ASSP

ADVANCED SENSOR SIGNAL PROCESSOR

FINAL TECHNICAL REPORT

June 1984

Sponsored by

Defense Advanced Research Project Agency
ARPA Order No. 4324, PAN

Monitored by

U.S. Army Missile Command
Contract DAAH01-82-C-A106
Effective Date: 1 December 1984
Expiration Date: 31 August 1984

Project Manager:

T. E. Dilworth, Jr.
Systems Development Office
Assault Breaker Division
Army Missile Laboratory
(205) 876-3171

Northrop Research and Technology Center
One Research Park
Palos Verdes Peninsula, CA 90274

Northrop Electro-Mechanical Division
500 East Orangethorpe Avenue
Anaheim, CA 92801

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

REPORT DOCUMENTATION PAGE

Form Approved
OPM No. 0704-0188

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	JUNE 1984	FINAL TECHNICAL REPORT JUNE 84

4. TITLE AND SUBTITLE
ASSP ADVANCED SIGNAL PROCESSOR

5. FUNDING NUMBERS
CONTRACT
DAAH01-82-C-A106

6. AUTHOR(S)
J. J. REIS + J. A. REED

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NORTHROP RESEARCH AN ELECTRO-MECHANICAL DIV
ONE RESEARCH PARK
PALOS VERDES PENINSULA, CALIF 90274

8. PERFORMING ORGANIZATION REPORT NUMBER
NONE

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
DARPA ORDER NO. 4324
U.S. ARMY MISSILE COMMAND
REDSTONE ARSONAL, AL 35898

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NONE

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
NO LIMITATIONS - PUBLIC AVAILABILITY

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Develop a suitable top-down infrared target signature data base to allow adequate training and exercising of the real-time processor and its algortihms.

Build and demonstrate a real-time signal processor for Northrop's proprietary image processing algorithms to locate and classify military targets.

Demonstrate the robustness of the Northrop approach using an airborne Captive Flight Test to locate and classify real targets in the field.

14. SUBJECT TERMS
INFRARED TARGET SIGNATURE, REAL TIME SIGNAL PROCESSOR

15. NUMBER OF PAGES
293

16. PRICE CODE
N/A

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

ASSP

ADVANCED SENSOR SIGNAL PROCESSOR

FINAL TECHNICAL REPORT

Prepared for:

Defense Advanced Research Project Agency
DARPA Order No. 4324
U.S. Army Missile Command
Contract DAAH01-82-C-A106

June 1984

J. J. Reis
Northrop Research and Technology Center
One Research Park
Palos Verdes Peninsula, CA 90274

J. A. Reed
Northrop Electro-Mechanical Division
500 East Orangethorpe Avenue
Anaheim, CA 92801

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF FIGURES

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF TABLES

ACKNOWLEDGMENTS

The Advanced Sensor Signal Processor (ASSP) Program was a Defense Advanced Research Project Agency (DARPA) sponsored effort monitored by the Systems Development Office and the Army Missile Laboratory, U.S. Army Missile Command (MICOM) Redstone Arsenal, Alabama. The DARPA technical representative was Col. James Ticklin: the MICOM support included Mr. Tom Dilworth and Mr. Ron Hall, Program Management, Mr. Ray Farmer, and Mr. Monty Helton, Engineering, and Mr. Tom Lamar, Test. We would like to acknowledge the critical participation and direction of these personnel vital to the successful completion of the program. We would also like to acknowledge the support of Dr. James Tegnalia who initially sponsored the effort while the Director of Tactical Technology Office at DARPA.

Northrop successfully completed the program tasks through a joint technical effort at the Northrop Research and Technology Center and the Electro-Mechanical Division. The effort was a continuation of joint in-house technology development ventures. We would like to acknowledge the participation of the technical staffs of both organizations plus the support of our senior management staff.

The ASSP algorithms were developed by Tony Lucero, Anthony Luk, Dave Garber, and Jim Reis. Software support was provided by Glenn Martin, Ken Davis, Marty Cohen, Jeff Cameron, and Nina Edmondson. The ASSP hardware was designed by Jim Reis, Henry Launspach, Steve Ohlsen, Craig Hackerd, Bruce Rapp, and Larry Shoemaker. Firmware coding was performed by David Ohl, Guy Parker, Glenn Martin, Steve Ohlsen, Craig Hackerd, and Jeff Cameron. Runtime software was developed by Poming Leo, Gordon Spear, and Lonni Thayer. Flight operations for Data Collection and the Captive Flight Test were coordinated by Joe Kriek.

DISCLAIMER

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Research Projects Agency, the U.S. Army Missile Command, or the U.S. Government.

DISTRIBUTION

This document contains classified information SECRET-XGDS-3 as noted. This document also contains Restricted Rights as established by Contract DAAH01-82-C-A108, Section H-4. As such, distribution of the document will be limited to U.S. Government agencies only. Only requests for this document must be referred to the Commander, U.S. Army Missile Command, Attn: DRSMI-O, Redstone Arsenal, AL 35898.

SECTION 1.0

PROGRAM DESCRIPTION

1.1 Introduction

The Advanced Sensor Signal Processor (ASSP) Program is a Defense Advanced Research Project Agency (DARPA) sponsored development effort directed by the U.S. Army Missile Command, Redstone Arsenal under U.S. Army Contract DAAH01-82-C-A106 issued December DY 1981. This program was issued to demonstrate the capability of a series of Northrop developed algorithms to acquire armored class targets in natural clutter. The program had four prime objectives:

a) Develop a suitable top-down infrared target signature data base to allow adequate training and exercising of the real-time processor and its algortihms.

b) Build and demonstrate a real-time signal processor for Northrop's proprietary image processing algorithms to locate and classify military targets.

c) Demonstrate the robustness of the Northrop approach using an airborne Captive Flight Test to locate and classify real targets in the field.

d) Prove the viability of a new gyroless strapdown guidance technique using a Proportional Airborne Digital Simulator via six-degree-of-freedom digital simulation and hardware-in-the-loop simulation.

Northrop Corporation has been actively involved in the development and implementation of autonomous weapon system technologies for several years. Considerable in-house efforts have been expended to analyze and generate new seeker, signal processor, airframe controls and guidance techniques that could be applied. The Northrop Research and Technology Center has been a focus for autonomous target detection algorithm development. This work started in the visual spectrum using terrain board models. The resulting algorithms proved to be powerful tools in detecting vehicular targets in natural clutter. These algorithms are unique in that they are image based and detect targets using correlated target features and

reject false targets by using uncorrelated false alarm features. These visual based algorithms were documented and constitute the technology baseline used in the Advanced Sensor Signal Processor (ASSP) Program.

Simultaneously with development of the algorithms, Northrop determined that a conversion to the infrared spectrum was required. This required the selection of a seeker and the acquisition of a suitable data base. The Northrop Research and Technology Center and the Northrop Electro-Mechanical Division have an ongoing joint Internal Research and Development Program to develop a 128 x 128 staring infrared focal plane array sensor. This sensor and its performance characteristics were selected as the input baseline for the processor. As the array sensor was in development, a Northrop owned common module FLIR was used for data collection. The FLIR output was reformatted to simulate the performance capability of the focal plane array in mean resoluble temperature (MRT) and modulation transfer function (MTF).

These Northrop assets were integrated in support of the ASSP contract. The FLIR sensor and support recording equipment were furnished to the contract effort to enable measurement of target infrared signatures to train the algorithms. The Northrop algorithm baseline was furnished to the contract for conversion to the infrared region. The contract scope, therefore, included conversion of the algorithms to the infrared, implementation of the converted algorithm set into a near real time brassboard signal processor, and the Captive Flight demonstration of this processor over real targets in natural clutter.

In addition, the contract effort included the measurement and data reduction of an image based data set for algorithm training. This set was obtained using the Northrop FLIR sensor, a ground truth television camera, analog and digital recorders, and IRIG timing. These data have been reduced and formatted and are deliverable to the Government.

Northrop had also previously developed a new guidance technique that replaces the conventional gyro stabilized guidance package with a solid state electronic processor. The scheme works much like a 6 degree-of-freedom laboratory simulation in that the airframe flight characteristics are preprogrammed into the processor and perform the guidance computations.

DARPA included a simulation task in the contract effort to validate this approach and determine its worth to low-cost precision munition concepts. This work was concluded with both 6 DOF and hardware-in-the-loop simulations.

The contract scope was structured to provide specific milestone demonstrations of new technologies that can be applied to autonomous weaponry. This Final Technical Report describes the work performed and results obtained. This report is being prepared and issued in accordance with the provisions of the Contractor Document Requirements List, Item A015.

1.2 Summary of Accomplishments

Each of the four main contract objectives was exceeded. The detailed results are described in Section 2.0 of this report but are summarized here:

a) The Northrop algorithms were successfully converted to operate in the infrared spectrum. Three algorithms were selected and used. They are correlated in their ability to make target decisions. They are uncorrelated in the manner they detect false targets. They are combined in a unique hierarchical manner that results in high probabilities of detection. The training set used is relatively small but effective due to the procedure used for final target detection.

b) A new form of pattern classifier was utilized which optimally mapped feature vectors in target probabilities. This classifier is capable of accommodating non-stationary target signatures and it extrapolates beyond the training set when presented with non-representative test data.

c) The near real time brassboard processor is a loosely coupled computing architecture that formats the input FLIR imagery, filters out anomalies, digitizes the image and passes this image to the individual algorithms for processing. After processing, the processor displays the results on a TV screen and passess all information to a digital recorder. The processor has been incorporated into a UH-1D helicopter and flown in November 1983 in a Captive Flight Test. The Captive Flight Test resulted in 141 independent images being processed with a P_D = 92% that the number one ranked object is a target.

d) In May 1982, the Northrop data collection equipment was used to collect suitable target signature data for algorithm training. 927 frames were selected, formatted, of which 150 were used to develop the algorithm set. The equipment has been designed to fit any UH-1, Bell 212, or Bell 205 class helicopter, is transportable to any place where data measurement is needed, and has been classified as airworthy.

e) The solid state guidance scheme was validated through simulation. The resulting data shows that this approach has application to precision munitions where the terminal end game is short and defined. The guidance accuracy is less than a one meter miss distance.

The Captive Flight results covered the initial top-down scenario and were expanded to include scene analysis of other targets, terrain, and lower aspect angles. The test results show that target detections were made at 25° from horizontal, at ranges of 3,000 feet to 200 feet, with target sizes one-half to four times the trained images and against both hot and cold targets. The ASSP algorithms are robust and efficient. The unique hierarchical combination for final target decisions results in high probabilities of detection against a variety of targeting conditions.

1.3 Conclusions and Recommendations

The ASSP processor performs beyond expectations. This processor is able to detect both hot and cold armored targets under a variety of conditions and in natural clutter. Additional work to increase its capability to find targets is merited and should include:

a) Continued development of algorithm extensions and refinements to acquire high-value targets, missile launch sites, and assembly areas. This will also require additional data collection efforts.

b) Expansion of the hierarchical combining logic of the ASSP to incorporate target classification by vehicle type and incorporate contextual information to further enhance system performance.

c) Improvement of the signal processor brassboard hardware to increase throughput and incorporate new algorithms as they become available.

d) Increased efforts to develop laboratory tools in handling data formatting and to enable assessment of algorithm worth. These tools were started and used during the ASSP effort but need to be generalized for future application.

e) The development, fabrication, and testing of an air vehicle that can dynamically demonstrate the unique technology features of ASSP. This vehicle would use the focal plane array as the seeker integrated to the processor. The controls would be integrated to the gyroless guidance processor.

The ASSP program was an effort to demonstrate new innovative solutions to old established and unresolved system level problems. The data presented in the following sections will show that an imaged based signal processor integrated to an imaging infrared seeker and using a solid state guidance technique provide a new next generation approach to terminally guided munitions. This program resulted in successful flight demonstration of brassboard hardware. This work should be continued culminating in a dynamic system level test that fully validates the approach and demonstrates the full system potential of the technologies involved.

SECTION 2.0

2.1 NATS Background

2.1.1 Introduction to ASSP/NATS

The Advanced Seeker Signal Processor, ASSP, also known as the Northrop Advanced Tactical System, NATS, is a sophisticated processing system based upon a network of computing elements. This network is designed and programmed to solve a class of image processing problems which are relevant to autonomous target detection and tracking.

ASSP/NATS is the product of a Northrop IR&D program, which has been ongoing for the last seven years. Based upon IR&D work on staring focal plane arrays, we were convinced that image quality infrared sensors would be available by the early 80's. Therefore, the real image processing problem would be one of detecting targets in the presence of highly resolved, high contrast background clutter. Since image quality infrared data was not available for the IR&D work, we utilized television imagery of accurate terrain board models based upon reconnaissance photographs of Eastern European scenes. Four types of data were collected: cultural and rural, in summer and winter. Tank targets were resolved to 12 pixels along its length, and the signal to noise ratio from the television sensor was excellent. The phenomenological differences between infrared and visible were felt to be a second order issue. The NATS algorithms were therefore initially designed to operate with these high resolution television images.

The NATS algorithms were subjected to blind tests with excellent results. With this as a background we sought Government support to extend and validate these algorithms into the infrared spectrum.

2.1.2 System Overview

The NATS/ASSP scenario is similar to Assault Breaker, namely a cannister of submunitions is sent to an area in which target activity has been detected by a stand-off sensor platform. The cannister dispenses its submunitions over the area. Each submunition must autonomously detect and hit-to-kill a target within its field of view.

The end game scenario is depicted in Figure 2.1. Around 1500 feet, the submunition penetrates the cloud ceiling to gain visibility of the ground. A GROUND-LOCK algorithm acquires the ground directing the submunition to

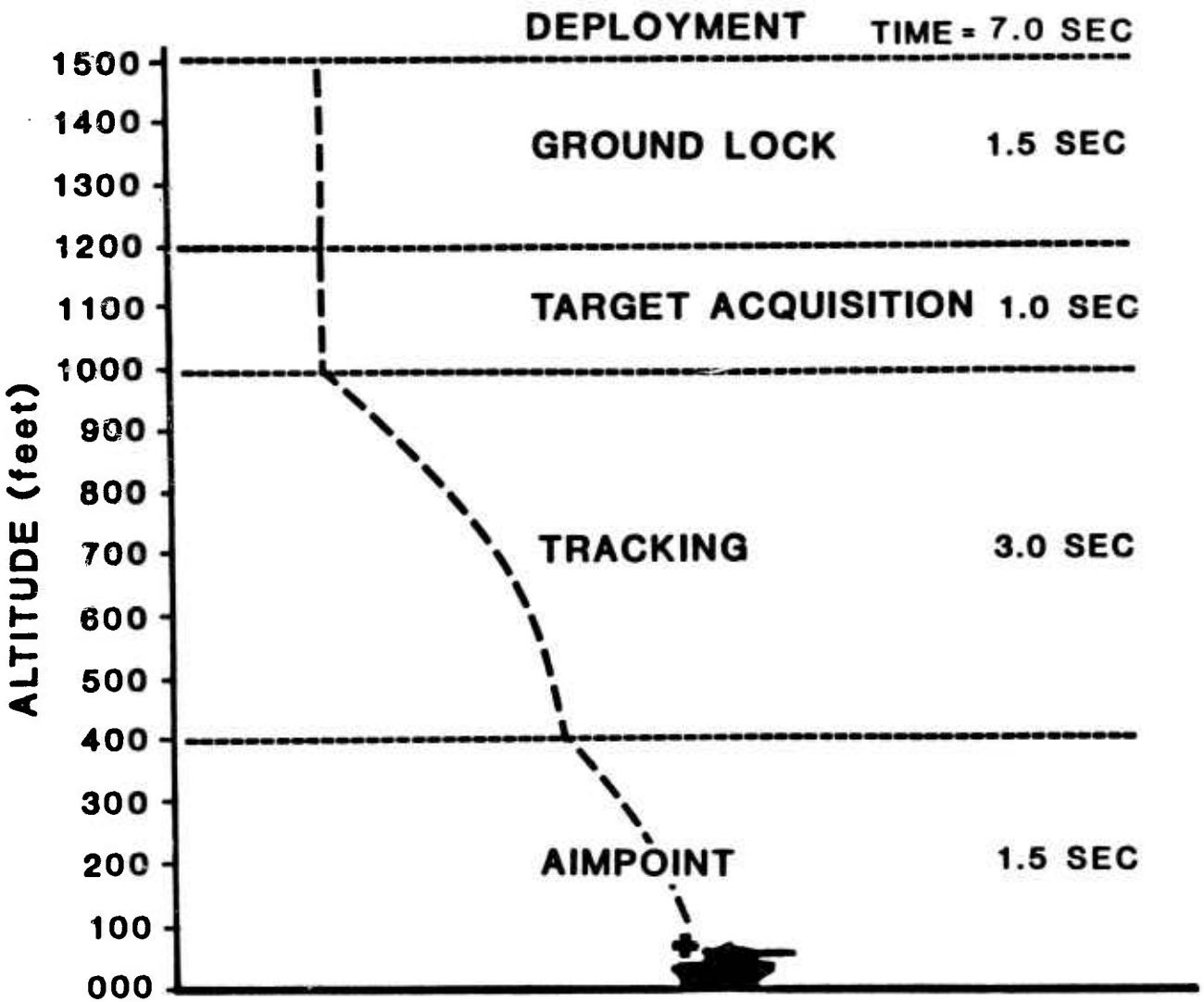


Figure 2.1. End Game Scenario.

motion stabilize the imaging sensor. Altitude is estimated by measuring the radial divergence of the image and by knowing the submunition descent rate. TARGET-ACQUISITION is activated around 1200 feet. The field of view is scanned and all possible targets within this field of view are ranked according to probability of target. The top-ranked detection is selected for tracking. The TRACKER and AIMPOINT-SELECTION algorithms guide the submunition to a particularly vulnerable spot on the target, ensuring a high probability of kill. This scenario defines the performance envelope over which the target detection algorithms must operate. The scenario constraints have been incorporated into the design of the signal processing algorithms in the form of apriori information and initial conditions. For example, the look angle is constrained to within 20 degrees of the vertical, the descent rate of the submunition is now within 5 percent, and target size is 12 pixels in length (plus or minus 10 percent). As will be shown later, the performance of the algorithms under captive flight test conditions greatly exceeded the design performance envelope and proved to be very robust and generic in nature.

The target acquisition algorithms consist of a parallel set of three independent detection algorithms operating independently and simultaneously to find potential target-like clusters, as shown in Figure 2.2. Each algorithm views the image from a different point of view:

1) MLC is a Maximum Likelihood Classifier which utilizes joint edge events in a statistical matching procedure to find targets.

2) GTIR is a modification to target-to-interference ratio matched filter. It is designed to find compact anomalous regions which are often associated with targets.

3) VSC is a Video Spatial Clustering algorithm which examines a target-sized region according to its grey-level statistics. An optimal threshold is computed and certain geometrical features computed and compared against those associated with known targets.

4) HPL is a Hierarchical Prioritization Logic which combines the output from the above three algorithms to form a composite measure of target probability. HPL also provides a confidence measure for each detection.

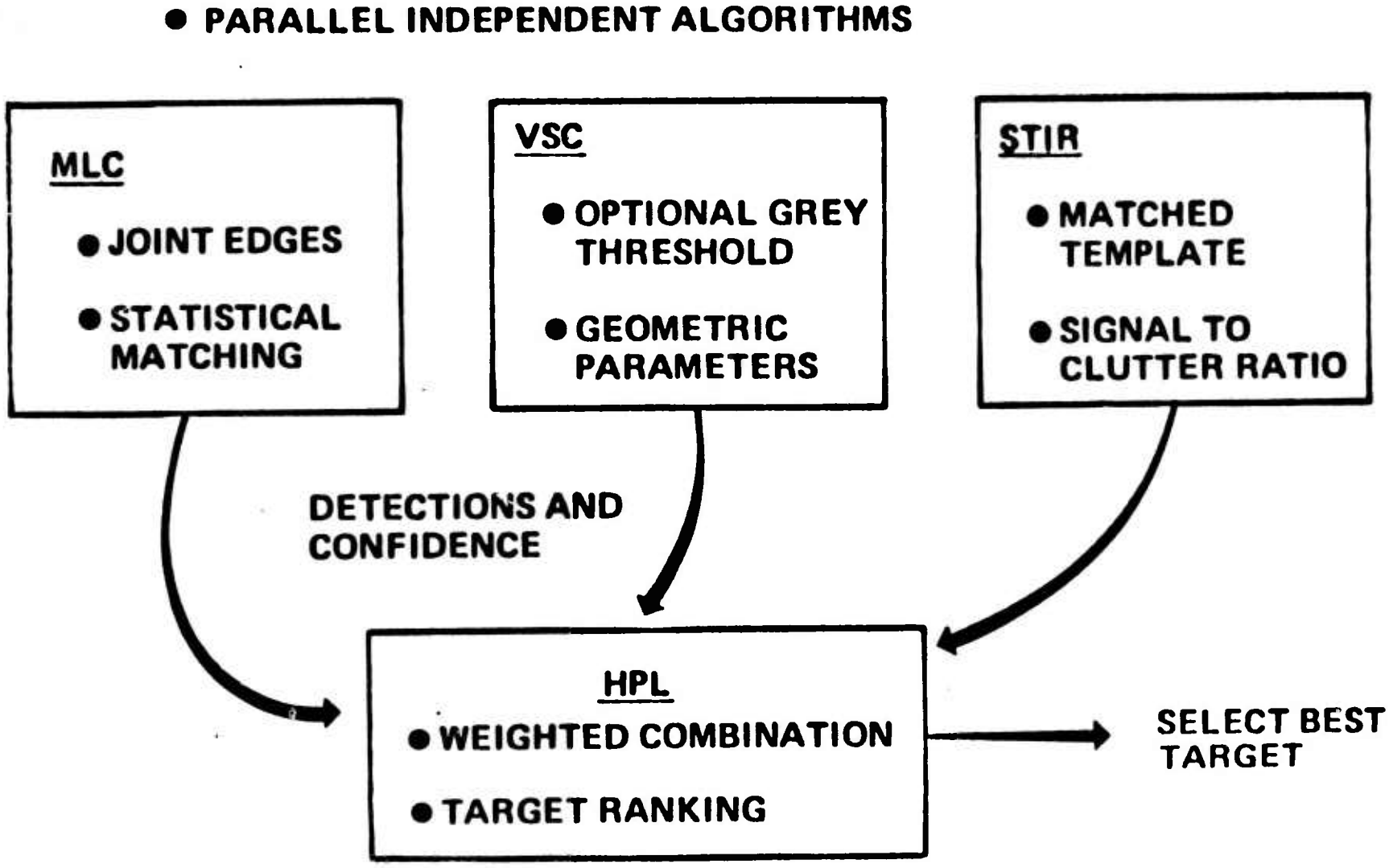


Figure 2.2. NATS/ASSP Autonomous Acquisition Processor.

The NATS/ASSP real-time processor is mechanized as a distributed computing net tailored to the processing requirements of the ASSP/NATS algorithms. The overall block diagram for the system is given in Figure 2.3.

2.1.2.1 Data Collection/Data Base Preparation

Integration of the Northrop visual based algorithms into the infrared spectrum requires image quality infrared data for training. Prior to contract award, an extensive survey of existing Government data bases was conducted to locate image quality infrared data from a top-down viewpoint. No such data was available. Therefore, a special data base needed to be collected which would meet these requirements. Under Northrop asset, a versatile platform was constructed which could be quickly installed into any standard UH-1 helicopter. A full description of this asset and the data collection effort is presented in Section 2.3 Test.

Field data was collected using the Northrop Pilot's Night Vision System, FLIR, common module FLIR. Chow boards were installed in order to reduce the horizontal streaking inherent in the common module sensor. However, residual streaking of up to 15% of the dynamic range remained. This residual streaking was removed by application of a novel data directed statistical filter called the GARBER-FILTER. A complete description of this filter can be found in Appendix A.

Laboratory calibration of the FLIR sensor indicated that in order to simultaneously meet the image quality specifications of 0.1 degrees MRT and 50% pixel-to-pixel step response, the resulting infrared image could be no larger than about 140 x 140 pixels. The planned infrared sensor for a tactical sized autonomous munition will be a staring focal plane array. Northrop is developing a 128 x 128 element staring array with an anticipated 0.05 to 0.10 MRT. The standard image size was therefore set to 128 x 128 pixels, with 8 bits per pixel. A data base was digitized according to this format. Raw data was digitized at a 17 MHz rate to accommodate the 875 line scan rate inherent in the FLIR sensor, producing a 512 x 384 pixel image. This image was GARBER-FILTERED to remove streaking and reduced 3:1 in size by averaging in both the horizontal and vertical dimensions, followed by subsampling to maximize the signal-to-noise ratio. The final image was cropped to the required 128 x 128 pixel standard.

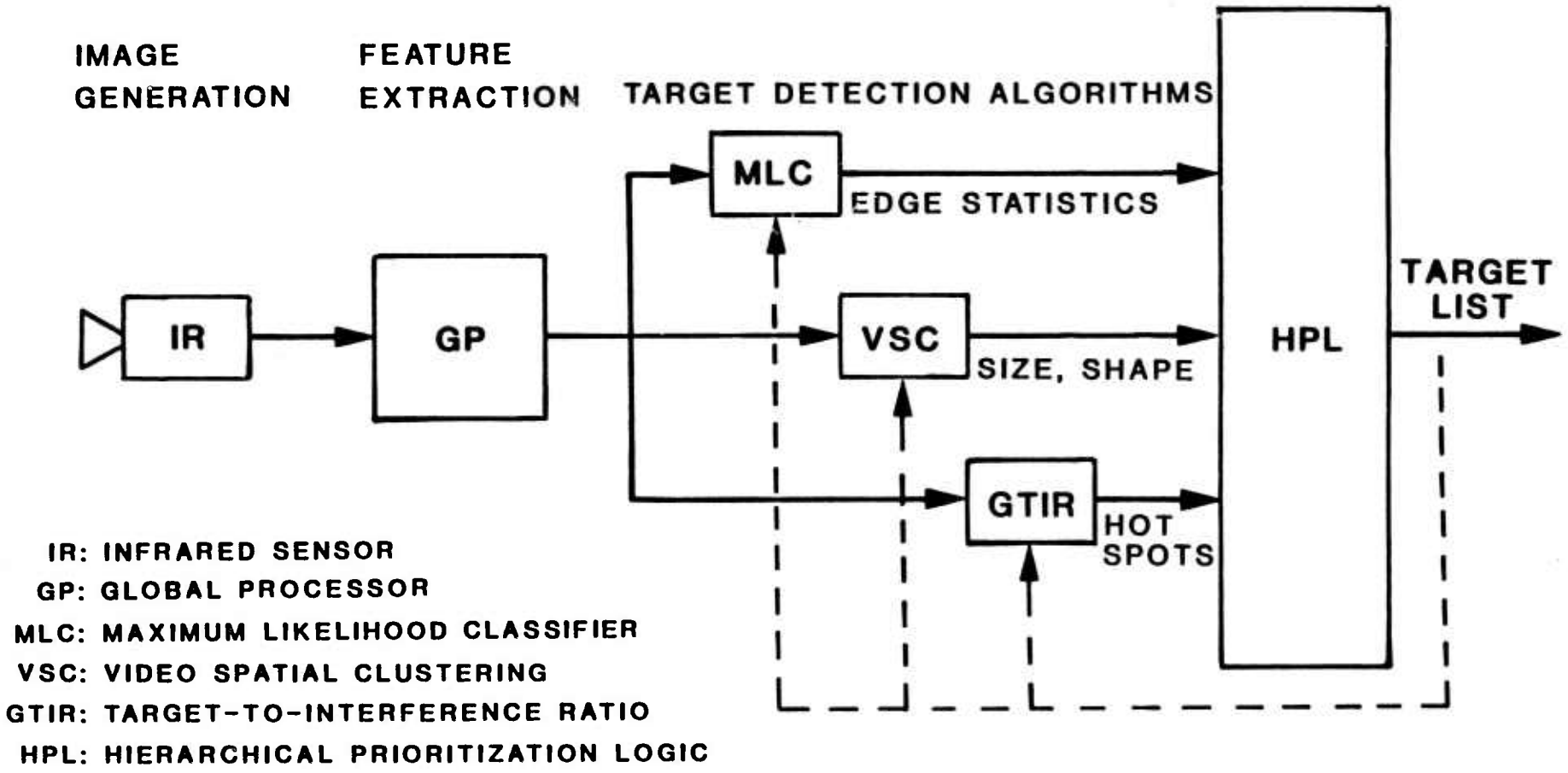


Figure 2.3. NATS Processor Functional Block Diagram.

The data base for training and testing consisted of 935 images covering a wide range of scenario conditions. These included tanks and a few trucks and jeeps, early morning and afternoon flights, and sunny, overcast, and rainy days. The data base was ground truthed by vehicle type. Some of the data base was ground truthed according to different types of background, e.g., buildings, trees, roads, clutter, tank tracks, cultural regions, fields, etc.

The data base consists of two separate file sequences which are designated in a prefix-index nomenclature. The first sequence consists of the filtered images, which have the prefix PNVS or PNKD and the index 1xxx, where xxx is the file index number which ranges from 1 to 935. The PNVS data is stored on line in the computing system in an extended format known as RTPAC. This format requires that the image be represented as a vector with four components: 1) raw video, 2) gradient magnitude, 3) edge direction map quantized to eight directions, 4) joint-edge direction map quantized to eight joint-edge directions. The Compass Gradient Edge Detection method is used for calculating the image gradients and edge directions. The PNKD data is stored in a similar format, except that the calculations for gradients, edges, and joint-edges are identical to the manner used by the NATS/ASSP processor. The second sequence of files contain the ground truth information. These files have the prefix PNTR and indexes which correspond with the PNVS and PNKD files.

2.1.3 Algorithm Description

This section describes the NATS/ASSP algorithms as embodied in the real time processor and tested during the captive flight test of November 1983. The algorithms are shown in their canonical form to emphasize the information processing aspects rather than the actual computational process.

2.1.3.1 GTIR Canonical Form

The GTIR (Garber's Target-to-Interference Ratio) algorithm is an anomoly detector for locating compact hot and cold regions. The filter is decomposable into a cascade of one-dimensional convolutions, hence it is computationally efficient. The features used to discriminate targets from the background are contrast, GTIR output amplitude, and cluster size.

Figure 2.4 illustrates the feature extraction process for the GTIR algorithm. This is a convolutional filter. Every point in the image is processed. An 18 x 18 pixel target-sized window is selected for filtering. This window is large enough to completely contain a tank which is 12 pixels in length for any angle of rotation.

The grey level variation, $\tilde{\sigma}$, in the pixels near the border of the window is calculated from a horizontal component, H_D, and a vertical component, V_D. These components are obtained by summing the absolute values of the differences in intensities between corresponding pixels along the horizontal and vertical boundaries of the 18 x 18 window. A lookup table functionally combines the horizontal differences, H_D, and vertical differences, V_D, into an estimate of this grey level variation. The functional form for the combining is given by:

$$f(h_D,V_D) = 3/4\ \mathrm{Min}(H_D,V_D) + 1/4\ \mathrm{Max}(H_D,V_D) = \tilde{\sigma}$$

This functional form was developed to discriminate against adjacent clutter such as roads and trees. The estimated grey level variation is associated with the point at the center of the 18 x 18 window.

The mean of the window border points is also required as is the average of the 3 x 3 pixel region in the center of the 18 x 18 window. The absolute value of their difference is a measure of the local contrast difference between the target and the background. Note that the center 3 x 3 region is small enough to encompass only target pixels, regardless of target orientation. It is mandatory that the statistics extracted by the background and target windows not come from mixed populations.

The actual GTIR measure is obtained by lookup table. The local contrast difference is divided by the grey level variation of the background resulting in a Z-score statistical test. This operation is similar to applying an optimal least mean squared error matched filter for square objects.

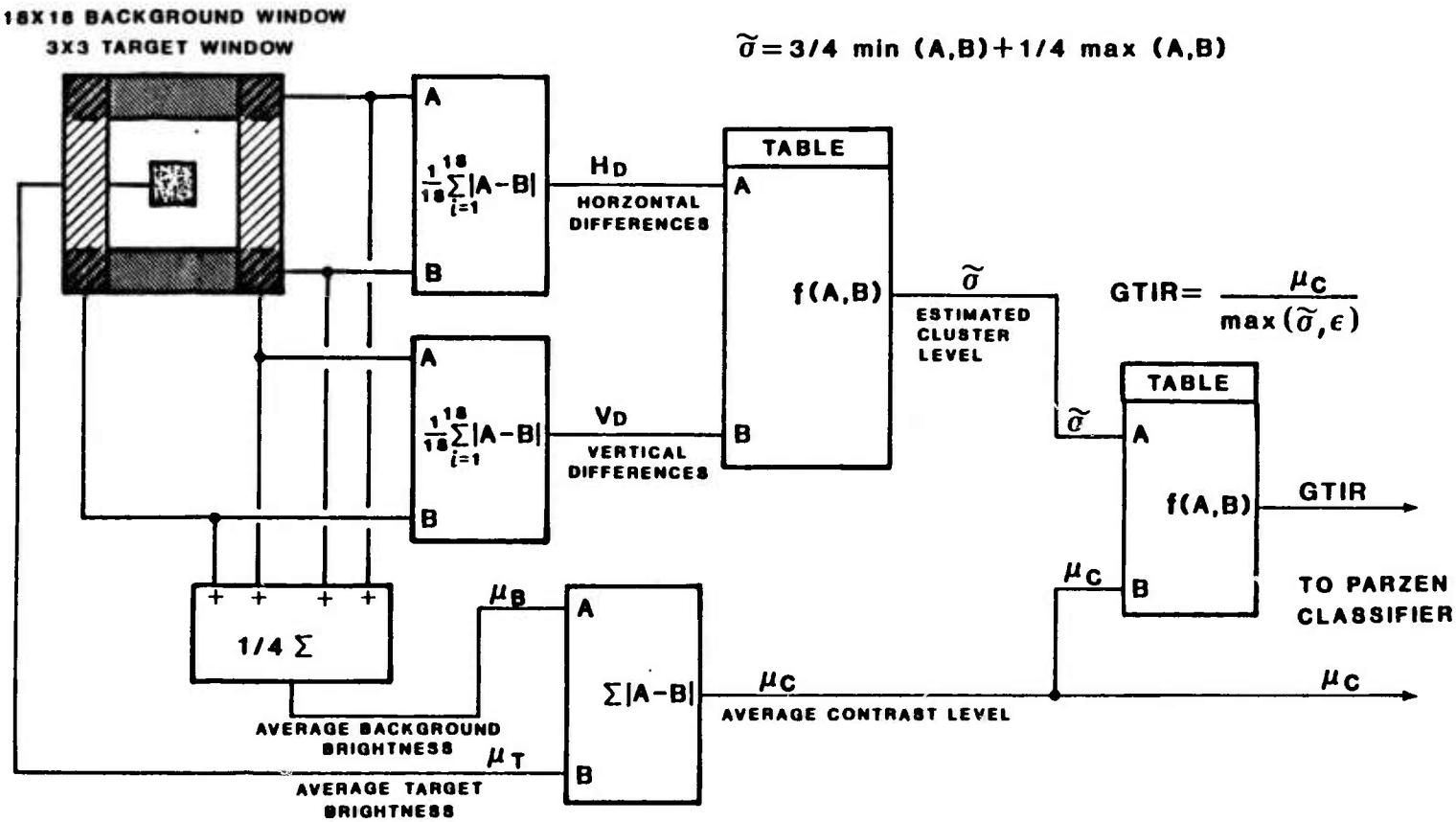


Figure 2.4. GTIR Canonical Form.

Since the GTIR operation is applied convolutionally, small clusters of correlated GTIR values occur around target sized anomolies. In Figure 2.5, the GTIR map is thresholded at 1.75 to eliminate poor signal to noise ratio responses and the remaining points are clustered. A feature vector for each compact cluster is formed consisting of: 1) Cluster size, 2) Maximum GTIR within the cluster, and 3) the local contrast difference at the centroid of the cluster. This feature vector indexes a 3-dimensional Parzen based classifier to produce an estimate of probability of target and confidence. Parzen based classification is fully described in 2.1.3.6.

The results of the classification is passed to the Hierarchical Prioritization Logic (HPL) for combining with the estimates made by the other algorithms.

2.1.3.2 MLC Canonical Form

The Maximum Likelihood Classifier, MLC, exploits edge connectivity information for recognition of targets under diverse thermal or environmental conditions. MLC utilizes a 4-dimensional feature vector to discriminate targets from clutter:

1) Major Straightness
2) Minor Straightness
3) Clockwise Curvature
4) Counter Clockwise Curvature.

Before describing the details of MLC, we must first establish some necessary background in edge detection.

Edge Detection and Joint Edge Generation

At each pixel of the input IR image, the gradient intensity and the gradient (or edge) direction need to be known. The SOBEL operator was used to extract both gradient and edge information:

$$HG = \begin{bmatrix} 1 & 1 & 2 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & -1 & -2 & -1 \end{bmatrix} \quad VG = \begin{bmatrix} 1 & -1 & 0 & 1 \\ 1 & -2 & 0 & 2 \\ 1 & -1 & 0 & 1 \end{bmatrix}$$

$$GRAD = SQRT\ (HG^2 + VG^2) \qquad \text{Gradient Magnitude}$$

$$EM = QUANTIZE\ \big(ATAN2(HG,VG)\big) \qquad \text{Edge Direction Map}$$

Where QUANTIZE labels the edge directions in a counterclockwise direction in increments of 22.5 degrees, as shown in Figure 2.6.

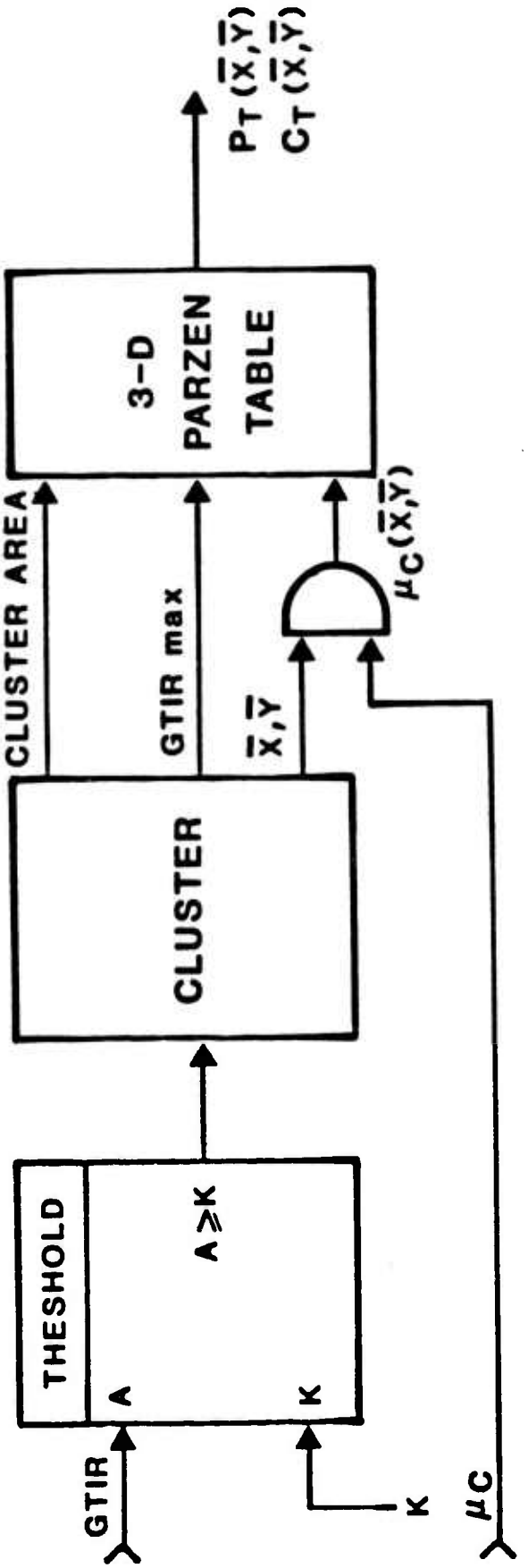


Figure 2.5. GTIR Canonical Form.

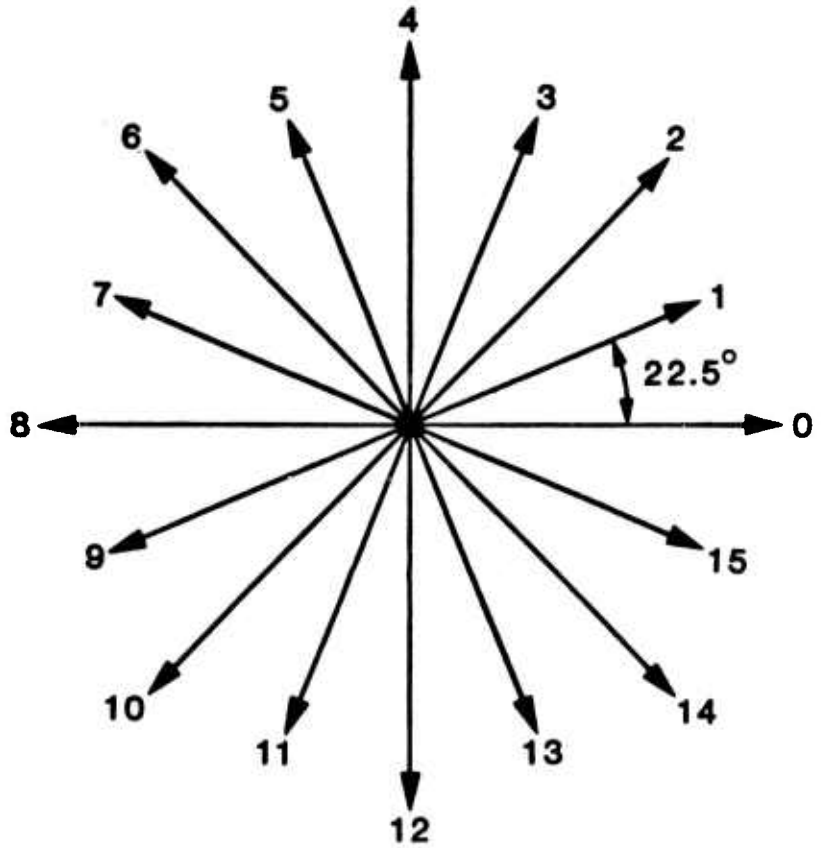


Figure 2.6. Quantized Edge Directions.

The accuracy of the edge direction measurement depends heavily upon the local signal-to-noise ratio. Theoretical work (Reis, et. al., 1978) showed that for additive Gaussian noise, the contrast-to-noise ratio (CNR) must exceed 4 to be 95% confident that the measured edge direction is within 22.5 degrees of the true edge direction 95% of the time. This is sufficient justification to discard those edge direction measurements associated with weak gradient magnitudes, since they are clearly noise dominated and detrimental to decision making.

The above operators are applied to the entire 128 x 128 image. In addition to the edge map, a joint edge map is also required. The joint edge map provides first order connectivity and forms the basis for a powerful rotationally invariant set of recognition features. A joint edge map, JEM, is computed from the edge map, EM, by applying a 3 x 3 convolutional operator which uses the center pixel's edge direction to select a corresponding edge direction from those of neighboring pixels, thus forming an ordered

pair of edge directions for each pixel in the 128 x 128 image. This ordered pair therefore consists of

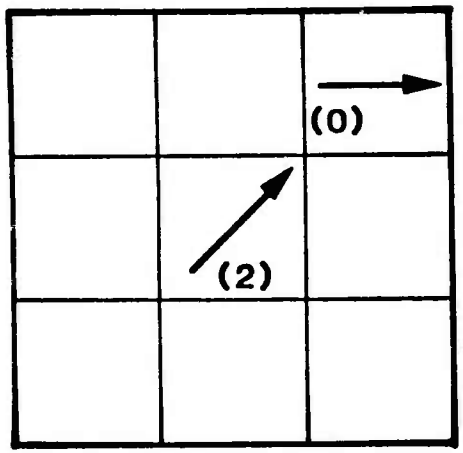
JEM = (reference-pixel's-direction, pointed-to-pixel's-direction).

The process is depicted in Figure 2.7. Figure 2.7A shows the center pixel's edge direction pointing exactly to a neighboring pixel, resulting in that pixel's edge direction becoming the second component of the ordered pair. Figure 2.7B shows the center pixel's edge direction pointing half way between two adjacent neighbors, which happens for each of the odd numbered edge directions. In these cases, the edge directions from both neighboring pixels are examined, and the one that is closest to the center pixel's edge direction is chosen as the second component of the ordered pair. The "average" edge direction is used for those cases where the "pointed-to" neighbors are equal distance from the center pixel's direction.

