTION: 1. GET OLD TEST PLAN NAME FROM OPERATOR
* 2. GET NEW TEST PLAN NAME FROM OPERATOR
* 3. CHECK FOR EXISTANCE OF NEW TEST NAME
* 4. INSERT TEST INTO DATA BASE

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

AS-113
SUBROUTINE MOTOR(MOT, SPEED, POS, IER)

PARAMETER MSEC=1000
INTEGER MOT, SPEED, POS, STATUS, DIR
INTEGER SLWDN, STRBRK, TOL, DIF, VALUE, VALUE1

*INSERT SYS:COM>CRTCTRL

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

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

SET UP THE PARAMETERS
IER=0

IF SPEED IS ZERO THEN BYPASS
IF(SPEED .EQ. 0)GO TO 999

FIND THE CURRENT POSITION
20 CALL POSIT(MOT, VALUE, STATUS)
IF (STATUS-1) 30, 950, 990

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

SEE IF THERE YET
C
40 CALL ROTION(MOT, DIR, SPEED, STATUS)
45 CALL POSIT(MOT, VALUE, STATUS)
IF (STATUS-1) 45, 950, 990

ALMOST THERE! SLOW DOWN.
100 IF(DIF .LT. STRBRK)GO TO 200

SDIF=DIF
CALL ROTION(MOT, DIR, 20, STATUS)
AS-114
| 3 of 3 | DATE | FILED | 6-181 | OTCI |
C MOTOR: CONTROLS MOTION OF A MOTOR

110 CALL POSIT(MOT,VALUE,STATUS)
   IF (STATUS-I) 115 950 990
115 IF(POS .GT. VALUE)DIF=POS-VALUE
   IF(DIF .LT. STRBRK)GO TO 20
   IF(POS .LE. VALUE)DIF=VALUE-POS
   IF(DIF .LT. STRBRK)GO TO 200
   GO TO 110

C PUT ON THE BRAKES

200 CALL MOTION(MOT,DRT,OSTATUS)
210 CALL POSIT(MOT,VALUE,STATUS)
   IF (STATUS-I) 215 950 990
215 VALUE1=VALUE

C PAUSE FOR 1 SEC
   CALL SLEEP(ASEC)

C CALL POSIT(MOT,VALUE,STATUS)
   IF (STATUS-I) 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 SOMEONE PROBABLY SWITCHED FROM REMOTE TO LOCAL IN THE MIDDLE OF MOVING TO THE TARGET LOCATION. SHAME ON THEM!

950 CONTINUE
   IER=1
   GO TO 999

C ERROR ON GPIB

990 CONTINUE
   IER=2

999 RETURN
END
MOVTAP: VERIFY AND MOVE TAPE

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

* CALLED BY: RVING AND RVINSPECTION *
* FUNCTION: 1. IF A NEW TAPE, MAKE SURE ON-LINE, AND REWIND *
* 2. MOVE THE TAPE TO THE CORRECT POSITION *

SUBROUTINE MOVTAP(NEW, NFIL, IERR)

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

LOGICAL NEW
INTEGER STATUS(3), STAT1, STAT2, UPDN, STATR(3), STATRI

EQUIVALENCE (STATUS(1), STAT1), (STATUS(2), STAT2)
EQUIVALENCE (STATR, STATRI)

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

UPDN = 1
MOVE = 122000  /* MOVE FORWARD ONE FILE
IF (.NOT. NEW) GOTO 40

NEW TAPE, VERIFY THAT TAPE UNIT IS SET UP

CHECK THE TAPE FOR ONLINE

20 TFCNT = 1  /* PRESENT FILE NO.
CALL TMAT(0, LOC(TREC), 0, 100000, STATUS)
CALL MSUP('SELECT', 6, STATUS) /* ***CHECKOUT
IF (AND(:300, STAT2) .NE. :300) GOTO 9200

REWIND TAPE

30 CALL TMAT(0, LOC(TREC), 0, 40, STATR)
CALL MSUP('REWIND', 6, STATR) /* ***CHECKOUT
32 IF(AND(STATRI, 1) .EQ. 0) GOTO 35
CALL TMAT(0, LOC(TREC), 0, 100000, STATUS)
GOTO 37
35 N = NFIL - TFCNT
GOTO 70

TAPE ALREADY INSTALLED POSITION IT

40 N = NFIL - TFCNT
IF (N .GE. 1) GOTO 70
UPDN = -1
MOVE = 201000  /* BACKWARD ONE FILE
N = N - 1

MOVE TO CORRECT PLACE ON TAPE

70 IF (N .EQ. 0) RETURN
CALL TMAT(0, LOC(TREC), 0, MOVE, STATUS) /* MOVE TAPE 1 FILE
IF (STAT1 .EQ. 0) GOTO 90
CALL TMAT(0, LOC(TREC), 0, 100000, STATUS)
IF (STAT1 .NE. 0) GOTO 80

90 N = N - 1
TFCNT = TFCNT + UPDN
GOTO 70

A5-116
C  NOUTAP: VERIFY AND MOVE TAPE

C  RECORD NOT FOUND

200  WRITE(1,9210)
9210  FORMAT(IX,'PLEASE MOUNT TAPE AND PUT ONLINE')
      DO 9211 I = 1,3
      CALL TOOCT(STATUS(I))
9211  CALL INOA(' ',I)
      CALL PAUS(J)
      GOTO 20
END
C PIMG:PROCESS THE IMAGE

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

**CALLED BY:** DINSF

**FUNCTION:**
1. INITIALIZE IMAGE DIRECTORY (CLEAN)
2. SET IMAGE PROCESSOR TO FREERUN
3. ALLOW OPERATOR TO ADJUST MOTORS, CAMERA
4. DO EACH IMAGE PROCESSING REQUEST (CMDR70)
5. IF CODE = 0: DO NEXT PROCESS IMMEDIATELY
   1: WAIT FOR OPERATOR
   2: WRITE IMAGE TO TAPE
   3: LET OPERATOR POSITION MOTOR,
      THEN TELL HIM MOTOR POSITION

SUBROUTINE PIMG(ITEST, NOTAPE, IERR)

C ITEST = TEST SEQUENCE NO
C NOTAPE = FLAG FOR NOTAPE
C IERR = ERROR CODE RETURNED

C INTEGER S500(N), ISTAT(2), DIGIT(40), COMND, DIRCK(40)