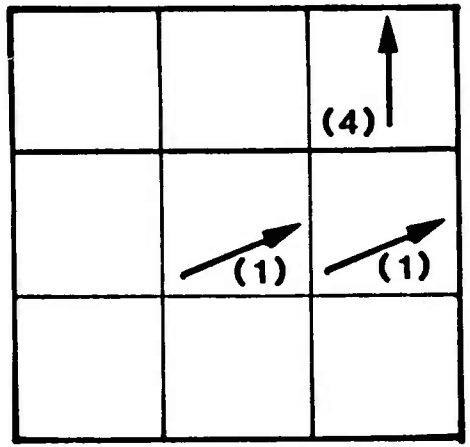
The original edge directions were quantized in 22.5 degree increments to reduce aliasing of joint edge directions for image objects oriented at such angles. For example, suppose a tank is oriented at a 22.5 degree angle relative to the horizontal axis of the sensor. If the edge directions are quantized in 45 degree increments, then the slightest noise is sufficient to cause the boundary pixel's edge direction to change from horizontal to diagonal, or vice versa. This would in turn cause the joint edge events to change randomly.

The resulting ordered pair of joint edge events are mapped into rotationally invariant features as shown in Figure 2.8.

The straightness feature is divided into Perpendicular Straightness and Diagonal Straightness. This provides for accumulation of joint edge events corresponding to rectangular objects rotated at 45 degree intervals. The +C and -C curvature features are indicative of dark objects on lighter backgrounds and lighter objects on darker backgrounds, respectively. These types of slowly changing edge directions are frequently associated with man-made objects. The remaining joint edge possibilities, labeled



A: CENTER PIXEL EXACTLY POINTS TO A NEIGHBOR
JOINT EDGE = (2,0)



B: CENTER PIXEL POINTS EXACTLY HALF-WAY BETWEEN NEIGHBORS
JOINT EDGE = (1,1)

Figure 2.7. Joint Edge Map Generation.

		EDGE DIRECTION IN "POINTED-TO" PIXEL															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
C	0	P	P	-C	-C	R	R	R	R	R	R	R	R	R	R	+C	+C
E	1	P	P	-C	-C	R	R	R	R	R	R	R	R	R	R	+C	+C
N	2	+C	+C	D	D	-C	-C	R	R	R	R	R	R	R	R	R	R
T	3	+C	+C	D	D	-C	-C	R	R	R	R	R	R	R	R	R	R
E	4	R	R	+C	+C	P	P	-C	-C	R	R	R	R	R	R	R	R
R	5	R	R	+C	+C	P	P	-C	-C	R	R	R	R	R	R	R	R
	6	R	R	R	R	+C	+C	D	D	-C	-C	R	R	R	R	R	R
D	7	R	R	R	R	+C	+C	D	D	-C	-C	R	R	R	R	R	R
I	8	R	R	R	R	R	R	+C	+C	P	P	-C	-C	R	R	R	R
R	9	R	R	R	R	R	R	+C	+C	P	P	-C	-C	R	R	R	R
E	10	R	R	R	R	R	R	R	R	+C	+C	D	D	-C	-C	R	R
C	11	R	R	R	R	R	R	R	R	+C	+C	D	D	-C	-C	R	R
T	12	R	R	R	R	R	R	R	R	R	R	+C	+C	P	P	-C	-C
I	13	R	R	R	R	R	R	R	R	R	R	+C	+C	P	P	-C	-C
D	14	-C	-C	R	R	R	R	R	R	R	R	R	R	+C	+C	D	D
N	15	-C	-C	R	R	R	R	R	R	R	R	R	R	+C	+C	D	D

P = PERPENDICULAR STRAIGHTNESS D = DIAGONAL STRAIGHTNESS
+C = CLOCKWISE CURVATURE -C = COUNTER CLOCKWISE CURVATURE
R = RESIDUALS

Figure 2.8. Joint Edge Mapping Into Features.

Residuals, are associated with very rapid or discontinuous spatial changes in image intensity found in textured backgrounds. Hence, the basis vectors for MLC are particularly well suited for extracting locally connected contour information and they are easily interpreted in terms of image content.

Extraction of MLC Primitives

The extraction of the joint edge features used by MLC is illustrated in Figure 2.9. A target sized window of 15 by 15 pixels is convolutionally scanned across the image and the MLC target detection algorithm is applied to each pixel location. (Actually, the evaluation is performed at every 5th horizontal and vertical pixel to reduce the computational load without exceeding the width of the MLC response function.)

An average neighborhood gradient value is computed for each non-overlapping block of 5 by 5 pixels for use in the global-local adaptive threshold test. The average gradient value for the entire 128 x 128 image is also required.

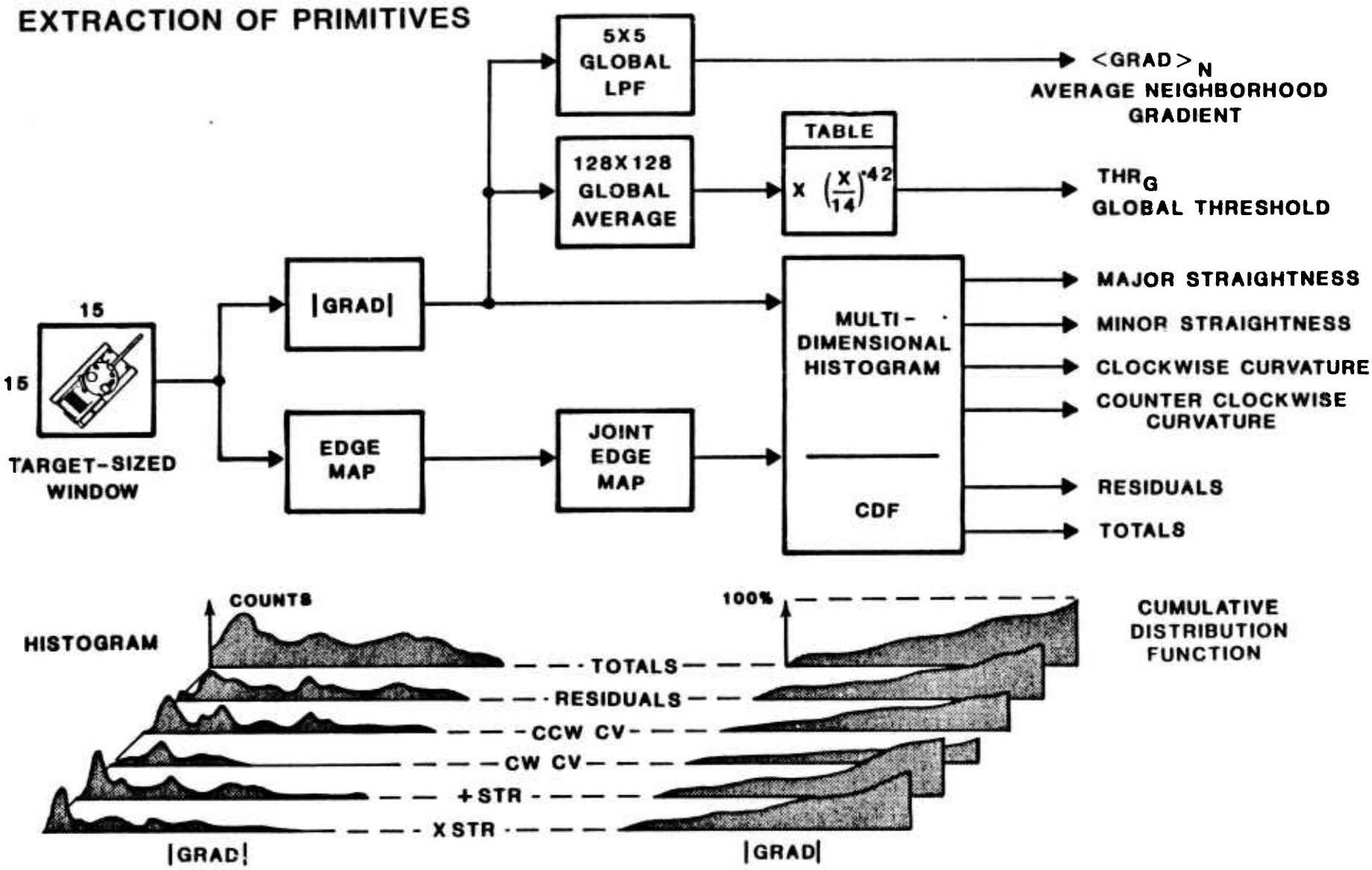


Figure 2.9. MLC Canonical Form (1 of 5).

A global threshold value is functionally related to this average global gradient and is computed via lookup table. The coefficients 14 and 0.42 were determined experimentally.

Histograms of the Joint Edge Features for the current 15 x 15 pixel target sized window are computed as a function of gradient magnitude. Since typically less than 50 events are available for computing the joint edge pdf's, the 8-bit gradient magnitude is reduced to 5-bits in dynamic range by division by 8. This scaling reduces the number of histogram bins from 0:255 to 0:31, reflecting the underlying precision of the pdf's being estimated. These histograms are converted into cumulative density functions by integration.

MLC Global-Local Adaptive (GLAD) Threshold Calculation

As stated earlier, it does not make sense to apply the MLC algorithm to areas of insufficient contrast. Also in the presence of high contrast interference, such as from a paved road, the target may not be the highest contrast feature. As a result a sophisticated thresholding algorithm that incorporates global as well as local adaptivity had to be developed. It is called GLAD, Global-Local Adaptive thresholding algorithm. The GLAD threshold calculation is shown in Figure 2.10. The GLAD clutter removal threshold value is determined by MIN $[T_A, T_B]$. T_B, in turn, is the product of three factors,

$$T_B = T_G T_\mu T_\sigma \quad = \text{Background Threshold}$$

$$T_G = [\mu_G/14]^{0.42} \quad = \text{Global Component}$$

where μ_G = global average of gradient.

T_μ = average of 16 gradient values in neighborhood window.

$T\sigma = 6.5\, T_w^{-0.36}$ = estimate of standard deviation,

where $T_w = 1/2\,[MAX_2 - MIN_2]$

= an order statistic estimate proportional to the standard of the 16 gradient values in the neighborhood of the MLC window.

$T_A = T_\mu + 5\, T_w$

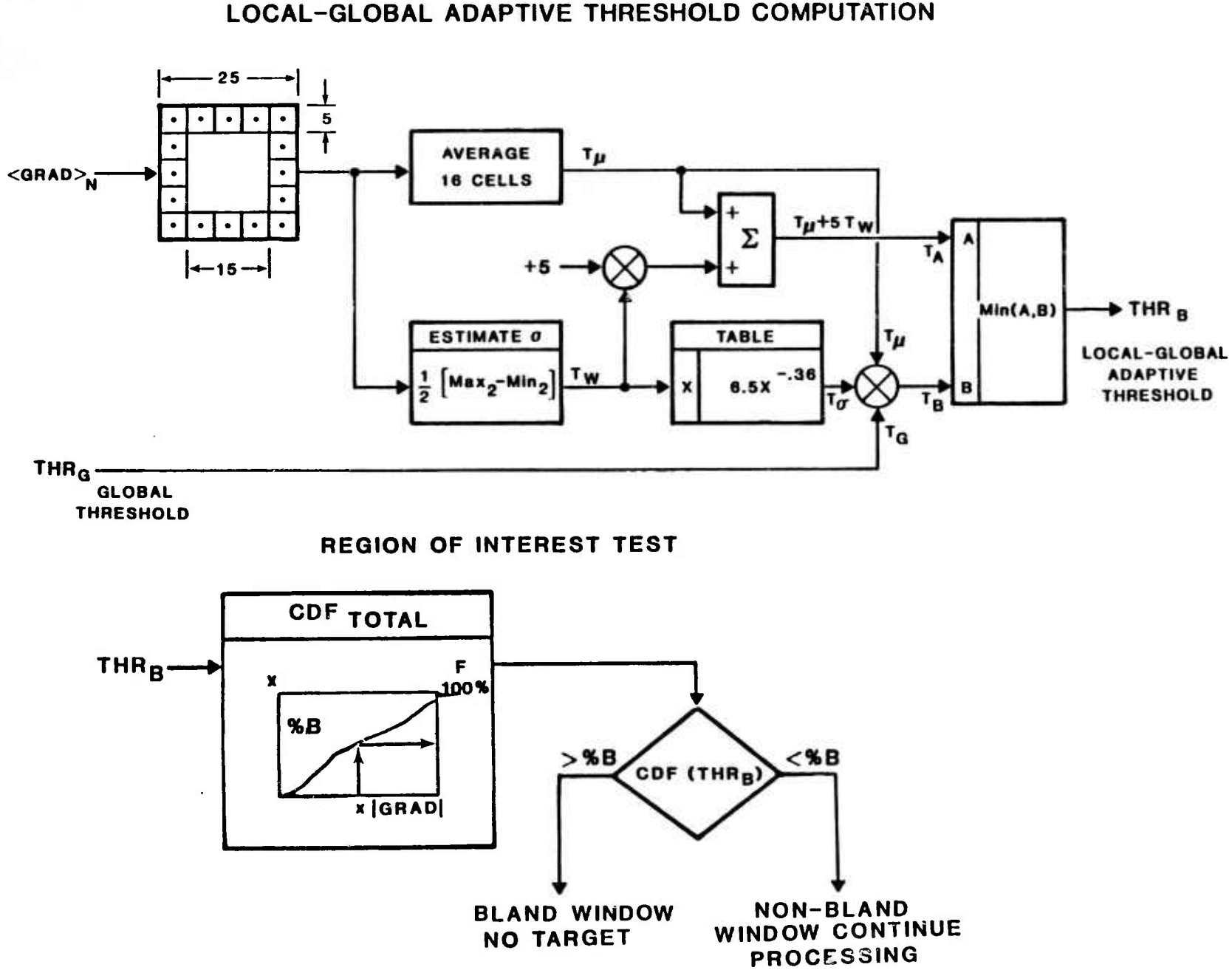


Figure 2.10. MLC Canonical Form (2 of 5).

For a target sized window of 15 by 15 pixels, the window is rejected as a region of interest if fewer than 12 pixels have gradient values that exceed MIN $[T_A, T_B]$. This result is insensitive to linear scalings in the original image.

The functional forms for T_G and T_σ were determined by noting the desired monotonicities among the parameters and then fitting actual measurements to candidate functional forms. T_A was defined to serve as a reasonable limit on T_B, in the event that T_w became too small.

MLC Feature Vector Formation

Feature vectors are calculated for those windows which survive the GLAD thresholding algorithm. The gradient magnitude range is divided into three overlapping intervals, designated lower, middle, and upper. These intervals are defined in terms of percentiles on the total number of histogrammed events as shown in Figure 2.11. Table 2.1 defines the intervals used by the real time processor. Multiple intervals are used to help detect targets adjacent to high contrast clutter or targets partially occluded by background clutter such as trees.

Table 2.1. Interval Definitions.

i	INTERVAL	LOWER %	UPPER %
1	LOWER	62	87
2	MIDDLE	75	99
3	UPPER	50	99

The feature vectors are formed as follows.

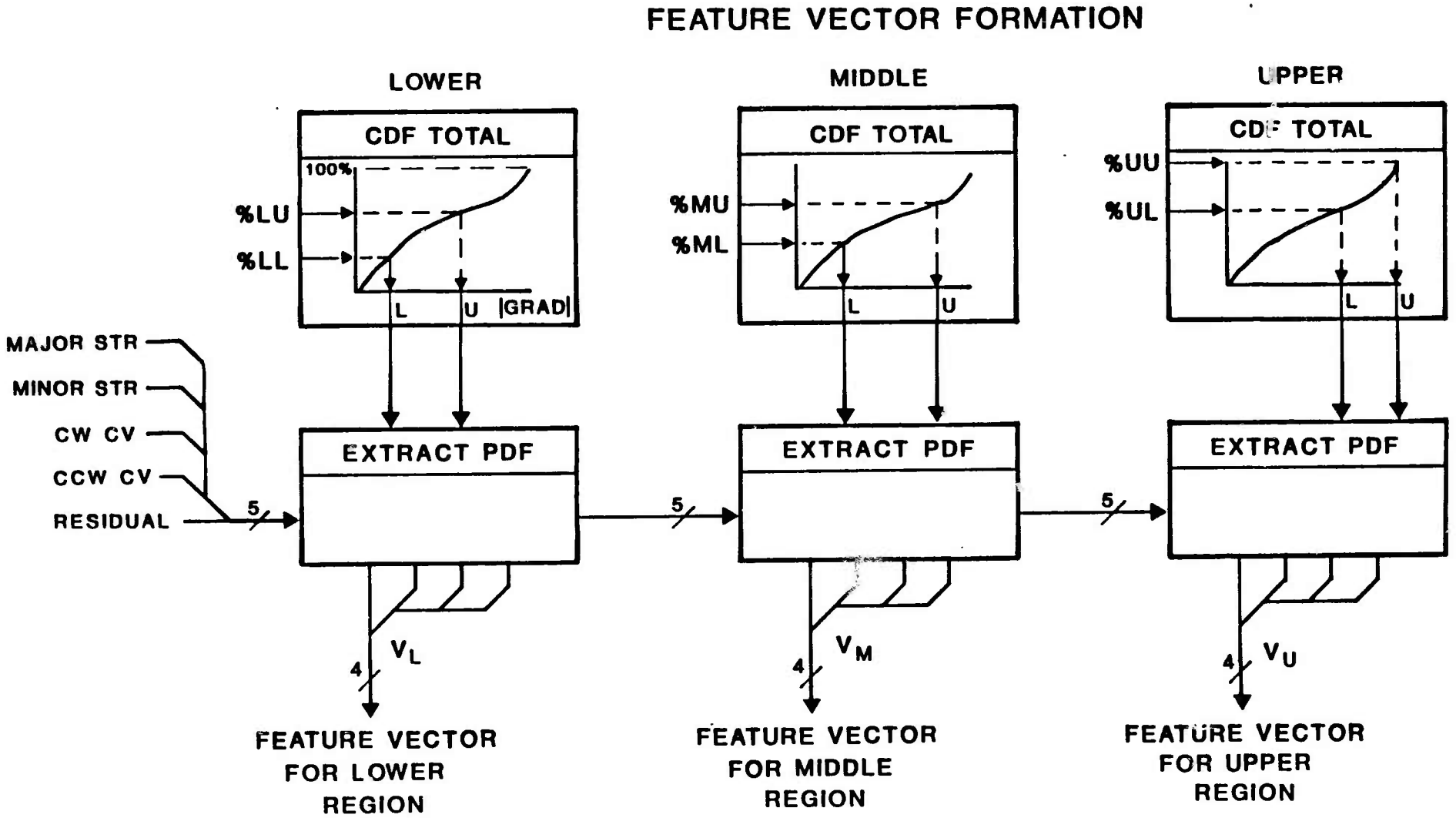


Figure 2.11. MLC Canonical Form (3 of 5).

Define the multi-dimensional CDF, F(g,k),

F(g,k) = Cumulative Density Function for the kth Primitive

k = 1 Major Straightness (MAJ STR)
k = 2 Minor Straightness (MIN STR)
k = 3 Clockwise Curvature (CW CV)
k = 4 Counter Clockwise Curvature (CCW CV)
k = 5 Residuals
k = 6 All of the above
5 = Scaled Gradient Magnitude (0:31)

For each of the three intervals defined in Table 2.1,

L(i) = Gradient Magnitude corresponding to the Lower Percentile for the ith interval

U(i) = Gradient Magnitude corresponding to the Upper Percentile for the ith interval

The feature vectors are defined by

$$V(j,i) = [F(U(i),j) - F(L(i),j)] / [F(U(i),6) - F(L(i),6)]$$

For j = 1,2,3,4 (Primitives)
And i = 1,2,3 (Intervals)

The numerator of V(j,i) is the 'probability' of the jth primitive for the ith interval. The denominator is the total 'probability' for all primitives in the ith interval, which normalizes the sum of the components of the feature vector to unity.

Maximum Likelihood Classification

The feature vectors for the three intervals are used to index a lookup table which contains the probability of target for every possible vector, as shown in Figure 2.12. This table utilizes the Parzen density estimation technique described in Section 2.1.3.6 to map feature vectors into probability of target, P_T.

The maximum and minimum P_T over the three intervals is compared against $P_T = 0.5$, with the largest deviate determining the class label and the P_T for the current window. Bland windows are assigned $P_T = 0.0$. Those windows

MAXIMUM LIKELIHOOD CLASSIFICATION

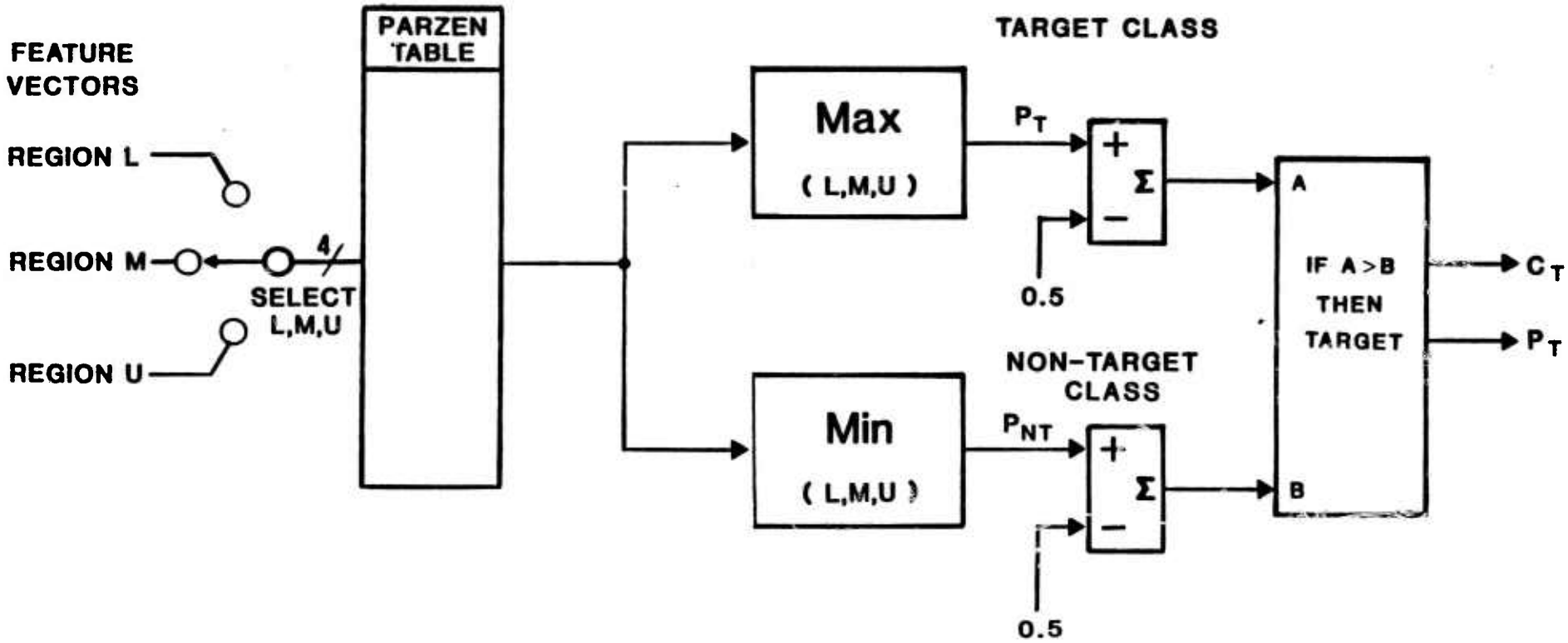


Figure 2.12. MLC Canonical Form (4 of 5).

which have a $P_T > 0.5$ are sent to the post processor for clustering. The confidence estimate, C_T, was not implemented.

MLC Post Processor

Since several window positions respond to the same target, a post processor is used to cluster these individual window level detections to form a cluster level detection, Figure 2.13. The resulting cluster size is mapped via a Parzen table into the probability of target for the cluster, which is sent onto HPL for combining with similar detections made by the other acquisition algorithms.

2.1.3.3 VSC Canonical Form

The Video Spatial Clustering algorithm exploits grey level information for the detection of targets. A four-dimensional feature vector incorporates two global grey level components, one local grey level component, and one component corresponding to cluster size. The VSC algorithm is normally cued by one or both of the other acquisition algorithms since it is computationally slower.

The canonical form for VSC is shown in Figure 2.14. An 18 by 18 pixel target sized window is histogrammed in preparation for computing the optimal threshold. This threshold is calculated as follows:

$$T_{opt}(k_1,k_2) = (k_1,k_2) \text{ such that } \sigma_b^2 \text{ is maximized}$$

where σ_b^2 = Between Class Variance $= w_0\mu_0^2 + w_1\mu_1^2 + w_2\mu_2^2 - \mu_T^2$

for $0 \le k_1 < k_2 \le 255$

$$P_i = n_i/N \qquad \mu_T = \sum_{i=0}^{255} (i+1)p_i$$

n_i = number of histogrammed pixels of grey level i

N = total number of pixels in the histogram

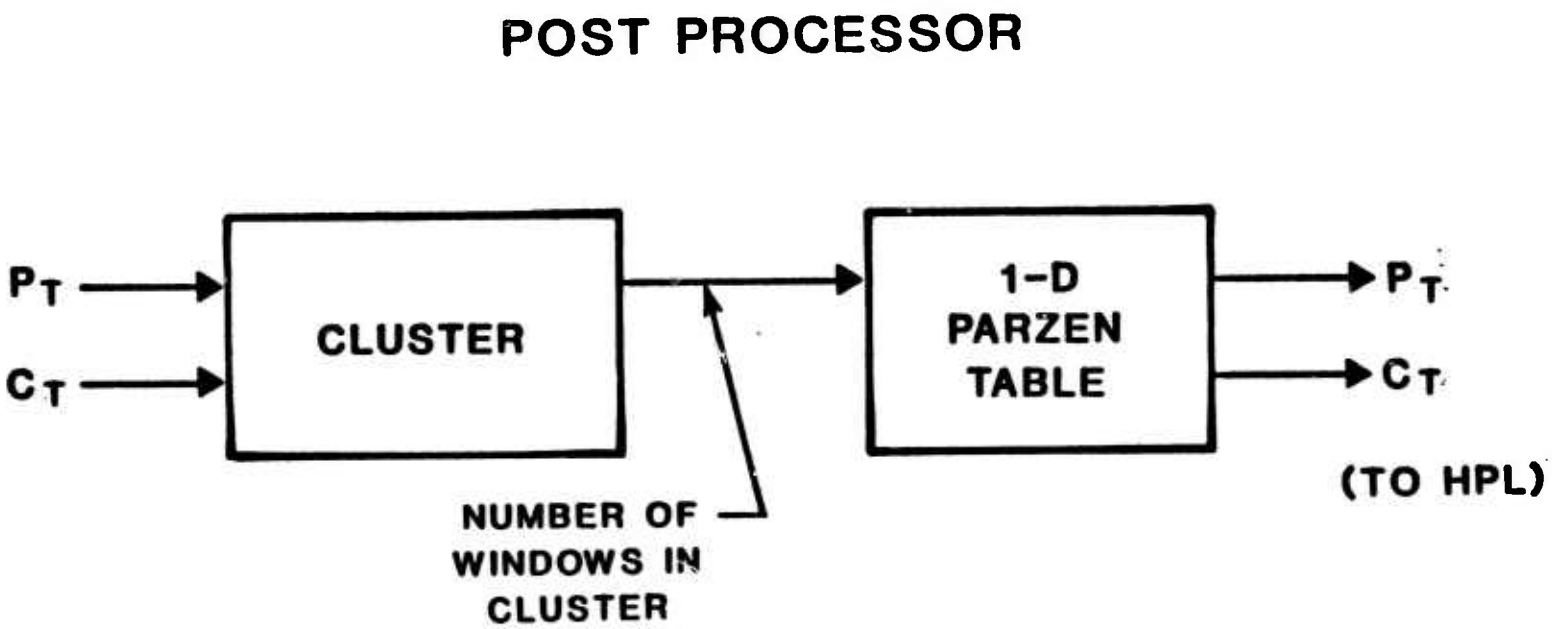


Figure 2.13. MLC Canonical Form (5 of 5).

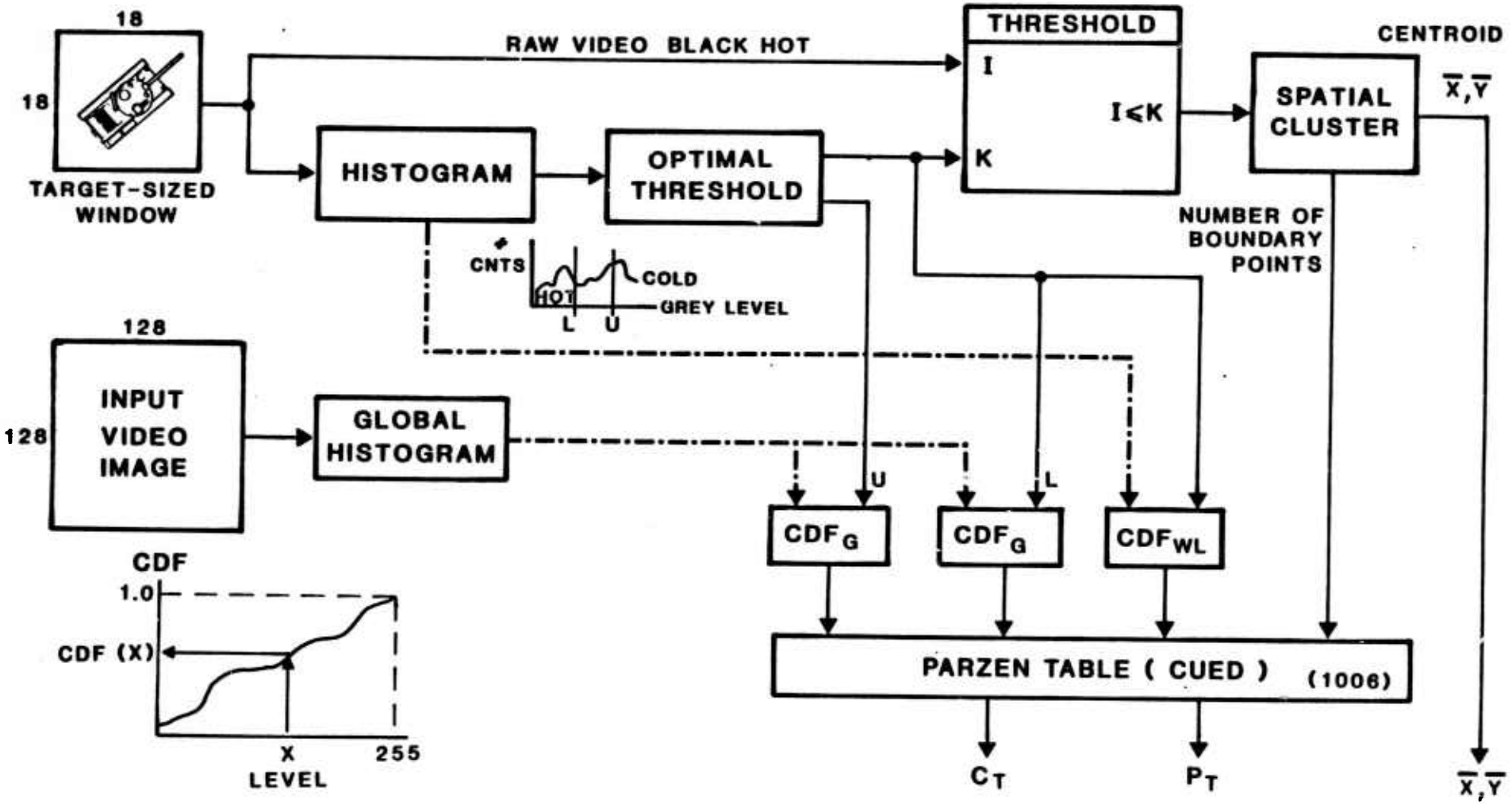


Figure 2.14. VSC Canonical Form.

$$w_0 = \sum_{i=0}^{k1} p_i \qquad w_0\mu_0 = \sum_{i=0}^{k1} (i+1)p_i$$

$$w_1 = \sum_{i=k1+1}^{k2} p_i \qquad w_1\mu_1 = \sum_{i=k1+1}^{k2} (i+1)p_i$$

$$w_2 = \sum_{i=k2+1}^{255} p_i \qquad w_2\mu_2 \quad \sum_{i=k2+1}^{255} (i+1)p_i$$

The first two components of the feature vector are derived from the grey level percentiles from the global 128 by 128 pixel histogram for optimum threshold values k_1 and k_2.

The grey level percentile which corresponds to the optimum threshold value k_1 for the target sized window is the third component of the feature vector.

The target sized window is thresholded by value k_1 to isolate black-hot clusters. That cluster, if any, which contains the center point of the window is evaluated for size and centroid location. The number of boundary points in this cluster is the fourth component of the feature vector.

Probability of target is found by table lookup. The table was computed via the Parzen density estimation technique and was based upon a training set consisting of all target clusters detected by MLC or GTIR. It was anticipated that computationally VSC would be the slowest of the three target detection algorithms. Therefore VSC would always be in a "cued" mode, i.e., VSC would be directed to only examine those clusters found by MLC or GTIR (or both).

Since VSC incorporates clustering directly into the feature vector extraction process, no post processor is required. The probability of target is sent directly to HPL for incorporation into the final ranking.

2.1.3.4 HPL Canonical Form

The Hierarchical Prioritization Logic, HPL, combines individual algorithm detections into a single measure of target probability to determine composite target rankings for the image as a whole.

The HPL algorithm inputs the cluster-level target detection reports generated by the three independent acquisition algorithms. Since these individual reports are generated asynchronously with respect to each other, the combining of spatially correlated reports takes place in random order. As an aid in generating a composite target detection list as quickly as possible, the target acquisition algorithms are cueable by one another to areas of likely target activity, via HPL. When an algorithm locates a target-like object in the image, it reports it to HPL for incorporation into the composite rankings. Internally, HPL keeps an ordered doubly linked list of all target detections that have been reported by the detection algorithms. When an acquisition algorithm reports a detection to HPL, HPL generates a new composite position cue which is derived from all previous reports of the same detection by the other algorithms. When HPL cues a new algorithm to a detection by other algorithms, the algorithm being cued immediately processes that location. In the absence of a cue, the algorithm is free to choose for itself the next location to be processed. This (stochastic) method of joint mutual cueing enables the three acquisition algorithms to examine those regions of the image which are most target-like first. In this manner, the composite ranking list is built up as quickly as possible.

The canonical form for HPL is shown in Figure 2.15. The combining methodology is based upon a decision tree. When a new report is received by HPL it is examined for its information content. Some reports contain null information where an algorithm could not or does not make a probability of target assessment. If the report comes from MLC (M), GTIR (G), or VSC (V), and is non-null, then it moves to the 'single algorithm detection' decision level. For example, if the report comes from MLC then an attempt is made to merge this new report with previous reports from other algorithms. If the new report cannot be merged (e.g., a new detection), then a one-dimensional MLC Parzen table is utilized to map the input cluster-level report into a global-level report. In similar fashion, those reports which can be merged with two or three other algorithms are mapped by a two or three-dimensional Parzen table, respectively.

Since VSC is always cued (see 2.1.3.3), its one-dimensional single-algorithm detection Parzen table was not mechanized in the real time processor.

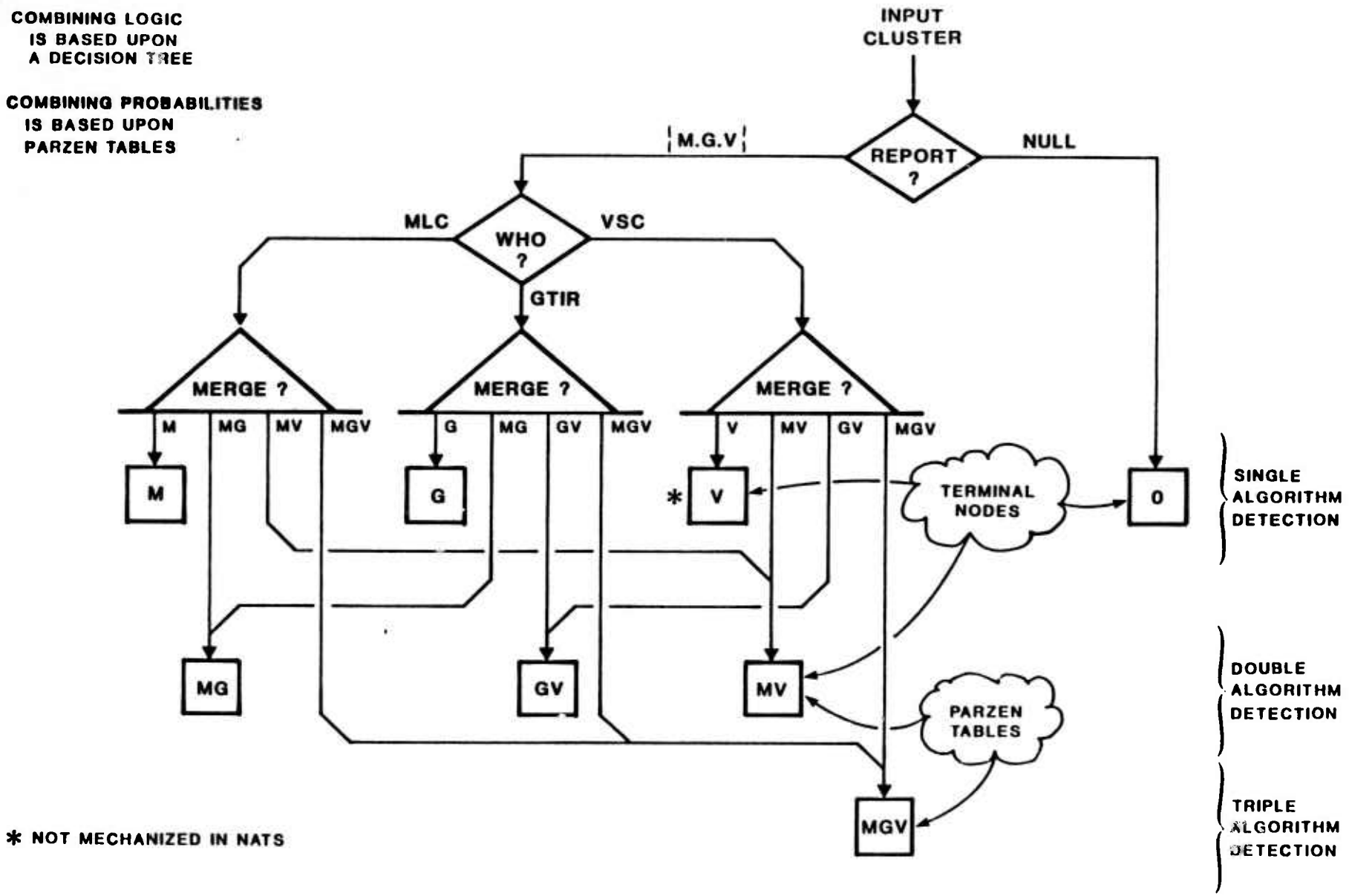


Figure 2.15. HPL Canonical Form (1 of 2).

The rule for merging separate reports into a combined detection report is simple. If the centroids of the MLC and GTIR cluster are within 9 pixels of each other, then the two reports are merged together as a double-algorithm detection. The merged (output) centroid is the average of the input centroids. Cued VSC reports are handled in exactly the same manner. When all of the image has been examined, or when time has run out, the composite target probabilities at the terminal nodes are sorted in descending order to determine final rankings, as shown in Figure 2.16.

2.1.3.5 Parzen Table Training Data

The amount of data used to train the various Parzen tables used in the real time processor is shown in Figure 2.17. One hundred and fifty images were selected from the data base as representative of the scenario. Processing of these images by GTIR produced 1621 feature vectors, MLC produced 1766 vectors, and VSC produced 1006 vectors. Remembering that VSC was always operated in the cued mode, only those individual or joint locations found by MLC and GTIR were candidate training samples (2377 such locations were defined, of which 1006 survived the VSC clustering operation).

The actual organization of HPL is somewhat different from that shown in the canonical form. Since VSC would be always cued, it was convenient to split HPL into two levels. HPL level I generates cues for VSC by combining where possible MLC and GTIR target detection reports. If VSC is successful in clustering the cued window, its report is combined by HPL Level II with the HPL Level I report to produce a composite probability of target. If VSC does not successfully cluster the cued window, then the HPL Level I report is combined with the fact that VSC failed through HPL Level II to produce a final composite probability of target.

There are three tables in HPL Level I which perform the combining of MLC and GTIR target reports. Two of these tables are for single algorithm detections, and one is for detections by both algorithms. For GTIR, of the 1621 input vectors derived from the training set, 956 vectors were not within the 9 pixel spatial limit for merging with MLC, hence, they were used to train the "GTIR ONLY" Parzen Table. Similarily, of the 1766 training vectors from MLC, 592 vectors could not be merged with GTIR, hence they were used

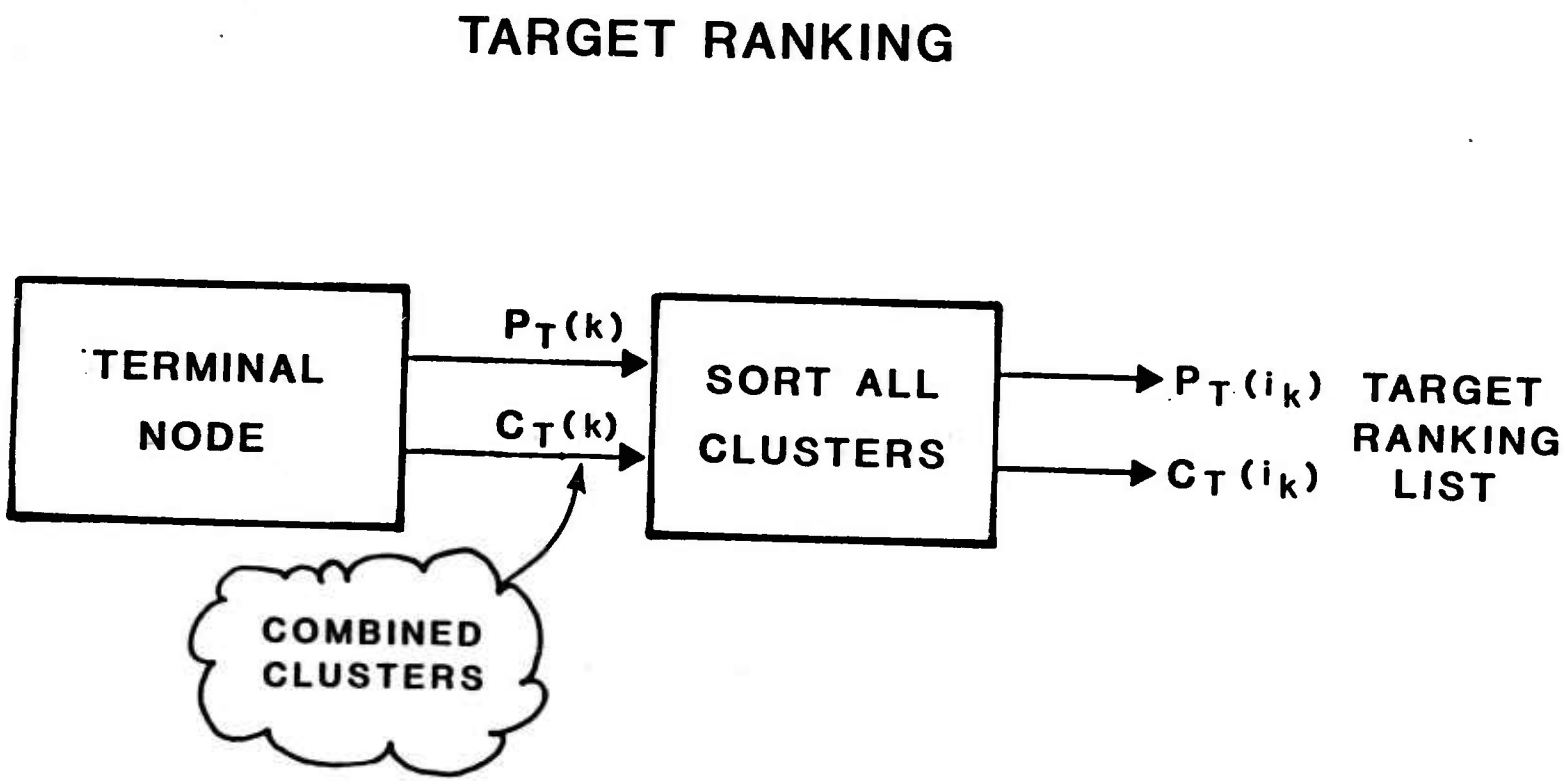


Figure 2.16. HPL Canonical Form (2 of 2).