C LOGICAL NOTAPE
DATA DIGIT/'>DIGITIZE>$A; ',33*' '/
DATA DIRCK/'>DIRCK; ',36*' '/

C CLEAN THE IMAGE DIRECTORY
CALL CLEAN

C SET IMAGE PROCESSOR TO FREERUN AND GIVE OPERATOR A CHANCE TO FINE
C TUNE THE MOTORS AND CAMERA
CALL CMDR70(DIGIT, IERR)
WRITE(1,5)
5 FORMAT(IX, 'ADJUST MOTORS AND CAMERA AS NECESSARY - HIT A KEY WHEN
1 READY' , '/) CALL PAUS(J)

C DO EACH IMAGE REQUEST
IBIAS = 177
DO 100 I = 1, PIMG
   COMND = PTREC(IBIAS)
   IF (COMND .EQ. 0) RETURN
   IF (COMND .EQ. ') GOTO 8
   IFLAG = FL$RET
   CALL NEXT$(CCHN, CREC, PTREC(IBIAS), CARR, IFLAG, 9000, 0, 0, 0)
   DO 7 J = 1, CNPROC
      IBIAS = 11 + (J-1) * CNWI
      ICAND = CREC(JBIAS)
      IF (ICAND .EQ. 0) GOTO 7
      IF (ICAND .EQ. ') GOTO 7
      CALL CMDR70(CREC(JBIAS), IERR)
   CONTINUE
7 NEXT = IBIAS + 40
   IBIAS = IBIAS + CNWI
   MOVE = PTREC(NEXT)
   IF (MOVE .LE. 0) GOTO 100
   IF (MOVE .GT. 4) GOTO 98

A5-118
C PROCESS THE IMAGE

C GOTO (10, 20, 30, 40), MOVE

10 CALL PAUS(J)
GOTO 100

C SAVE IMAGE ON TAPE
20 IF (NOTAPE) GOTO 100
CALL ZMVD(RTHN, SS0ON, 16)
CALL SVINT(SS0ON)

C CREATE THE TAPE ID RECORD
CALL SVTAPR(IERR)
GOTO 100

C ALLOW OPERATOR TO POSITION MOTOR, THEN TELL HIM WHERE HE'S AT
30 CALL PAUS(J)
CALL POSIT(2, IVAL, ISTAT)
WRITE(1, '11)IVAL
FORMAT(1X, 'CURRENT POSITION IS ', I5)
CALL DMORE(MORE)
IF (MORE) GOTO 30
GOTO 100

C DIRCK

40 CALL CMDA7D(DIRCK, IERR)
CALL TNOUA(PTREC(IBIAS), 40)
CALL PAUS(J)
GOTO 100

C ERROR OCCURRED
C
98 WRITE (1, 99) MOVE, I
99 FORMAT(*ERROR** PLANNED NEXT MOVE = ', I3, ' TEST', I3)
C
100 CONTINUE
C
C SET IMAGE PROCESSOR TO FREERUN
CALL CMDM7O(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
GOTO 6 /* TRY AGAIN
9900 IF (CERR .NE. 7) GOTO 9905
JBIAS = IBIAS + 9
WRITE(1, 9901)(PTREC(I), I = IBIAS, JBIAS), CHNAM
9901 FORMAT(1X, 'TEST', I0A2, ' 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), CHNAM
9910 FORMAT(1X, 'MIDAS ERROR ', I3, ' IN TEST ', 16A2/I, ' FROM DATA BASE ', 16A2)
RETURN
END

A5-119
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:
    CHEWP - CREATE A NEW PLAN
    CHEWT - CREATE A NEW TEST PLAN
    MKFMP - MAKE A NEW PLAN FROM 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
    DLTY - 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

SUBROUTINE PLAN(IER)

IER = 0
CALL GPFNCT(MENU)
IF (MENU .LE. 0) RETURN
GOTO (100,200,300,400,500,600,700,800,900,1000,1100,1200,1300,1400,1500,1600)
GOTO 5

100 CALL CHEWP(IER)
IF (IER) 9000,5,9000

200 CALL CHEWT(IER)
IF (IER) 9000,5,9000

300 CALL CNEWC(IER)
IF (IER) 9000,5,9000

400 CALL MKFMP(IER)
IF (IER) 9000,5,9000

500 CALL MKFMT(IER)
IF (IER) 9000,5,9000

600 CALL MKFMC(IER)
IF (IER) 9000,5,9000

700 CALL MDFYP(IER)
IF (IER) 9000,5,9000

800 CALL MDFYT(IER)
IF (IER) 9000,5,9000

900 CALL MDFYC(IER)
IF (IER) 9000,5,9000

A5-120
C DELETE A PLAN
1000 CALL DLTV(IER)
   IF (IER) 9000,5,9000
C
C DELETE A TEST PROC.
1100 CALL DLTV(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 DSPLC(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: DINSIP
*FUNCTION: 1. POSITION A MOTOR (MOTOR)
*         2. IF MOTOR CONTROL PUT TO MANUAL DURING
*         POSITIONING, PAUSES UNTIL IN REMOTE

SUBROUTINE PMOT

$INSERT PICO

C TURN ON MOTORS WHICH ARE TO RUN CONTINUOUSLY DURING POSITIONING

IPOS = 167
NDIR = 1
DO 5 I = 1,PNMOT
  NSPEED = PTREC(IPOS+1)
  NPOS = P1REC(IPOS)
  IF (NPOS .NE. -1) GOTO 7
  M = I - 1
  CALL MOTION(M,NDIR,NSPEED,IERR)
  IF (IERR .NE. 0) GOTO 30
  IPOS = IPOS + PNW
5 CONTINUE

C POSITION MOTORS WHICH HAVE ENCODERS

IPOS = 167
DO 20 I = 1,PNMOT
  NSPEED = PTREC(IPOS+1)
  IF (NSPEED .NE. 0) GOTO 10
  NPOS = PTREC(IPOS)
  IF (NPOS .LE. 0) GOTO 10
  M = I - 1
  CALL MOTOR(N,NSPEED,NPOS,IERR)
  IF (IERR .NE. 0) GOTO 30
  IPOS = IPOS + PNW
20 CONTINUE

C TURN OFF MOTORS WHICH WERE RUNNING CONTINUOUSLY DURING POSITIONING

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 + PNW
25 CONTINUE
RETURN

C 30 IF (IERR .NE. 1) GOTO 60
CALL TNOUA('PUT MOTOR CONTROL IN REMOTE, HIT ANY KEY ',41)
CALL PAUS(J)
GOTO B
C
60 WRITE(1,61)N,NSPEED,IERR
61 FORMAT(1X,'MOTOR # ',I3,' SPEED ',I3,' ERROR ',I3)
RETURN
END

A5-122
SUBROUTINE POSINT(ENC, IER)
   INTEGER ENCDR, ENC
   INTEGER DIR(10), SPEED, VALUE, VALUE1
   DATA DIR / 0, 1, 0, 0, 0, 0, 0, 0, 0, 0 /
   IER = 0

   MOVE THE MOTOR IN THE DIRECTION SPECIFIED
   10 CALL MOTION(ENC, DIR(ENCDR+1), 127, IER)
   IF (IER-1) 20, 15, 20
   15 CALL THOUA('PUT MOTOR CONTROL IN REMOTE, HIT ANY KEY', 41)
   CALL PAUS(1GO)
   GO TO 10

   CHECK TO SEE IF THE ENCODER IS STILL CHANGING
   20 CALL POSIT(ENC, VALUE1, IER)
   IF (IER-1) 30, 25, 20
   25 CALL THOUA('PUT MOTOR CONTROL IN REMOTE, HIT ANY KEY', 41)
   CALL PAUS(1GO)
   GO TO 20

   STOP THE MOTOR
   30 CALL SLEEPS(MSEC)
   CALL POSIT(ENC, VALUE, IER)
   IF(VALUE1-VALUE .EQ. 0) GO TO 50
g0 TO 20

   CLEAR THE ENCODER
   ENC = OR(ENC, 100000)
   CALL POSIT(ENC, VALUE, IER)
   CALL POSIT(ENC, VALUE, IER)

   SET ERROR FLAG (FOR ENCODER NOT CLEARING)
   IF (VALUE .NE. -1) IER = 1

RETURN
END
*CALLED BY: A1
*FUNCTION : 1. GET RETRIEVAL FUNCTION
*  2. INITIALIZE THE IMAGE PROCESSOR (INTIP)
*  3. CALL ONE OF THE FOLLOWING SUBROUTINES:
*     RVIMG - GET ONE IMAGE
*     RVIST - RECREATE ONE TEST
*     RVINS - RECREATE AN INSPECTION
*     DSPRT - DISPLAY TAPE RECORD
*     DSPR - DISPLAY RESULTS
*     DSPT - DISPLAY TAPE RECORD
*     LISTIX - LIST DATA BASE BY INDEX

ARG 1: ierr INTEGER
SUBROUTINE RETR(EQUIP, ierr)

INTEGER SPCHN, DIRCK(40)
LOGICAL EQUIP
DATA DIRCK/"DIRCK ;',36*'

C SET SPOOL CHAN = 0 FOR DISPLAY ONLY
SPCHN = 0

C GET RETRIEVAL FUNCTION
10 CALL GRFNCT(MENU)
   IF (MENU .EQ. 0) RETURN
   IF (MENU .GT. 3) GOTO 20
   IF (EQUIP) GOTO 20

C INITIALIZE THE IMAGE PROCESSOR
CALL INTIP(IERR)
   IF (IERR .EQ. 0) GOTO 20
   WRITE(1,15) IERR
   FORMAT(1X, 'ERROR ', 13, ' IN INITIALIZING IMAGE PROCESSOR')
   RETURN
20 GOTO (100,200,300,400,500,600,700), MENU

C RETRIEVE IMAGE FOR DISPLAY
100 CALL RVIMG
   GOTO 10

C RECREATE ONE TEST RESULT
200 CALL RVIST
   GOTO 10

C RECREATE AN INSPECTION
300 CALL RVINS
   GOTO 10

C DISPLAY MAIN RESULTS
400 CALL DSPR(SPCHN)
   GOTO 10

C DISPLAY TEST RESULTS
500 CALL DSPRT(SPCHN)
C RETR: RETRIEVE HISTORICAL RECORD

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

C RANAB: REMOVE NON-ALPHABETICAL CHARAC PAGE 0001

RANAB: REMOVE NON-ALPHABETICAL CHARACTERS

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

*GENERAL PURPOSE SUBROUTINE TO REMOVE NON-ALPHABETICAL
*CHARACTERS FROM A STRING

SUBROUTINE RANAB(IBUF,NCHAR)
C IBUF = TEXT STRING
C NCHAR = NO. CHARMS IN TEXT, MAX = 80 CHARMS.
$INSERT SYS.COM> A$KEYS
$INSERT SYS.COM> KEYS.F

INTEGER JBUF(40)

C INITIALIZE PARAMETERS
K = 0
N = NCHAR
CALL ZFIL(JBUF,80,' ')
IF (N.GT.80) N = 80

C PACK INTO TEMP BUFFER
DO 100 I=1,N
   IC = RB($CHR%A(IBUF,1),B)
   IF (IC.LT.1280 .OR. IC.GT.332) GOTO 100
   IF (IC.GT.1271 .AND. IC.LT.301) GOTO 100
   K = K + 1
   CALL NCHR%A(JBUF,K,IC,2)
100 CONTINUE

C NOW REPLACE THE ORIGINAL STRING WITH THE PACKED STRING
CALL MSTR%A(JBUF,N,IBUF,N)
RETURN
END
SUBROUTINE RVIMG

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

LOGICAL ZCM, NEW, MORE
INTEGER IBUF(36), TAPE(IO), IRAGE(25), FILEENTER(40), SELECT(40)

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

DATA ENTER/’ENTER>’A’,35*’/ 
DATA SELECT/’A>SELECT;’,35*’/

OPEN SCREEN TEMPLATE

CALL VOPEN(’DEMO.SCREEN.FT’, 14, 1, 1CHIERR)
IF (IERR .NE. 0) GOTO 9000

GET TAPE NAME, IMAGE NAME, OR FILE NO

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

GET RECORD BASED ON PEDIT INFO

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

FIND TAPENAME REQUESTED IN MIDAS
IFLAG = FL$RET
CALL NEXT$(TCHN, TREC, TAPE, TPARR, IFLAG, 100, 0, 0, 0, 0)

NO FILE NUMBER SPECIFIED: FIND CORRECT IMAGE BY IMAGE NAME
IF (FILE) GOTO 130, 50
5 DO 10 N = 1, TNWT
IBIAS = 12 + (N-1)*TNWT
IF (ZCM(IMAGE, 50, TREC(IBIAS), 50, ICODE)) GOTO 5R
10 CONTINUE