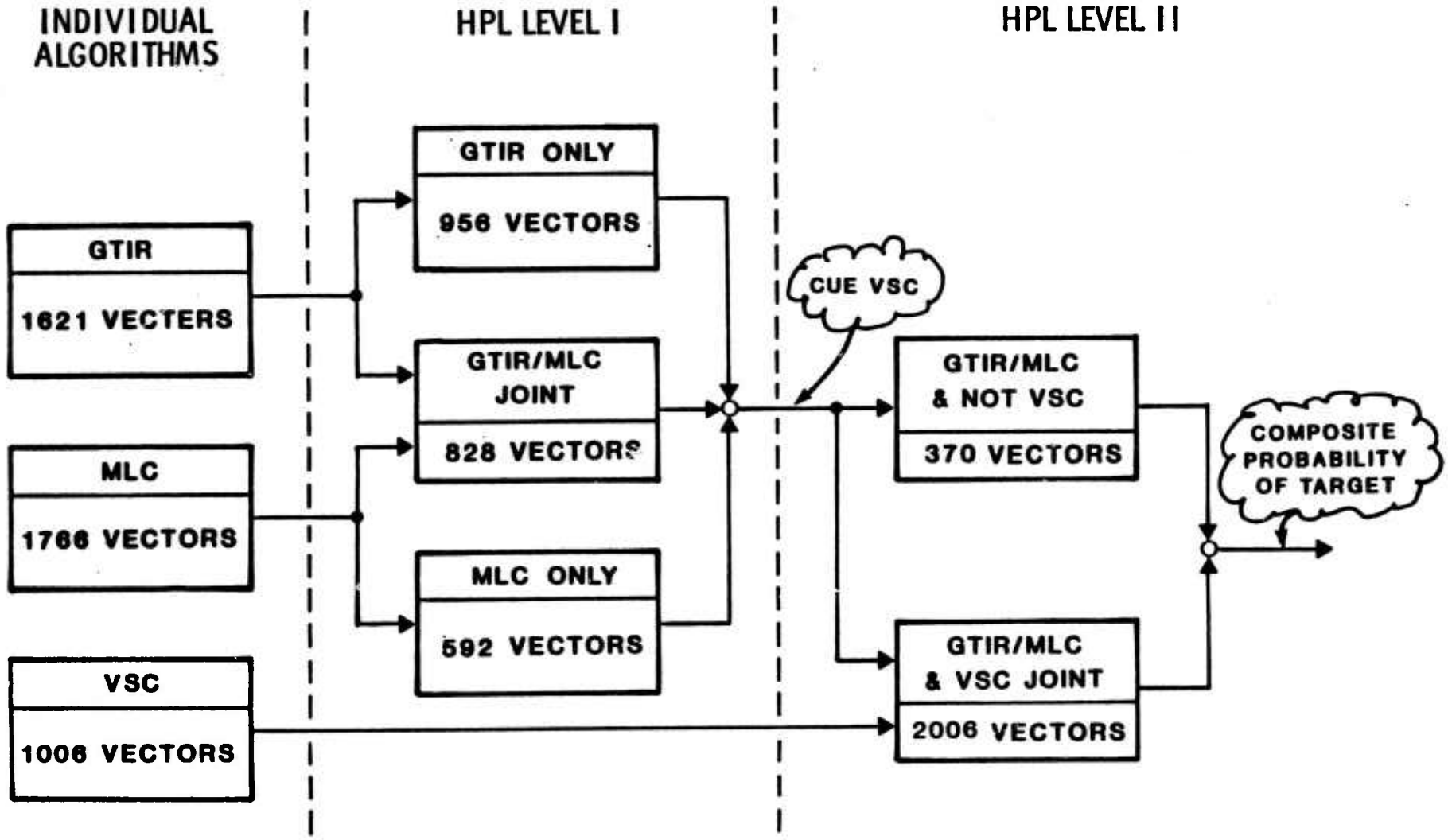


Figure 2.17. Parzen Tables Training Data.

to train the "MLC ONLY" Parzen Table. Therefore, from the available total of 3387 training vectors, 1548 correspond to single algorithm detections. The remaining 1839 training vectors are used to train the "GTIR/MLC JOINT" Parzen table. Due to mottling, it is very common for the individual acquisition algorithms to detect a single target more than once. These multiple detections were merged by HPL Level I resulting in a reduction of training vectors from 1839 to 828.

HPL Level II was trained upon the output from HPL Level I. Of the 2376 HPL Level I training vectors, 2006 produced valid VSC target detection reports and were used to train the "GTIR/MLC & VSC JOINT" Parzen table. The remaining 370 vectors were used to train the "GTIR/MLC & NOT VSC" Parzen table.

2.1.3.6 Parzen Based Classification

The core of the NATS/ASSP classification approach is based upon Parzen probability density estimation. This estimation technique was introduced in 1962 [2]. In spite of its early genesis, it apparently has not been previously applied to problems in automatic target recognition. The technique is nonparametric, meaning for example, that parameters such as mean and covariance are not used as global parameters in a functional form to describe an entire population. However, parametric functional forms may be used in a spatially adaptive way to describe local structure. From a signal processing point of view, the Parzen technique consists of defining a variable kernel with unit volume in feature space and convolving this kernel over a data set. A one-dimensional example is depicted in Figure 2.18.

The Parzen technique is essential for solving the difficult problem of estimating probability density functions (pdf's) for random variables obtained from evolutionary stochastic processes. Such processes are commonly encountered in real world data. For example, the variations of infrared signatures that occur with changes in time, range, and aspect are evolutionary stochastic processes. These variations cause corresponding variations in the features used for classification, as well as in the target probabilities output by the lower-level algorithms.

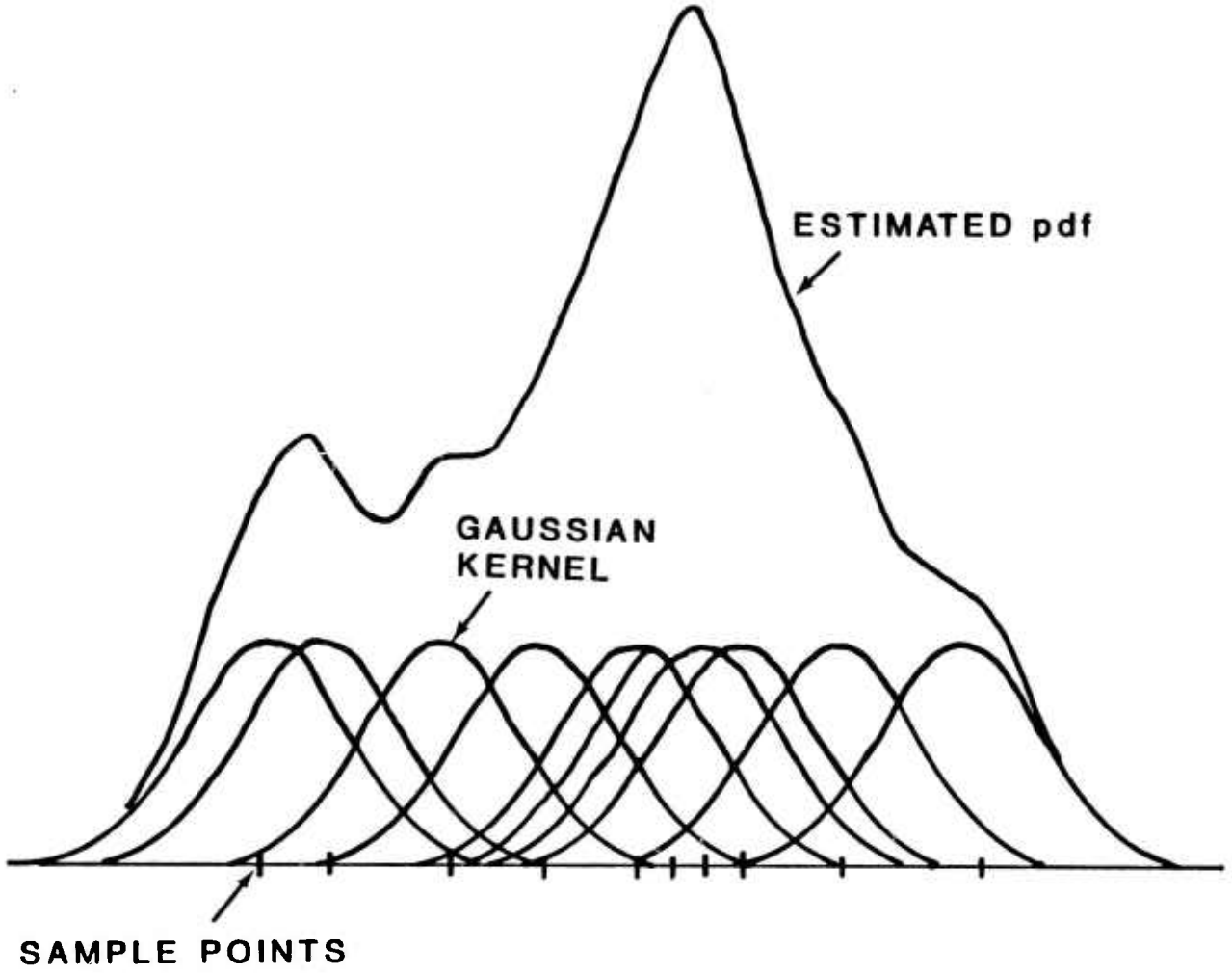


Figure 2.18. One-dimensional Example of Parzen Estimate of pdf for a Single Class.

Gaussian kernels were used exclusively, although other kernels may be just as effective. The covariance matrix used to specify each multivariate gaussian kernel was determined by categorical partitioning of the data sets. For example, to estimate the pdf for clutter objects, we heuristically categorized clutter into five categories: trees, roads, buildings, cultural, and bland areas. The features (used for discriminating targets from clutter) were extracted from the image for each clutter object in the training data base. For each of the five categories of clutter objects, the inverse covariance matrix is computed and, thereby, a Gaussian kernel is defined. Now for any point X in feature space, the pdf for clutter objects, $p(X|C)$ is defined as shown below:

$$p(X|C) = \frac{1}{M}\sum_{i=1}^{M}\left[\frac{1}{(2\pi)^{n/2}} \frac{1}{h_{k(i)}^{2}\,|\phi_{k(i)}|^{1/2}} \exp\left[\frac{-1}{2\,h_{k(i)}^{2}} (X-Z_i)^{T} \phi_{k(i)}^{-1} (X-Z_i)\right]\right],$$

where:

X	=	the continuous random vector in feature space,
C	=	the class of clutter objects,
i	=	the ith sample of the clutter training set,
Z_i	=	the location of the ith sample in feature space,
M	=	the number of samples in the training set of clutter objects,
n	=	the dimensionality of feature space,
$k(i)$	=	the categorical clutter subclass k to which sample i belongs,
$\phi_{k(i)}$	=	the covariance matrix for subclass k,
$h_{k(i)}$	=	a free scaling parameter for the kth covariance matrix.

The pdf for targets $p(X|T)$ is defined in an analogous manner.

The classification decision at any point X in feature space is based on Bayes' Test for Minimum Risk:

$$P(T)\ c(MD)\ p(X|T) \underset{\text{Clutter}}{\overset{\text{TARGET}}{\gtrless}} P(C)\ c(FA)\ p(X|C),$$

where P(T) and P(C) are the a priori probabilities of target and clutter, and c(MD) and c(FA) are the costs of missed detections and false alarms.

The probability of target at X is estimated by

$$P(T|X) = \frac{P(T)\ p(X|T)}{P(T)\ p(X|T) + P(C)\ p(X|C)}.$$

The actual form of the decision variate that was used in the NATS/ASSP system is the cost-weighted probability of target Q(T|X):

$$Q(T|X) = \frac{P(T)\ c(MD)\ p(X|T)}{P(T)\ c(MD)\ p(X|T) + P(C)\ c(FA)\ p(X|C)}.$$

The way in which Q(T|X) is computed and used in the form of a classification lookup table is depicted in Figures 2.19 and 2.20. Fourteen such classification tables were used in various parts of the NATS/ASSP system.

2.1.3.7 Dual-Mode Tracker

A multiple independent hierarchical tracking algorithm was developed for the ASSP. The same concept that proved so effective in the target acquisition algorithms was incorporated in this novel approach to target tracking.

The software block diagram for the tracker is shown in Figure 2.21. Upon entry into the module, two concurrent processes are started - image data is taken from the sensor and formatted as necessary and the Track Point Combining Logic (TPCL) is activated. Image data is processed by the GTIR algorithm (Section 2.1.2) to produce a GTIR map for the current input image. When the GTIR map is available, two new concurrent tasks are spawned, the TIR TRACKER and the NORMALIZED PRODUCT CORRELATION (NPC). These tasks each process the scene from their own point of view. TIR performs a GTIR target acquisition function and reports the target's centroid to TPCL. NPC performs the normalized product correlation algorithm for comparing a reference template taken from a recent image against the current image for some specified region.

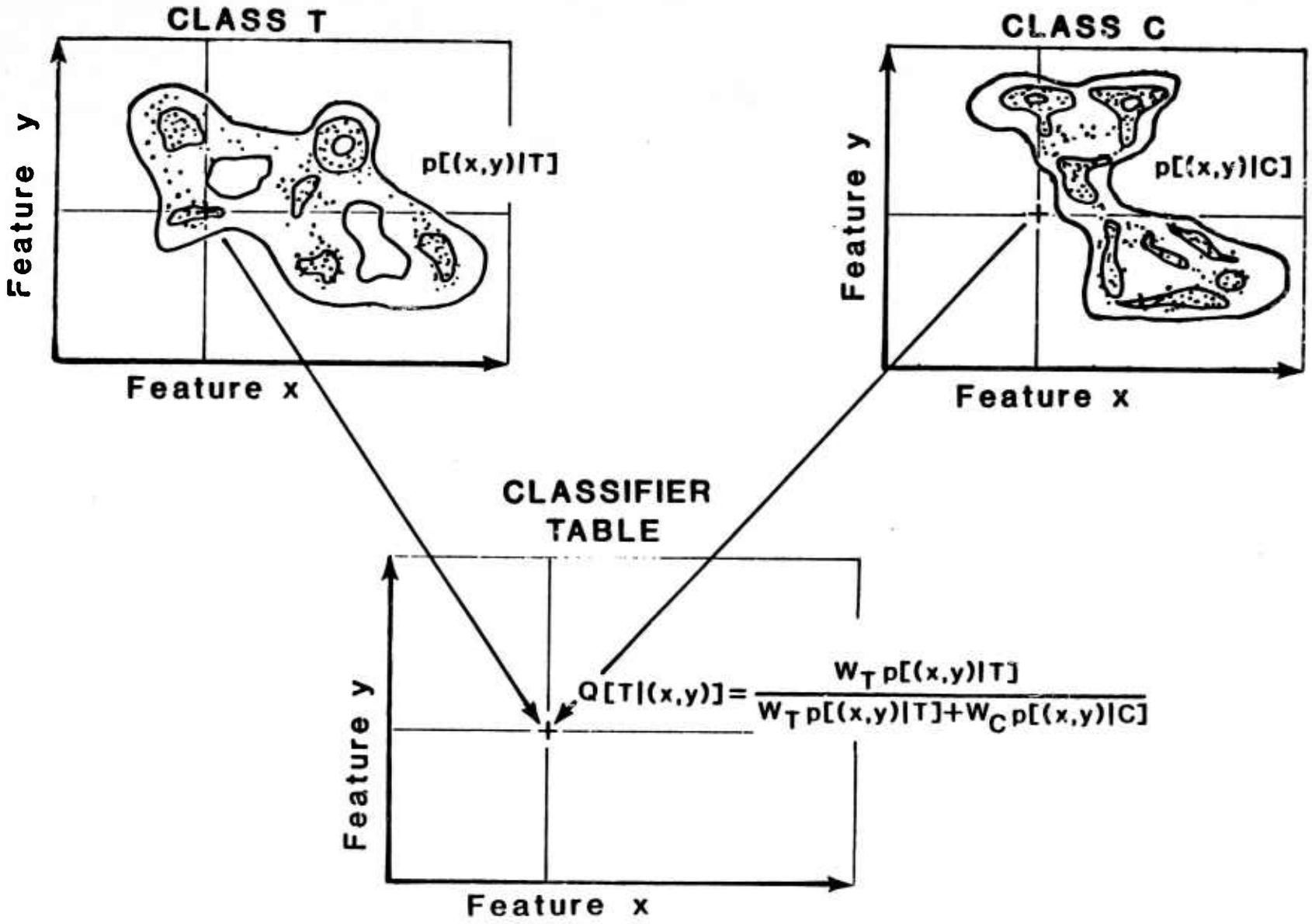


Figure 2.19. Illustration of Classifier Table Generation.

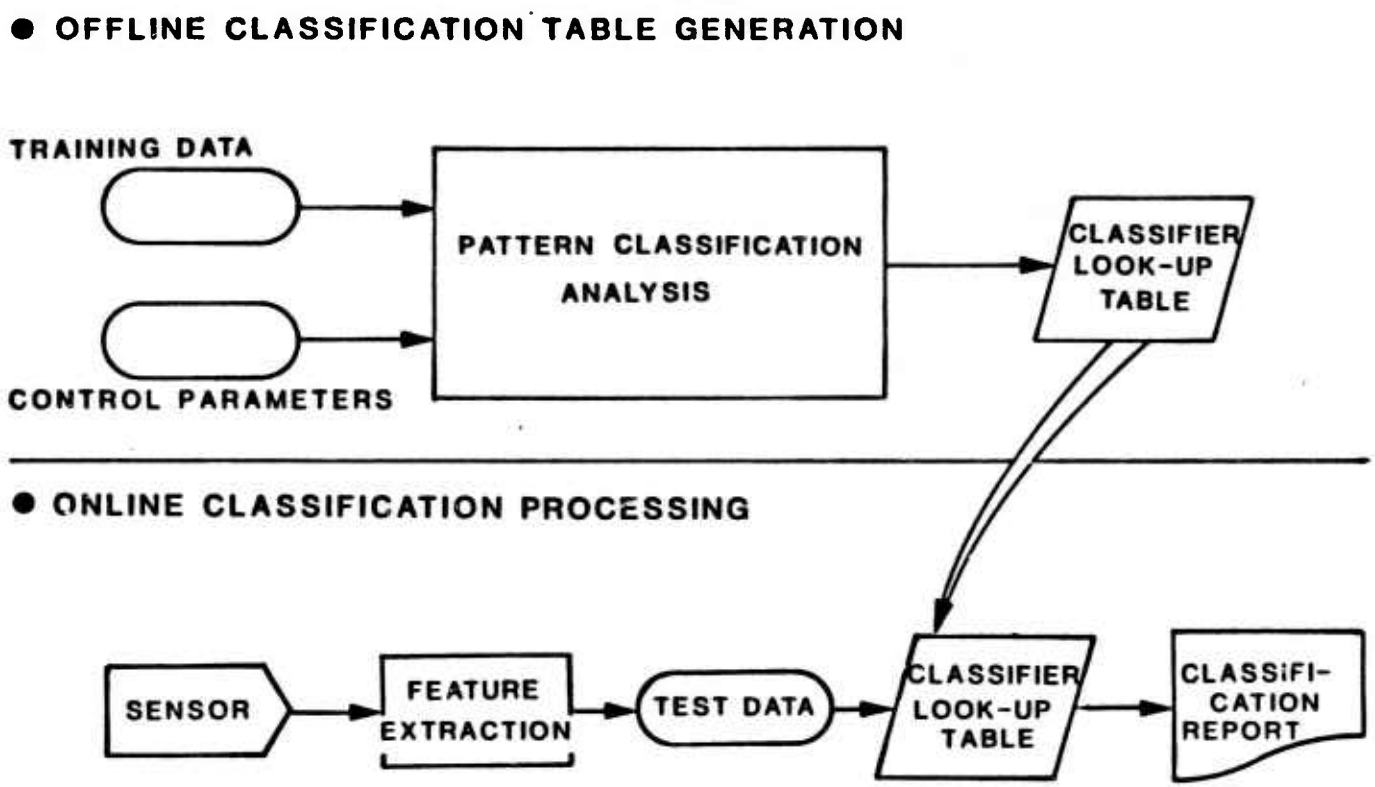


Figure 2.20. Overview of Generation and Use of Classifier Look-up Tables.

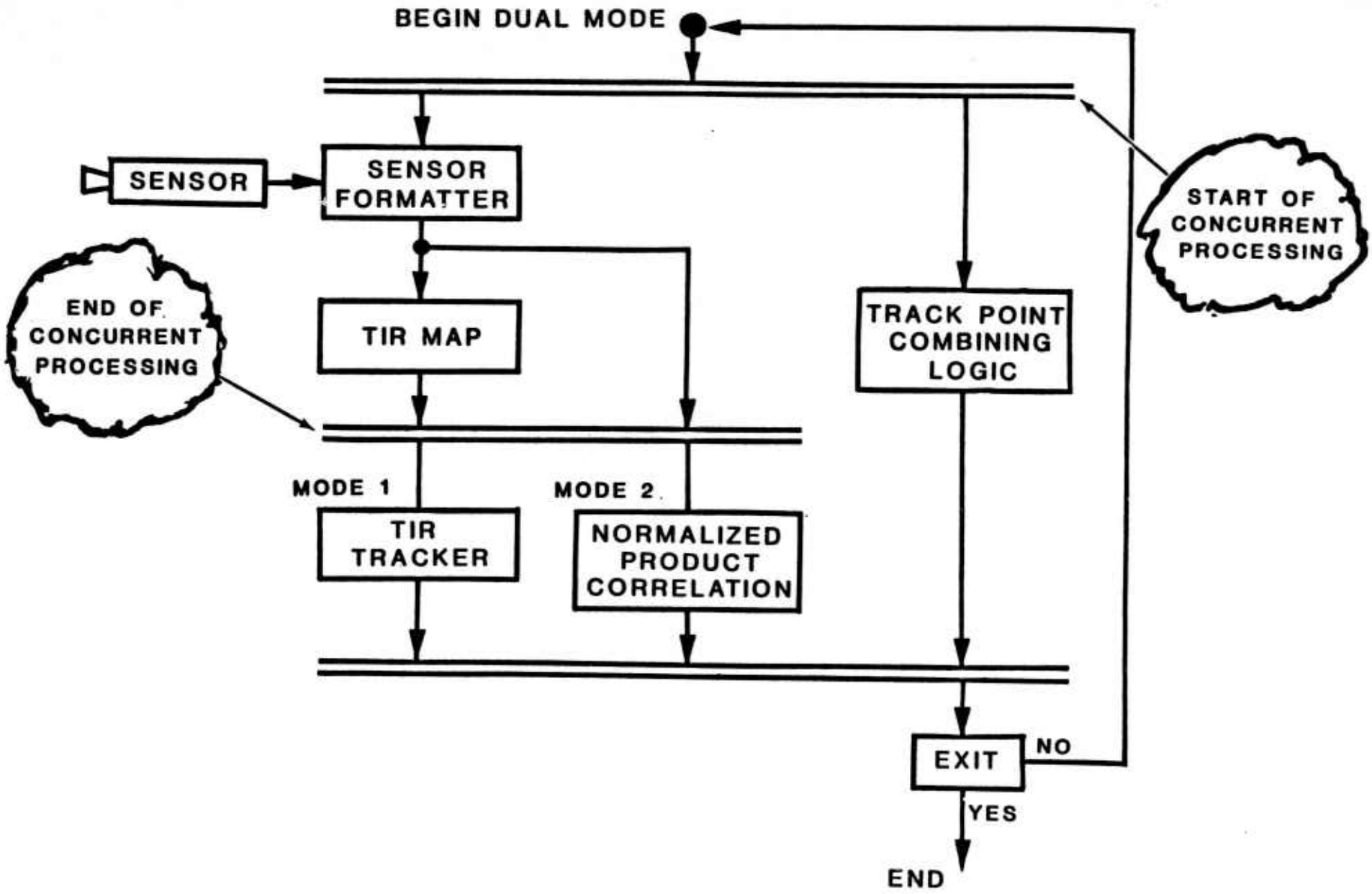


Figure 2.21. Classical Inertially Stabilized Seeker Proportional Navigation Guidance.

$$NPC(x,y) = \text{MAXIMUM SQRT}\left[\frac{\text{COVARIANCE}\ (\ R(x,y),\ S(x,y)\ )}{\text{STDEV}(R)\ \ \text{STDEV}(S)}\right]$$

over region

where: R = reference image, S = current sensed image

A functional block diagram for the ASSP dual mode tracker is shown in Figure 2.22, and the hardware block diagram is presented in Figure 2.23.

Referring to Figure 2-23, the Sensor Formatter digitizes a new image frame from the PNVS sensor, reformats it, and performs the Garber streak removal process. The Global Processor generates the GTIR and edge maps of the filtered input scene. The MLC stage is not used except that it passes on the GTIR map to its VM02 (model number for a commercial single board computer), which executes the GTIR track algorithm.

The SL/TT pipeline processor implements the Normalized Product Correlation (NPC) Track Mode. Its VM02 processor executes the Track Point Combining Logic (TPCL) algorithm. The latter investigates intelligently the target track points reported by the GTIR and NPC trackers, and decides upon a composite track point.

The CD pipeline processor and its VM02 continue to perform the global system control and I/O as usual.

The TPCL architecture is specially designed to offer the capability in executing multi-tracking algorithms in dedicated processors. In ASSP, the pipeline and VM02 processors which were initially configured for target acquisition are switched to the tracking algorithms. Each individual tracking algorithm and the TPCL are loaded to different segments of the local memory of the particular processors. This multi-tracker scheme follows the same information processing strategy of the overall ASSP architecture. Its salient feature reinforces the concept of independent observation from multiple points of view from separately preprocessed or derived information. More importantly, the same architecture readily accommodates the multi-sensor fusion in either target acquisition or tracking.

MULTI-MODE DIGITAL TRACKER
FUNCTIONAL BLOCK DIAGRAM

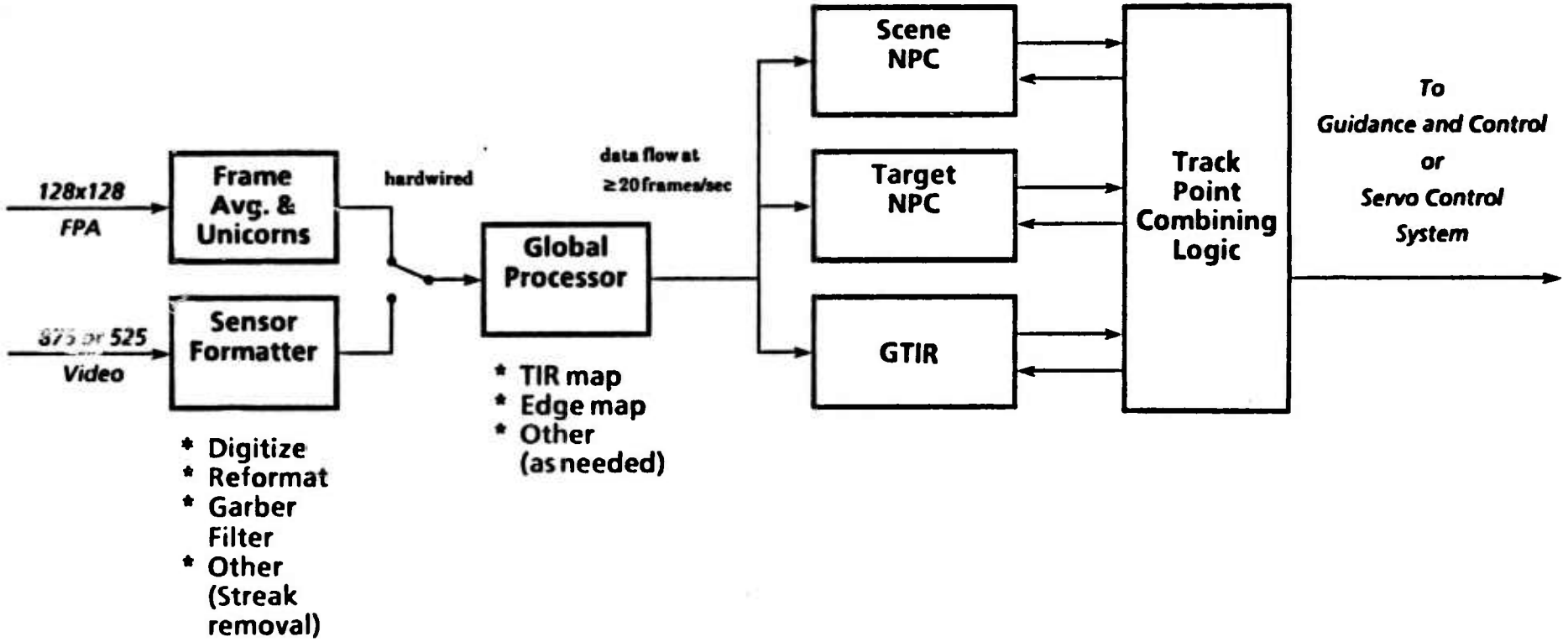


Figure 2.22. ASSP Dual Mode Tracker Functional Block Diagram.

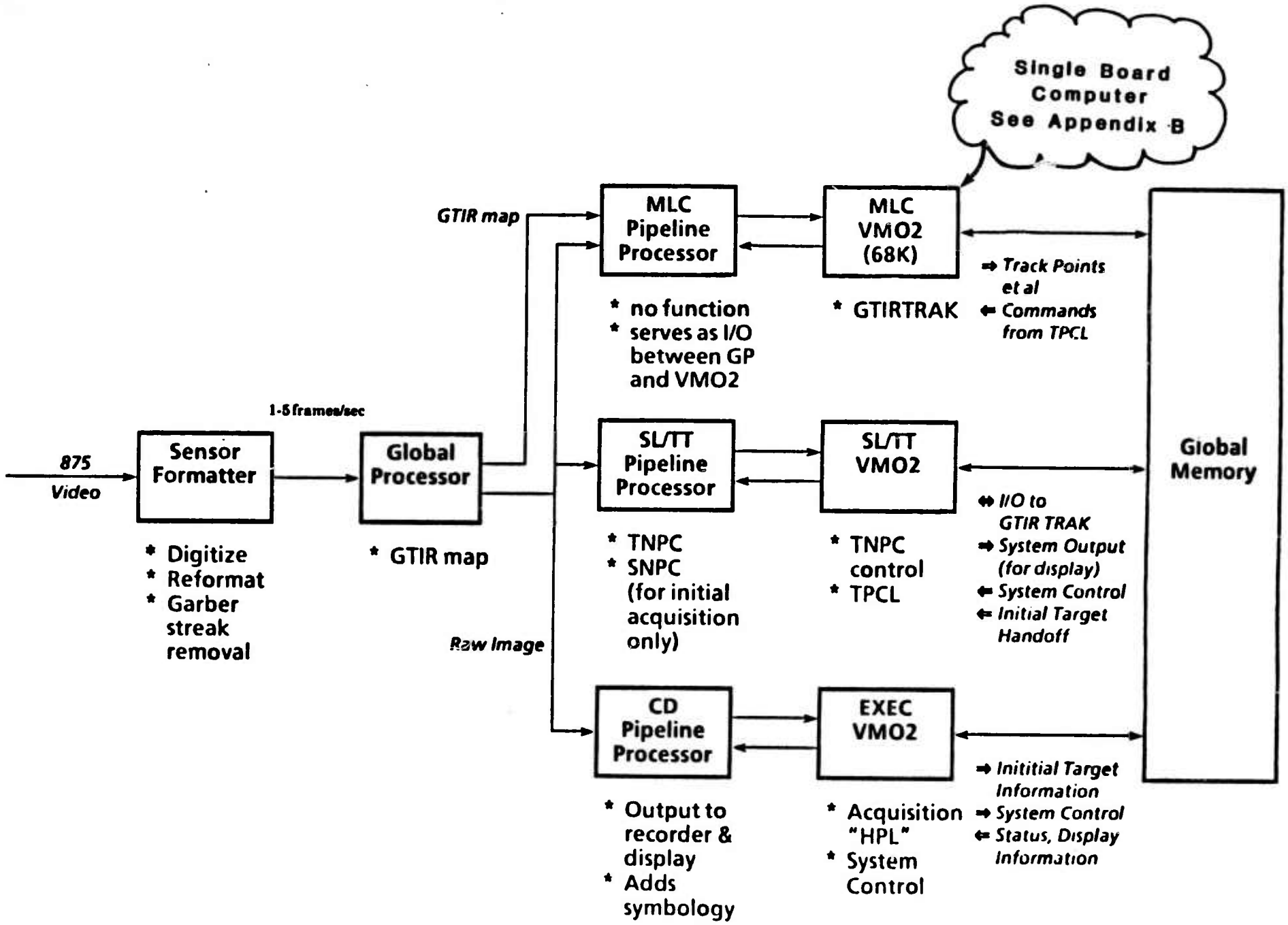


Figure 2.23. ASSP Dual Mode Tracker Hardware Block Diagram.

In the multi-mode tracker, each tracking mode returns a list of candidate target locations along with their confidence values. In the current system, the NPC tracker returns only its highest peak using the correlation coefficient as the confidence measure, while the GTIR tracker returns up to five tracked points.

The GTIR tracker performs the same functions as its acquisition algorithm in a dynamically updated and smaller search area instead of the entire image field of view. It keeps a record of the past history of its tracked points. Based on a spatial threshold on target tracked locations between consecutive frames, a preliminary criterion of successful tracking can be established. The past history of the tracked points are then utilized to readjust the corresponding confidence measure to reflect how well the tracked target location and its displacement vector match the statistics of the desired target.

Similarly, the NPC tracker uses a search area centered at the most confident target location as reported by HPL and locates the best match position employing the reference target template according to the coefficient of the normalized product correlation. The reference target template is acquired during target handoff from HPL and subsequently updated based on the TPCL report.

The Target Position Combining Logic (TPCL) algorithm initializes itself when it receives the handoff signal. In one version of TPCL, it always uses the handoff target location from HPL. In another version which aims to solve the problem of sudden shift in sensor field of view, TPCL waits for GTIR to reacquire in the present frame either the originally detected target from HPL or an object within a bounded neighborhood of the HPL target location having the highest confidence. At this point, TPCL enters its track mode. Two globally set parameters are the size of the search area for both GTIR and NPC and the size of the reference target template.

Once in the track mode, the NPC and GTIR trackers process the windowed data of the current image. When both finish processing, they report their result to TPCL which continuously accumulates the latest output from the NPC and GTIR trackers. Utilizing these target location inputs from the two separate

tracking algorithms and any valid GTIR past history, TPCL selects the highest confident GTIR target track location from the candidate list of GTIR track points and obtains a composite confidence measure for the tracked target between the two modes.

The composite confidence measure is assigned one of the three values described below:

(a) It is assigned a value of 2 when both trackers agree on a tracked position to within a specified maximum separation.

(b) It is assigned a value of 1 when both trackers do not agree on the tracked position but do agree on the relative motion of the target between the current and the last examined scene.

(c) Otherwise, it is assigned a value of 0.

According to the NPC coefficient, C, the NPC confidence measure is assigned one of three values described below:

(a) It is assigned a value of 2 when $C > 0.75$ and the change in the correlation coefficient between the present and the last time is less than or equal to 0.08.

(b) It is assigned a value of 1 when EITHER $C > 0.75$ and the change in the correlation coefficient between the present and the last time is greater than 0.08 OR the change in the correlation coefficient between the present and the last is less than 0.08.

(c) ELSE, it is assigned a value of 0.

According to the NPC coefficient, C, the NPC confidence measure is assigned one of three values described below:

(a) It is assigned a value of 2 when $C > 0.75$ and the change in the correlation coefficient between the present and the last time is less than or equal to 0.08.

(b) It is assigned a value of 1 when EITHER C > 0.75 and the change in the correlation coefficient between the present and the last time is greater than 0.08 OR the change in the correlation coefficient between the present and the last time is less than 0.08.

(c) ELSE, it is assigned a value of 0.

Similarly, according to the GTIR measure, G, which ranges from 0 to 32, the GTIR confidence measure can be one of the three values described below:

(a) It is assigned a value of 2 when G > 21.

(b) It is assigned a value of 1 when 21 ≥ = G ≥ 12.

(c) It is assigned a value of 0 when G < 12.

The three confidence measures are encoded into an integer, which ranges from 0 to 26. This is achieved by treating each of the three confidence measures as one bit in a three-bit based-3 number. This joint confidence code is used to index into a table that specifies which one of the eleven cases the TPCL should next proceed to. The eleven cases and their brief descriptions are shown in the Table 2-2.

The case-select code is then used to index into another series of tables which contain the decision for each case. These tables give the weights used in the equation to calculate the new composite target track position (TX,TY), whether to refresh the NPC tracker (REFN=1) or not (REFN=0), and whether to refresh the GTIR tracker (REFG=1) or not (REFG=0). This information is shown in Table 2-3. It is noted from Table 2-3 that the flexibility is there which allows the composite target location of TPCL (TX,TY), the NPC new position (NX,NY), and the GTIR new position (GX,GY) be all different. However, in the present ASSP system, both the NPC and GTIR trackers will receive the composite target location from TPCL as an update for their new center of search. Further study should evaluate the advantage of permitting individual algorithm to track its own most confident update track point provided that the corresponding confidence measure agrees within certain threshold with that of TPCL.

Table 2-2. TPCL Decision Encoding-1.

TNPCST	GTIRST	CLOSE	Weights	JOINT	CASE
FAILED	FAILED	FAR	0 + 0 + 0	0	1
FAILED	FAILED	POOR	0 + 0 + 9	9	1
FAILED	FAILED	GOOD	0 + 0 + 18	18	1
LOCONF	FAILED	FAR	1 + 0 + 0	1	2
LOCONF	FAILED	POOR	1 + 0 + 9	10	2
LOCONF	FAILED	GOOD	1 + 0 + 18	19	2
HICONF	FAILED	FAR	2 + 0 + 0	2	3
HICONF	FAILED	POOR	2 + 0 + 9	11	3
HICONF	FAILED	GOOD	2 + 0 + 18	20	3
FAILED	LOCONF	FAR	0 + 3 + 0	3	4
FAILED	LOCONF	POOR	0 + 3 + 9	12	4
FAILED	LOCONF	GOOD	0 + 3 + 18	21	4
LOCONF	LOCONF	FAR	1 + 3 + 0	4	5
LOCONF	LOCONF	POOR	1 + 3 + 9	13	6
LOCONF	LOCONF	GOOD	1 + 3 + 18	22	5
HICONF	LOCONF	FAR	2 + 3 + 0	5	8
HICONF	LOCONF	POOR	2 + 3 + 9	14	8
HICONF	LOCONF	GOOD	2 + 3 + 18	23	7
FAILED	HICONF	FAR	0 + 6 + 0	6	9
FAILED	HICONF	POOR	0 + 6 + 9	15	9
FAILED	HICONF	GOOD	0 + 6 + 18	24	9
LOCONF	HICONF	FAR	1 + 6 + 0	7	10
LOCONF	HICONF	POOR	1 + 6 + 9	16	11
LOCONF	HICONF	GOOD	1 + 6 + 18	25	5
HICONF	HICONF	FAR	2 + 6 + 0	8	7
HICONF	HICONF	POOR	2 + 6 + 9	17	7
HICONF	HICONF	GOOD	2 + 6 + 18	26	7

JOINT = TNPCST + 3*GTIRST + 9*CLOSE

TNPCST,GTIRST:

- 0: FAILED
- 1: LOCONF
- 2: HICONF

CLOSE:

- 0: Far (Not Close)
- 1: Poor (Close, poor Attrib)
- 2: Good (Close, good Attrib)

GTIRA1

LO: 12

HI: 21

Change in Correlation of >0.08 causes a decrease in confidence.

Table 2-3. TPCL Decision Encoding-2.

CASE	TX = (TNPCX*WTN(CASE) + TNPCXO*WTNO(CASE) + GTIRX(JBEST)*WTG(CASE) + GTIRXO(JBEST)*WTGO(CASE) + TPCLXO*WTTO(CASE) + DXS*WTDS(CASE))/DT(CASE)						
1	0	0	0	0	1	0	1
2	F	– F	0	0	1	0	1
3	K	1 – K	0	0	0	0	1
4	0	0	F	– F	1	0	1
5	1	0	1	0	0	0	2
6	0	0	1	0	0	0	1
7	1	0	1	0	0	0	2
8	3	0	1	0	0	0	4
9	0	0	K	1 – K	0	0	1
10	1	0	3	0	0	0	4
11	0	0	1	0	0	0	1

CASE	NX = (TNPCX*WNN(CASE) + TNPCXO*WNNO(CASE) + GTIRX(JBEST)*WNG(CASE) + GTIRXO(JBEST)*WNGO(CASE) + TPCLXO*WNTO(CASE) + DXS*WNDS(CASE))/DN(CASE)							REFN
1	0	0	0	0	1	0	1	0
2	F	– F	0	0	1	0	1	1
3	K	1 – K	0	0	0	0	1	0
4	0	0	F	– F	1	0	1	1
5	1	0	1	0	0	0	2	1
6	0	0	1	0	0	0	1	1
7	1	0	1	0	0	0	2	0
8	3	0	1	0	0	0	4	0
9	0	0	K	1 – K	0	0	1	1
10	1	0	3	0	0	0	4	1
11	0	0	1	0	0	0	1	1

CASE	GX = (TNPCX*WGN(CASE) + TNPCXO*WGNO(CASE) + GTIRX(JBEST)*WGG(CASE) + GTIRXO(JBEST)*WGGO(CASE) + TPCLXO*WGTO(CASE) + DXS*WGDS(CASE))/DG(CASE)							REFG
1	0	0	0	0	1	0	1	1
2	F	– F	0	0	1	0	1	1
3	K	1 – K	0	0	0	0	1	1
4	0	0	F	– F	1	0	1	2
5	1	0	1	0	0	0	2	2
6	0	0	1	0	0	0	1	2
7	1	0	1	0	0	0	2	2
8	3	0	1	0	0	0	4	2
9	0	0	K	1 – K	0	0	1	2
10	1	0	3	0	0	0	4	2
11	0	0	1	0	0	0	1	2

for now: F = K = 1

In Table 2-3 there are two parameters, F and K, which can be used in conjunction with the other values to "tune" the TPCL logic. Presently, F and K are set to 1. Further study could substitute a smaller fractional value for F and K in the target positional update computation.

The seven variables in Table 2-3 are used as follows:

WTN: weight for the new correlation position
WTNO: weight for the old correlation position
WTG: weight for the new GTIR position
WTGO: weight for the old GTIR position
WTTO: weight for the old composite TPCL position
WTDS: weights for the target displacement
DT: divisor for the product of a scale factor and the sum of the above weights

Note that WTDS is set to zero. Otherwise, this weight is intended to implement the ability to coast when track loss becomes imminent and when reacquisition is needed.

In those situations where a very high confidence measure from one tracker contrasts with a very low confidence measure from the other tracker, the composite target location is supplied by the tracker with the higher confidence measure. When both trackers present equal confidence measures, the average of the separate target positions is used as the composite target location. When a high confidence measure and a medium confidence measure are received by TPCL, the composite target location is computed with weights of 3:1, the larger weight is given to the target location of the higher confidence.

While TPCL loops in consecutive tracking frames, the refresh signal of the trackers will be determined by the array entries selected by the particular case. When refresh occurs, the particular tracker will operate with the new data being sent by the global processor.

Simulation on our VAX computer using scene sequences with varying degree of difficulties demonstrated that the NPC and GTIR trackers did successfully complement each other, especially in situations when one tracker failed to continue tracking. When both trackers fail for an extended period of time,

the GTIR tracker becomes a target detector for reacquisition while the NPC tracker tries to match the data residing in its reference target template. The track state is halted from external interrupt.

2.1.4 ASSP Hardware Description

The ASSP processor is a loosely coupled computing net of parallel processors, as shown in Figure 2.24. A detailed description of the hardware can be found in Appendix B. The Global Processor performs data transformations which are applicable to the entire image, such as the computation of gradients, edge and joint-edge directions, and the TIR filtering function. These processed images are sent via a packet protocol to their respective processors: MLC utilizes gradient and joint edge information, VSC utilizes the unprocessed video, and TIR uses the TIR data.

Each of the nine processors show in Figure 2.24 are independent and they run asynchronously except during message passing. The bipolar processors are designed to efficiently process algorithmic specific operators, e.g., MLC requires multi-dimensional histograms, and the NPC tracker requires the computation of the normalized correlation coefficient. Associated with each processor is single board computer (CPU) which executes FORTRAN 77 code for high level processes such as HPL and TPCL. Global memory is provided for the exchange and storage of symbolic information and data.

The ASSP System Software Block Diagram is shown in Figure 2.25. The parallel lines indicate the beginning (or ending) of concurrent processing. At START time, the processor loads itself with its runtime programs and performs a SELF TEST. The processor halts with an error code if all is not well. The SENSOR FORMATTER and the IMAGE DISPLAY tasks are activated and continue to run until the processor is halted. When the POINT OF INTEREST is specified (i.e., the operator directs the processor to enter the acquisition mode), HPL, VSC, TIR, MLC, and the GLOBAL PROCESSOR are simultaneously activated for concurrent parallel processing. When these processors have finished the Target Acquisition Task, the TRACKER and GLOBAL PROCESSOR tasks are activated in parallel. The tracking function continues for five or six seconds and automatically terminates (for the Captive Flight Test) activating the ALGORITHM DUMP task which records and displays the Ground Truth and the algorithm results. This process is repeated until directed to STOP.

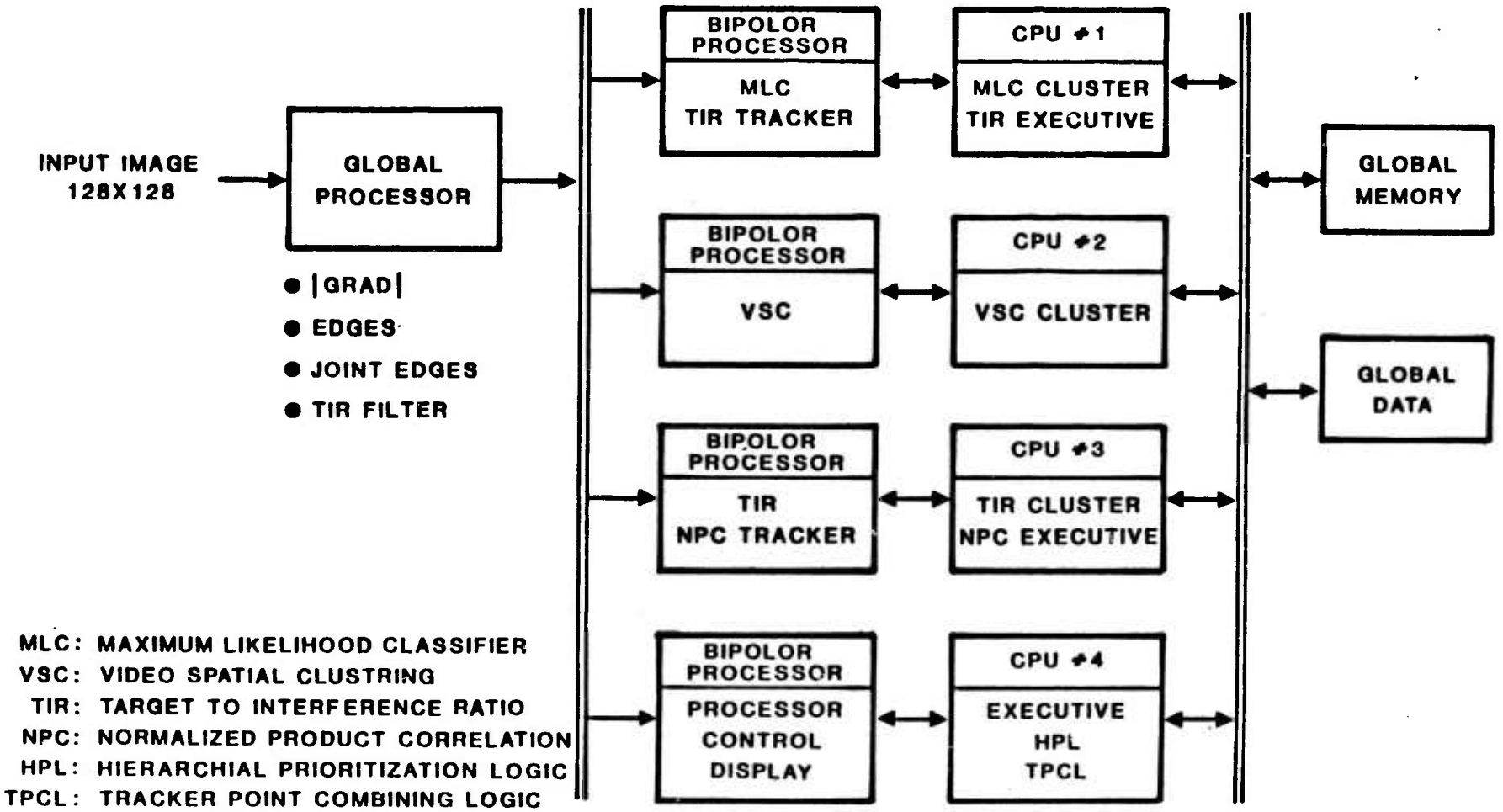


Figure 2.24. Hardware Block Diagram.

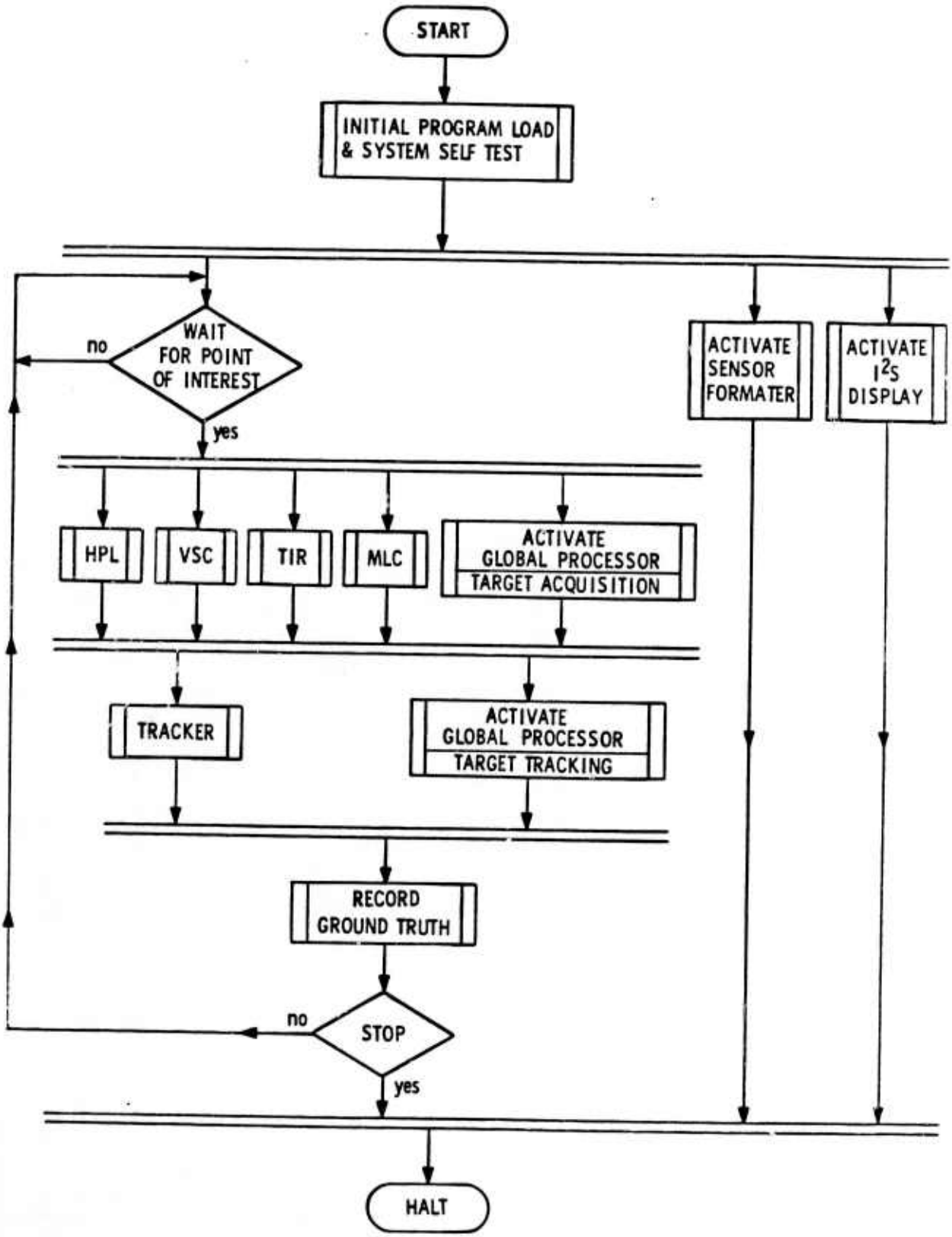


Figure 2.25. ASSP System Software Block Diagram.

Table 2-4 summarizes the chip count and wire count for the ASSP, and Table 2-5 summarizes the amount of computer code generated in support of the algorithm development and real-time mechanization.

2.1.5 Results from ASSP Captive Flight Test

In November 1983, the Advanced Seeker Signal Processor was flown in a series of captive flight tests over Test Range 6 at the Army Missile Command (MICOM), Redstone Arsenal, Huntsville, Alabama. In three separate flights, 249 images were taken with depression angles from 20 degrees to 90 degrees (vertical lookdown) and target sizes from 1 pixel to 33 pixels in length. Target types included tanks, trucks, APC's, in both the cold and hot thermal states.

The ASSP was run against all images. Some images were not included in the performance measure due to 1) excessive platform vibration or motion resulting in image breakup, blurr, or distortion, 2) targets 5 pixels or less in length, 3) no targets in the field of view.

The entire Captive Flight Test data set is included in APPENDIX C, Figures C-1 to C-29. Where applicable, the number one ranked object has been indicated by a circle 12 pixels in radius and the corresponding ground truth annotated in the lower left-hand corner of each 128 x 128 image. Correlated images are marked by "  " in the upper left-hand corner and are not included in the performance measure. The need for independent images in the performance measure is apparent since inclusion of correlated samples skews the resulting performance statistics for better or for worse, depending upon whether the processor finds a target or a false alarm for the correlated images. For example, suppose 200 frames of acquisition data have been taken, of which 51 frames are of the same scene under the same conditions. Suppose the acquisition system scores a Pd of 1.00 for the 51 correlated frames and a Pd of 0.60 for the 149 remaining uncorrelated frames. If the correlated frames are included, the overall performance measure would be

$$Pd = (0.60 * 149 + 51)/(149 + 51) = 0.70.$$

However, if the correlated samples are counted as a single event, the overall performance would be

$$Pd = (0.60 * 149 + 1)/(149 + 1) = 0.60.$$

Table 2-4. Hardware Statistics.

PROCESSOR	CHIPS	WIRES
SFV	283	4460
SFN	309	4447
GPA	233	3439
GPB	237	3498
MLCA	325	4791
MLCB	88	1305
SLTA	140	1935
SLTB	312	4434
CDA	314	4909
CDB	86	1269
TOTAL	2327	34,487

Table 2-5. Approximate Lines of Computer Code.

TASK	LANGUAGE		
	FORTRAN	ASSEMBLY	MICRO-CODE
DATA BASE	5,800	400	
ACQUISITION ALGORITHMS	59,800	8,000	33,200
HANDOFF & TRACKER	3,600		
ANALYSIS	24,000		
SUPPORT	47,800	13,600	
TOTAL	141,000	22,000	33,200

NOTES:

1. LINE COUNTS INCLUDE COMMENTS
2. MICRO-CODE IS AT THE MACRO LEVEL

Clearly, the use of correlated samples in the computation of the performance measure must be avoided. However, the determination of which images are correlated is not always an easy process, since it is subject to interpretation. Northrop utilized a two-member panel to make the selection of independent image samples based upon a careful comparison of the images and the available ground truth.

2.1.5.1 ASSP Captive Flight Test Performance Summary

A summary for the ASSP captive flight test performance is presented in Table 2.6. This data is compiled completely from independent captive flight test images. All depression angles have been included, which in some cases were as small as 30 degrees.

Table 2-6. ASSP Captive Flight Test Summary.

TARGET SIZE (pixels)	SAMPLE SIZE	PROBABILITY OF CORRECT ACQUISITIONS (%)	PROBABILITY OF FALSE ACQUISITIONS (%)
*10-16	75	93	7
6-40	141	91	9

* BASELINE SCENARIO

The ASSP achieved a 93% probability of correct classification over the baseline scenario for which it was trained. The robustness of the acquisition algorithms is demonstrated by a 91% probability of correct acquisition for target sizes well outside the nominal training limits of 12 pixels in length

± 20%, and the ability to correctly process acquired targets for depression angles as low as 20 degrees.

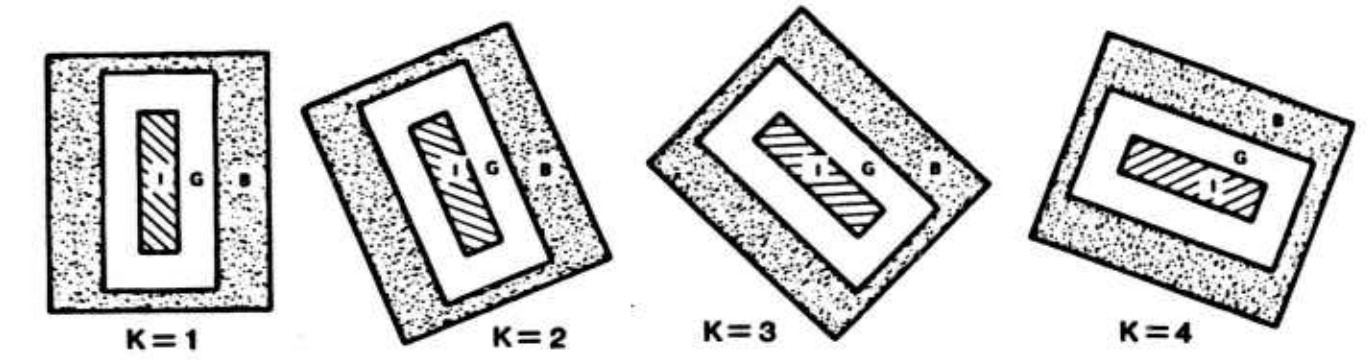
2.1.5.2 Analysis of Captive Flight Test Data

Data from the captive flight test is summarized in Figures 2.27 to 2.30 for the Baseline Scenario, and Figures 2.31 to 2.34 for the Extended Scenario. Data is summarized in four ways. First, the probability density function and the cumulative density function for both targets and non-targets are plotted as a function of the Acquisition Figure of Merit, i.e., the output from HPL Level II. Second, the probability density function and the cumulative density function for both targets and non-targets are plotted as a function of the Target to Interference Ratio (TIR). Third, the probability density function and the cumulative density function for both targets and non-targets are plotted as a function of the TIR Z-Score per Image. Fourth, the Probability of Correct Acquisition is plotted against the Probability of False Acquisition.

The Acquisition Figure of Merit is related to the overall detection confidence, and is bounded between 0 and 100 percent. This figure of merit may be used as a metric for thresholding detections. For example, all detections with a figure of merit greater than, say, 75% are considered equal for purposes of target assignment. (This produces a very effective anti-clustering algorithm when submunitions share a common field of regard.)

Target to Interference Ratio is a spatially matched filter for locating rectangular objects in natural imagery. This filter convolves the eight templates shown in Figure 2.26 and selects the maximum response as the TIR value for each pixel. The templates are rotated in increments of 22.5 degrees to accommodate target rotation in the natural imagery. TIR is an excellent measure of scene difficulty and is useful in prediction algorithm performance prior to their being run on the images.

TIR is an absolute measure independent of the target non-target mix. Weak targets in a bland scene should be more easily detected than weak targets in a highly cluttered scene. One way of measuring this dependency is to normalize the TIR on a per image basis by forming the Z-Score between the TIR value for the number one ranked object and the TIR statistics for the image as a whole.

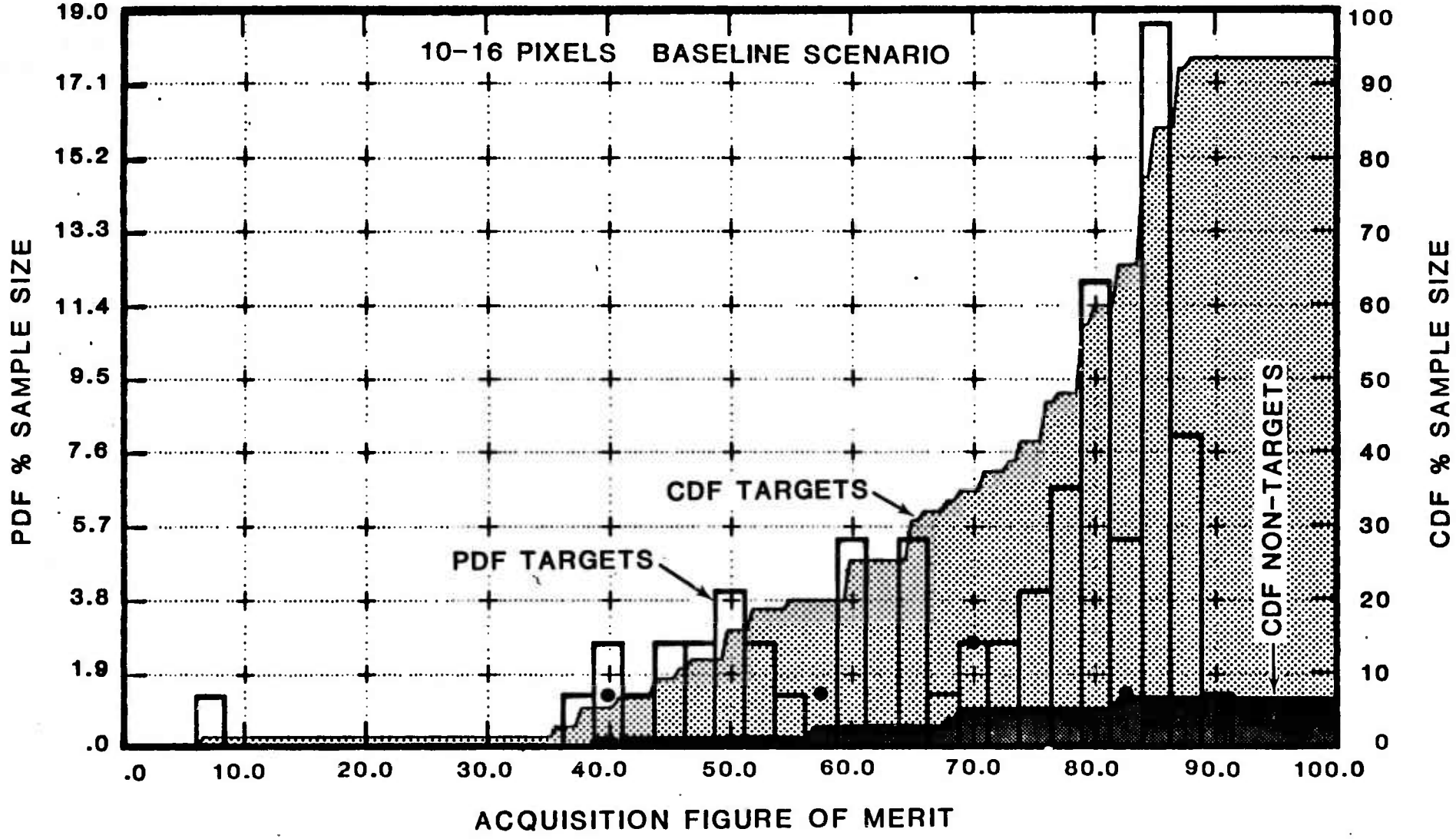


I = INTERIOR AREA
G = GUARD AREA
B = BACKGROUND AREA

$$TIR = \max_{K=1,8} \left| \frac{\mu_{KI} - \mu_{KB}}{\sigma_{KB}} \right|$$

Figure 2.26. Directional Masks for Target-to-Interference Ratio.

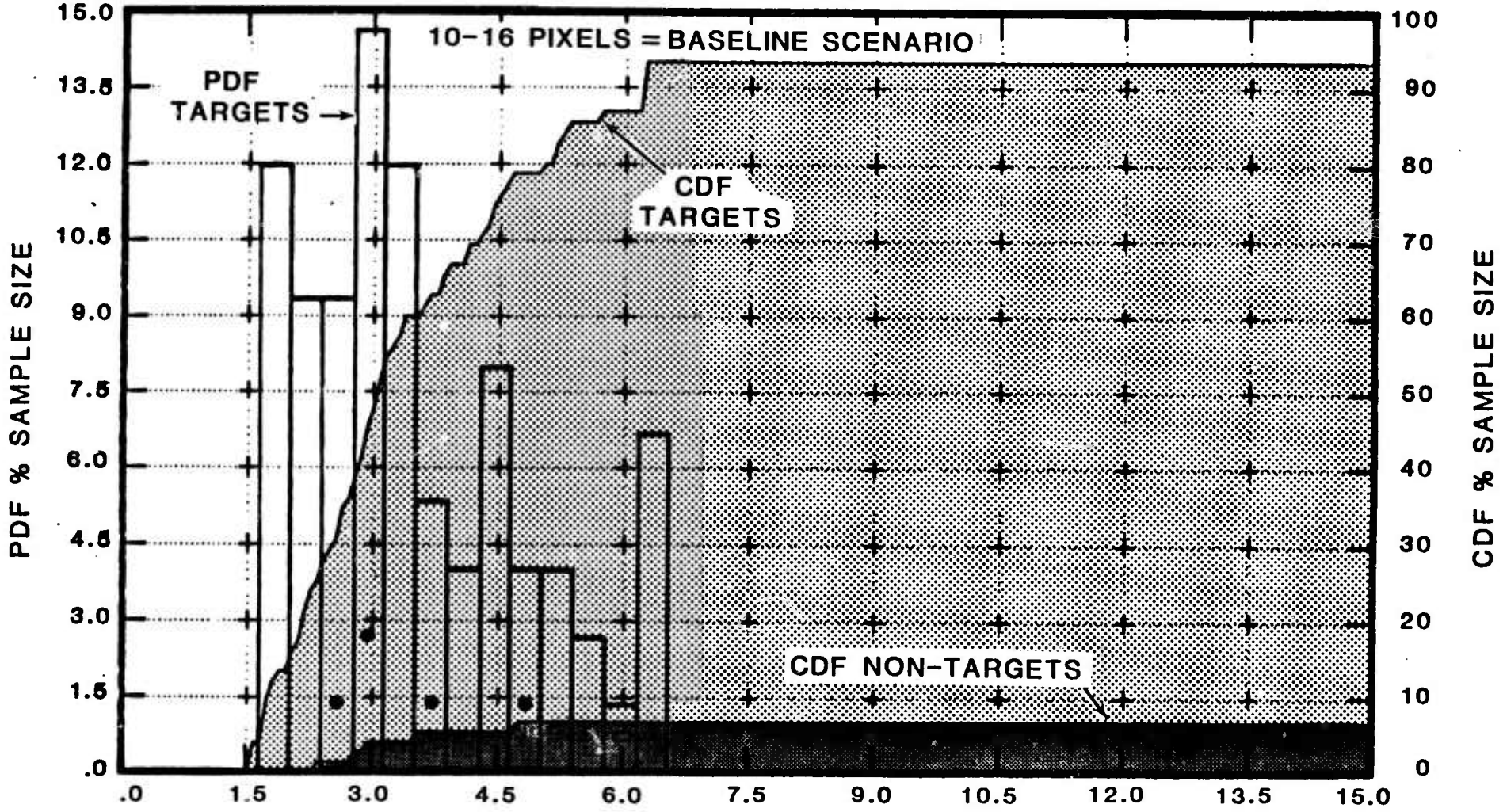
Figure 2.27. ASSP Performance for First Ranked Object (1 of 8).



TARGET SIZE 10 TO 16 PIXELS
SAMPLE SIZE = 75 INDEPENDENT SAMPLES
TOTAL TARGETS = 70 . TOTAL NON TARGETS = 5

● = NON TARGET
ALL ASPECTS ANGLES

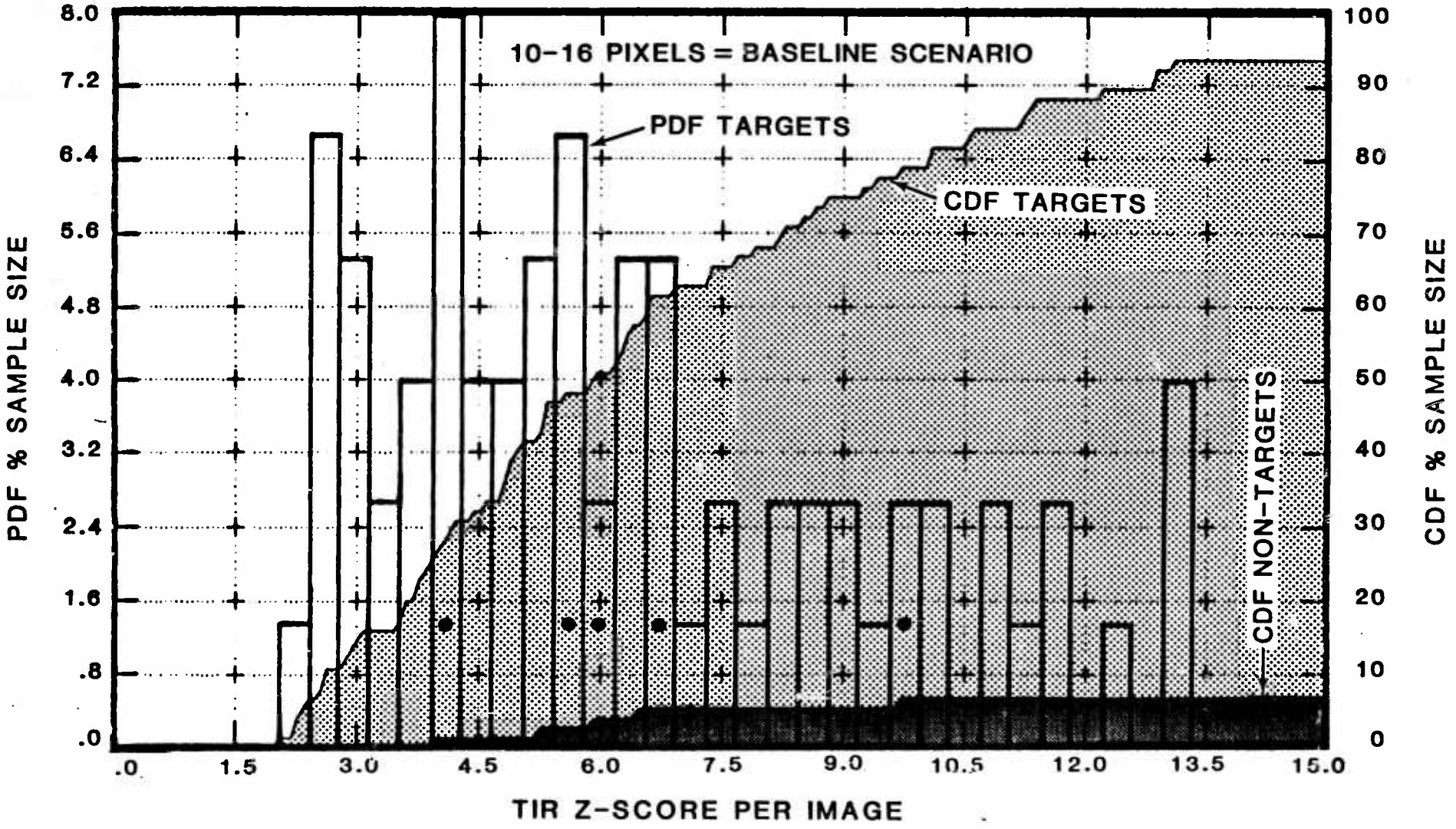
Figure 2.28. ASSP Performance for First Ranked Object (2 of 8).



TARGET SIZE 10 TO 16 PIXELS
SAMPLE SIZE = 75 INDEPENDENT SAMPLES
TOTAL TARGETS = 70 . TOTAL NON TARGETS = 5

● = NON-TARGET
ALL ASPECTS ANGLES

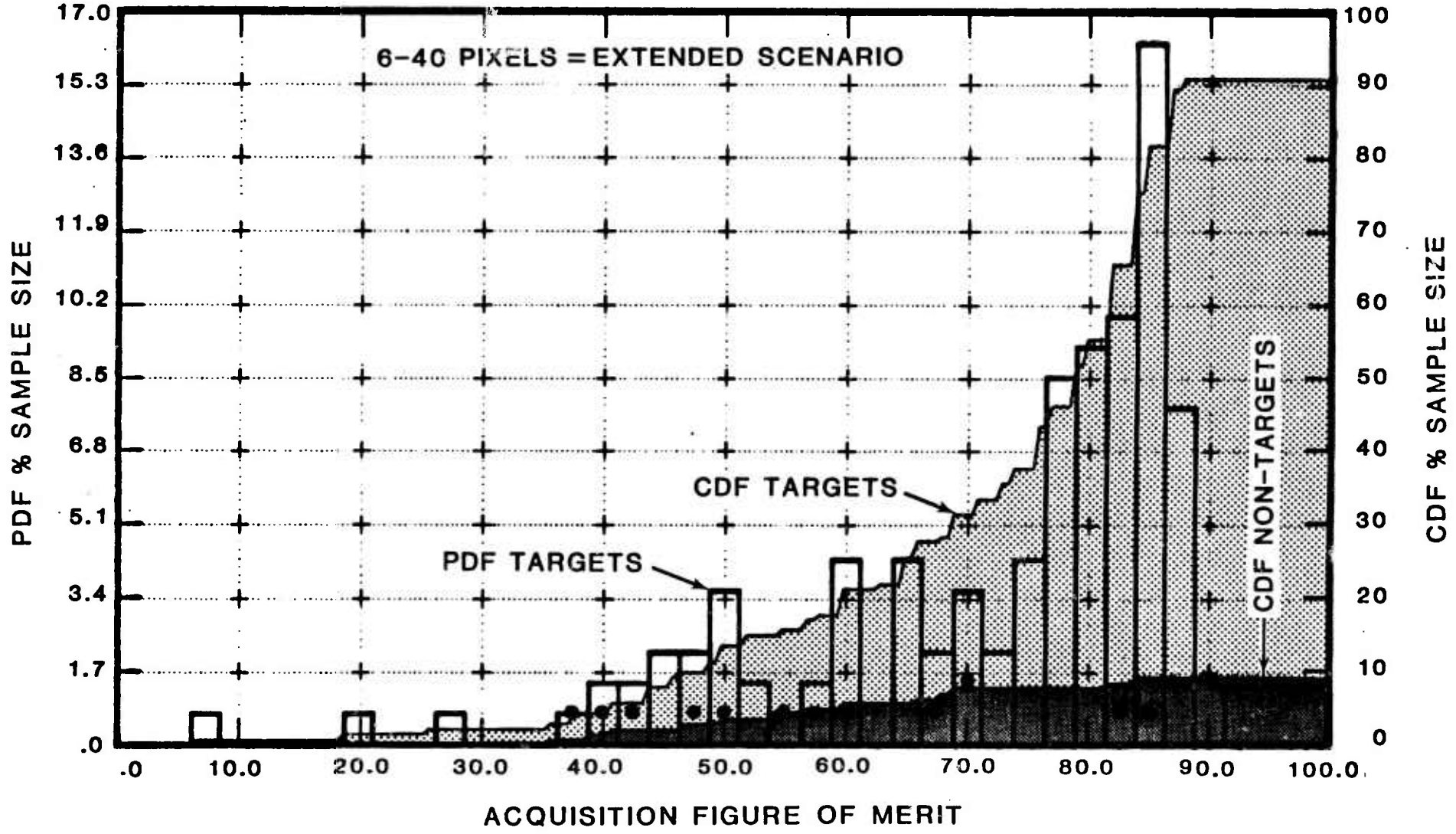
Figure 2.29. ASSP Performance for First Ranked Object (3 of 8).



TARGET SIZE 10 TO 16 PIXELS
SAMPLE SIZE = 75 INDEPENDENT SAMPLES
TOTAL TARGETS = 70 . TOTAL NON TARGETS = 5

● = NON-TARGET
ALL ASPECTS ANGLES

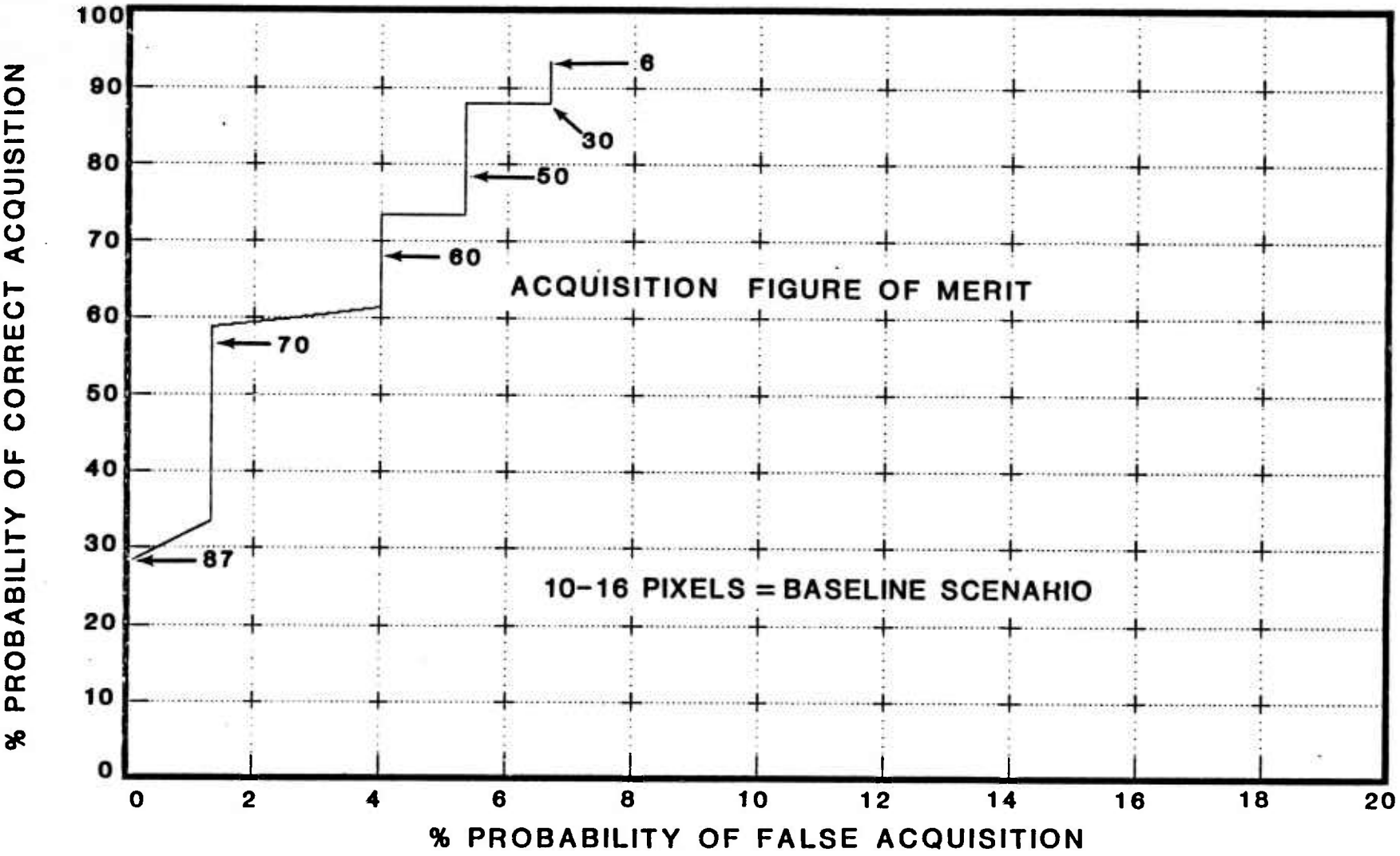
Figure 2.30. ASSP Performance for First Ranked Object (4 of 8).



TARGET SIZE 6 TO 40 PIXELS
SAMPLE SIZE = 141 INDEPENDENT SAMPLES
TOTAL TARGETS = 128 . TOTAL NON TARGETS = 13

● = NON-TARGET
ALL ASPECTS ANGLES

Figure 2.31. ASSP Performance for First Ranked Object (5 of 8).



TARGET SIZE 10 TO 16 PIXELS
SAMPLE SIZE = 75 INDEPENDENT SAMPLES
TOTAL TARGETS = 70 . TOTAL NON TARGETS= 5

● = NON-TARGET
ALL ASPECTS ANGLES

Figure 2.32. ASSP Performance for First Ranked Object (6 of 8).

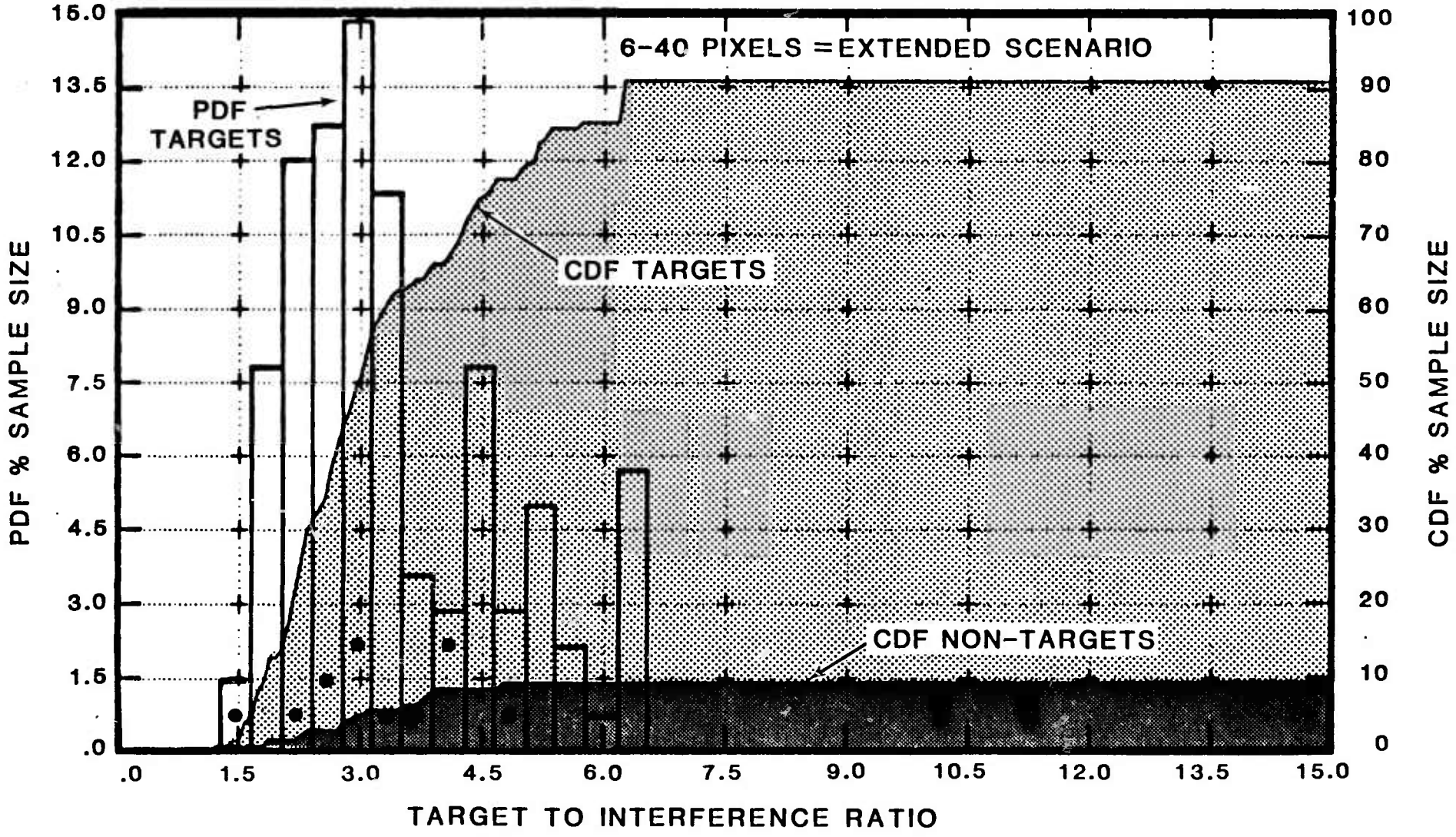
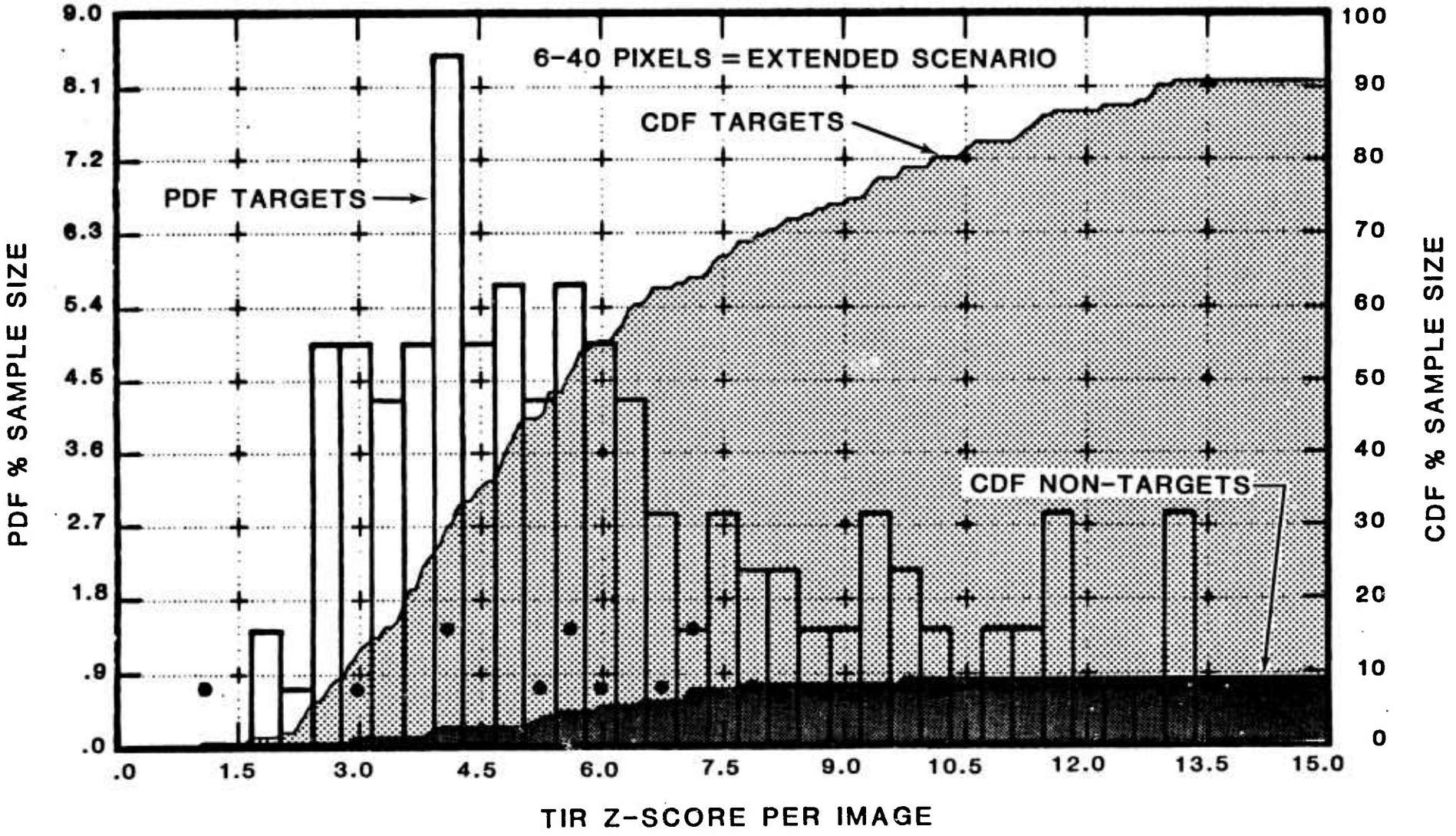


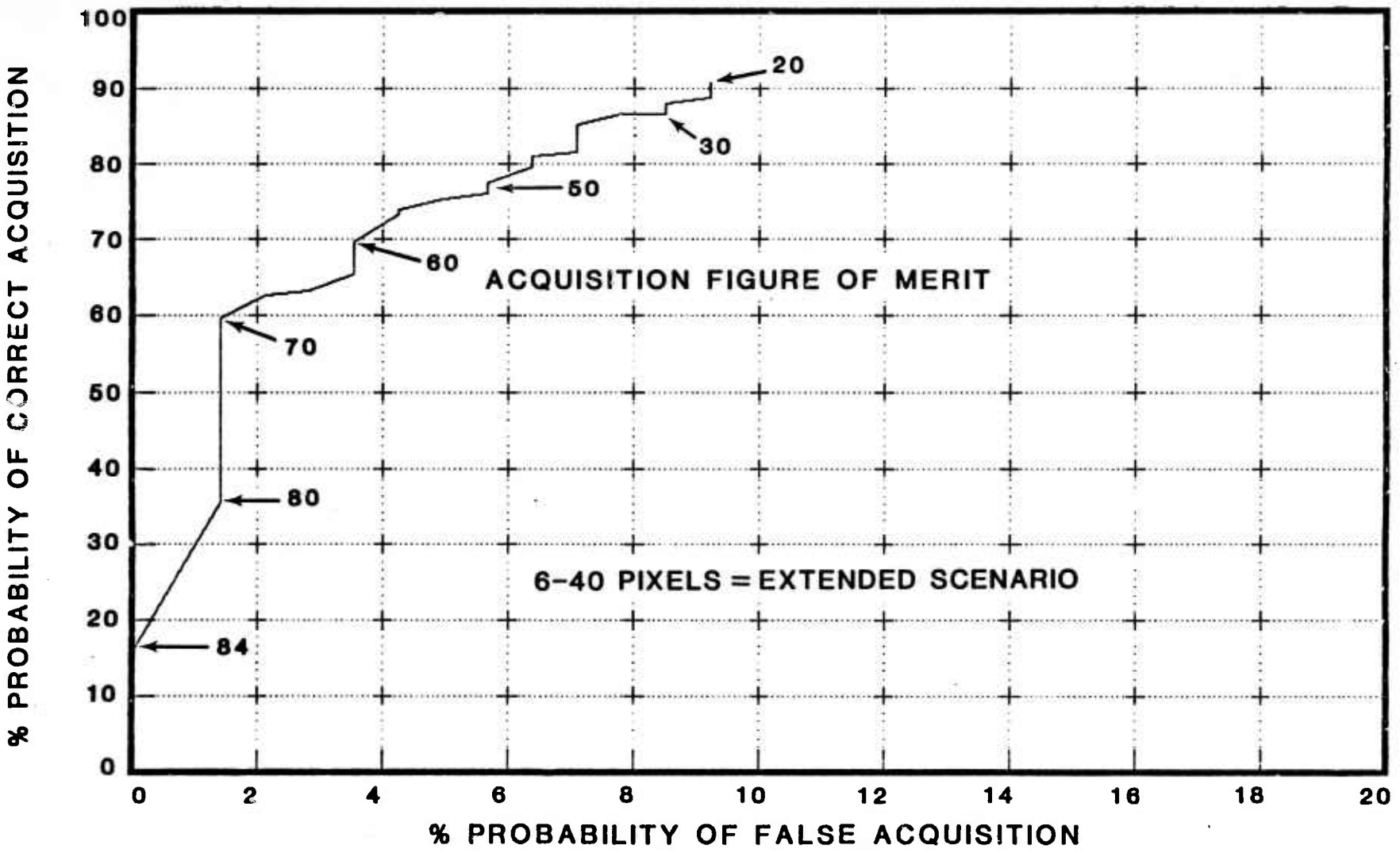
Figure 2.33. ASSP Performance for First Ranked Object (7 of 8).