CORRECT IMAGE NOT FOUND

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

FILE NUMBER SPECIFIED: FIND CORRECT IMAGE BY FILE NUMBER

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

WRITE(1,60) TNAM
A5-126
FORMAT('MOUNT TAPE: ',10A2,' & HIT ANY KEY WHEN READY')
CALL PAUS(J)
CALL MOVMTAP(NEW,FILE,IERR)
IF (IERR .NE. 0) GOTO 9100
CALL CLEAN
CALL DIRCK /***CHECKOUT
CALL CMD70(ENTER,IERR,
CALL CMD70(SELECT,IERR)
CALL DIRCK /***CHECKOUT
TFCN7 = TFCNT + 1
GOTO 500

TAPE RECORD NOT FOUND
WRITE(1,110)TAPE
FORMAT(IX,'TAPE RECORD ',10A2,' NOT FOUND')
GOTO 500

TAPE IS CURRENT ONE: WHAT IS FILEN?
IF (FILE) 130,150,70
FILEN LESS THAN ZERO - GIVE 'EM ANOTHER CHANCE
CALL TNOU('PLEASE GIVE MORE SPECIFIC DATA',29)
CALL PAUS(J)
CALL TNOU('PLEASE GIVE MORE SPECIFIC DATA',29)

DO 160 N = 1,TNFL
IBIAS = 12 + (N-1)*TNUT
IF (ZCM(IMAGE,50,TREC(IBIAS),50,ICODE)) GOTO 70
CONTINUE
GOTO 20

MORE?
CALL DMORE(MORE)
IF (MORE) GOTO 3

CLOSE SCREEN TEMPLATE
CALL CLOS*A(ICH)
RETURN

WRITE(1,9010)IERR
FORMAT(IX,'ERROR ',13,' IN OPENING DEMO.SCREEN.FT')
RETURN

WRITE (1,9101) FILE,IERR
FORMAT('ERROR IN LOCATING FILE',13,' ERROR =',13)
GOTO 600

END
SUBROUTINE RUINSP

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

A5-128
C RVST: RECREATE ONE TEST RESULT

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

*CALLED BY: RETR
*FUNCTION : 1. GET PART#, INSPECTION NAME, SERIAL#, TEST NAME (PEDIT)
*  2. CREATE KEYWORD (CRPKEY)
*  3. FIND TEST RESULT RECORD IN DATA BASE
*  4. CORRECT TAPE MOUNTED?
*  5. POSITION TAPE (MOTAP)
*  6. DISPLAY TEST RESULT RECORD ON SCREEN (RPTGEN)
*  7. OUTPUT IMAGE TO IMAGE PROCESSOR
*  8. PROCESS IMAGE AS WAS DONE ORIGINALLY

SUBROUTINE RVST

C $INSERT RTCOR
C $INSERT TCOM
C $INSERT CCOM
C $INSERT SYSCOM>PARM.K
C $INSERT SYSCOM>#$KEYS
C
INTEGER IID(5), PART(IO), IN(IO), SNO(IO), TN(IO), IBUF(45), RKEY(20)
INTEGER ENTER(40), SELECT(40)
LOGICAL NEW, ROREYSNO$A, ZCR
EQUIVALENCE (IBUF(IID)), IID
EQUIVALENCE (IBUF(PART)), PART
EQUIVALENCE (IBUF(IN)), IN
EQUIVALENCE (IBUF(SNO)), SNO
EQUIVALENCE (IBUF(TN)), TN

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

OPEN SCREEN TEMPLATE
CALL UOPEN$('S.RS. SCREEN. FT, 14, I1, IPCHIERR)
IF (IERR .NE. 0) GOTO 9000

GET PART#, INSPECTION NAME, SERIAL#, TEST NAME
CALL ZFIL(IBUF,90,0)
IS = 0
IE = 6
NEWSCR = 2 /#DO NOT ERASE SCREEN TO START
CALL PEDIT(IPCHIBUFISIENESCR)

CREATE RTCOM KEYNAME, FIND ITS LENGTH
CALL CRPKEY(PART,IO,IN,IO,SNO,IO, TN,IO, IBUF,45, RKEY,20)
NCHAR = LSIZE(RKEY,40)
IFLAG = FLRET + FLBIT
CALL TMOUT(RKEY,40)
CALL TDUMP(RKEY,40)

FIND THE RECORD FOR RTCOM
CALL ZFIL(RTREC,RTSZB,0)
CALL NEXT$(RTCHN,RTREC,RKEY, RTARR, IFLAG,$9100,0,0,0,NCHAR)
IFLAG = FLBIT + FLRET + FLUSE
IF (RTTNR(l).NE. 0) GOTO 10

CHANGE PASS/FAIL CODE TO ASCII
RTPF = RTPF * :260

IS CORRECT TAPE MOUNTED?
NEW = .FALSE.

A5-129
C RVTST: RECREATE ONE TEST RESULT

IF (TNAM(1),EQ. 0) GOTO 15
15 NEW = .TRUE.; CALL ZFIL(TM,IXZB,0)
IFLAG = FL*RET
CALL NEXTS(CTCN,TRECC,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(I)

CALL MONTAP(NEW,RTTNM,IERR)
IF (IERR .NE. 0) GOTO 9300

READ IMAGE TO IMAGE PROCESSOR
CALL CLEAN
CALL CMDM70(ENTER,IERR)
CALL CMDM70(SELECT,IERR)
TFCTN = TFCTN + 1

DISPLAY TEST NAME, DESCRIPTION, PASS/FAIL, COMMENTS, TAPENAME, FILEN, IMAGENAME
IS = 304
IE = 310
NEWSCR = 2 /* DO NOT ERASE SCREEN TO START
CALL RPTGEN(0,IPCH,RTREC,LINES,IS,IE,NEWSCR)

FIND IMAGE COMMAND THAT CAUSES IMAGE TO BE WRITTEN TO TAPE

NBIA$ = 217
DO 50 I = 1,8
   IMG = I
   NEXT = RTREC(NBIAS)
   IF (NEXT .EQ. 2) GOTO 60
   NBIAS = NBIAS + RNWI
50 CONTINUE

RECREATE THE IMAGE PROCESSING
60 NBIAS = NBIAS + 1
DO 70 I = IMG,8
   NBIAS = NBIAS + RNWI
   ICAD = RTREC(NBIAS)
   IF (ICAD .EQ. ' ') GOTO 66
   IF (ICAD .EQ. 0) GOTO 80
   IFLAG=FL*RET
62 CALL NEXTS(CCTCN,CREC,RTREC(NBIAS),CARR,IFLAG,$9500,0,0,0,0)
64 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 CMDM70(CREC(JBIAS),IERR)
      IF (IERR .NE. 0) GOTO 9400
64 CONTINUE
IF (RTREC(NBIAS) .EQ. 0) GOTO 66
CALL PAUS(I)
70 CONTINUE