TARGET SIZE 6 TO 40 PIXELS
SAMPLE SIZE = 141 INDEPENDENT SAMPLES
TOTAL TARGETS = 128 . TOTAL NON TARGETS = 13

● = NON-TARGET
ALL ASPECTS ANGLES

Figure 2.34. ASSP Performance for First Ranked Object (3 of 8).



TARGET SIZE 6 TO 40 PIXELS
SAMPLE SIZE = 141 INDEPENDENT SAMPLES
TOTAL TARGETS = 128 . TOTAL NON TARGETS = 13

● = NON-TARGET
ALL ASPECTS ANGLES

$$Z(TIR) = \frac{\text{ABS(TIR(\#1 ranked object) - TIR(mean for image))}}{\text{TIR Standard Deviation for image}}$$

Thresholding the Z-Score TIR is analogous to the constant false alarm method of target detection since it allows the image threshold to be specified by the number of standard deviations the target lies above the background.

The Probability of Correct Acquisition versus Probability of False Acquisition Curves are generated by varying the Acquisition Figure of Merit as a threshold and computing the percentages of correct and incorrect classification with Figure of Merits greater than or equal to the threshold. Several threshold values have been annotated on these curves. The curves are somewhat ragged due to the low false acquisition rates.

Data from the Captive Flight Test is summarized in Figures 2-27 to 2-30 for the Baseline Scenario and Figures 2-31 to 2-34 for the Extended Scenario. Note that for both the baseline and the extended scenarios, the Probability of Correct Acquisition is 28% and 16%, respectively, at a false acquisition rate of 0.%.

2.1.5.3 Pattern Sensitivities

The overall performance of the ASSP acquisition algorithms is excellent, especially considering the fact that this is their first application to field data. However, there are three patterns encountered in the Captive Flight Test which are troublesome. These patterns are illustrated in Appendix C: 1) Hole-in-the-trees, Figure C-21, row 1, column 1; 2) Square-building, Figure C-21, row 3, column 1; 3) Burning-oil-drum, Figure C-13, row 2, column 2.

The hole-in-the-trees pattern has the rectangular tank-like signature common to many of the target signatures in the training set. The square-building signature is nearly identical to known vehicle signatures such as the tanks in Figures C-6, row 3, column 3, and Figure C-7, row 3, column 2. The burning-oil-drum signature is similar to the square building.

These pattern sensitivities were wholly expected, in fact many more such sensitivities were anticipated. In each case the pattern is very similar to existing patterns in the training set which are designated as target-type signatures. It is felt that these patterns can easily be accommodated by the current acquisition algorithms, but to do so will require a retraining. For the first field test, the ASSP has been remarkably robust.

2.2 Weaponization

The Advanced Sensor Signal Processor developed under this contract was designed for the top-down attack of vehicular targets by multiple small-caliber (4 inch diameter) submunitions. These submunitions need to be low cost. Control and guidance hardware, therefore, needs to be minimized. Northrop has generated a new guidance scheme using solid state electronics in place of gyros. The contract included an engineering simulation task to explore the validity of this approach.

2.2.1 Description of Approach

Strapdown seeker proportional navigation guidance for tactical weapons has been the lofty ideal of guidance and control engineers for many years. Potential savings of 20 percent or more of the total weapon cost in this high volume "expendable" sensor market place as well as reduced complexity and increased reliability all enhance the desirability of this technically intriguing issue. The problem is to devise a way to replace the complex and potentially fragile inertially stabilized seeker of a conventional proportional navigation (PRONAV) guidance system selected for the submunition (Figure 2.2-1), with a potentially rugged body fixed (i.e., "strapdown") seeker (Figure 2.2-2).

Since an inertial line-of-sight angle (LOS) rate ($\dot{\lambda}_I$) is required for the PRONAV guidance law, an estimate of body attitude ($\hat{\theta}$) is combined with the body fixed (seeker look angle) LOS to the target (λ_B), then differentiated, to yield an estimate of the inertial LOS rate ($\hat{\dot{\lambda}}_I$). The remainder of the guidance law is then implemented in the classical proportional manner, wherein $\dot{\lambda}_I$ is driven to zero by virtue of the "proportional" vehicle turn rate command ($\dot{\gamma}_{CM}$). The weapon will converge on the target whenever the navigational constant $K_N \geq 2$.

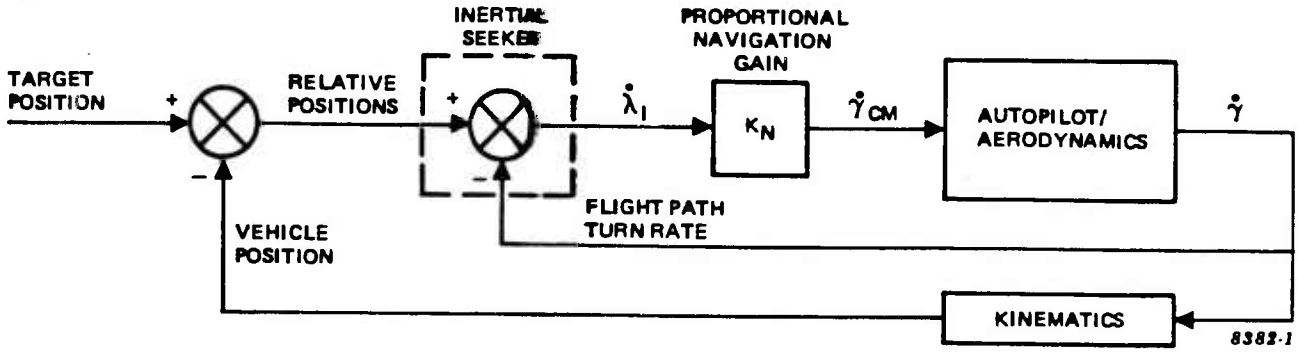


FIGURE 2.2-1. CLASSICAL INERTIALLY STABILIZED SEEKER PROPORTIONAL NAVIGATION GUIDANCE

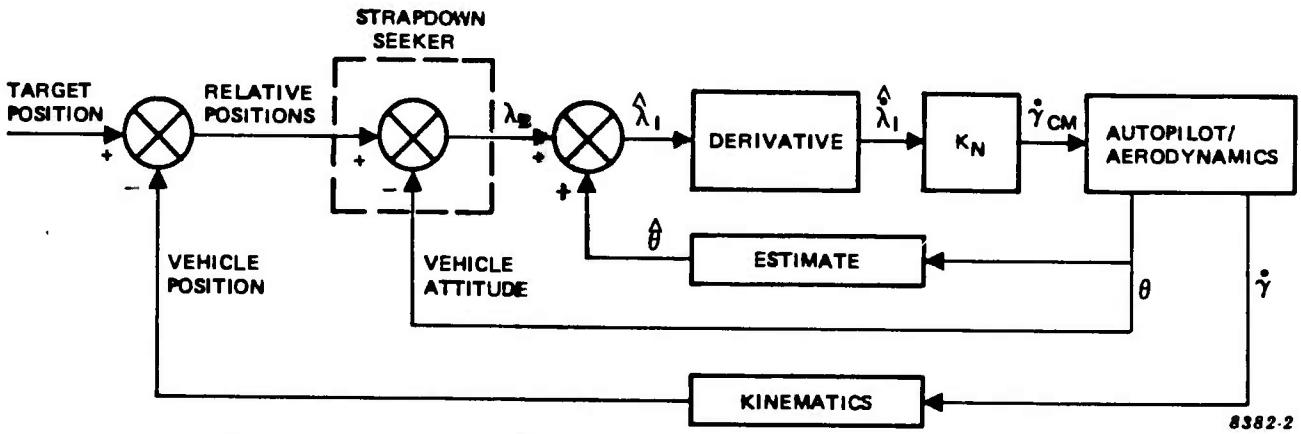


FIGURE 2.2-2. STRAP-DOWN SEEKER PROPORTIONAL NAVIGATION GUIDANCE

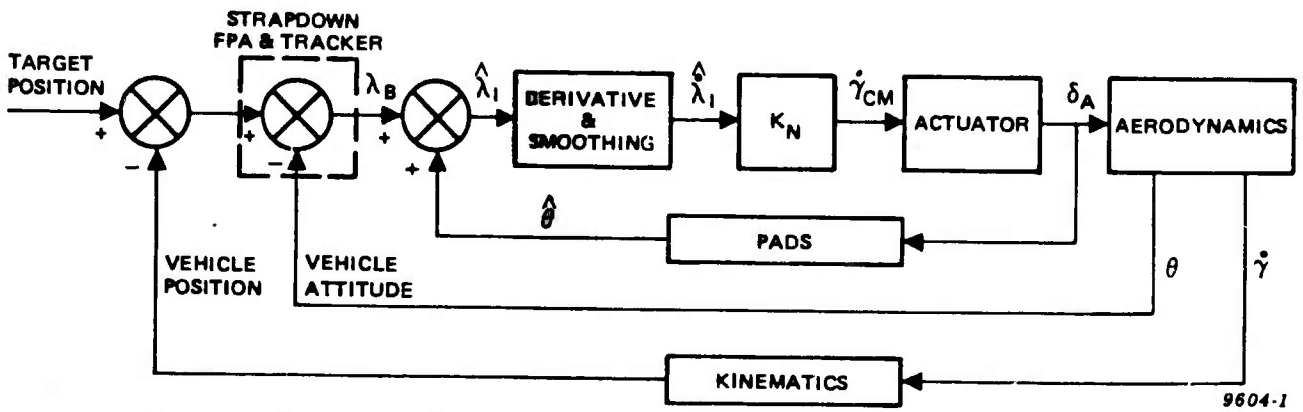


FIGURE 2.2-3. PADS – BASED, STRAP-DOWN SEEKER PROPORTIONAL NAVIGATION GUIDANCE

The standard approaches to strapdown PRONAV have used inertial instrumentation (typically gyros) to estimate the vehicle attitude, and so call "linear" detectors for the strapdown seeker. The linear detectors were typically one of two types: single element spot detectors wherein electrical voltage differentials between four or more collectors indicate spot position off center, or imaging or pseudo-imaging sensors (like TV cameras) with electronically scanned patterns and trackers for target location. Predicted performance of these types of systems has been widely studied including a definitive 1978 report on the subject by Emment and Ehrich[1], sponsored by the U.S. Air Force. They noted that even if seeker resolution versus field of regard trades could be acceptably solved, seeker dynamic gain variation coupled with gyro gain variations can cause positive feedback instability even for very tight component tolerances (< 5 percent net variation caused instability for the configurations studied). Some clever suggestions to "minimize" these effects were postulated, but the generic difficulties in the traditional approach are now well documented.

The advent of a new Focal Plane Array (FPA) staring detector technology and advances in microelectronic processor technology have allowed the formulation of a new strapdown seeker PRONAV approach for a substantial class of "restricted" tactical mission applications. One of the two major precepts of the approach also renders it gyroless and thereby accrues additional cost savings. The guidance law is again configured as in Figure 2.2-2, with the strapdown seeker being a truly linear focal plane array (when tied to an imaging tracker). If the initial conditions of the engagement are well known (e.g., vehicle attitude, velocity, control configuration, etc.) then the estimate of body attitude can be provided by a classical six-degree-of-freedom vehicle simulation computed in real time and updated on the basis of guidance command history. For simple airframes $\dot{\gamma}_{CM}$ is proportional (by the airframe gain (K_{AF})) to the control command (δ_{CM}) and the airframe response is predictable from the actual instantaneous control condition (δ_A). This is effectively an open-loop integration predictor scheme, and has been named the Proportional Airborne Digital Simulator (PADS). The PADS-based control loop is shown in Figure 2.2-3.

[1] Emment, R. I., and Ehrich, R. D. - Strapdown Seeker Guidance for Air to Surface Tactical Weapons AFATL-TR-78-60, 1978 - Eglin AFB, Florida

The new approach replaces potentially destabilizing positive feedback conditions with open-loop integration and a heavy dependence on well-known and controlled aerodynamics, kinematics, atmospherics and engagement conditions. The PADS premise is that it can provide a low-cost, low-mechanical complexity, high reliability, potentially rugged strapdown-seeker gyroless proportionally controlled guidance approach for a significant class of restricted mission applications with:

- Well-known initial conditions
- Well-behaved velocity profile
- Limited maneuver requirements
- Well-known aerodynamics and kinematics
- Short engagement times

This section summarizes the analytical studies which validate the above premise for a specific top-down, autonomous-acquisition guided anti-armor mission. The study results, which include performance margin evaluation on the basis of parametrically degraded conditions, indicate a marked ability to readily achieve almost twice the required accuracy (1σ). This implies a strong potential for application to most similar engagements for extensions to other types of missions.

2.2.2 Scenario Assumptions for PADS Study

The PADS premise was validated against a specific application whose engagement scenario is as shown in Figure 2.2-4. The weapon is deployed in the vicinity of a moving or stationary target, and slowed to a terminal velocity of 250 feet per second by virture of a drag device. On the basis of this known velocity, the front mounted strapdown Focal Plane Array Seeker and associated image processor "locks on" to arbitrarily selected patches of the ground to measure roll rate and estimate altitude. The weapon is a CG-mounted cruciform wing-controlled vehicle with stabilizer tails. While attached to a drag device, the vehicle controls to zero roll rate using separately actuatable wings and seeker ground lock roll inputs as feedback. This roll control is maintained throughout the engagement. Once roll rate is zeroed, the seeker locates and begins to track to the most likely armor-type target in the field of view. (The

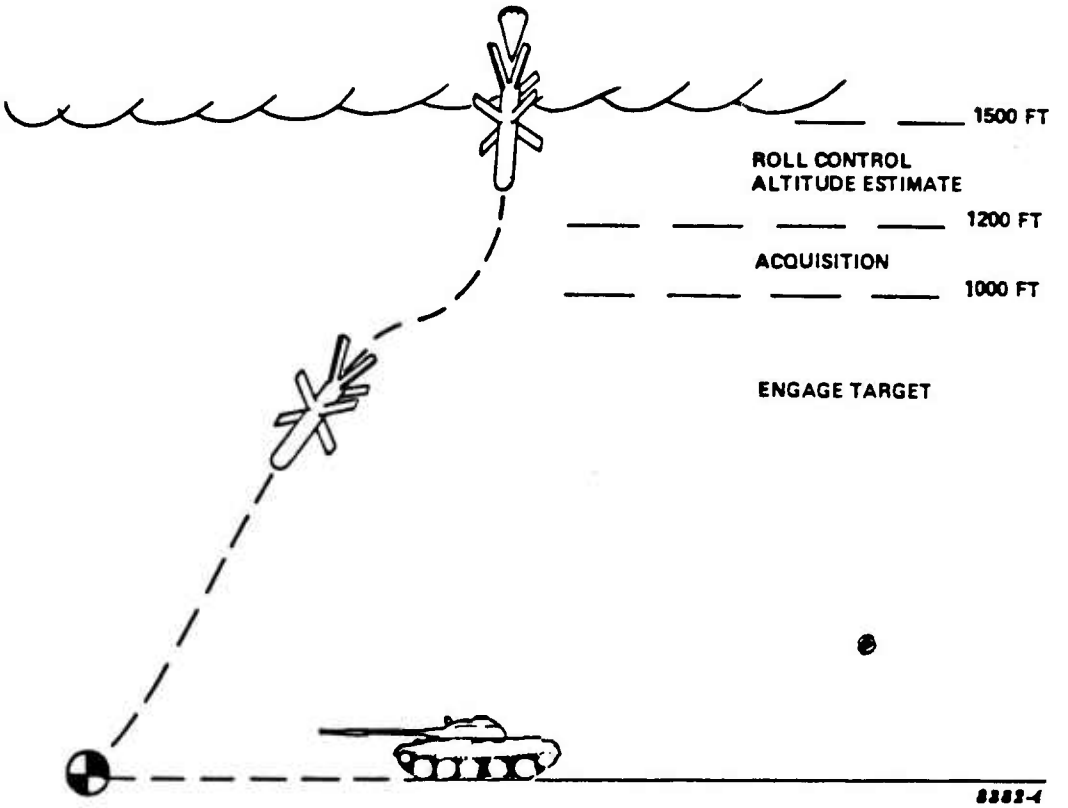


Figure 2.2-4. PADS Engagement Scenario

methods for lock-on and tracking are not directly related to the guidance study and, as such, will not be considered herein except as their accuracies, computational characteristics and sample rates affect guidance accuracy.) At the onset of tracking (approximately 1,000 feet above ground level), the drag device is released and PADS-based guidance begins.

The "Known" nominal conditions, therefore, are:

Speed:	250 feet per second
Attitude:	o° (vertical descent - both control axes)
Control:	o° net - X and Y axes
Acceleration:	1G less nominal aerodynamic drag
Aerodynamics:	Nominal
Kinematics:	Nominal

Max Control:	±15° - each axes
Max Footprint:	~ ± 500 feet each axes
Field of Regard	±15°
Time to Impact:	~ 4 seconds
Guidance Stop:	3.5 seconds after start (blind range of seeker at 100 feet above target)

The control footprint allows engaging any armor target on the boundary of the acquisition field of regard.

2.2.3 PADS Validation

In addition to target effects (position, speed and maneuvers), there is one major generic issue that could cause the system to fail this mission: PADS open-loop predictor failure. This could be caused by any one of three types of problems, or combinations thereof.

1) Aerodynamic mismatch - Manufacturing variations (e.g., bent or oversized wings) from one round to the next or variations in meteorlogical conditions (e.g., air density) will result in non-nominal aerodynamic performance. Major differences between the nominal characteristics programmed into the PADS onboard computer and the actual vehicle response could yield classical system instability. Minor variations could degrade accuracy. Errors in the assumed initial velocity by virtue of altitude estimation errors or drag device variation can also cause aerodynamic mismatch.

2) Sampling, Delays and Noise - The system consists of a digitally sampled seeker (FPA), computation heavy digital tracker and finite processing time digital PADS simulation. The sample rates and transport delays of a realizable system could make the PADS impractical or impossible (especially when adding seeker and electronic noise problems). Compromise in round offs, filters and approximations could degrade accuracy.

3) Atmospherics and Open-Loop Integration - The nominal wind condition assumed is 0° fps in each direction. While this is typically not the case, wind speed is not instrumented and will cause an unanticipated initial motion relative to the target. Wind gusts will cause the airframe to undergo unanticipated transients. In either case, as well as most of the non-catastrophic effects caused by error types (1) and (2) above, small unaccounted for errors will be perceived by the seeker as apparent target motion. These will be integrated across the engagement time in open-loop fashion and will typically lead to terminal impact inaccuracies.

The validation was accomplished in four steps: 1) classical linear stability analysis (Root Locus) of the basic control loops including estimation of the stability limits of generic system mismatches in gain, frequency and damping, 2) mechanical and meteorlogical tolerance and variation estimation to compare expected mismatches to stability limits, 3) formal six-degree-of-freedom, time-based simulation modeling to explore non-linear sampling, delay and open-loop integration effects as well as detail a baseline design for the PADS computer, and 4) Monte-Carlo type evaluation of baseline system sensitivity to parametrically varied contributing error sources.

2.2.3.1 Linear Analysis, Stability Limits

The center of gravity mounted cruciform wing airframe with stabilizer tails was analyzed aerodynamically to estimate linearized transfer of its control features. At 250 fps, the airframe is very lightly damped (ratio ≃ .15) with a natural frequency of around 3 Hz. The frequency and damping do not change substantially over the flight regime, where impact velocities are about 300 fps. A filtered (or smoothing) derivative and a time scheduled gain product (K_N/K_{AF}) were determined to provide adequate stability. The primary dynamics of the linearized model can be identified from Figure 2.2-3 and are the aerodynamics, the similar aerodynamics predicted in PADS, the system kinematics, and derivative smoothing and the actuator. (It was shown that with sufficiently high response, the actuator effects could be neglected.)

The kinematics term in all PRONAV control loops is a pole in the right half of the S-plane, whose position is dependent on the inverse of the instantaneous time before impact, or $1/t_{go}$. As $t_{go} \rightarrow 0$, the open-loop pole position tends towards positive infinity, that is, increasingly unstable. In a perfect PRONAV loop (with perfect differentiator and no smoothing) a $K_N \geq 2$ will keep the pole stable. However, all realizable PRONAV systems will be driven to an unstable condition by this effect some time immediately before impact. This numerical instability can be seen intuitively by considering the magnitude of the body fixed seeker angle on a non-maneuvering vehicle falling exactly one foot away from a target. At long ranges (e.g., 1,000 ft), the angle is very small and grows very slowly. Just before impact (e.g., 2 or 3 feet away), the angle starts to grow very rapidly, until it reaches 90° at impact. This analogy is also useful in understanding that numerical instability just before impact does not imply a significant miss distance.

The nominal PADS-based system has a blind range (i.e., the target is so large that the seeker can no longer see it) about .25 seconds before impact, at which time no further control action is taken. Analyses verified that the kinematic pole was the primary destabilizing effect in the linearized PADS-based model. Therefore, stability was more difficult for small t_{go}. As such, t_{go} = .5 seconds was selected as the critical stability point; that is, if all control loop poles are still in the stable (left-half) region of the S-plane at .5 seconds before impact, the particular conditions being studied are adequately stable.

Using this criteria, the system generic stability limits on the basis of gain, damping and frequency mismatch between PADS and true aerodynamic responses were estimated for the subject system. Figure 5 is a plot of the damping ratio of the closed loop positions of the dominant pole pairs (kinematic and airframe) for the baseline guidance law and t_{go} = .5 seconds (seconds before impact). The damping ratios are plotted as a function of the percentage of mismatch in airframe gain between the nominal gain anticipated by PADS and the actual gain experienced by the dynamic airframe.

Once numerous copies of the baseline airframe are actually tested and the PADS is programmed to their nominal value, the likelihood of substantial round-to-round variation is small. Such variations, if any, could be caused by manufacturing tolerances (especially in the wings) or variations in air density or airspeed. Figure 2.2-5 shows that the system is dynamically stable for actual airframe gains of 20 percent below to 15 percent above the nominal PADS value, or a ± 17.5 percent stability band (or "bandwidth") about the central stability point.

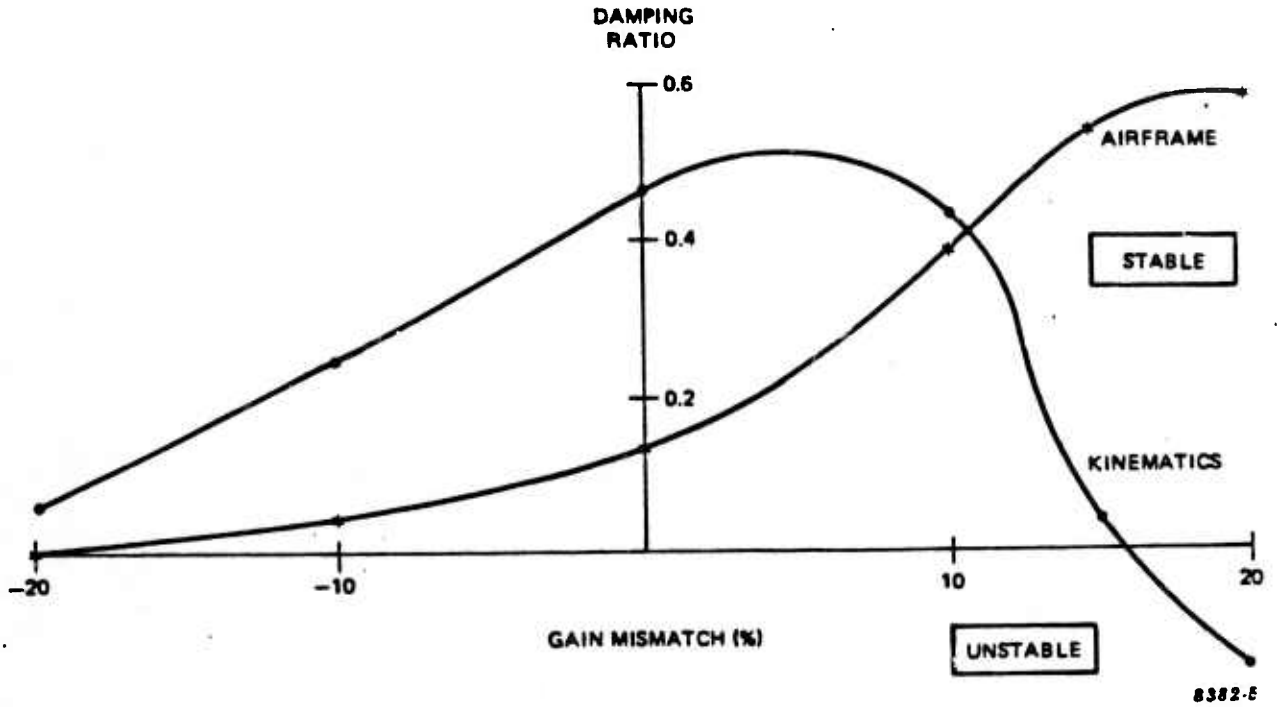


Figure 2.2-5. Gain Mismatch Stability Limits

Figure 2.2-6 shows a similar plot for percentage differences between nominal and actual airframe open-loop damping estimates. Figure 2.2-7 shows the limits of stability for airframe undamped natural frequency mismatches. In both cases, it can be observed that a mismatch in dynamic terms such as damping and natural frequency gives rise to a third set of dominant poles, directly attributableto the PADS feedback. The stability bandwidth for damping mismatch is ±15 percent about the central stability point, and for natural frequency mismatch it is ±6 percent.

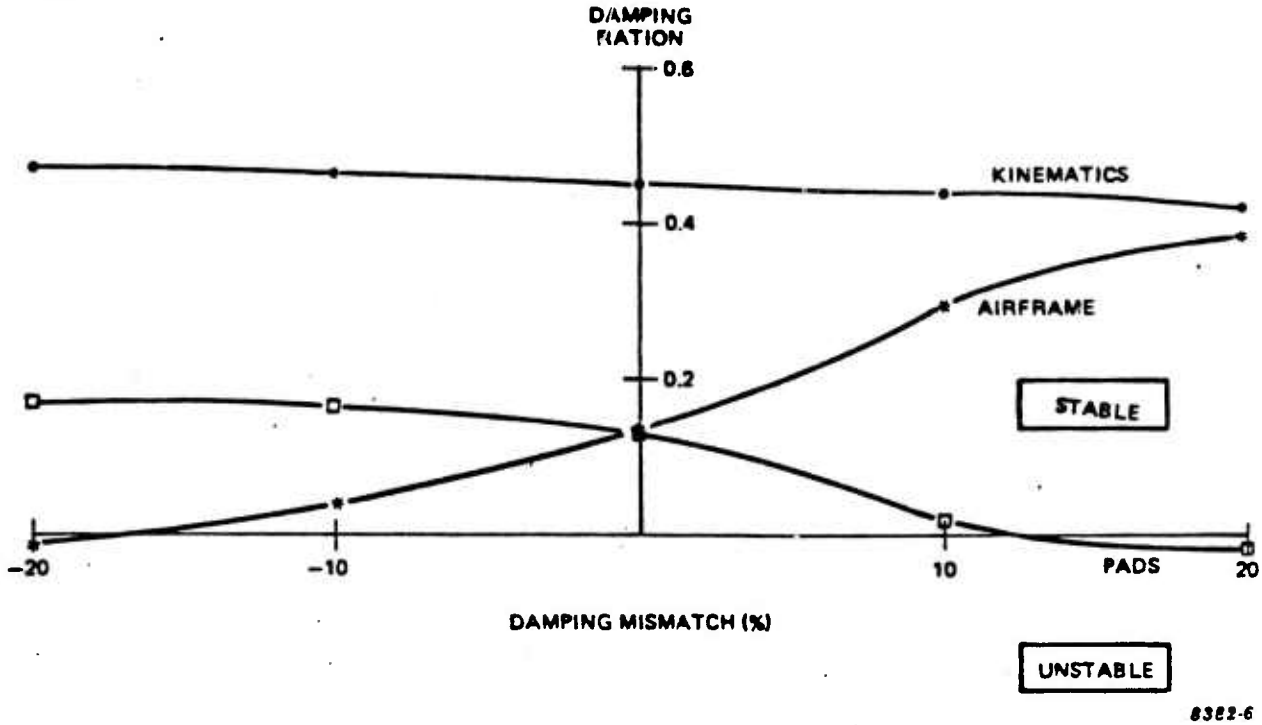


Figure 2.2-6. Damping Mismatch Stability Limits

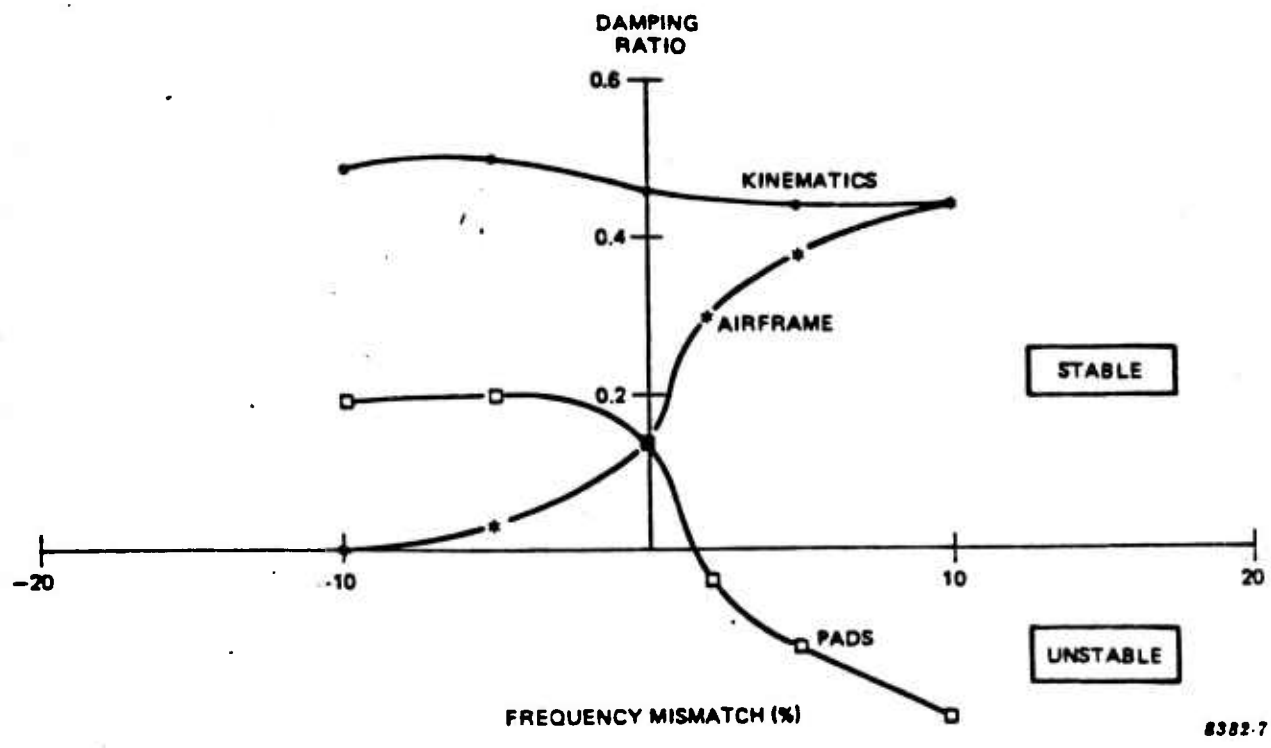


Figure 2.2-7. Natural Frequency Mismatch Stability Limits

It is interesting to note that these stability bandwidths are heavily related to the baseline airframe. This is significant since the baseline airframe open-loop characteristics are very lightly damped and, therefore, barely stable. A more conventional airframe would be expected to have wider stability bandwidths and hence be even more suitable for PADS-based guidance.

2.2.3.2 Tolerance Stability

In order to assess the significance of the determined stability bandwidths, it is necessary to evaluate those manufacturing and meteorological conditions that could cause round-to-round variations in each of the generic mismatch parameters. Potential variations in system parameters due to manufacturing tolerances were estimated by means of a manufacturability study. In the study, a production specialist estimated the reasonable tolerances achievable without special tooling and costly assembly techniques. Tolerances on each airframe component were estimated. Preliminary aerodynamic estimates on the effects of having any one element at tolerance showed low sensitivity. Therefore, the mechanical tolerances were clustered additively (not RSS) into two classes: Geometric tolerances (wing length, chord, tail spacing, etc.), and mass balance tolerances (weight, c.g. longitudinal, c.g. axial, etc.). Recognizing that inspections would reject most components out of tolerance, the study conservatively allowed the tolerance value to be considered 1σ and tripled the individual values of the clustered parameters to achieve a 3σ estimate. The aerodynamics of the plus and minus 3σ airframes were then compared to the nominal airframe and the plus or minus 3σ percentage variation in the generic parameters (gain, frequency and damping) were assessed.

Similar techniques were also used to estimate the plus or minus 3σ variations due to a mismatch of estimated initial airspeed and a mismatch of assumed air density (for air density the limits of earth measured air density "from the peak of Mt. Everest to Death Valley" was used for the 3σ limits). Table 2.2-1 summarizes the generic parameter most affected by each of the major error sources, the width of the variation band due to plus and minus 3σ conditions, the width of the PADS stability band (from Step 1) for the generic parameter

MAJOR ERROR SOURCES	GENERIC PARAMETER	±3σ BANDWIDTH	PADS TOLERANCE BANDWIDTH	STABILITY ASSESSMENT
GEOMETRIC TOLERANCES	NATURAL FREQ.	3%	12%	STABLE
MASS BALANCE TOLERANCES	NATURAL FREQ.	4.8%	12%	STABLE
AIR DENSITY	FREE A/F DAMPING	9.6%	30%	STABLE
AIRSPEED AT ACQUISITION	A/F GAIN	10.2%	35%	STABLE
	NATURAL FREQ.	5.2%	12%	STABLE
RSS OF ALL SOURCES	NATURAL FREQ.	7.8%	12%	STABLE
	FREE A/F DAMPING	11.0%	30%	STABLE
	A/F GAIN	14.2%	35%	STABLE

8382-25

Table 2.2-1. Stability Tolerance Summary.

and an assessment as to the stability expected using a 3σ condition. Not only did all four independent error sources fail to cause an unstable condition, but as also shown in Table 2-1, a root sum of squares composite of all 3σ tolerances and variations is also predicted to yield a stable system.

Thus, the tolerance stability concludes that, on the basis of the classical linearized model, dynamically stable system operation is expected of a PADS-based system operating under nominal to plus or minus 3σ conditions.

2.2.3.3 Time Based 6 DOF Simulation

Classical linear analysis techniques are useful in establishing basic anticipation of system stability. Certain aspects of system performance are not, however, conveniently accomplished through linear analysis. Specifically, effects of sample rates (the seeker/tracker outputs a λ_B at a 20 Hz rate), image processor transport lag (around 1/20 second), discrete updates of the PADS equations (for realizability of the real-time PADS processor a 20 Hz update rate was assumed), open-loop integration of net errors (incorrect initial conditions, biases, mismatch errors), and blind range terminated guidance. In addition, the general topic of total system delivery accuracy against realistic targets is not convenient.

Time based six-degree-of-freedom vehicle and two-degree-of-freedom target simulation were selected as the medium for studying these effects. Figure 2.2-8 shows the block diagram of the simulation and the interconnection of the seven major modules: Target, Seeker, Guidance and Control Computer, Actuator, Airframe, Atmospherics, and Miss Distance. A short description of each model is also available at the end of this section. As indicated in Figure 2.2-8, the engagement starts at the point of initial target acquisition by the sub-missile. As the target moves and the submissile descends, a relative line of sight is measured (and distorted) by the seeker model. This signal, in turn, is processed by the Guidance and Control Computer (including the

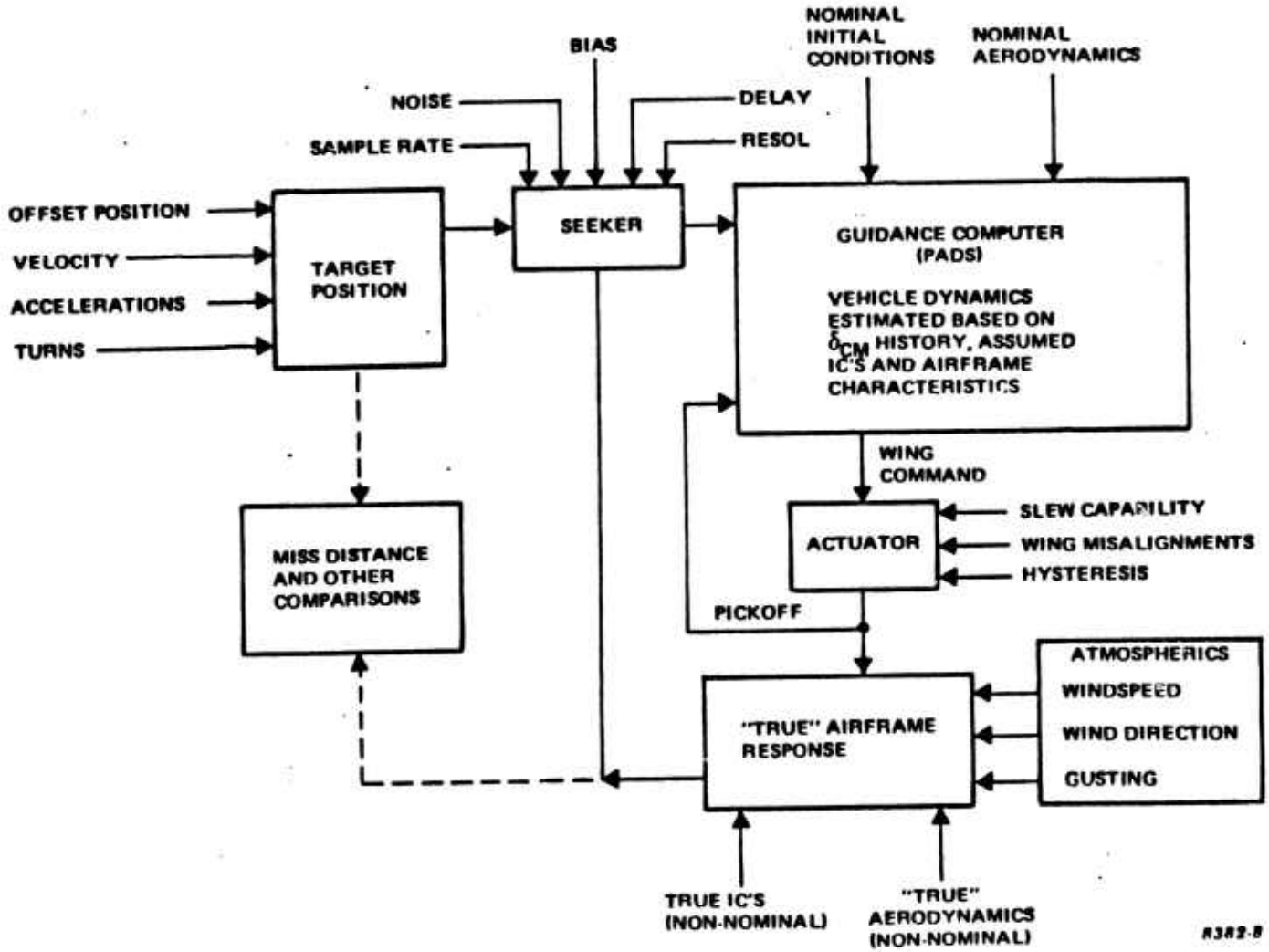
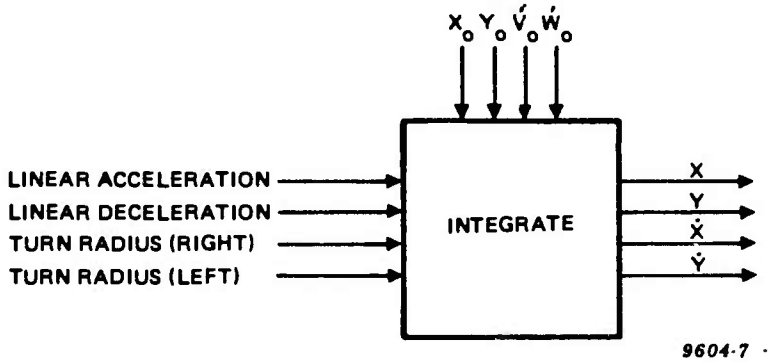


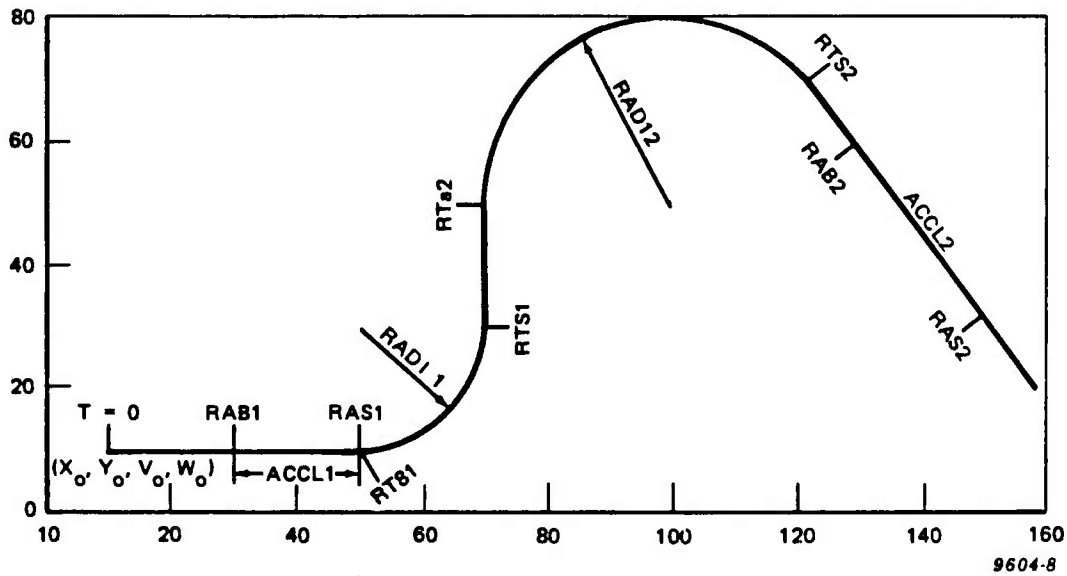
Figure 2.2-8. 6-DOF Block Diagram

MODEL: TARGET

BLOCK DIAGRAM



SAMPLE RUN

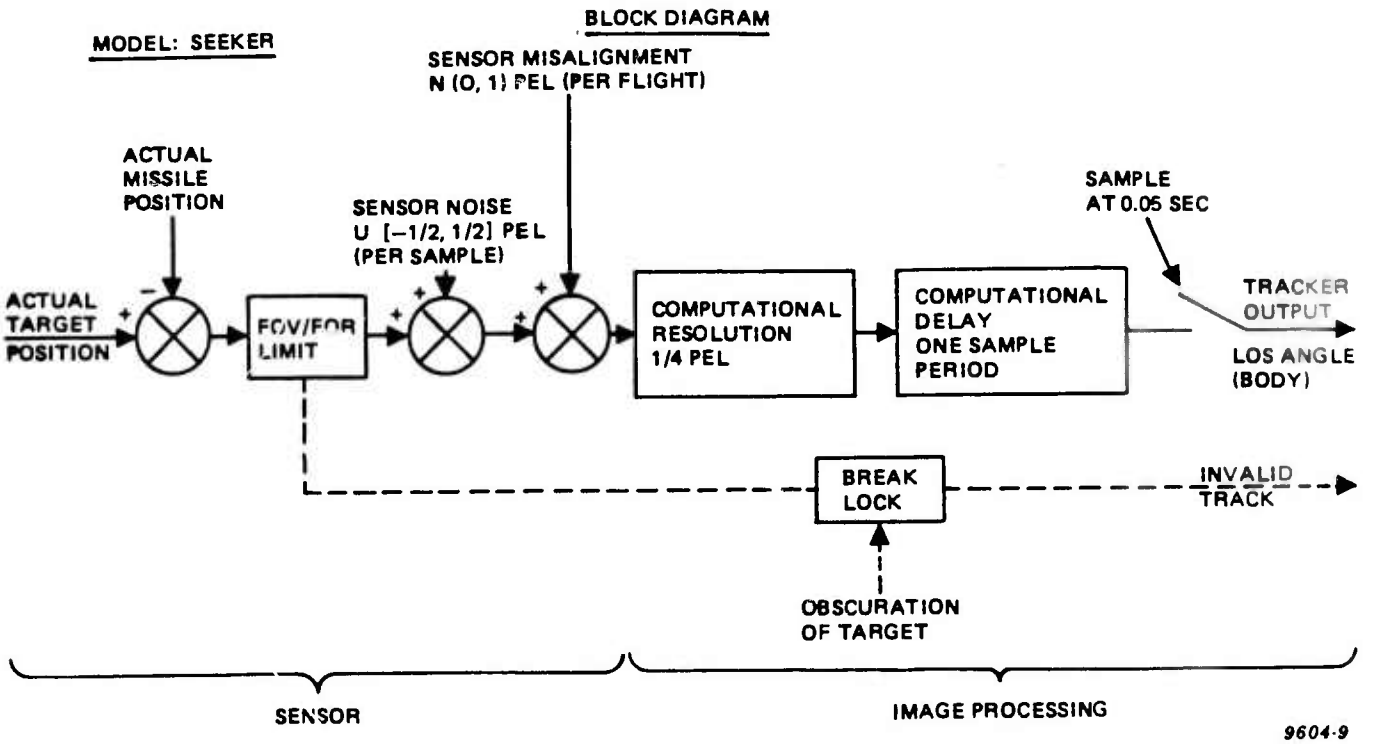


DESCRIPTION

THE TARGET IS CAPABLE OF MANEUVERING WITH CONSTANT VELOCITY , LINEAR ACCELERATION OR DECELERATION (TWO DIFFERENT ACCELERATIONS CAN BE APPLIED DURING EACH RUN) AND TURN RIGHT OR LEFT WITH A CERTAIN RADIUS (TWO DIFFERENT TURNS CAN BE CONDUCTED DURING EACH RUN). THE INPUTS OF THE MODEL ARE INITIAL POSITION, SPEED AND HEADING, TWO SETS OF STARTING AND STOPPING TIMES FOR ACCELERATION OR DECELERATION, AND ANOTHER TWO SETS OF STARTING AND STOPPING TIMES FOR TURNING LEFT OR RIGHT. THE OUTPUTS OF THE MODEL ARE TARGET POSITION AND VELOCITY IN THE X AND Y AXES.

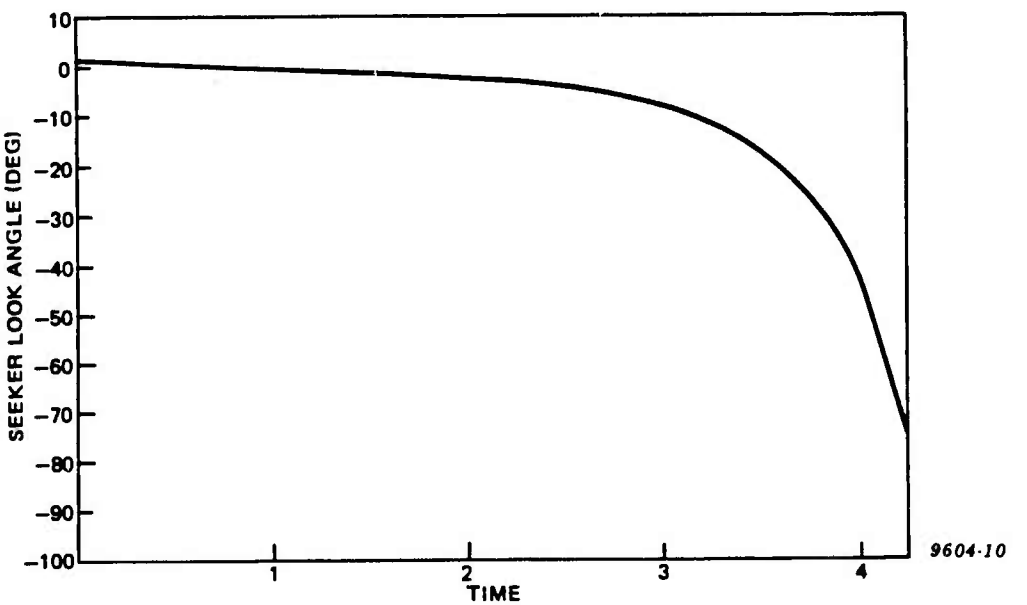
MODEL: SEEKER

BLOCK DIAGRAM



9604-9

SAMPLE RUN



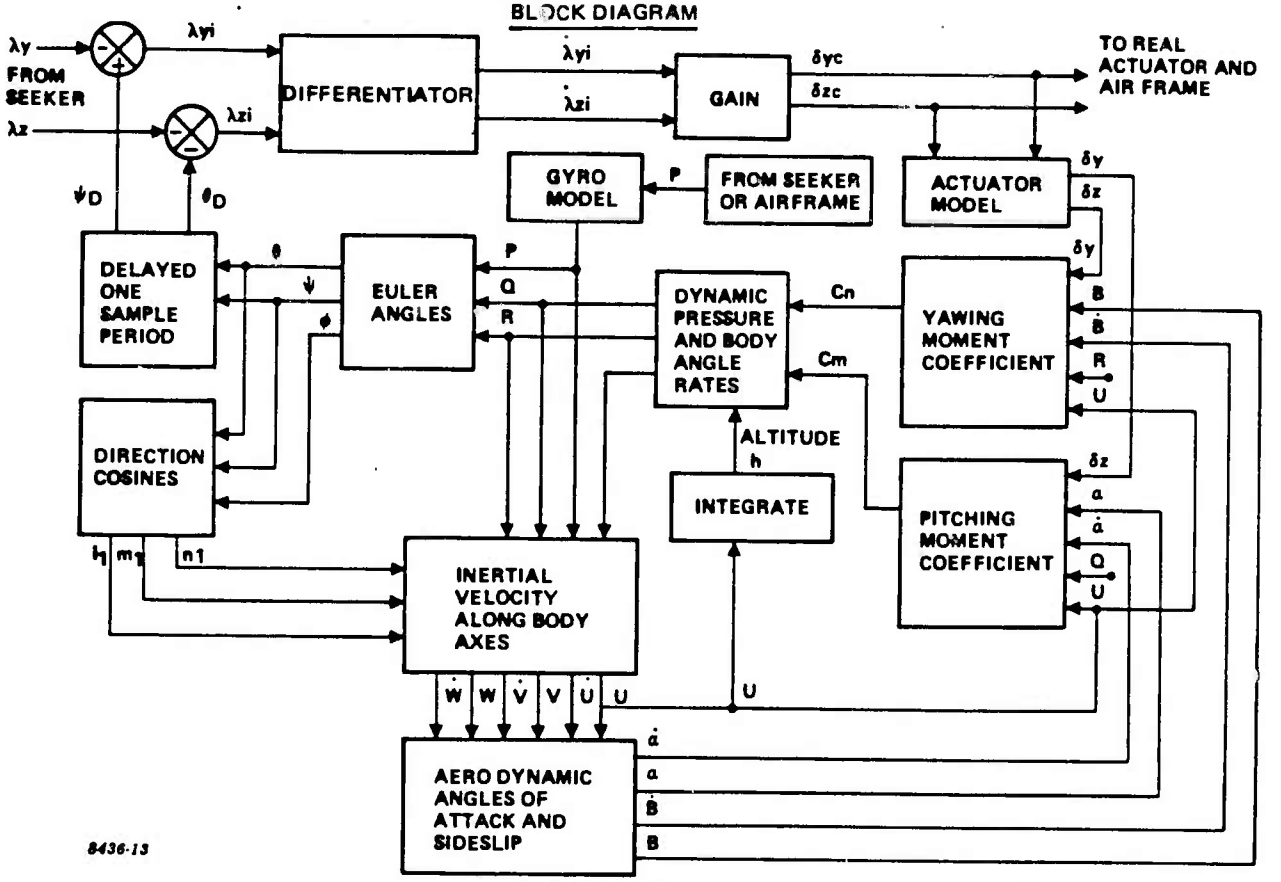
9604-10

DESCRIPTION

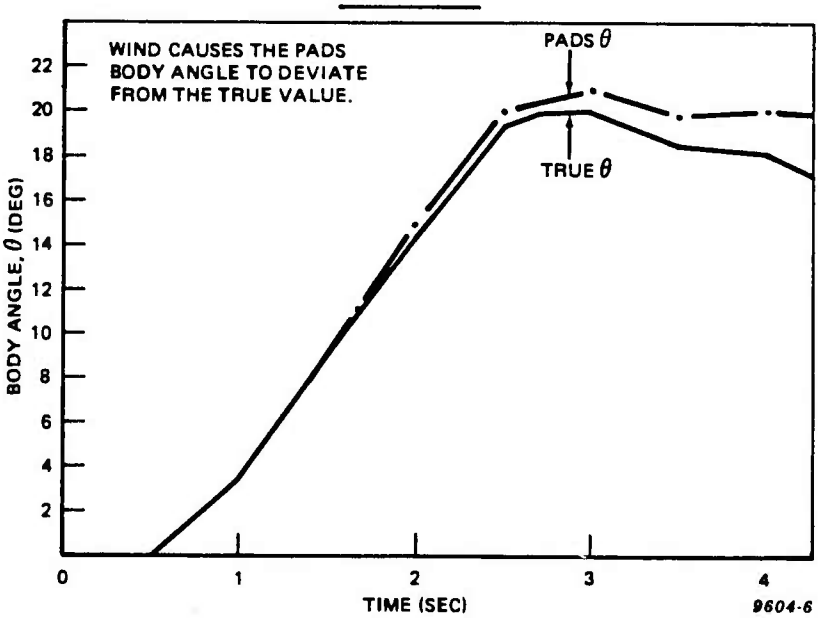
THE SEEKER OUTPUTS A CORRUPTED AND DELAYED LINE OF SIGHT ANGLE RELATIVE TO THE BODY BASED ON ACTUAL TARGET AND MISSILE POSITION INPUTS. THE SEEKER MODEL CONTAINS SENSOR AND IMAGE PROCESSING CHARACTERISTICS; SUCH AS, FOV LIMITATION, SENSOR NOISE, COMPUTATIONAL RESOLUTION AND DELAY, SAMPLING AND BREAK LOCK.

MODEL: SIMULATED AIRFRAME RESPONSE (PADS)

BLOCK DIAGRAM



SAMPLE RUN

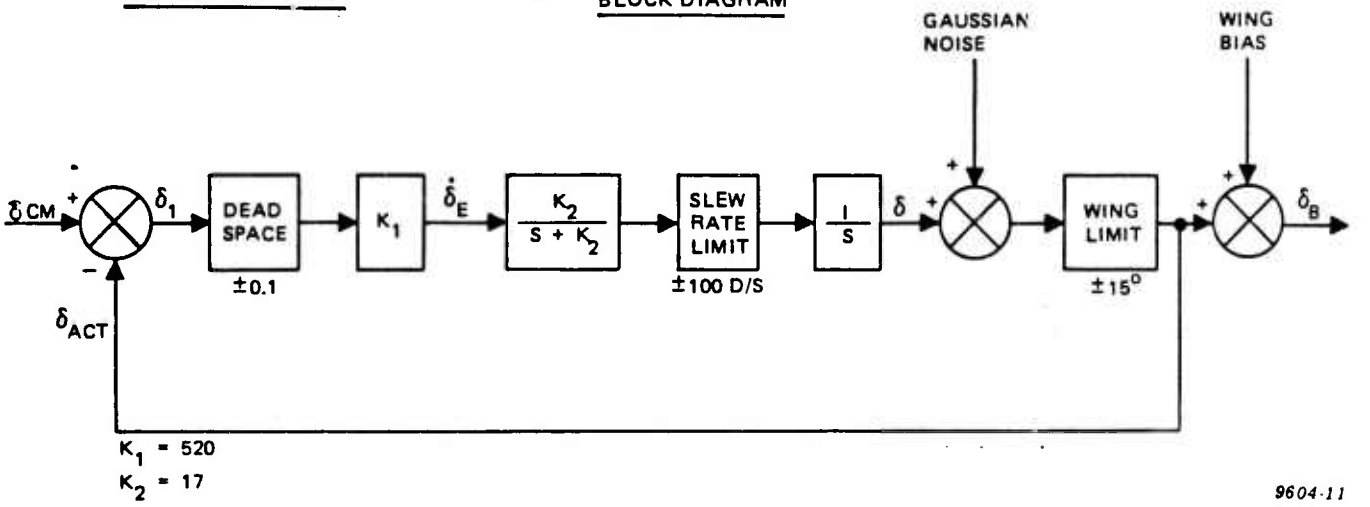


DESCRIPTION

THE BODY LINE-OF-SIGHT PRODUCED BY THE SEEKER IS COMPARED AGAINST THE ESTIMATED BODY ANGLE, COMPUTED BY PADS TO PRODUCE AN INERTIAL LINE-OF-SIGHT ANGLE WHICH IS FED INTO A LEAST SQUARES FILTER TO PRODUCE A FILTERED INERTIAL LINE-OF-SIGHT ANGLE RATE. THE GUIDANCE COMPUTER USES THE SMOOTHED INERTIAL LINE-OF-SIGHT ANGLE RATES TO GENERATE WING COMMANDS TO THE ACTUATORS (IN ORDER TO MANEUVER THE SUBMISSILE TOWARDS THE TARGET). THE MODEL THEN ESTIMATES THE VEHICLE DYNAMICS BASED ON A HISTORY OF WING COMMANDS, AIRFRAME CHARACTERISTICS, AND ASSUMED INITIAL CONDITIONS.

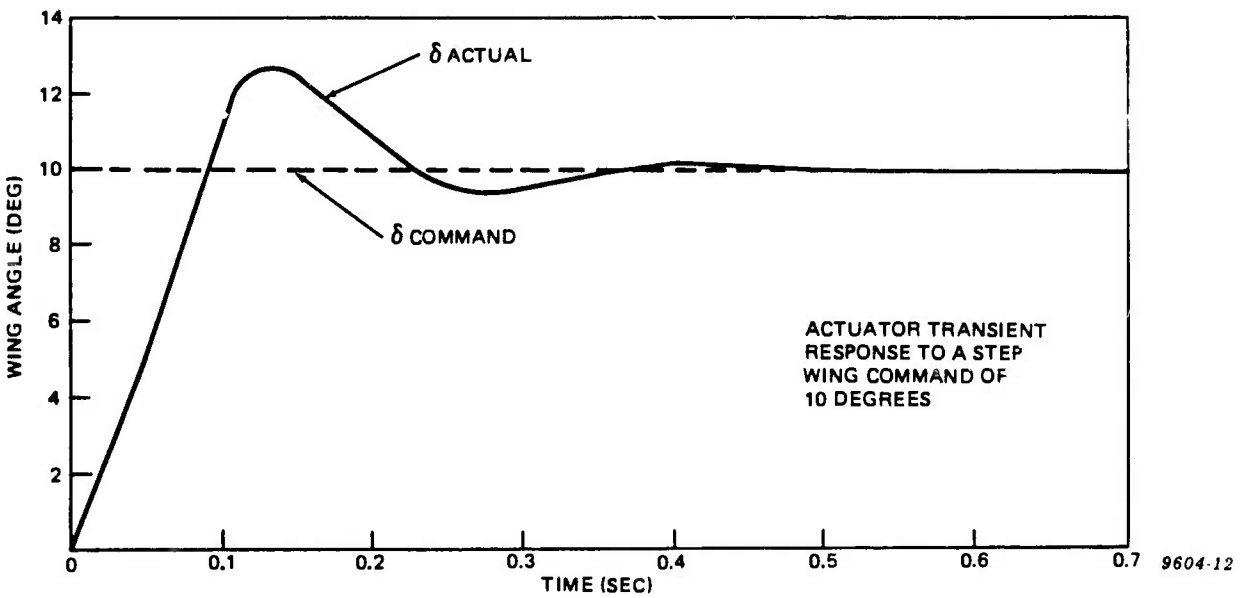
MODEL: ACTUATOR

BLOCK DIAGRAM



9604-11

SAMPLE RUN



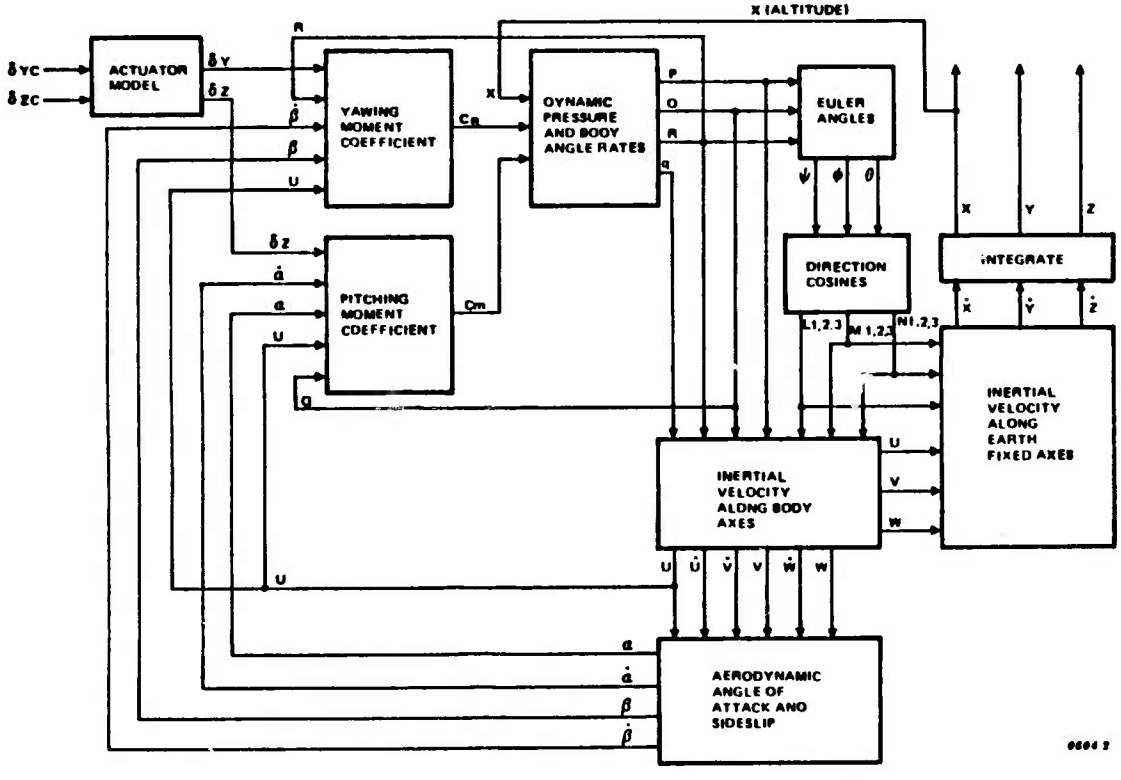
9604-12

DESCRIPTION

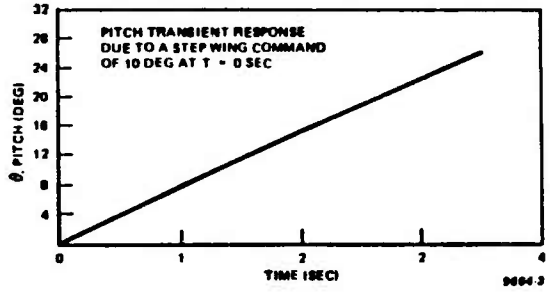
THE INPUTS TO THE ACTUATOR MODEL ARE COMMANDED WING DEFLECTIONS, FROM THE GUIDANCE COMPUTER, WHICH ARE COMPARED WITH THE ACTUAL POSITION AND, HENCE PRODUCE AN ERROR TERM. THIS OUTPUT IS THEN FED TO A DEAD SPACE FUNCTION WHICH WILL SIMULATE A HYSTERESIS SINCE IT IS POSITIONED IN A FEEDBACK LOOP SYSTEM. THE OUTPUT IS THEN MULTIPLIED BY A GAIN WHICH WILL CONVERT IT TO A RATE. THE TORQUE RESPONSE OF THE ACTUAL SYSTEM DRIVE IS MODELED BY A FIRST ORDER TRANSFER FUNCTION AND SLEW RATE LIMITER. THE OUTPUT IS THEN INTEGRATED TO YIELD THEORETICAL WING POSITION. A GAUSSIAN NOISE IS ADDED TO THE WING RESPONSE, AND THE POSITION IS LIMITED AT THE ACTUATOR POSITION STOPS. THIS OUTPUT (POSITION PICKOFF) IS THEN FED BACK TO THE BEGINNING OF THE LOOP TO BE SUBTRACTED FROM THE WING COMMAND. FINALLY, A WING BIAS IS ADDED TO THE WING LIMITER OUTPUT; THIS CORRUPTED AND DELAYED WING RESPONSE BECOMES THE OUTPUT OF THE ACTUATOR MODEL.

MODEL: "TRUE" AIRFRAME RESPONSE

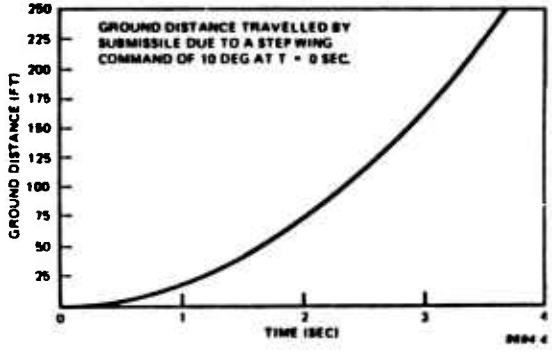
BLOCK DIAGRAM



SAMPLE RUN



SAMPLE RUN



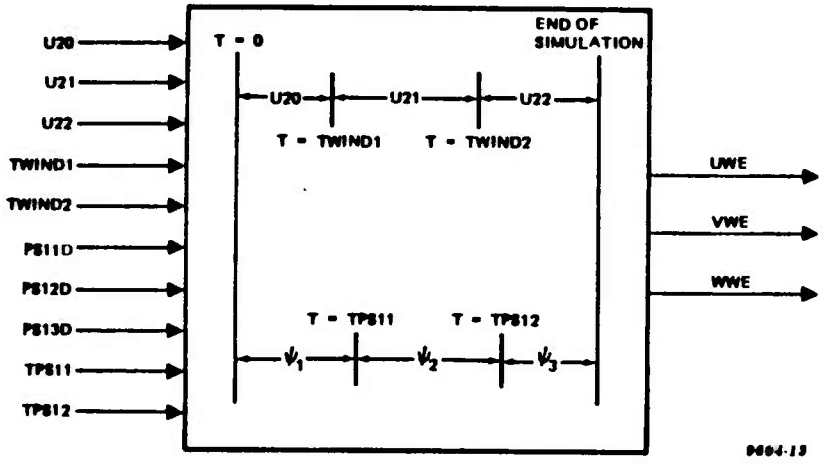
DESCRIPTION

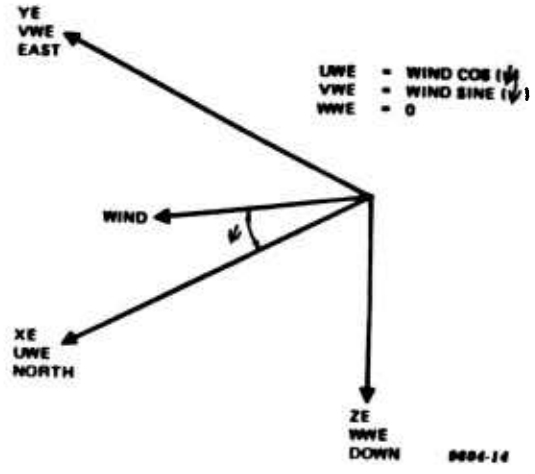
THE MODEL SIMULATES A SUBMISSILE FALLING UNDER THE FORCE OF GRAVITY TAKING INTO ACCOUNT THE AERODYNAMIC FORCES, MOMENTS, WINDS, GUSTS AND WING DEFLECTIONS. THE MODEL KEEPS TRACK OF THE SUBMISSILE'S "TRUE" POSITION, VELOCITY, ACCELERATION, ATTITUDE ANGLES, BODY RATES, AND WING DEFLECTIONS WHICH LATER SERVES AS A COMPARISON AGAINST THE COMPUTED VALUES BY THE PAOS ONBOARD COMPUTER

9604 5

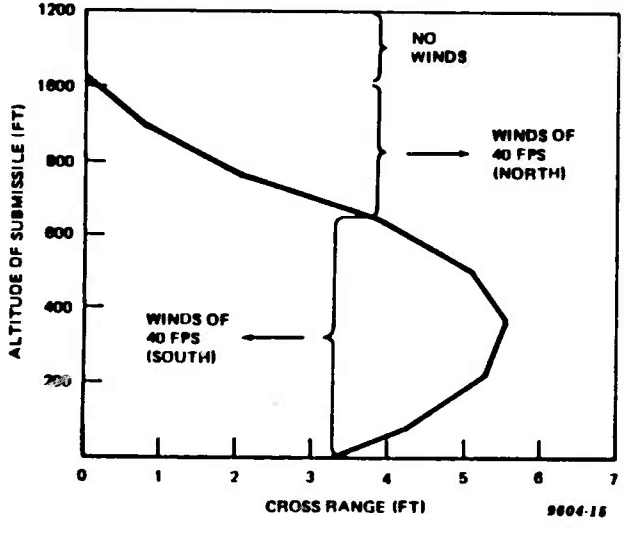
MODEL: ATMOSPHERICS

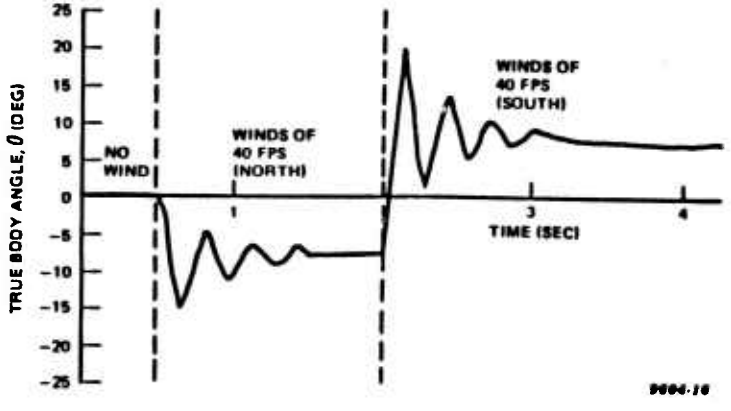
BLOCK DIAGRAM





SAMPLE RUN



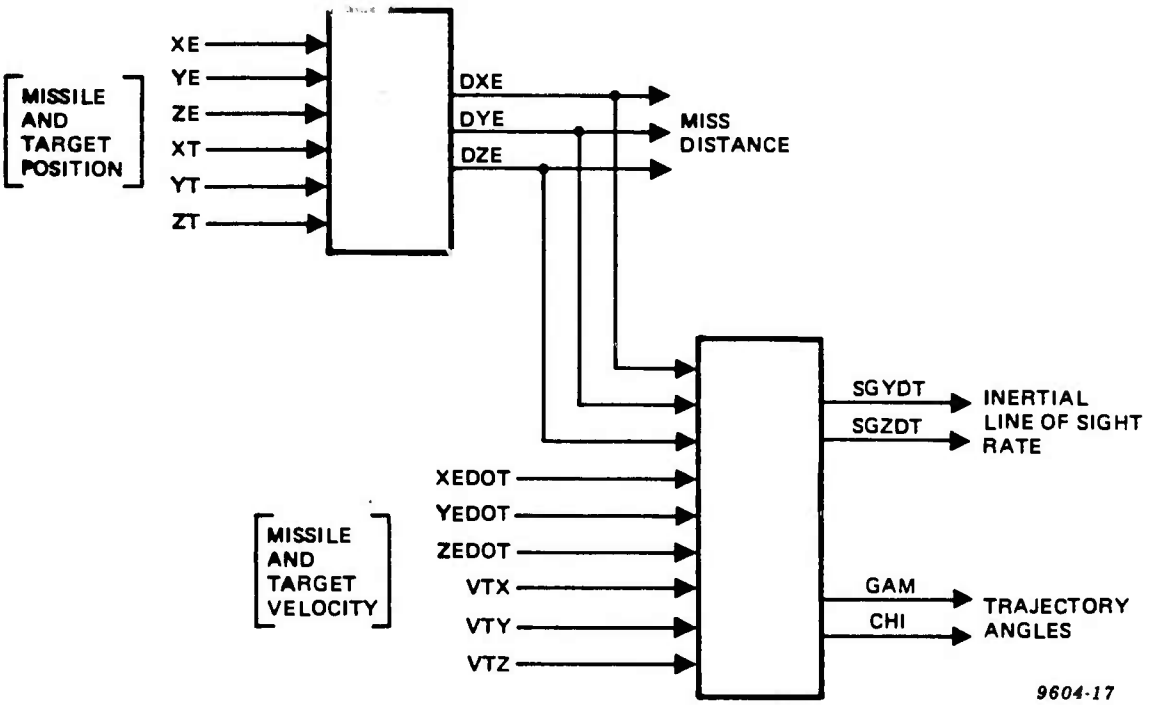


DESCRIPTION

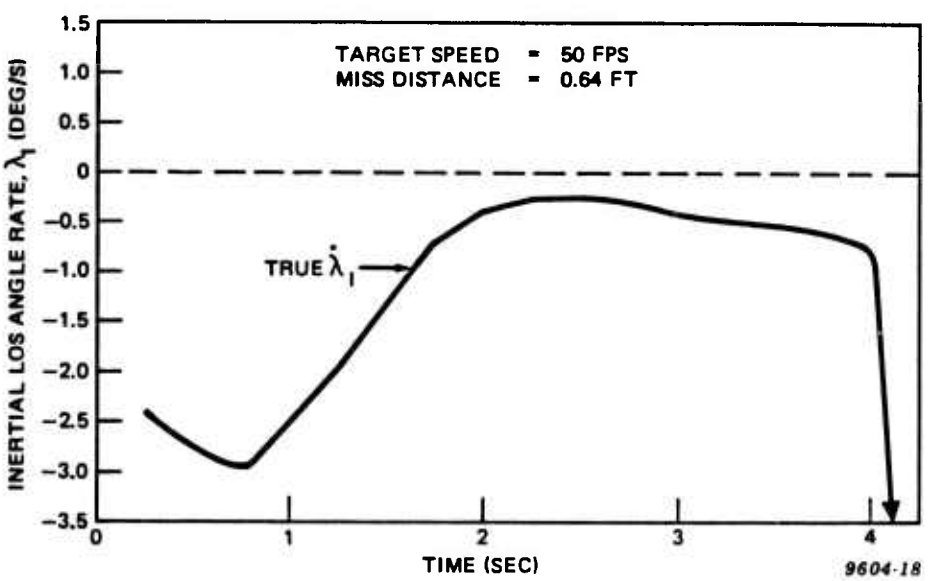
THE INPUTS TO THE MODEL ARE THREE WIND VELOCITIES AND DIRECTIONS. THE OUTPUTS ARE WIND COMPONENTS (EARTH AXIS) AT THE APPROPRIATE TIME AND DURATION. AT THE START OF THE ENGAGEMENT, A WIND OF VELOCITY "U20" FT/S IS BLOWING FROM THE "PSI1O" DIRECTION. AT TIME = TWINO1, THE VELOCITY CHANGES TO "U21" AND AT TIME = TS12, THE DIRECTION CHANGES TO "PS12D." THE FINAL CHANGE IN WIND SPEED AND DIRECTION OCCURS AT TIME = TWIND2 WHERE THE WIND SPEED CHANGES TO "U22" AND AT TIME = TPS12, THE WIND DIRECTION CHANGES TO "PS13O."

MODEL: MISS DISTANCE AND OTHER COMPARISONS

BLOCK DIAGRAM



SAMPLE RUN



DESCRIPTION

THE INPUTS ARE THE TARGET AND MISSILE POSITION AND VELOCITY. THE OUTPUTS ARE MISS DISTANCE, INERTIAL LINE-OF-SIGHT RATE, AND MISSILE TRAJECTORY ANGLES.

Proportional Airborne Digital Simulator - PADS - Equations). The G&C computer model will generate commands to the wing actuators (in order to maneuver the submissile towards the target). A small actuator model represents the dynamics of this servo, and provides to the "True" aerodynamics the "actual" wing deflection (including errors). The Aero-Model computes the submissile moments based on aerodynamic conditions including ambient atmospheric events (wings, etc.). The equations of motion are exercised to yield actual submissile motion, and the loop is closed.

2.2.4 Benchmark Simulation

The PADS simulation has undergone thorough validation and system testing as a part of the series of "benchmark" or typical case engagement runs. The benchmark runs validate the simulation design and the aerodynamics models. The 6 DOF was also used to optimize the guidance system design. Figures 2.2-9 through 2.2-14 show typical results from single engagements using the 6 DOF.

A stationary target, initially 100 feet from the weapon's nominal drop position is engaged in Figures 2.2-9 and 2.2-10. The submissile's position in space is plotted as time varies in Figure 2.2-9. Guidance starts at 1,000 feet altitude and continues until blind range at feet (3.5 seconds of guidance). Dead reckoning carries the weapon to .2 feet from the nominal aimpoint.

Figure 2.2-10 shows the control parameter (the inertial line of sight (LOS) rate - $\dot{\lambda}_I$) versus time for the same engagement. For comparison, the inertial LOS rate that would have been measured by an ideal conventional gimballed inertially stabilized seekers is also shown in Figure 2.2-10. Several items of interest are worth noting in this Figure. An initial 1.5 degree per second guidance error is corrected by the PADS-based PRONAV divergence just before impact predicted in the linear analysis is clearly observed, conveniently near blind range.

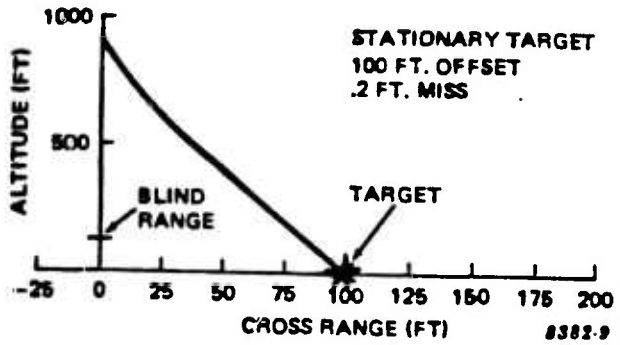


Figure 2.2-9. Stationary Target: Cross Range

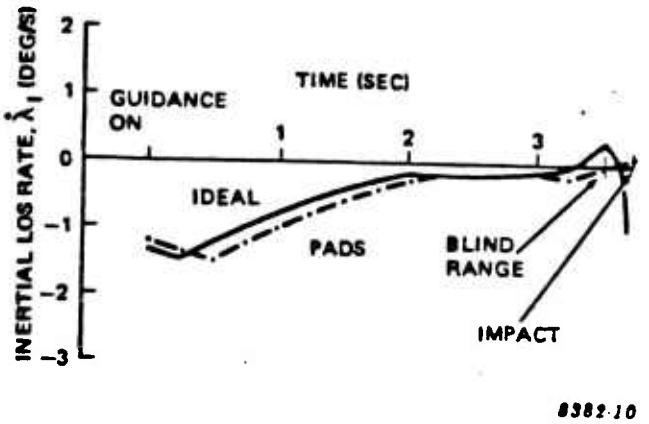


Figure 2.2-10. Stationary Target: LOS Rate

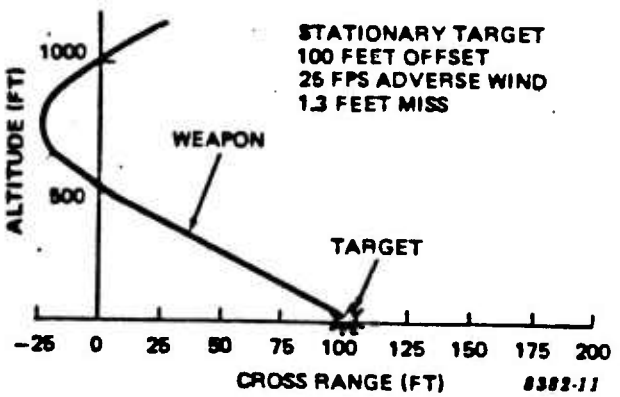


Figure 2.2-11. Wind, Cross Range

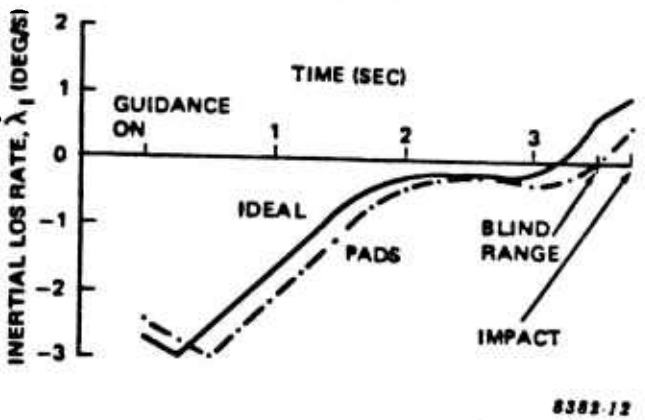


Figure 2.2-12. Wind: LOS Rate

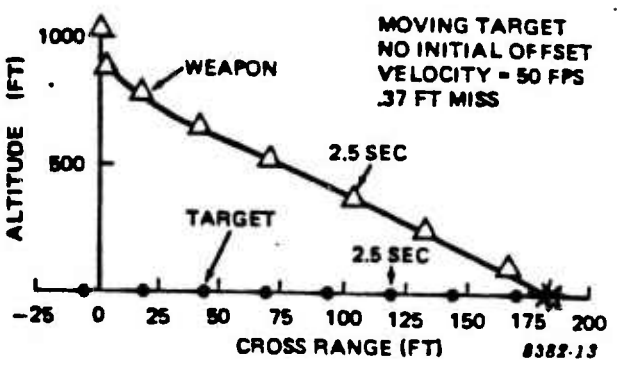


Figure 2.2-13. Moving Target: Cross Range

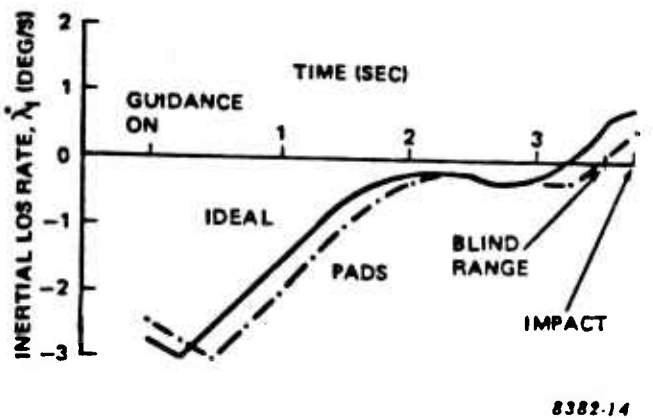


Figure 2.2-14. Moving Target: LOS Rate

Similar items of interest appear consistently for various engagements. Figures 2.2-11 and 2.2-12 are the analogous of 6 DOF results for the same engagement with 25 feet per second ambient winds initially blowing the weapon away from the target. Figures 2.2-13 and 2.2-14 consider a target racing at virtually top speed (50 feet per second) away from the weapon. Figure 2.2-13 also shows target and weapon position at corresponding points of time.

The basic simulation results validated the feasibility of operating a PADS-based PRONVA under realistic non-linear conditions. Further, accuracies achieved in the basis simulation were consistent with requirements for top-down, anti-armor attack. Specifically, this is taken to be no more than one meter (1σ) off the normal aimpoint. As the seeker is capable of selecting as its aimpoint, the vehicle centroid or a known vulnerable area instead of just the brightest hotspot, a PADS-based system achieving better than one meter accuracy will indeed be a lethal system.

2.2.5 System Sensitivity

The final step in validating PADS is to determine system accuracy under typical non-ideal conditions, then explore the degree to which further deteriorization in those conditions must extend in order to degrade system accuracy beyond acceptable limits. The 6 DOF simulation was used in a "psuedo Monte Carlo" fashion to evaluate these performance sensitivities, where one meter, 1σ, radial miss distance was considered to be the limit of acceptable performance.

Eight critical variables were identified as the dominant degrading parameters affecting system accuracy. A nominal condition set (1σ errors) of the eight parameters is listed with the error sources below:

1) Seeker/Tracker Noise - random noise of ±1/2 picture element (analogous to ±1/2 TV line) added to the true seeker output, with a new value selected each tracker update (20 times per second).