ANY MORE MATCHES?
80 IFLAG = FL*BIT + FL*RET + FL*USE
MORE = YSN0$("LOOK FOR MORE MATCHES",21,A$ND0)
IF (MORE) GOTO 10
GOTO 9900

ERRORS
9000 WRITE(1,9001)IERR
9010 FORMAT(IX,'ERROR ','13,' IN OPENING DEMO.SCREEN.FT')
RETURN

9100 IF (RTERR .EQ. 7) GOTO 9900

A5-130
IF (RTERR .NE. 22 .AND. RTERR .NE. 24) GOTO 9110
CALL RECYC1
GOTO 10
9110 WRITE (I, 9120) RTERR, RKEY
9120 FORMAT (1X, 'NIDAS ERROR ', I3, ' IN FINDING ', 16A2)
GOTO 9900
C
9200 IF (TPERR .NE. 7) GOTO 9208
9208 WRITE (I, 9201) RTINM
9201 FORMAT ('DATA BASE ERROR: TAPE ', 10A2, ' NOT FOUND. ')
GOTO 9900
9210 WRITE (I, 9210) RTNMM
9210 FORMAT (1X, 'NIDAS ERROR ', I3, ' IN FINDING TAPE ', 10A2)
CALL TDUMP (RTINM, 20)
GOTO 9900
C
9300 WRITE (I, 9300) IERR, RTNMM, RTFMN
9310 FORMAT (1X, 'MOV'TAP ERROR ', I3, ' IN MOVING TAPE ', 10A2, ' TO FILE '
C, 13)
GOTO 9900
C
9400 WRITE (I, 9400) IERR, IBIAS
9410 FORMAT (1X, 'CAD07O ERROR ', I3, ' IN IMAGE PROCESS ', 13)
GOTO 9900
C
9500 IF (CERR .NE. 7) GOTO 9510
9501 JBIAS = IBIAS + 9
9502 WRITE (I, 9501) RTREC (I), I = IBIAS, JBIAS, CCNAM
9510 FORMAT (1X, 'TEST ', 10A2, ' NOT FOUND IN DATA BASE ', 16A2)
CALL TDUMP (RTREC (IBIAS), 20)
GOTO 9900
9520 IF (CERR .NE. 22 .OR. CERR .NE. 24) GOTO 9520
CALL RECYC1
GOTO 62 /* TRY AGAIN
9521 IF (CERR .NE. 22 .OR. CERR .NE. 24) GOTO 9520
9522 WRITE (I, 9521) CERR, RTREC (I), I = IBIAS, JBIAS, CCNAM
9523 FORMAT (1X, 'NIDAS ERROR ', I3, ' IN TEST ', 18A2, ' FROM DATA BASE ', 16A2)
GOTO 9900
C
CLOSE SCREEN TEMPLATE
C
9900 CALL CLOS$A (IPCH)
RETURN
END
SUBROUTINE SPOLIT(SPCHN,SPFNAM)

INTEGER ICHN(2), BSIZE, NSIZE

PARAMETER BSIZE = 2000, NSIZE = 7
PARAMETER NBTS = NSIZE*2

INTEGER BUFFER(BSIZE), INFO(12), SPCHNSPFNAM(NSIZE)

LOGICAL DEL

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

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

/* CHECKOUT */
IERR = IERR
JSIZ = BSIZE
WRITE(1, 9000) SPCHN, KEY, SPFNAM, INFO, JSIZ, IERR

9000 FORMAT('SPOOL CHAN = ',13,/
'KEY = ',13,/
'FILE NAME = ',7A2,/
'INFO = ',314,3A2,14,3A2,14,14,/
'BUFFER ',16,/
'ERROR = ',13)

CALL SPOOLS(KEY, SPFNAM, NBTS, INFO, BUFFER, BSIZE, IERR)
IF (IERR .EQ. 0) GOTO 100
WRITE(1,10)IERR,SPFNAM

10 FORMAT(1X,'ERROR ',13,' IN SPOOLING ',7A2)

CALL DELE$A(SPFNAM,14)
IF(DEL) RETURN
WRITE(1,150)IERR,SPFNAM

150 FORMAT(1X,'ERROR ',13,' IN DELETING ',SPFNAM,7A2)
RETURN
END

A5-132
**REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM**

*VERSION 1.0 JUNE 1, 1980*

*BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY*

**SUBROUTINE SVIND(NAME)**

CALL INOUA('SAVE IMAGE: ',12)
CALL INOUA(NAME,32)
CALL INOU(' NOT IMPLEMENTED!',17)
RETURN
END

**SUBROUTINE SVIMT(S500N)**

$S500N = SYSTEM 500 IMAGE NAME$

*FUNCTION: 1. ADD THIS IMAGE'S NAME TO THE TAPE RECORD*

*2. WRITE THE IMAGE TO TAPE (CRDM70)*

CALL INOUA('SAVE IMAGE: ',12)
CALL INOUA(NAME,32)
CALL INOU(' NOT IMPLEMENTED!',17)
RETURN
END

**REAL TIME XRAY - AUTOMATED INSPECTION SYSTEM**

*VERSION 1.0 JUNE 1, 1980*

*BOEING AEROSPACE QUALITY ASSURANCE TECHNOLOGY*

**SUBROUTINE SVIND(NAME)**

CALL INOUA('SAVE IMAGE: ',12)
CALL INOUA(NAME,32)
CALL INOU(' NOT IMPLEMENTED!',17)
RETURN
END

**SUBROUTINE SVIMT(S500N)**

$S500N = SYSTEM 500 IMAGE NAME$

*FUNCTION: 1. ADD THIS IMAGE'S NAME TO THE TAPE RECORD*

*2. WRITE THE IMAGE TO TAPE (CRDM70)*
SUBROUTINE SVARES(IERR)
$INSERT RCOR
$INSERT PCOM
$INSERT SYSCOR>PARR.K
C INTEGER RKEY(20),BIAS
C
CALL ZFIL(RKEY,40,'')
CALL CRPKEY(RPN,20,RIN,20,RSNO,20,0,0,RKEY,40) /*CR RES KEY
CALL APDAt(RKEY,40) /*APPEND DATE

CALL ADDI$(RLCHN,RREC,RKEY,RLARR,FL$USE,$9000,0,0,0)

ADD TEST NAMES TO DATA BASE AS SECONDARY KEYS
DO 100 I = 1,PNT
   BIAS = 43 + (I-1)*RNMT /* FIND TEST NAME
   CALL ADDI$(RLCHN,RKEY,RREC(BIAS),RLARR,FL$USE,$9050,1,0,0)
100 CONTINUE

IERR = 0
RETURN

ERROR IN INSERTING MAIN RECORD
9000 IERR = RLERR
9010 FORMAT(IX,'AIDAS ERROR ',12,' IN ADDING ',2OA2,' TO RESULT DB')
RETURN

ERROR IN INSERTING A SECONDARY KEY
9050 WRITE(1,9060)IERR,RREC(BIAS),I
9060 FORMAT(IX,'AIDAS ERROR ',12,' IN ADDING ',25A2,' INDEX: ',12)
RETURN
END

AS-134
SUBROUTINE SVRES(DEFAULT, NOTAPE, ILOOP, RTKEY, IERR)

LOGICAL DEFAULT, NOTAPE
INTEGER DIGIT(40)
DATA DIGIT/'DIGITIZE', '33 ' /

IF (.NOT.DEFAULT) GOTO 40

C DEFAULT PLAN - SAVE RESULTS? TAPE OR DISK?

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

C DEFAULT PLAN - SAVE RESULTS ON DISK

30 CALL SVIMD(RTIMN)
GOTO 100

C SAVE RESULTS ON TAPE

C OPEN SCREEN TEMPLATE

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

C SET UP PEDIT PARAMETERS

60 IS = 25
IE = 36
NEWSCR = 2  
C CLOSE SCREEN FILE
CALL CLOS4A(ICHN)
C SAVE TEST RESULTS
C
IF (DEFAULT) GOTO 100
CALL SVTSTR(RTKEY, IERR)
C SET IMAGE PROCESSOR TO FREERUN
C
100 CALL CLEAN
CALL CHOA70(DIGIT, IERR)
SUBROUTINE SVTSTR (KEY, IERR)

INTEGER KEY(25)

IF (RTPF .EQ. 0) GOTO 10
RPF = RTPF

C INSERT TEST RESULTS INTO DATA BASE
10 IFLAG = FL$RET
CALL ADDI(RTCHN,RTREC,KEY,RTARR,FL$RET,$9000,0,0,0)
RETURN

9000 IERR = RTERR
WRITE(19010)IERR,KEY
9010 FORMAT(1X,'MIDAS ERROR ',I2,' KEYWORD = ',25A2)
RETURN
END
SUBROUTINE SVTAPR(IERR)

C LOCK TAPE DATA BASE FOR EDITING
IFLAGS = FLSRET
10 CALL LOCK$(1CHN,1RPREC,TNAM,TPARR,IFLAGS,$9000,0,0,0)

C UPDATE TAPE DATA BASE RECORD
IFLAGS = FLSUSE
CALL UPDAT$(1CHN,TREC,TNAM,TPARR,IFLAGS,$9050,0,0,0,0)
IBIAS = 12 + (TNFL - 1)*TNWT
CALL ADDI$(1CHN,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 RECYC
GOTO 10  /* BUSY - DIAL AGAIN

C ERROR IN UPDATING
9050 WRITE(1,9050)TPERR,TNAM
9060 FORMAT(1X,'MIDAS ERROR ',12,' FROM LOCK$, TNAM='',10A2)
IERR = TPERR
RETURN

C ERROR IN ADDING IMAGE NAME
9200 WRITE(1,9200)TPERR
9210 FORMAT(1X,'MIDAS ERROR ',13,' 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:
- .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
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

BUILD TABLE OF OPTIONS
- NO. OF FRAMES
- DELAY
- BREAK POINTS
- AUTO RANGE

PARAMS
PARSE COMMAND LINE

"N" IN RANGE?
- NO
- YES

N = 32

"MIN" IN RANGE?
- NO
- YES

MIN = 32

"MAX" IN RANGE?
- NO
- YES

MAX = 32

DAVG
DO VIDEO AVERAGING

ESTAB.
IMAGE
PROC.
CONTROL.

SETUP
ESTABLISH
IMAGE

ICLOS
CLOSE OUTPUT
DISPLAY

RETURN
SAVE - PRIMITIVE TO SAVE OUTPUT OF PIPELINE

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

A6-5
SAVED - SAVE THE OUTPUT OF THE PIPELINE

ARGUMENTS: 5
- INTEGER: FCB
- INTEGER: ZERO
- INTEGER: NIDS
- INTEGER: NODS
- INTEGER: IBUFF

A6-6
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
       LUTCN WRITES LUT MASK TO ENABLE CHANNELS
       MNMAX 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 VERSION, DIGCHN, DIGMSK
INTEGER X, Y
INTEGER OUTSEL(8)
REAL SLOPE

INTEGER PBITS, DMASK /* EXTERNAL FUNCTIONS

GET VERSION INFO
CALL M700P (FCB, VERSION, NCHAN, LEVELS, IRMEM, I, I, I, 1,
           IFDBCK, I, I, I, I, I, I, I, I, 1)

GET DIGITIZER CHANNEL NUMBER
CALL GBITS (IRMEM, 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)

A6-7
CALL PROFIL (FCB, 0, 0, 0, 0, 0)
CALL ZFIL (BUFFER, 512, 0)
CALL GRAM (FCB, BUFFER, 0, 0)
CALL CRCTL (FCB, 0, 0, 0, 0, 0, 0, 0, 0)

IF N > 32 THEN DON'T NEED TO LOAD IFM
IF (N.GT. 32) GO TO 140
NN=N
IF (N .EQ. 32) NN=16
SLOPE=1./FLOAT(NN)

DO 120 J=1,NN
   DO 110 I=1,512
      ISTART=(J-1)*512
      IDX=ISTART+I-1
      BUFFER(I)=IFIX(IDX*SLOPE)
   CONTINUE

   CALL IFM(FCB, BUFFER, ISTART, 512, 0, 0, 0)

   FCB
   MAP = IFM CONTENTS
   START = ZERO RELATIVE POSITION OF STARTING POINT
   COUNT = THE NUMBER OF IFM ELEMENTS TO TRANSFER
   PACK = 1 IMPLIES PACKED MODE
   VRTRTC = READ = 0 IMPLIES WRITE, 1 IMPLIES READ