2) Wind Speed - 25 foot per second winds (meteorology tables indicate 25 fps winds or slower occur at 1,000 feet altitude 50 percent of the time that winds are measurable).

3) Wing Bias (Bent Wing) - 0° (no bias) was used as nominal, as the manufacturability study indicated sub-degree assemblies are readily achievable.

4) Air Density - 2.5 percent from world-wide average. (Meteorology tables indicate variations from 20 percent lighter (positive variance) density (Mt. Everest) to 10 percent heavier (negative variance) (Death Valley) are the measured world-wide extremes).

5) Actuator Slew Rate Mismatch - 5 percent variance between that assumed (PADS) and thatactually achieved (e.g., PADS assumes 100 degrees/sec unloaded slew rate, actual system delivers 95 degrees/sec).

6) Wind Gust - 0percent (none) was used as nominal, as instant step changes in wind velocity (sheer) is rare, however, should they occur, 10 percent ofambient wind speed would be typical.

7) Speed/Altitude Mismatch - 5 percent, where a non-nominal descent speed (say 262fps versus 250) would couple directly into an initial altitude error (1050 foot actual, while assuming 1000 feet) by virtue of the altitudeestimation processing. Faster/higher is considered a positive mismatch.

8) Target Motion - 20 miles per hour (29.33 fps) away from the weapon, with a 1 second turn at radius 156 feet. When motion sensitivity was evaluated, the turn was eliminated and replaced by a target full acceleration starting at one second after guidance initiation.

The composite effect of these simultaneous errors is shown for a single engagement in Figure 2.2-15. The Figure again shows the ideal inertial LOS rate that would have been measured by a stabilized gimballed seeker, along side the actual control parameter ($\dot{\xi}_{\lambda_I}$) generated by the PADS and focal plane array seeker. The most prominent difference between this and the prior engagements (Figures 2.2-9 - 2.2-14) is the noise effect of the seeker, and the control parameter variation observable from 2 - 3 seconds due to the target acceleration. For this particular engagement, the radial miss distance was 1.2 feet.

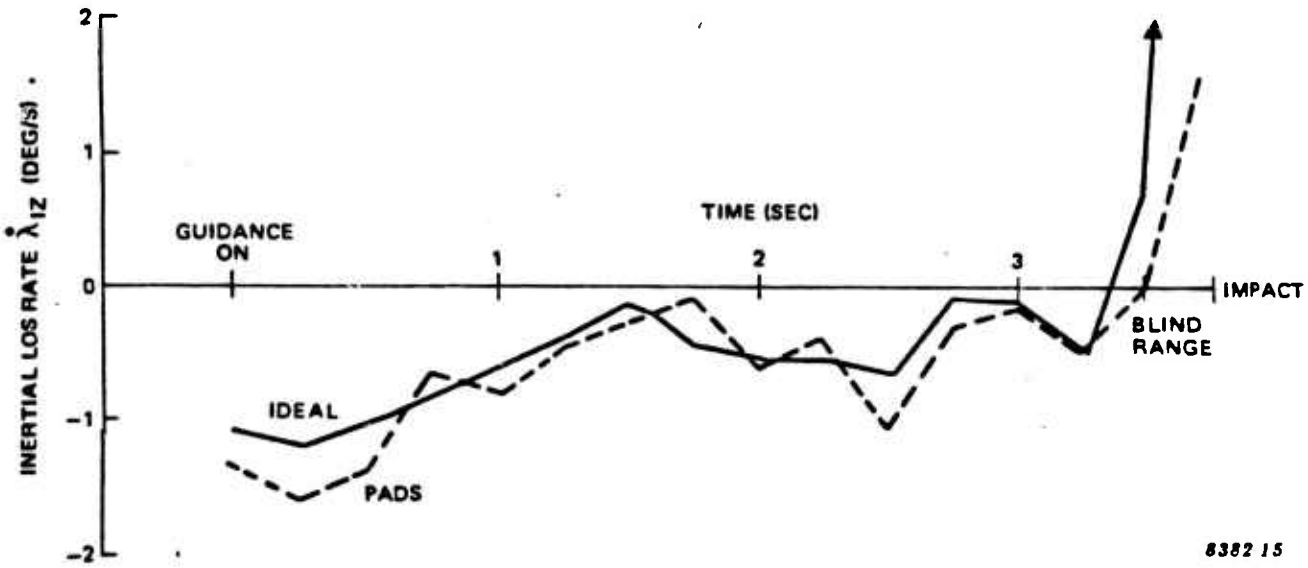


Figure 2.2-15. **Nominal Errors, Typical Engagement**

To determine sensitivities, a psuedo-Monte Carlo approach was used. Sensitivity to each parameter was evaluated with all other parameters held at their "1σ" error conditions (listed above). The most dynamic error parameter is the seeker/tracker noise for which a new error value is selected using a random number generator once every simulated guidance computer update cycle (20 Hz). Thus, for each set of error conditions (i.e., all 1σ except the study parameter at the particular 2σ or 3σ

or whatever value) three to ten individual engagements were run. Each run used a different seed for the seeker/tracker noise random number generator. The actual radial miss distance for each engagement was determined from the simulation results and the average miss for the considered conditions was computed. This method provides a good estimate of the performance statistics without the computer cost which would result from an exhaustive Monte Carlo study. The validity of this method was verified by spot checking points via 10 to 20 additional runs. In all cases, the larger sample mean was different from the smaller sample mean by less than 10 percent.

Figures 2.2-16 - 2.2-23 show the system sensitivities to each of the eight critical parameters. The curves show that when subjected to the 1σ error set (called out as "nominal" on each curve), the system accuracy is just over one foot, or three times better than the required one meter accuracy. The most critical parameter appears to be speed/altitude mismatch (Figure 2.2-22) where negative mismatches less than -10% cause large misses. The probability of exceeding the -10% mismatch and thereby causing large misses, is less than 5% (i.e., -10% mismatch is about a 2σ value). If more margin against negative mismatches is desired, then the PADS equations can be designed to have a positive speed/altitude error bias. The next critical parameter is ambient winds (Figure 2.2-17) where winds greater than 60 ft/s cause large misses but are not likely to occur (less than 5% of the time). It is interesting to note that no reasonable errors ($\leq 2\sigma$) from the eight critical parameters (Figures 2.2-16 - 2.2-23) can be expected to appreciably degrade system accuracy.

These studies validated PADS for the chosen application and indicated the types of bounds to its performance therein. A short study extension was added to the sensitivity work to attain some insight into PADS applicability to longer duration engagements. Specifically, the baseline system (with guidance gain scheduling and initial altitude assumption adjustments only, and assuming seeker resolution was adequate for tracking) was tested in

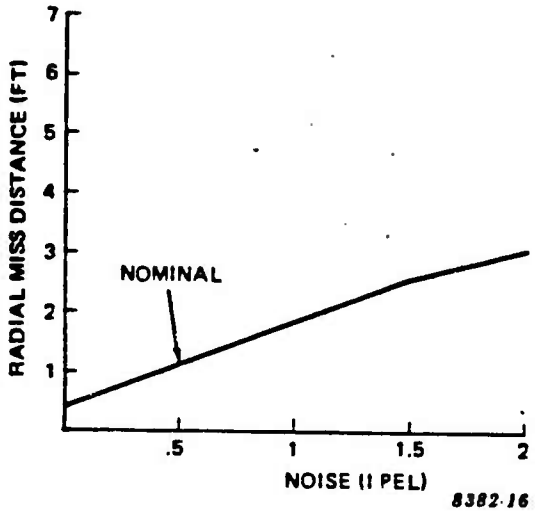


Figure 2.2-16. Seeker/Tracker Noise

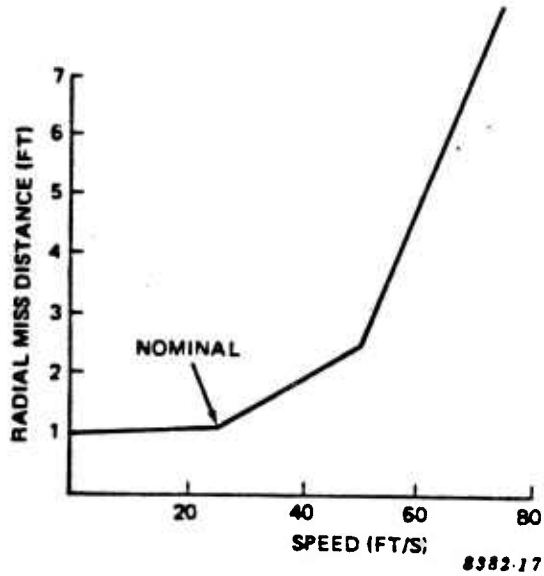


Figure 2.2-17. Wind Speed

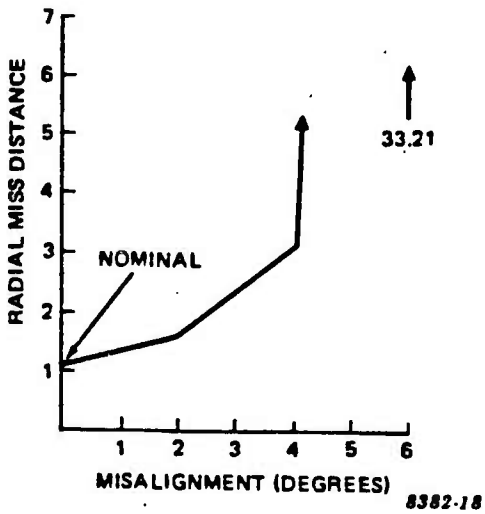


Figure 2.2-18. Wing Bias

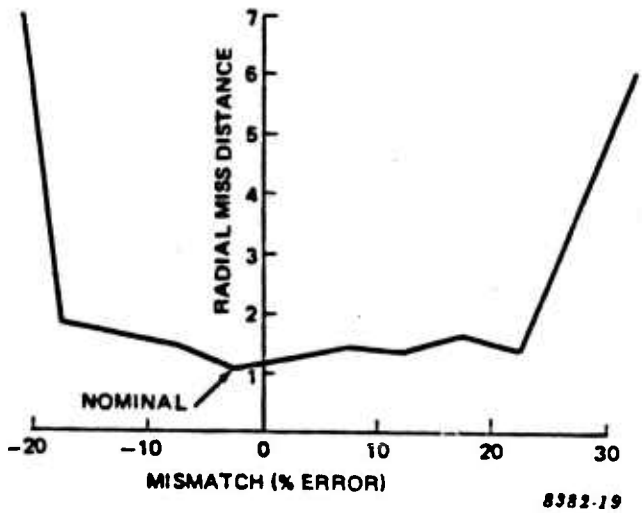


Figure 2.2-19. Air Density

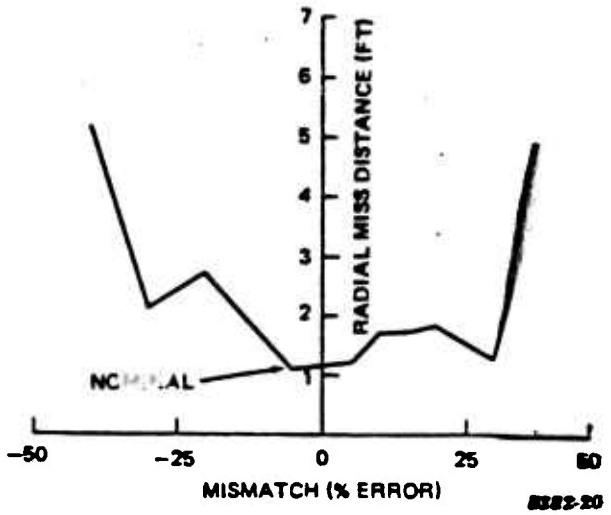


Figure 2.2-20. Actuator Slew Rate

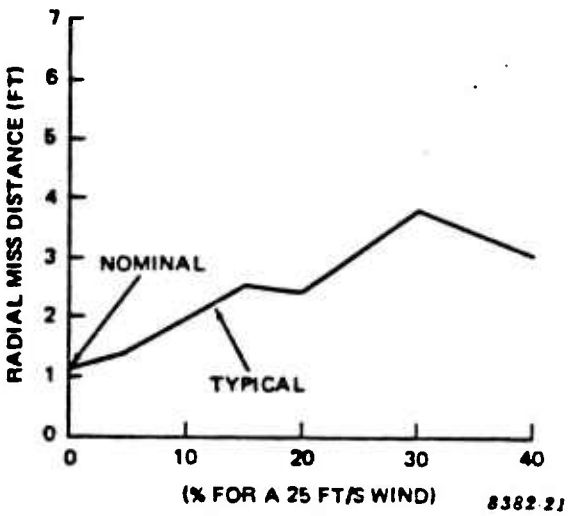


Figure 2.2-21. Wind Gust

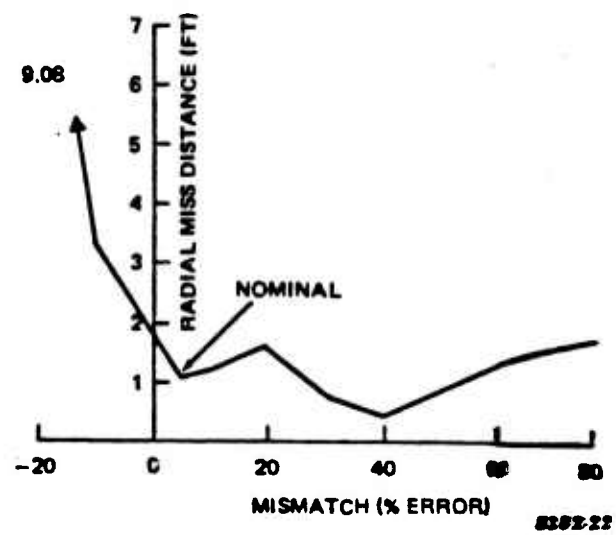


Figure 2.2-22. Speed/Altitude

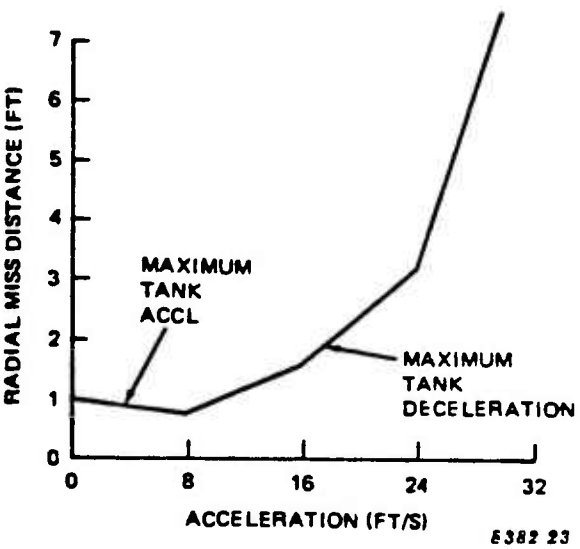


Figure 2.2-23. Target Motion

an anti-armor engagement starting at 3,000 feet (versus 1,000 feet nominal). With all error parameters at their above "1σ" values, and the addition of a second turn and two axial accelerations to the target maneuver, the typical engagement lasts about 10 seconds. A plot of control parameter versus time is shown in Figure 2.2-24. In this instance, the radial miss distance just exceeds two feet. Sensitivities to seeker/tracker noise and ambient winds at this long range were evaluated as before and showed only slightly greater sensitivity. One sigma errors yielded 1.6 foot radial misses and grew to 3 feet at 2 sigma conditions. An interesting aspect of the 10 second engagement (Figure 2.2-24) is that initially the ideal and PADS curves match well except for lag and noise (as before). However, towards the end of the engagement, the curves start to diverge. This is, no doubt, due to the now ten seconds of open-loop integration of imperfect assumptions and conditions. This observation tends to validate the validation process itself, since the residual effects cannot in theory be completely eliminated.

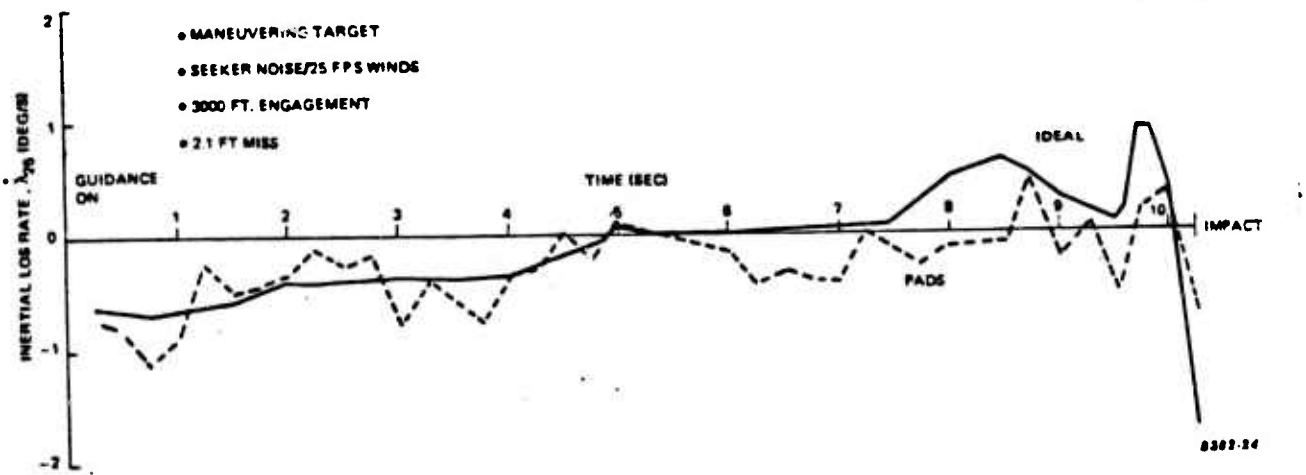


Figure 2.2-24. Ten-Second PADS Engagement

2.2.6 PADS Processor Development and HIL Demonstration

A real-time PADS processor (M68000) was developed and an HIL simulation was created in order to demonstrate PADS feasibility under real-time constraints. The following hardware was used to implement the simulation (Figure 2.2-25):

1) Digital computer (PADS)
2) Carco table
3) TV camera
4) Controller
5) Analog computer
6) VATS brassboard (tracker)

A three-axis rate table (Carco table) was used to simulate submissile motion as well as target motion. An operational seeker was not available; therefore, the ASSP seeker was simulated by a TV camera mounted on a rate stabilized Carco table. The camera video signal was then fed to a Northrop-built Video Augmented Tracking System (VATS) Preproduction Automatic Digital Tracker, to generate "body referenced" line of sight angle to the target. The VATS Tracker used here is quite similar to the conceptual ASSP Tracker and, therefore, is used as a good model. The tracker information was then read by the guidance (PADS) computer (a Motorola 68000 microprocessor) at a 20 Hz rate. Communication between the tracker and digital computer was carried out in assembly language. PASCAL language was used by the digital computer to estimate the airframe response and to compute wing commands. The "wing" commands were sent to the analog computer via two eight-bit dacs (y and z). The analog computer was used to simulate the submissile true airframe response (5 DOF) and target motion, thereby modeling the real world difference of "analog" airframe versus digital (PADS-based) "autopilot." Computed airframe body angles relative to the target were communicated to the Carco table through the controller, thus closing the loop.

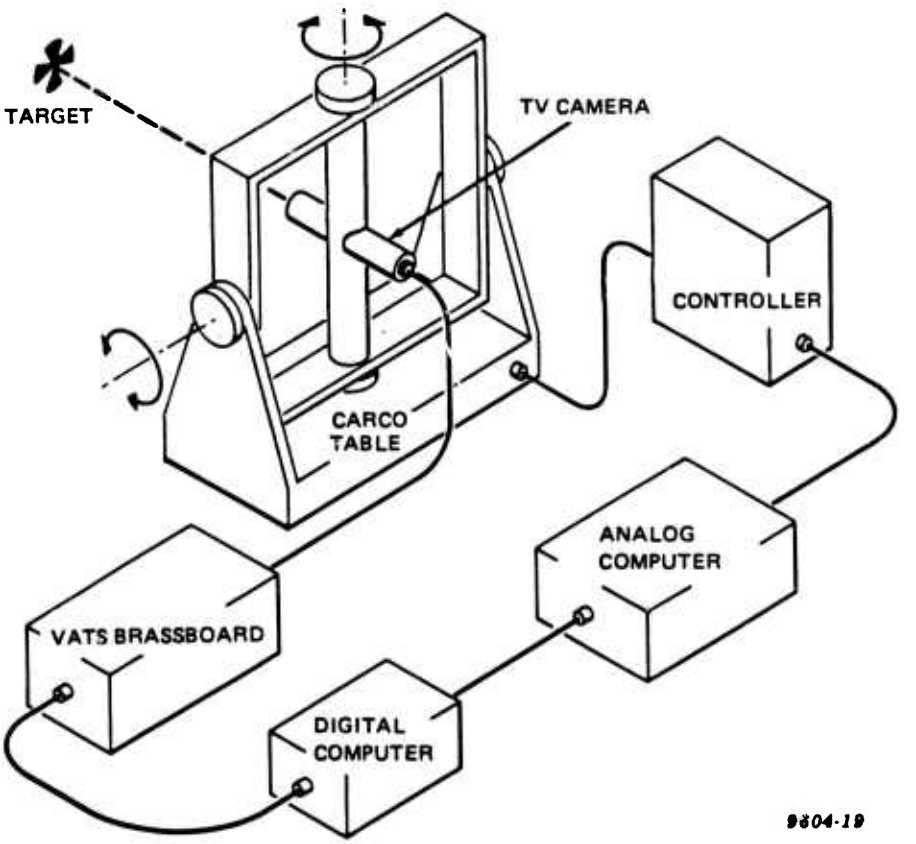


FIGURE 2.2-25. HIL SIMULATION

2.2.6.1 HIL Models

The purpose of the HIL simulation is to demonstrate the viability of building a real-time PADS processor. To accomplish this, an actual PADS guidance processor must be built and all the real-time stimuli it requires must be simulated. As such, four models comprise the HIL simulation complement. The include:

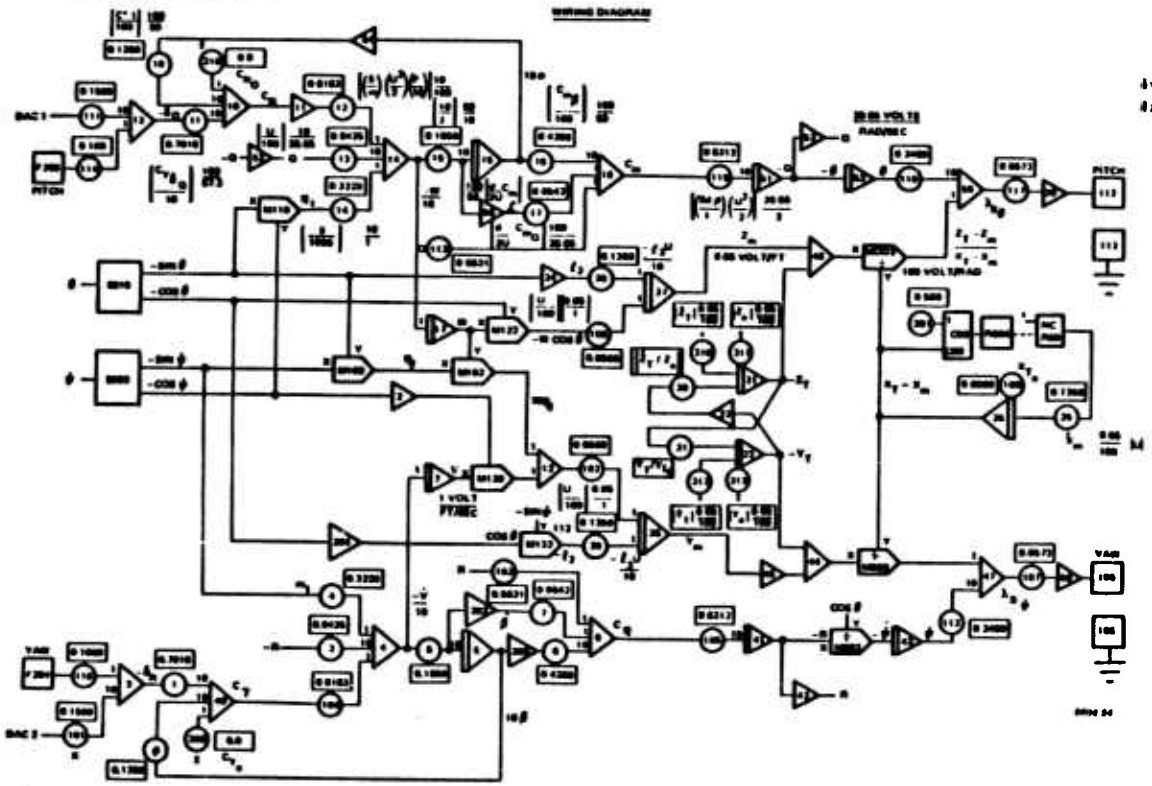
PADS-based Guidance Computer:	68000 Microprocessor
"True" Submissile Airframe Response:	COMCOR Analog Computer
FPA Seeker/NATS Processor:	TV Camera plus VATS Tracker
Target:	COMCOR Analog Computer and Paper Target

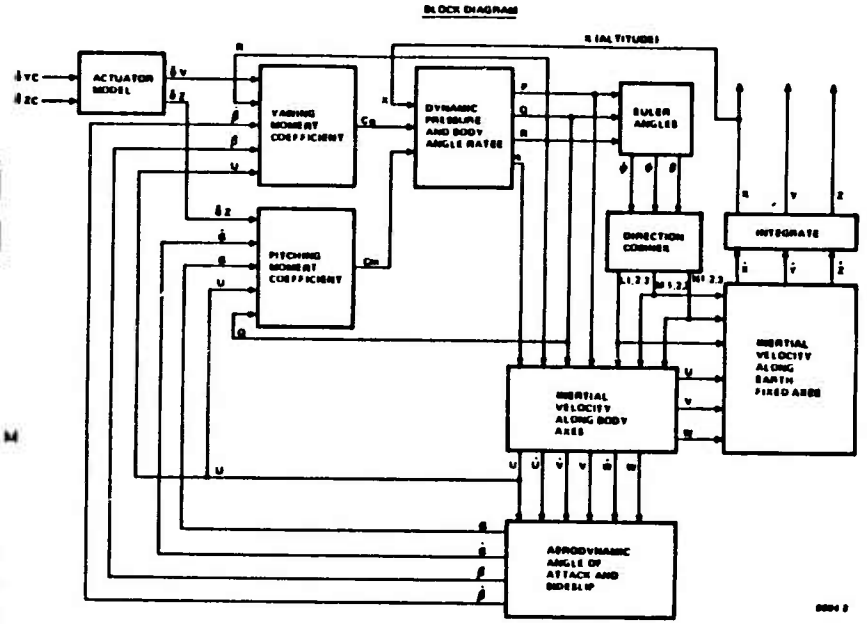
No wind effects or actuator effects were modeled and system performance statistics (miss distance, etc.) was derived manually from strip chart records. A short description of each model is also available at the end of the following section.

2.2.6.1.a Real-Time PADS

A guidance and control computer (PADS) real-time code has been developed in PASCAL and demonstrated on the target Motorola 68000 processor. The code is based on the time based simulation developed FORTRAN emulation and required major scaling type adjustments to optimize it for real-time operation in the 16-bit fixed point arithmetic computing system.

To validate and optimize the real-time code, the real-time analog computer model was first benchmarked against the FORTRAN model and a theoretically derived closed-form aerodynamic math model. The real-time PADS system was then developed against the "Actual Airframe" (the analog model with its generic imperfections). The baseline PADS was finally benchmarked (open loop) against the analog.



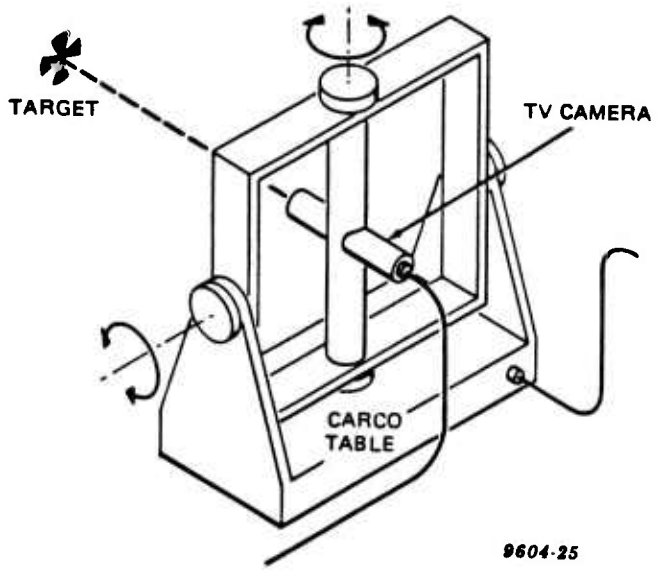


DESCRIPTION

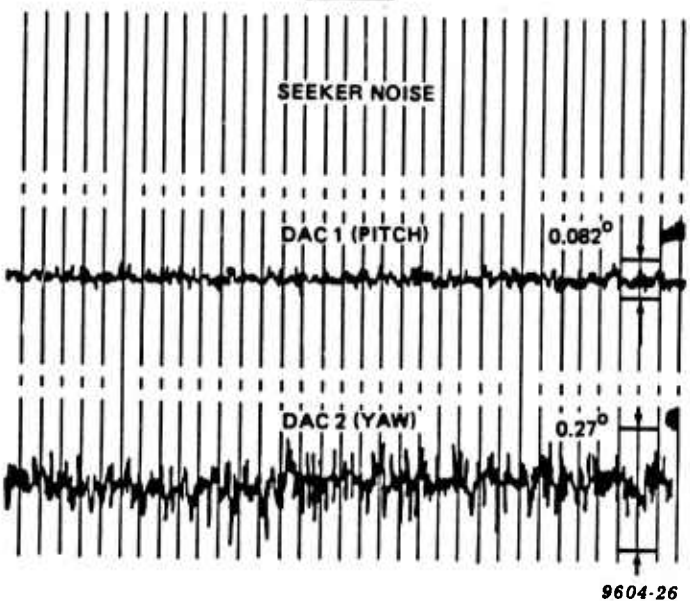
THE MODEL SIMULATES A SUBMISSILE FALLING UNDER THE FORCE OF GRAVITY TAKING INTO ACCOUNT THE AERODYNAMIC FORCES, MOMENTS AND WING DEFLECTIONS THE MODEL KEEPS TRACK OF THE SUBMISSILE'S "TRUE" POSITION, VELOCITY, ACCELERATION, ATTITUDE ANGLES, BODY RATES, AND WING DEFLECTIONS, WHICH ARE USED TO CLOSE THE SIMULATION LOOP AND LATER SERVES AS A COMPARISON AGAINST THE COMPUTED VALUES BY THE PADS ONBOARD COMPUTER.

MODEL: SEEKER

SKETCH



OUTPUT

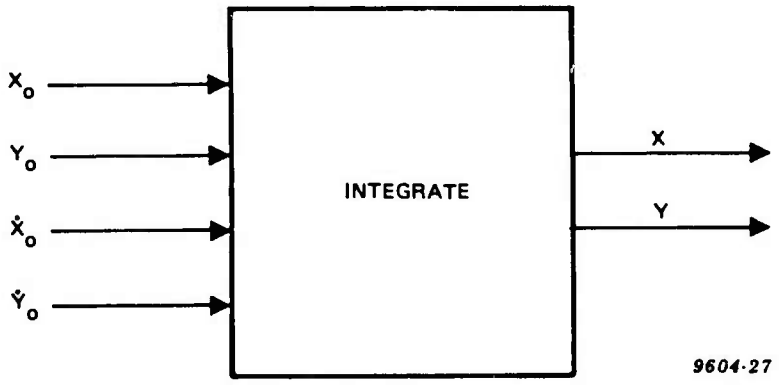


DESCRIPTION

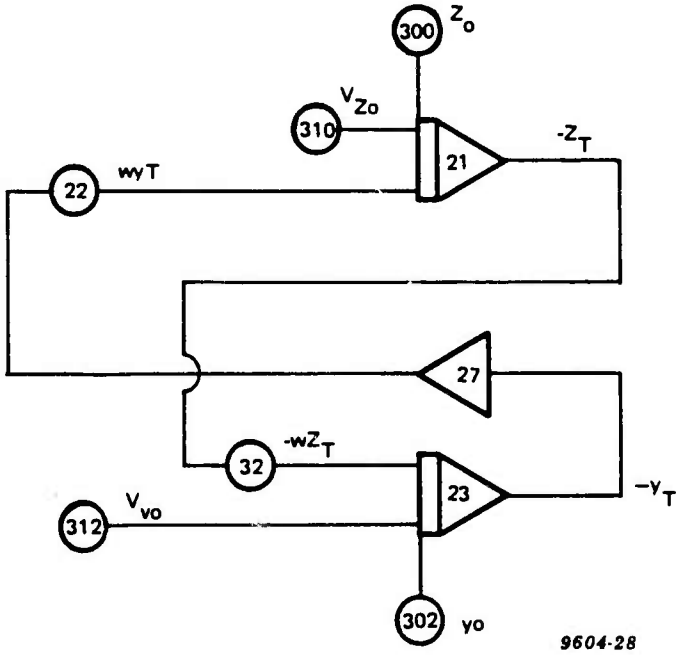
THE ASSP SEEKER IS SIMULATED IN THE HIL TEST CONFIGURATION BY A TV CAMERA CONNECTED TO A "VATS" DIGITAL TRACKER, MOUNTED ON A THREE-AXIS RATE TABLE (CARCO TABLE). A STRIP CHART RECORDING SHOWS THE DAC OUTPUTS WITH THE SYSTEM IN A STEADY-STATE TRACK MODE. IN THIS CASE, A SIMPLE UNITY GAIN ALGORITHM WAS USED; THEREFORE, THE TRACES SHOW THE SEEKER STEADY-STATE NOISE OUTPUT. NOTE THAT THE DAC 2 (YAW CHANNEL) IS ABOUT 3 TIMES WORSE THAN THE PITCH CHANNEL. SEEKER NOISE WAS MEASURED PEAK TO PEAK GIVING THE FOLLOWING VALUES: (A) 0.82° PITCH (PEAK TO PEAK) (B) 0.270° YAW (PEAK TO PEAK).

MODEL: TARGET

BLOCK DIAGRAM



WIRING DIAGRAM



DESCRIPTION

A PAPER TARGET IS MOUNTED ON THE FAR WALL OF THE LAB. APPARENT TARGET MOTION IS ACHIEVED BY COMMANDING RATES TO THE RATE TABLE WHICH ARE NOT "KNOWN" TO THE MISSILE AIR FRAME OR GUIDANCE. THE TARGET MODEL IS ABLE TO SIMULATE A TARGET OFFSET, LINEAR VELOCITY AND CIRCULAR MOTION. THE SEEKER WILL SENSE "TARGET MOTION" WHEN THE ANALOG COMPUTER SENDS A SIGNAL TO THE CARCO TABLE SINCE THE SEEKER IS ON THE CARCO TABLE. THIS MEANS THAT THE TARGET STAYS STATIONARY IN THE LAB AT ALL TIMES.

A comparison between the IBM "True Airframe," the theoretical result and the analog model is shown for the most critical parameters and benchmarked in Figures 2.2-26 - 2.2-28. The analog compares reasonably well against both prior results and, therefore, agreement is validated. The analog airframe, with its observable (and hidden) anomalies is now considered the "Actual Airframe" in the same way an actual submissile will be built and thoroughly tested. Therefore, the real-time PADS code is now adjusted to optimize agreement with the "Actual Airframe" rather than previous "Models."

2.2.7 HIL Demonstration

A successful transfer between the IBM simulation and the HIL simulation was achieved. In order to validate the HIL simulation, a set of benchmark runs were made similar to those in Section 2.2.4. The benchmark runs contain three different engagement scenarios and they are as follows:

1) Target Offset
2) Target Constant Speed
3) Target Maneuver

The above scenarios are typical "bad but not worst case" engagements which contain some or a combination of the following errors:

1) Seeker Noise

   a) ± .041° Pitch (Please see seeker model description for
   b) ± .135° Yaw discussion of non-symmetrical seeker noise characteristics)

2) Airframe Mismatch

   Generic Analog/Digital Mismatch

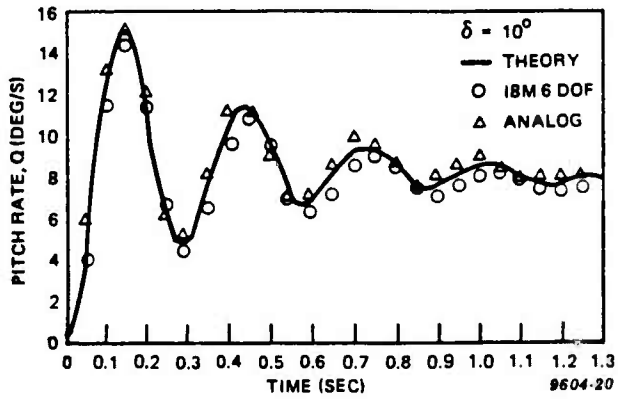


FIGURE 2.2-26. MODEL VERIFICATION

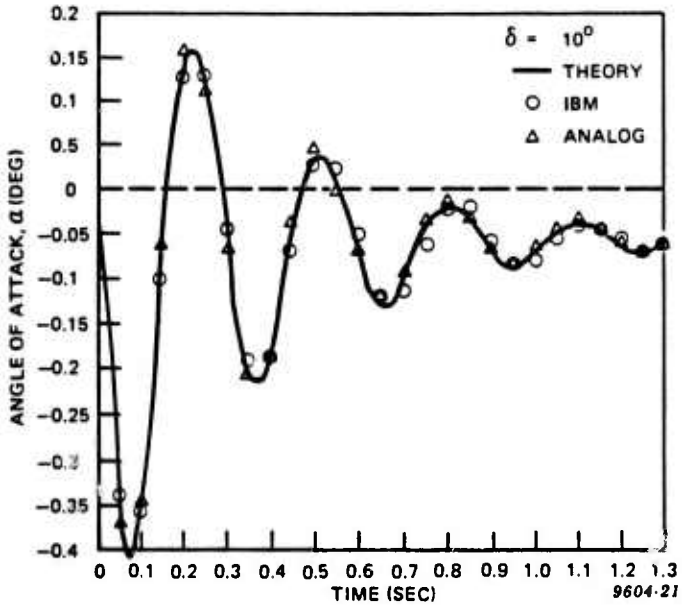


FIGURE 2.2-27. MODEL VERIFICATION

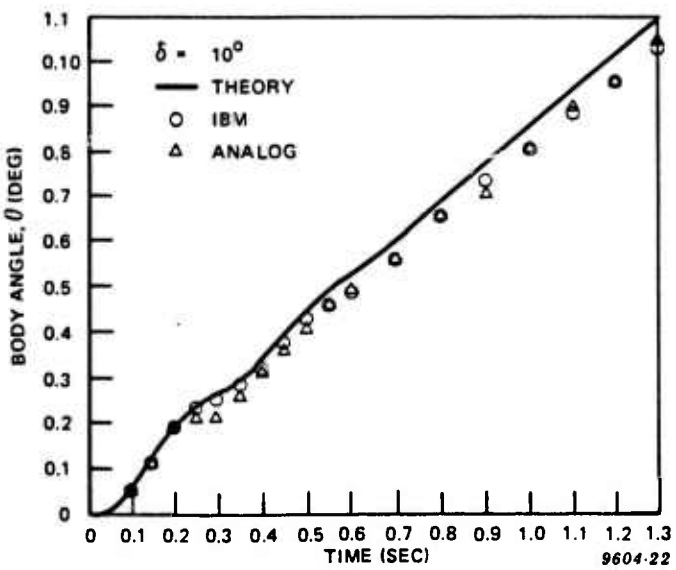


FIGURE 2.2-28. MODEL VERIFICATION

3) Velocity Mismatch (5%)

4) Air Density Mismatch (-2.5%)

Seeker noise and airframe mismatch are part of the system and will not change (unless the hardware changes); and, therefore, every run will have the above two errors. Velocity mismatch and air density mismatch were introduced by changing the necessary parameters in the analog computer to produce the desired mismatch with PADS (Motorola 68000 Processor). A 5% velicity mismatch and a -2.5% air density mismatch constituted the nominal error.

Results from a typical benchmark run are shown in Figure 2.2-29, a-d. The figures show the time histories of several critical simulation parameters from an engagement where the nominal set of submissile errors (mismatch, etc.) as well as a 20 foot per second (fps) target motion and a 15 foot initial delivery error were present. Figures 2.2-29a and 2.2-29b show the attitude, pitch rate and attack angles of the "true" airframe as compared to that which the onboard PADS has computed. Good, but clearly not perfect, agreement in these parameters (which are indeed computed in real time) lead to a stable and accurately controlled submissile which hits well within 3 feet of the aimpoint (Figure 2.2-29a). (This result adequately validates the HIL simulation as being reasonable and PADS as achievable.)

2.2.8 Sensitivity Studies

A nominal set of typical (1σ) was determined as the basis for a series of parametric error sensitivity studies as listed below:

1) Seeker Noise (20 Hz Sample Rate)
   a) Pitch ± .041 deg
   b) Yaw ± .135 deg

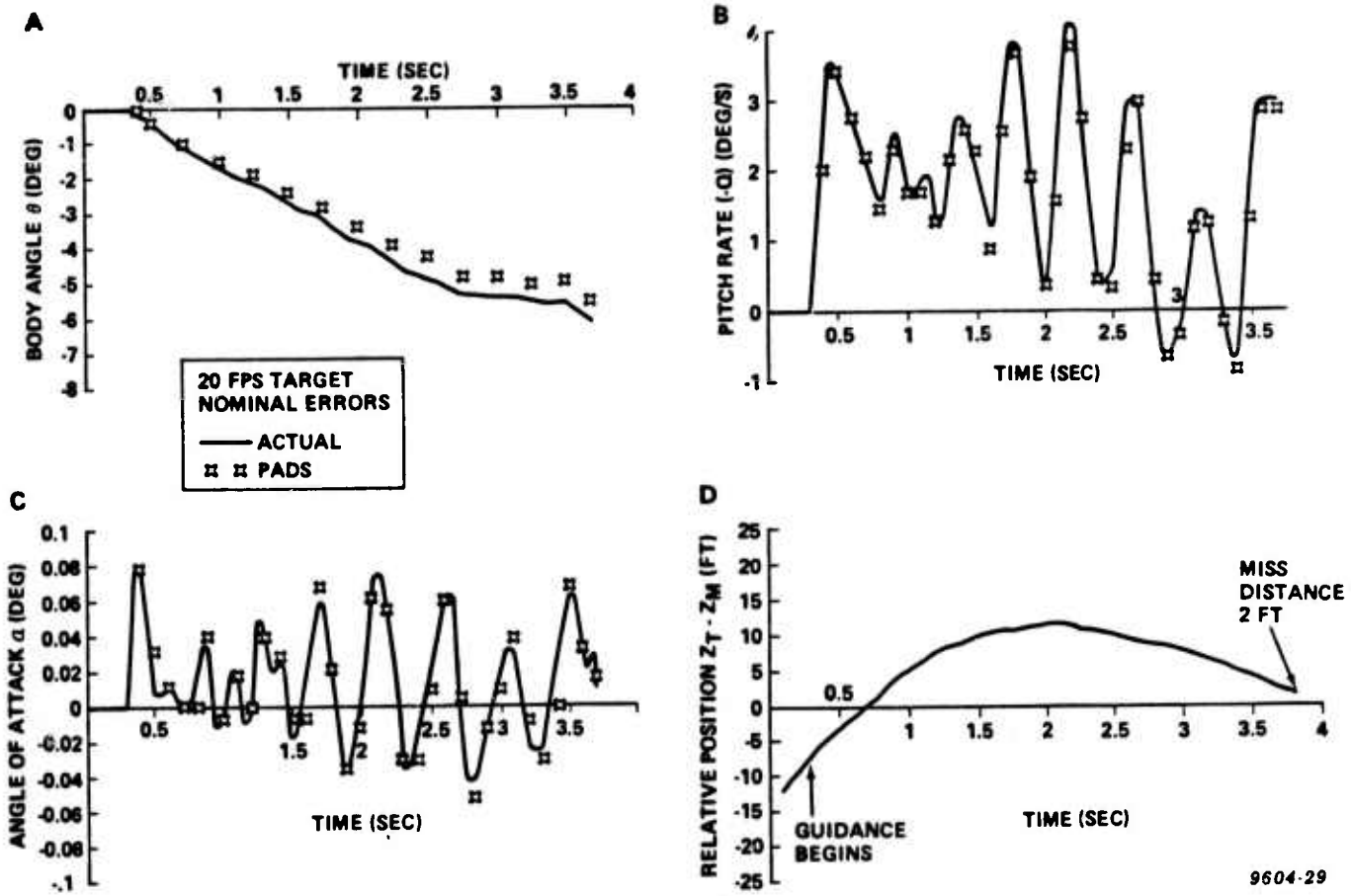


FIGURE 2.2-29. TYPICAL PADS HIL ENGAGEMENT

2) Airframe Mismatch (Generic)
   a) Imperfections of analog computer (i.e., amplifier drift, scaling, etc.)
   b) Limited PADS update rate and roundoff due to microprocessor throughput and fixed point arithmetic limitations

3) Velocity Mismatch (5%)
   a) PADS expects a constant velocity of 270 ft/s. Submissile is actually at a constant velocity of 283.5 ft/s.