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

   DEFINE AND LOAD LUT MAP FOR DIGITIZING CHANNEL FOR SLOPE OF 1

   DO 22 I=1,256
      BUFFER(I)=I-1
   CONTINUE

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

   CHECK IF WANT DIFFERENT SCALING
   IF (.NOT. MINMAX) GO TO 310

   ENABLE THE DIGITIZING LUT, GET MIN AND MAX, RELOAD LUT

   CALL LTCNT(FCB, DIGMSK, 7, 0, 0)
   CALL MNMAX(FCB, MIN, MAX, I, ZI, I, I)

   MINP1=MIN+1
   DO 305 I=1,MINP1
      BUFFER(I)=0
   CONTINUE

   SLOPE=255./(MAX-MIN)
   MINP2=MINP1+1
   DO 320 I=MINP2,MAX
      BUFFER(I)=IFIX((I-MINP1)*SLOPE+.5)
   CONTINUE

   MAXP1=MAX+1
   DO 340 I=MAXP1,256
      BUFFER(I)=255
   CONTINUE

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

   FCB
   MAP = MAP FUNCTION FOR LUT
   COLOR = BIT MASK FOR WHICH LUTS TO WRITE
   VRTRTC = READ = 0 IMPLIES WRITE, 1 IMPLIES READ
A6-8
LOAD THE OTHER LUTS

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

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

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

RELOAD OFMS WITH POSITIVE UNITY TRANSFORM

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

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

DO 80 I=1,N
   CALL FDBCK(FCB,2,3,-1,1,0,0,1,1,0)
80 CONTINUE
C DAVG: PRIMITIVE FOR AVERAGING VIDEO FRA PAGE 0004

C

C FCB COLOR = 4,2,1 FOR RED, GREEN, BLUE (ORDER?)
C CHANL = BIT MASK SELECTING DESTINATION CHANNEL
C BITP = BIT MASK SELECTING BIT PLANES
C Bypass = 0 IMPLIES USE IFM, 1 IMPLIES BYPASS IFM
C PIXOFF = EXTERN ZERO
C EXTERN ACCUM
C ADWRT

80 CONTINUE

C SET UP ALU OREF FOR ACCUMULATOR
C DO 50 I=1,8
C OUTSEL(I) = 10
50 CONTINUE
C CALL ALU(FCB, 0, :14, :14, BUFFER, OUTSEL, 0, 0, 0, 0, 0, 0)
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 CARRYIN =
C EQUAL =
C READ =
C IF(N .LE. 32) GO TO 600
C OTHERWISE SHIFT DATA UP TO THE MSB
C SET ALU FOR ALU OUTPUT
C DO 88 I=1,8
C OUTSEL(I) = 12
88 CONTINUE
C CALL ALU(FCB, 0, :14, :14, BUFFER, OUTSEL, 0, 0, 0, 0, 0, 0)
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 CARRYIN = SETS THE CARRY IN CONDITION
C CARRY =
C EQUAL =
C READ =
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 MULTIPLY BY TWO BY FEEDING BACK AND DOING A+A
C 500 CALL FDBCK(FCB, 0, :1, :1, 0, 0, 0, 0, 0, 0)
C 510 CONTINUE
A6-10
C DAUG: PRIMITIVE FOR AVERAGING VIDEO FRA PAGE 0005

MULTIPLY BY TWO BY FEEDING BACK AND DOING A+A
CALL FDBCK(FCB,0,3,-1,1,0,0,0,1,0)

SET ALU 0ST FOR OFM AND THEN FEED BACK THE 2ND CHANNEL TO THE 1ST
DO 99 I=1,8
   OUTSEL(I)=:11
   CONTINUE
   CALL ALU(FCB,0,:14,:14,BUFFER,OUTSEL,0,0,0,0,0,0,0)
   CALL LTcnt(FCB,2,7,0,0)
   CALL FDBCK(FCB,2,1,-1,1,0,0,0,0,0,0)
   GO TO 444

FEEDBACK 1 TIME THRU THE IFM TO ACCOMPLISH A DIVIDE BY N
   CALL FDBCK(FCB,2,1,-1,0,0,0,0,0,0,0)

ENABLE THE LUTS FOR THE CHANNEL THAT THE OUTPUT WAS FED BACK TO
   CALL LTcnt(FCB,1,7,0,0)
SCALE UP BY RELOADING OFMS WITH SLOPE OF 4
   DO 294 I=1,512
      BUFFER(I)=I+I+I+I-4
      BUFFER(512+I)=512+I+I+I-4
   CONTINUE

   CALL OFM(FCB,BUFFER,7,0,0)
TRANSFER COMMAND BUFFER TO MODEL 70
   CALL DEEXEC (FCB)
RETURN
END
DAVD: INTERFACE FOR VIDEO AVERAGING.

***************************************************************
* 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 WAI, 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 ADDRI, ADDR, ADDRS(6)

BUILD PARAMETER PROMPTS

SET N = 16 DEFAULT
CALL ZMVD ('N', INAMES(1), 16)
CODES(1) = 0
SIZES(1) = 1
ADDRS(1) = ADDRI (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) = ADDRI (WAIT)
COUNTS(2) = 0

WAIT = 10

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

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

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

CALL ZMVD ('BATCH', INAMES(41), 16)
CDAVG: INTERFACE FOR VIDEO AVERAGING. PAGE 0002

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

NPARMS = 6

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 = MAX
IF (MAX .LE. MIN) MAX = 255
IF (MAX .GT. 255) MAX = 255

IF (COUNTS(5).EQ.1) MINMAX = .TRUE.
IF (COUNTS(6).EQ.1) BATCH = .TRUE.

OPEN OUTPUT DISPLAY IMAGE
ODSRN = 1
DTYPE = 1
FTYPE = 2
NS = 512
NL = 512
NBANDS = 1

DO ACTUAL DIGITIZING AND SUMMATION
CALL DAVG (FCB, BUFFER, N, WAIT, MIN, MAX, MINMAX, BATCH)
OCHNNO(1) = 0
CALL SETUP (FCB, BUFFER, NBANDS, NLEVS, OCHNNO, O, 1)

CLEAN UP
CALL ICLOS (FCB, ODSRN)
RETURN
END
SUBROUTINE SAVE (FCB, BUFFER, ICHNNO, OCHNNO, NBANDS, BIAS, RSCALE)

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

DESCRIPTION - 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 FBCK WRITES THE FEEDBACK LOOP CONTROL WORD

CHANGES - COMPUTATION OF IFM RAMP CHANGED FROM '1*RSCALE' TO '(I-1)*RSCALE' JUN 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 GRRAN (FCB, BUFFER, 0, 0)
CALL CRCTL (FCB, 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
A6-1A
C --- SAVE: PRIMITIVE TO SAVE OUTPUT OF PIPE PAGE 0002

CALL ALU (FCB, 0, 0, BUFFER, OUTSEL, 0, 0, 0, 0, 0)
/* OUTSEL = :11 */ PIPELINE OUTPUT

DEFINE IFM RESCALE MAP
THE .25 = \( \frac{1}{2^{2}} \) WHICH SCALES 10 BIT TV DOWN TO 8 BIT MEMORY
DO 60 I=1,1024
BUFFER(I) = (I-1) * RSSCALE * .25
60 CONTINUE

CALL IFM (FCB, BUFFER, 0, 1024, 0, 0)
/* MAP = BUFFER */ RESCALE RAMP FUNCTION
/* START = 0 */ FIRST POSITION TO LOAD IN IFM
/* COUNT = 1024 */ NUMBER OF ELEMENTS TO TRANSFER
/* PACK = 0 */ 1 IMPLIES PACKED MODE TRANSFER
/* VRTRTC = 0 */ WAIT FOR VERTICAL RETRACE INTERVAL
/* READ = 0 */ 0 IMPLIES READ, 1 IMPLIES WRITE

COMPUTE OUTPUT CHANNEL MASK FROM CHANNEL CODE ARRAY
CALL DADRS (CHMASK, OCHNNO, CHCODE, NBANDS)

WRITE THE FEEDBACK LOOP CONTROL WORD
CALL FOBCK (FCB, 1, CHMASK, -1, 0, 0, 0, 0, 0)
/* COLOR = 1 */ 4, 2, 1, FOR RED, GREEN, AND BLUE
/* CHANL = CHMASK */ BIT MASK SELECTING DESTINATION CHANNEL
/* BITP = -1 */ BIT MASK SELECTING THE BIT PLANES
/* BYPIFM = 0 */ 0 IMPLIES USE IFM, 1 IMPLIES BYPASS IFM

RETURN
END
C SAVED: SAVE THE OUTPUT OF THE PIPELINE

SAVED: SAVE THE OUTPUT OF THE PIPELINE

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

*******************************************************************************

SUBROUTINE SAVED (FCB, ZERO, NIDS, NODS, IBUFF)

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

DESCRIPTION: SAVED calls the primitive SAVE to save the pipeline output of a display image in refresh memory.

C specify passed variables
INTEGER ZERO(1), FCB(1), IBUFF(1)
INTEGER NIDS, NODS

C specify input and output image variables
INTEGER ICHNNO(16), OCHNNO(16)
INTEGER IDSRN, ODSRN, DTYPE, FTYPE, NS, NL, NBANDS
INTEGER ID, NLEVS

C specify parameter system variables
INTEGER INAMES(16), CODES(2), SIZES(2), COUNTS(2), NPARMS
INTEGER BIAS, ADDR1, ADDR2
REAL RSCALE

LOGICAL LOCKED(16)

IF (I.LE.NIDS .AND. NIDS.LE.3) GOTO 100
CALL TNOUA ('ERROR: ',8)
CALL TNOU ('NO INPUT, OR MORE THEN THREE INPUTS SPECIFIED',45)
RETURN

C 100 CONTINUE

C check value of NODS
IF (NODS.LE.1) GOTO 200
CALL TNOUA ('ERROR: ',8)
CALL TNOU ('MORE THEN ONE OUTPUT SPECIFIED',30)
RETURN

C 200 CONTINUE

CALL ZMV'D ('BIAS ',INAMES(1), 16)
CODES(1) = 0
SIZES(1) = 1
ADDR1 = ADDR1 (BIAS)
COUNTS(1) = 0
BIAS = 0

CALL ZMV'D ('RESCALE ',INAMES(9), 16)
CODES(2) = 1
SIZES(2) = 1
ADDR2 = ADDR2 (RSCALE)
COUNTS(2) = 0
RSCALE = 1.0

NPAMRS = 2

CALL PARAMS (FCB, INAMES, CODES, SIZES, ADDR, COUNTS, NPARMS)

CALL IOPI (FCB, IBUFF, IDSRN, DTYPE, FTYPE, NS, NL, NBANDS)
A6-16
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 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 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

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 CHREP (FCB, ODSRN, OCHNNO, NLEVS)

700 CONTINUE
   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)

800 CONTINUE
   close all output images
   CALL ICLOS (FCB, ODSRN)

900 CONTINUE
   close all input images
   DO 910 IDSRN = 1,NIDS
      IF (.NOT.LOCKED(IDSRN)) CALL UNLOCK (FCB, IDSRN)
      CALL ICLOS (FCB, IDSRN)
910 CONTINUE

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,ALTTRN,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.
ALTTRN = RETURN TAKEN IF ANY ERROR OCCURS.
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.

CMD70 (COM,ERR)
CMD70 PASSES A COMMAND TO THE STC SYSTEM 500. THIS SUBROUTINE WAS WRITTEN TO BE COMPATIBLE WITH AUTOMATIC PROCESSES. THE ARGUMENTS ARE:
COM = COMMAND LINE IN SYSTEM 500 FORMAT.
ERR = ERROR CODE, 0 = NO ERROR.

CNTLD
CNTLD 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$ (BUFFER)
DATE$ 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.

DELETE (FILE, LEN)
DELETE 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 occurs.
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$ (BUFFER, BUFLEN, 'CHAR')
FILL$ 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$ (TEXT, POS)
GCHR$ 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 occurs.
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.

A7-3
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.

RCHRSA (DTEXT, DPOS, STEXT, SPOS)
RCHRSA 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.

MSTRSA (STEXT, LENS, DTEXT, LEND)
MSTRSA 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)

MSUBSA (STEXT, LENS, STARTS, ENDS, DTEXT, LEND, STARTD, ENDD)
MSUBSA 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 STRING.
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, ALTRTN, 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.
ALTRTN = RETURN TAKEN IF ANY ERROR OCCURS.
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 (1GO)
PAUS is a subroutine used to hold the program flow until the operator presses a terminal key. The argument is:
1GO = 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,NUN)
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.
NUN = 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.
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,FLN,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.
FLN = 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.
TINT (UNIT, ADDR, COUNT, COMND, STATUS)
TINT 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.

T1OU (CHAR)
T1OU 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.

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$.

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$.

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$.

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$.

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$.

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$.

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$.

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$.

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$.
VOPEN$ (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 STRING.
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
6-81
DTIC