4) Air Density Mismatch (2.5%)
   a) PADS expects a constant air density of .002342 Slugs/ft^3 true constant air density is .002283 Slugs/ft^3

5) Target Maneuver (Circle)
   a) Radius 80 ft
   b) Velocity 20 ft/s

The statistics of one parameter was varied while holding the statistics of each other error source constant at the nominal (e.g., 1σ) value indicated. This technique was previously used with good success in the digital 6 DOF efforts described in Section 2.2.5. Velocity mismatch, air density mismatch and target maneuver were selected as the sensitivity parameters based on previous testing and ability to mechanize such variations in the HIL simulation.

Figure 2.2-30, a-c, shows that system accuracy is consistently better than the required 1 meter misdistance for well greater than the indicated normal 1σ errors. Also shown are the previous results from the digital 6 DOF studies. The surprisingly high degree of commonality between the two greatly enhances the confidence in each simulation independently and strongly supports the

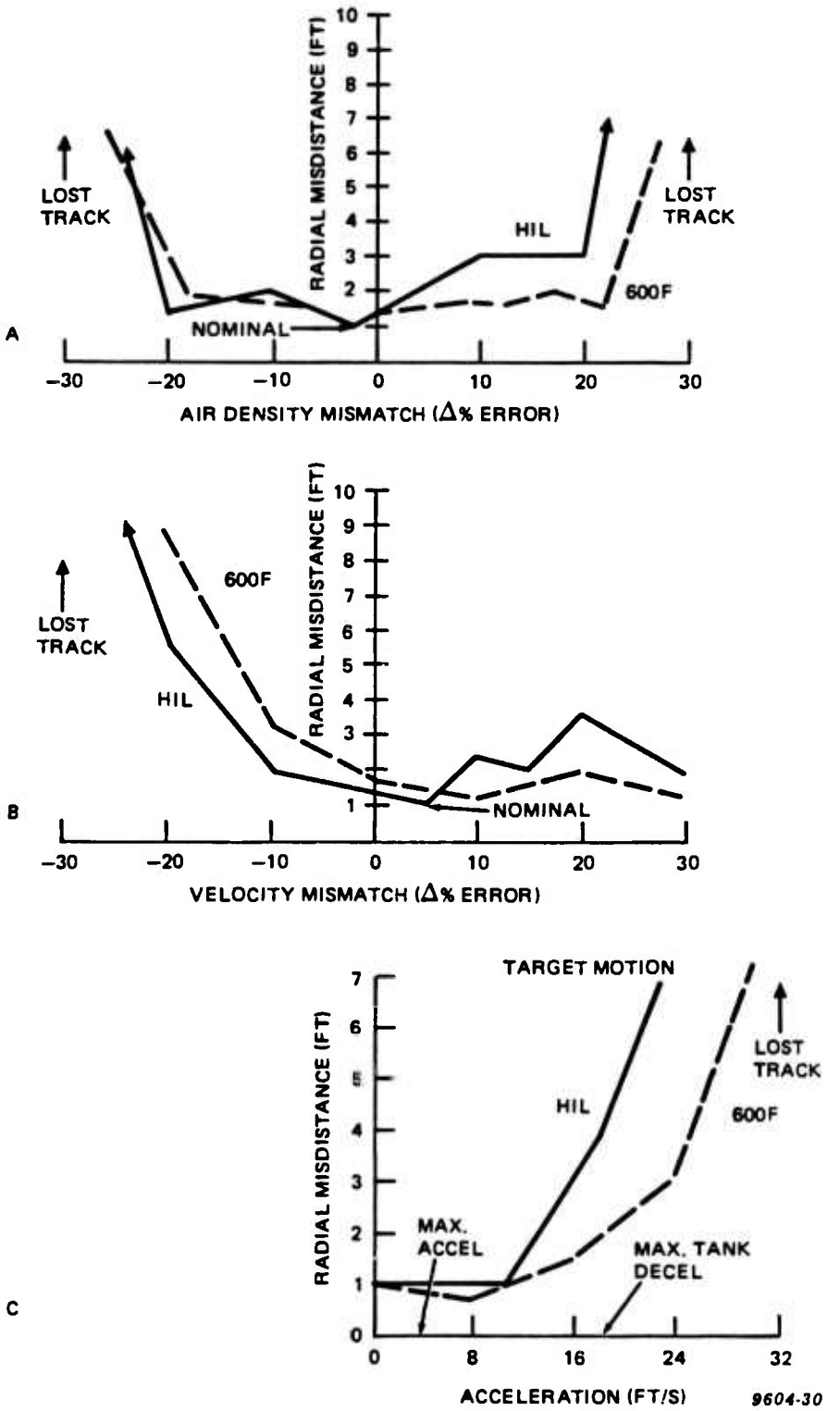


FIGURE 2.2-30. HIL SENSITIVITY STUDY RESULTS

analytical conclusions that for this class of mission, a PADS-based strapdown seeker gyroless proportional law can be implemented in available real time componentry and can provide lethally accurate, rugged, low-cost guidance.

2.2.9 PADS Studies Conclusions

The Proportional Airborne Digital Simulator (PADS) approach to strapdown seeker gyroless proportional navigation guidance is a new innovation made possible by the advent of focal plane array seekers and advances in microelectronic computer technology. The guidance approach has been analytically demonstrated to be feasible for a substantial class of "restricted" mission applications by means of a four-step validation analysis on one such selected application. The steps taken were classical linear stability analysis, realizability tolerance effects, non-linear, time based, six-degree-of-freedom simulation and system accuracy sensitivity to non-ideal conditions.

A real-time PADS Processor was developed and an HIL simulation was created in order to demonstrate PADS feasibility under real-time constraints. The HIL sensitivity studies showed a surprisingly high degree of commonality with the digital 6 DOF results, thereby enhancing the confidence in each simulation independently.

The sensitivity studies and extensions thereto clearly demonstrated that system performance can be consistently achieved under extended, non-nominal, and even some very difficult conditions. This general strength of the PADS approach shows it validated for many different mission applications wherein:

- Approximate initial conditions are well-known.
- Weapon velocity profile is reasonably predictable.
- Maneuvers required are not dramatic.
- Nominal weapon aerodynamics are known and relatively insensitive to manufacturing tolerances.
- Engagement times are short.

It is recommended that the potential weapon system cost savings complexity reduction increases in reliability and ruggedness clearly warrants further study of this approach. Such study must include fabrication of hardware and actual guided flight experiments as analyses and simulation can never fully model real-world conditions.

2.3 Test

Northrop completed a survey of existing image data bases of vehicular targets at the start of the Advanced Sensor Signal Processor contract effort. We had a set of autonomous detection algorithms based on visual spectrum data. We needed infrared based data to convert these algorithms to the infrared region. The Northrop survey showed a marked lack of image quality data especially for the top-down scenario. It was therefore apparent that these data would have to be obtained by test methods.

Northrop elected to build, as a capital asset, a modular set of data collection equipment to obtain suitable imagery to enable algorithm conversion. We built this equipment, qualified it as airworthy and provided the equipment in support of the contract data collection effort. The equipment and resulting test flights are described herein. The equipment has been built for installation into both commercial and military helicopter aircraft and provides 8-12 micron infrared data, visual television ground truth data, analog and digital recording of the data and IRIG timing for data reduction.

2.3.1 Data Collection Equipment Development

An airborne infrared data collection system was required to provide a suitable top down infrared signature data base to allow algorithm training. Northrop integrated and checked out an airborne infrared data collection system that successfully met these objectives. This system was certified as safe and an airworthiness report was submitted to the U. S. Army Aviation Research and Development Command (AVRADCOM). A flight release was obtained from AVRADCOM prior to any flight testing.

The data collection system requirements have been satisfied through the use of Northrop's common module FLIR and support equipment compatible with a Bell 205, 212 or UH-1H type helicopter (See figure 2.3.1-1). The system has been designed for maximum utility and includes the following features:

- Infrared sensor (8-12 micron), stabilized
- Closed circuit television camera (ground truth)
- Oscilloscope for adjustment of sensor video levels
- Airborne recorders compatible with standard lab recorders (FLIR recorder-improved bandwidth)
- Cockpit FLIR 875 line monitor for efficient pilot position of helicopter, day or night
- Compatible with military helicopters, UH-1H and UH-1N
- Compatible with commercial helicopters Bell 205 and Bell 212
- No helicopter airframe modifications required
- Multi-mission capability - top down and forward looking and compatible with four (4) rotary aircraft types
- Remote sensor positioning and adjustment
- Simultaneous recording/playback of infrared and visual data
- Day/night data collection capability
- Aircraft intercom communication and recording of intercom communication
- Uses standard UH-1H cabin seating
- Uses UH-1H standard external stores mount for infrared sensor mounting
- Equipment rack installation duplicates MICOM flight qualified UH-1H installation
- Capable of rapid installation/removal from aircraft.
- Capable of operating entirely from onboard aircraft electrical power (28 VDC).
- Northrop system supplies 115 VAC @ 1 Ø, and 115 VAC @ 400HZ @ 3 Ø electrical power
- System electrical load is within MICOM specification
- System weight and centers of gravity are within MICOM specification

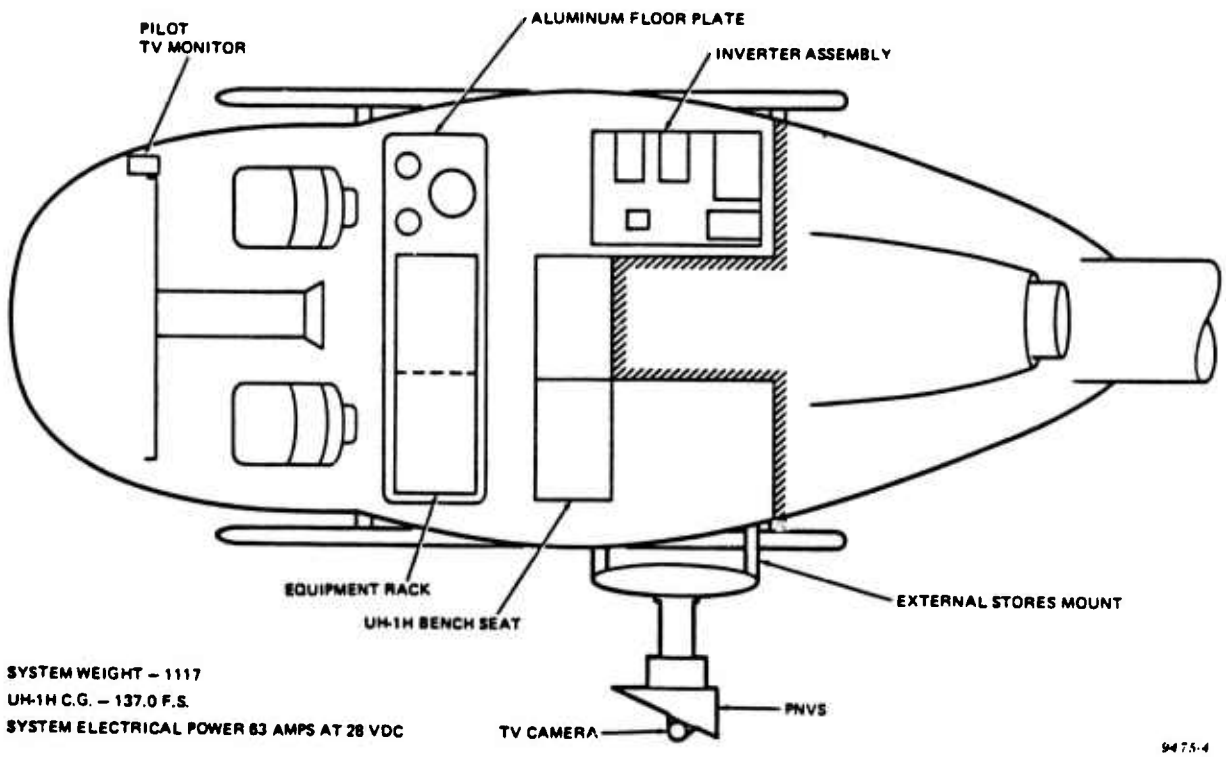


Figure 2.3.1-1 Data Collection System Installed on Standard UH-1H Helicopter

Extensive work was performed in preparation of the FLIR system and data collection support equipment for the data collection testing.

Special modifications of the FLIR system were performed to assure suitable recorded infrared data; these included removal of all symbology, addition of a manual FLIR gain control, and addition of external horizontal and vertical blanking circuits. Chow circuit boards have been obtained from NVEOL (through MICOM) and installed in the FLIR. These circuits eliminated DC restoration effects that may be encountered in certain infrared scenes. Special FLIR cables and a FLIR control box (see Figure 2.3.1-2) were designed and fabricated to allow field operation independent of lab equipment. The FLIR was installed in a Northrop designed and fabricated fixture. This fixture was machined from a one-piece block of aluminum. The FLIR (installed in the mounting fixture) was subjected to lab vibration testing prior to flight test to insure structural integrity and resonance location.

The FLIR and FLIR fixtures were attached to a UH-1H helicopter standard external stores mount (see Figure 2.3.1-3). Two of these standard mounts (left and right) were obtained from the California Air National Guard (through MICOM). These mounts were disassembled, stripped of paint, dye penetrant inspected, repainted and reassembled to assure structural integrity.

A closed-circuit televison camera was installed onto the FLIR cover plate to obtain the benefit of the FLIR stabilization. One of the two airborne video records was specially modified to obtain one megahertz higher bandwidth video records. This insured that the required infrared video recording of minimum resolvable temperature (MRT) and the modulation transfer function (MTF) was obtained during data collection tests. A 4-inch monitor compatible with the 875 line rate FLIR was located and checked out. This allowed the helicopter pilot to efficiently position the FLIR field of view over the target arrays during test flights.

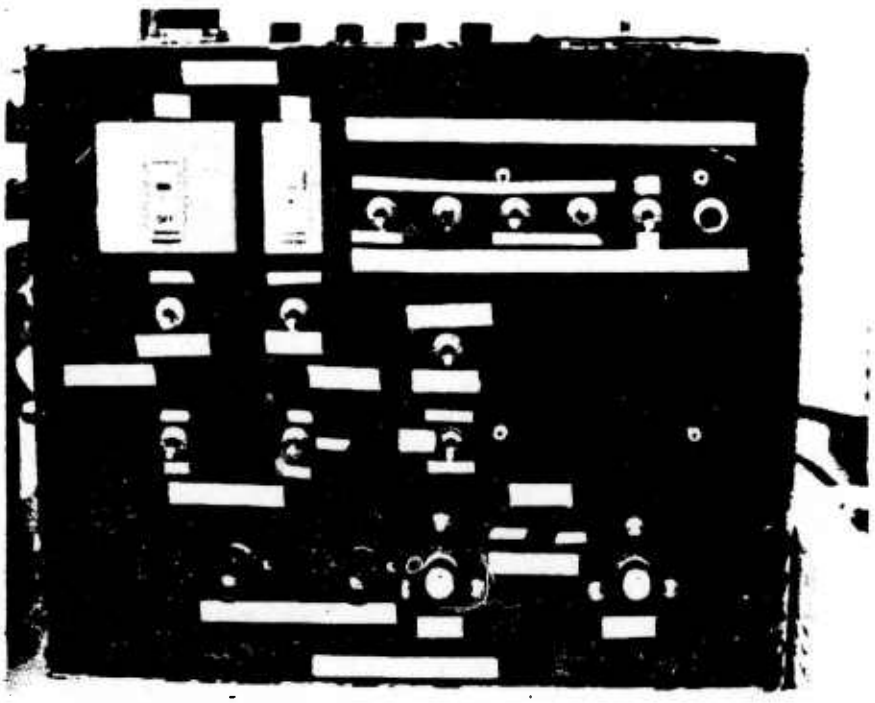


Figure 2.3.1-2 FLIR Control Box

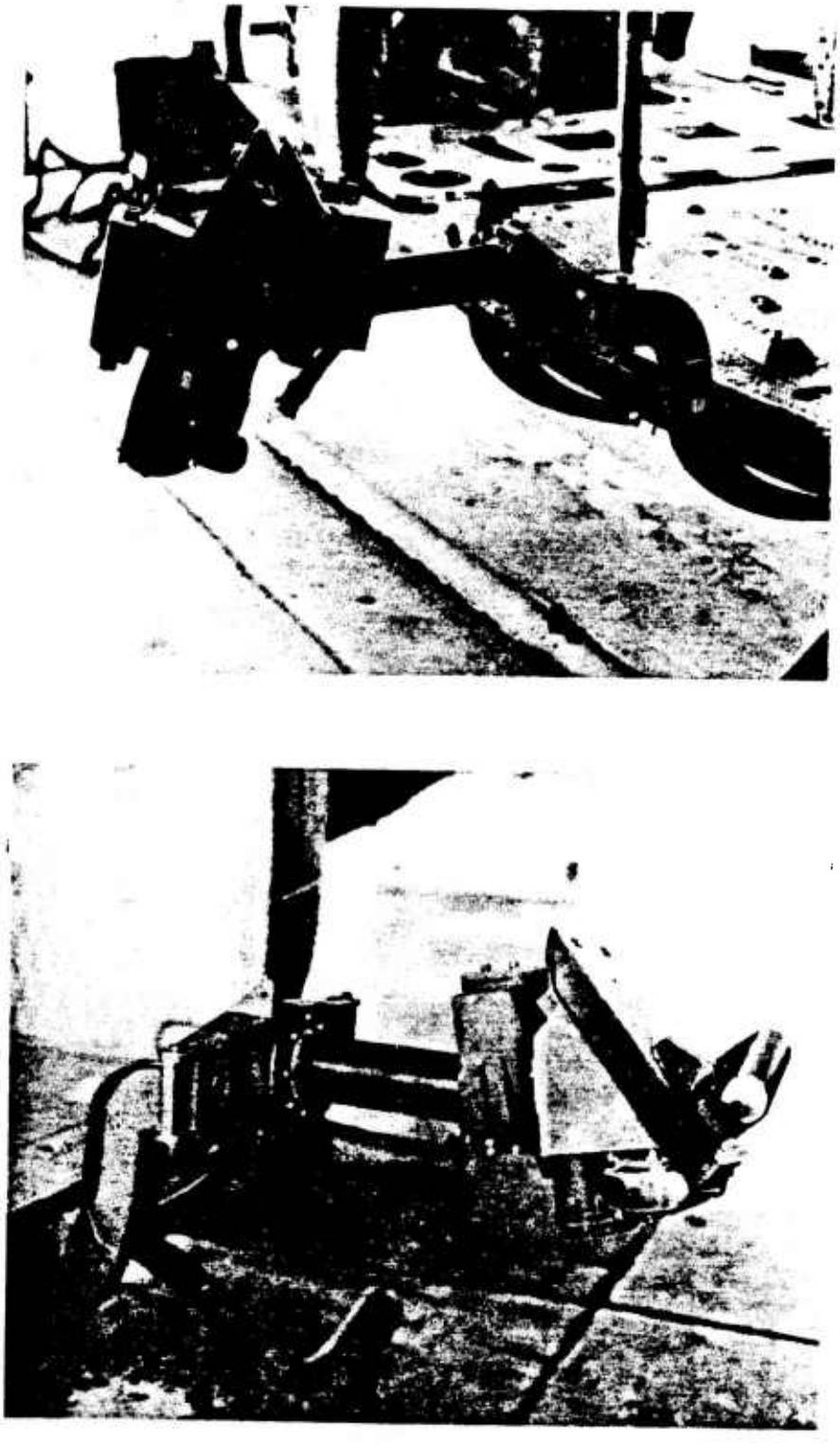


Figure 2.3.1-3 UH-1H Helicopter Externally Mounted Equipment

All video monitoring recording equipment was rack mounted (see Figures 2.3.1-4 and 2.3.1-5). This rack installation duplicated a previously qualified MICOM installation. This assured flight safety and aided in obtaining timely airworthiness approval from the Aviation Research and Development Command. The rack-mounted equipment effort included coordination with MICOM test personnel on installation design, obtaining two 19-inch racks and rack shelves, fabrication of an aluminum rack floor plate and aluminum angle, obtaining UH-1H cargo tiedowns, and shock mounting all equipment in the rack shelves.

The data collection system receives all required electrical power directly from the aircraft (28 vdc) and inverters that supply the 115 VAC @ 60HZ, @ 1 Ø, and 115 VAC @ 400HZ @ 3 Ø. All inverters were procured and mounted on a separate aluminum plate. This aluminum inverter plate was located in the UH-1H right aft cabin bay (Figure 2.3.1-6). The entire data collection system was operationally tested through the inverter assembly using simulated aircraft power.

All mechanical installations were designed to be compatible with a California Air National Guard UH-1H helicopter, a commercial 212 helicopter, and measurements taken from the MICOM UH-1H helicopters.

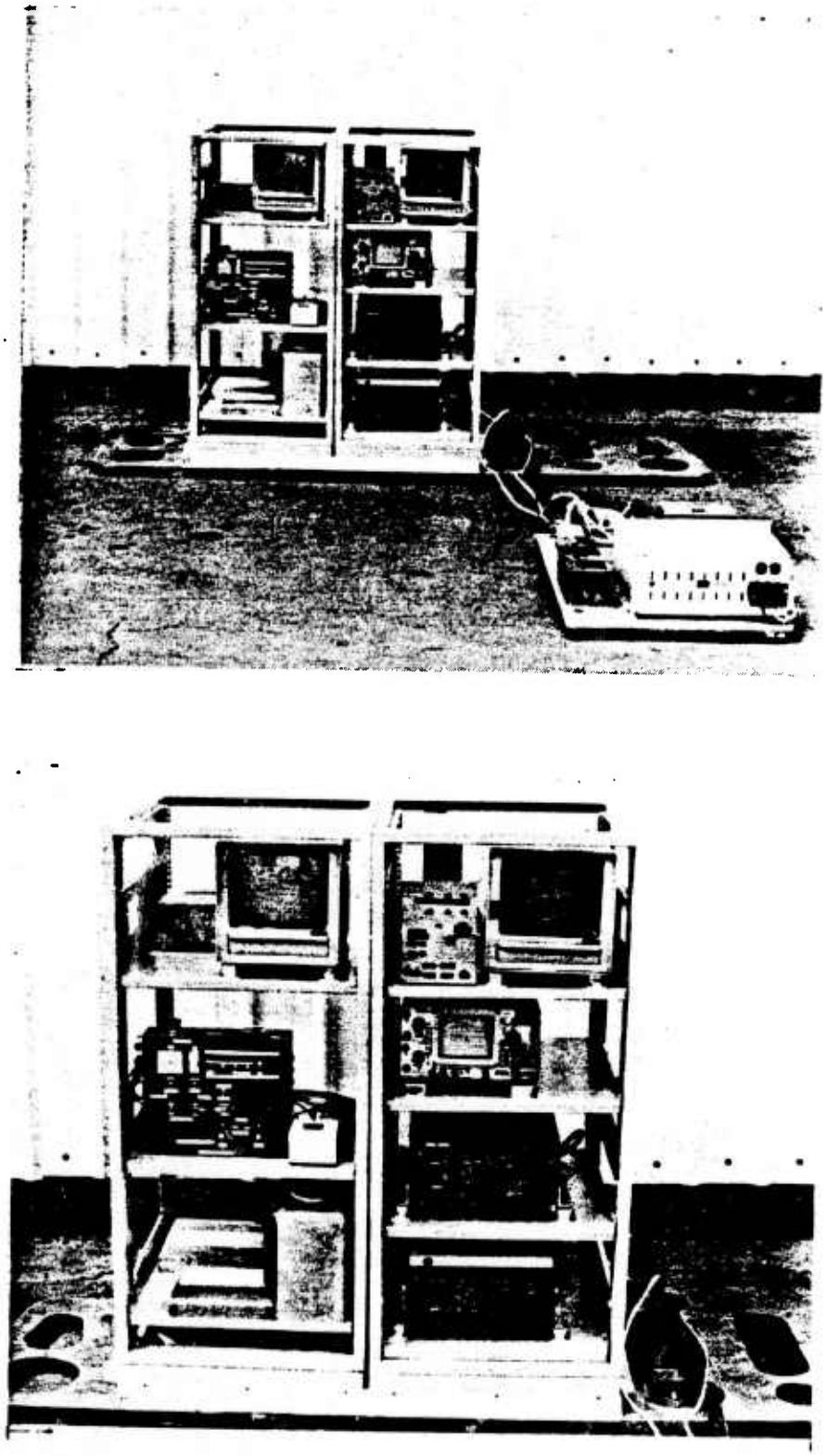


Figure 2.3.1-4 Data Collection System Rack Mounted Equipment

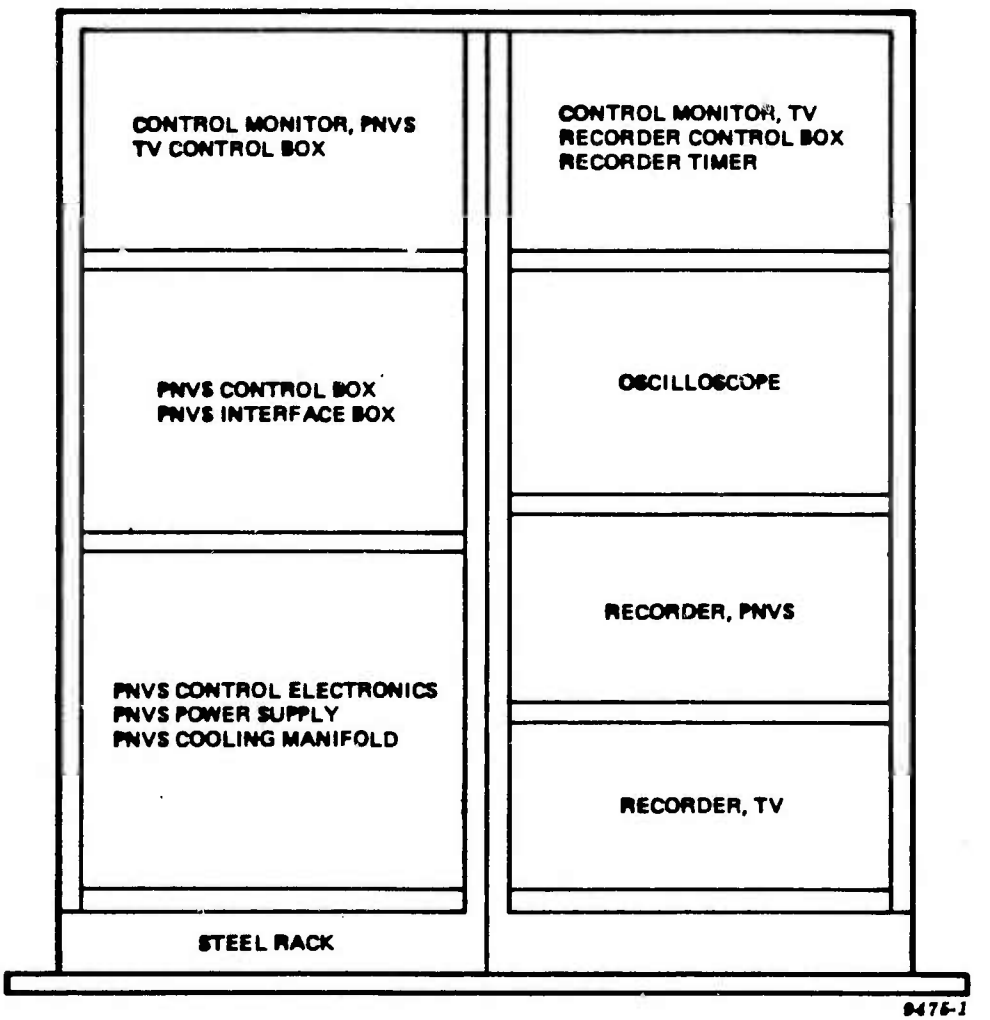


Figure 2.3.1-5 Data Collection Rack Equipment

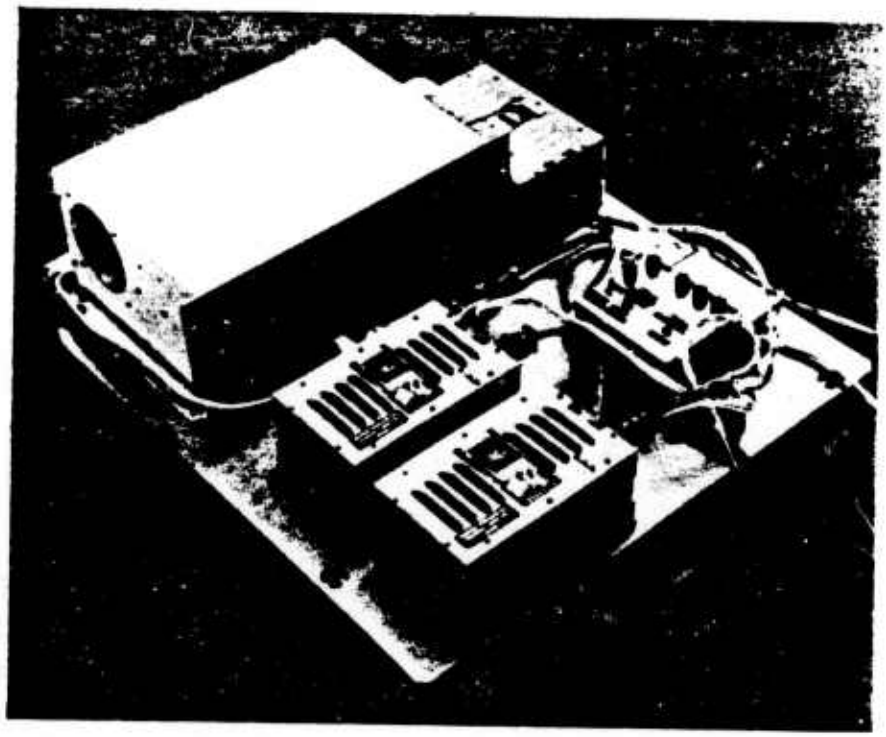


Figure 2.3.1-6 Data Collection System Electrical Power Inverter Installation

2.3.2 Data Collection Test

A data collection test was required to collect the infrared signature data base to be used for algorithm training. The data collection system described in Section 2.3.1 was installed on a commercial Bell 212 helicopter (Figures 2.3.2-1, 2.3.2-2, 2.3.2-3 and 2.3.2-4) and used to conduct airworthiness tests and preliminary data collection tests prior to testing at MICOM. Data was obtained by flying over tanks, trucks, other vehicles and background located at USMC, Camp Pendleton. The system performed to specification and had no adverse effect on the operation of the helicopter. Personnel from MICOM's Test Group visited Northrop on 5 March 1982 to review the system installation on the helicopter. The tests were conducted from Northrop 3 March through 5 March 1982.

Minor changes were made to the installation of the ground truth camera and intercom system to improve system operation and camera field of view. The system was packed and shipped to MICOM for the contract data collection tests on 19 March 1982.

The Data Collection Test activity required by the contract was completed successfully from 22 March through 6 April 1982. These tests were controlled by a Data Collection Test Plan, Northrop Document No. 342402. The equipment was certified as safe and an airworthiness report was submitted and approved prior to any data collection activities. (Northrop Document No. 342401)

Figure 2.3.2-1 Bell 212 Helicopter Data Collection System Installed at Northrop

Figure 2.3.2-2 Bell 212 Helicopter with Data Collection System Installed

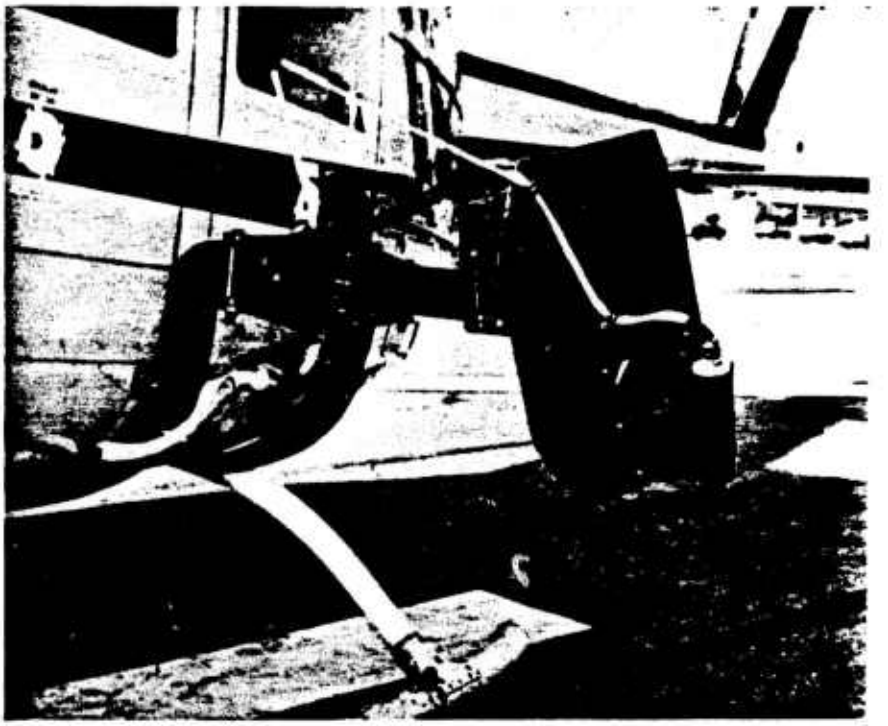


Figure 2.3.2-3 Close-up of Bell 212 Installed FLIR, Television Camera, FLIR mount

Figure 2.3.2-4 Bell 212 Helicopter Cabin Installed Data Collection System Rack and Electrical Power Inverter Assembly

2.3.2.1 Test Setup

Northrop's FLIR system, closed circuit television system, and monitoring/recording equipment were installed aboard MICOM helicopter UH-1H-22361. (See figures 2.3.2-5 and 2.3.2-6). Some additional MICOM test equipment was installed in the UH-1H helicopter to support the data collection tests. A 70 mm camera was mounted external to the helicopter on a left forward standard UH-1H external stores mount. Early in the testing, closed circuit television video was supplied to the pilot's panel mounted monitor. The operator control system located in the aft cabin had intercom communication with the pilot. System operators had direct communication with the ground test director via FM radio. All voice communication was recorded on the airborne video recorder's audio track.

IRIG timing signals were supplied to each recorder audio track simultaneously during test via the Datun Model 9310. A mini-ranger system wasset up to transmit continuous helicopter altitude information back to Test Area -1 Control. The complete data collection system was installed on a UH-1H helicopter and inspected by the MICOM Helicopter Group specialists. The helicopter was then weighed and centers of gravity determined. On 26 March 1982, a successful airworthiness flight test was conducted prior to any data collection flight testing. All modes of flight were verified including several autorotations. Following the completion of this airworthiness test, data collection flight testing commenced.

2.3.2.2 Test Area Description

The data collection individual testing time requirements were coordinated with both the Helicopter and Test Area Groups. One Northrop team member was assigned to the test area group to layout and direct the

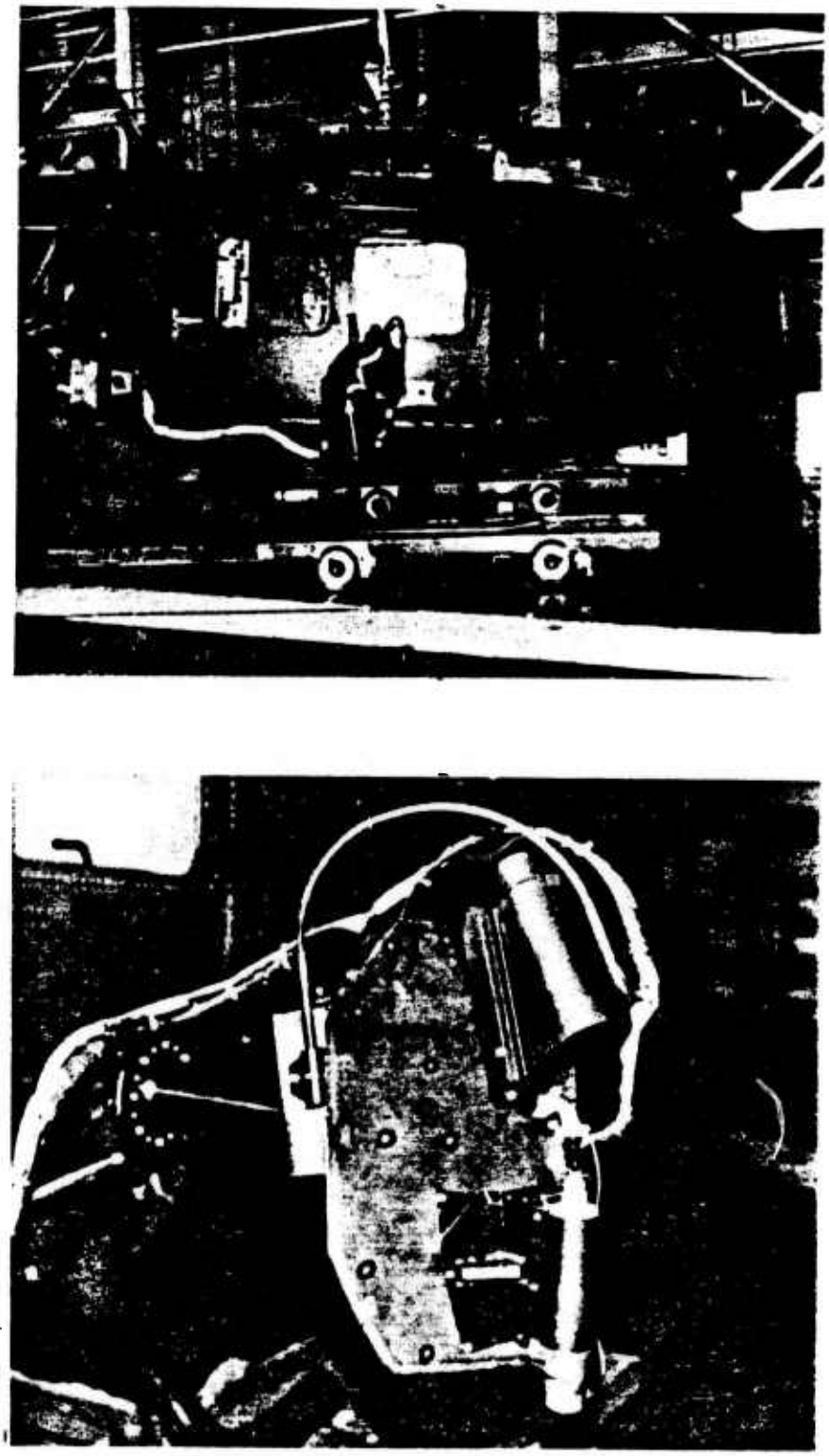


Figure 2.3.2-5 FLIR/TV Installed on a Micom UH-1H Helicopter

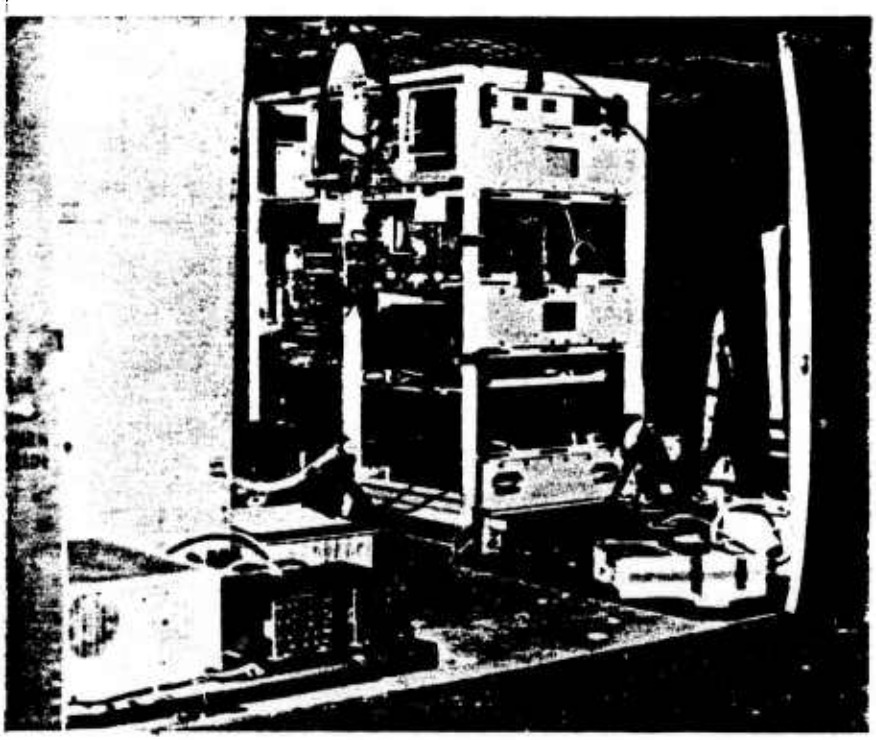


Figure 2.3.2-6 Data Collection System Equipment Installation on a Micom UH-1H Helicopter

use of the test area, test area military vehicles and to collect ground data. Two Northrop team members were assigned to the helicopter to collect in-flight data. A MICOM helicopter crew chief was aboard during each test flight.

A course was laid out on Test Area -1 for the tanks to drive on. This course maximized use of the helicopter and test area time and includes asphalt roads, dirt roads, gravel roads, open area, heavy forest, buildings and swamp area.

2.3.2.3 Test Procedure

A typical individual data collection test consisted of the following sequence:

1. Coordinate testing with the Helicopter Group and Test Area Group.
2. Dispatch team member to coordinate test area requirements with the test area group.
3. Discuss test with pilot(s).
4. Helicopter lift-off, system powered on, departure from high bay and establish communication with Test Area -1 Control and Redstone Airport tower.
5. Landing on Dodd Road ard calibration/checkout of Mini-Ranger System by Test Area Group personnel.
6. Proceed to test site.
7. Start up tanks and traverse the established course.
8. Following completion of the course, the tanks were turned off to allow cooling conditions.
9. The UH-1H helicopter pilot(s)maintained the desired targets in the sensor field of view during all hover and forward travel maneuvers at the pre-selected altitudes.

2.3.2.4 Test Data

Several types and quantities of data were collected. These are summarized in Table 2.3.2-1. After each test sequence, meetings were conducted to review collected data and to make decisions on upcoming testing to assure quality data.

Both infrared and closed circuit television video data tapes were generated aboard the UH-1H helicopter during testing. A summary of the video data tapes is provided in Table 2.3.2-2 and includes the test data, test time, tape identification number and comments. Tables 2.3.2-3 and 2.3.2-4 provide examples of details measured on the events that occurred during each test sequence. Each data tape has all helicopter intercom and radio communication on audio track 1 and range IRIG timing signals on audio track 2.

Airborne 70 mm photographs were remotely taken of test scenes of interest with the externally mounted camera. These photographs aided in the identification of terrain imagery.

The Mini-Ranger equipment provided continuous helicopter altitude information in feet during the flight testing. This information has been stored on magnetic tape. Hard copies of these data have been supplied to Northrop by the Test Area -1 Group. Both sets of data include the IRIG time code that are also on the video data tapes. This allows determination of altitude at any particular point on a video data tape.

Weather data was made available to Northrop by the National Weather Service at Huntsville Airport. These data were collected hourly and are highly detailed. Two data sheets per day were generated.

TABLE 2.3.2-1

SUMMARY OF DATA COLLECTION TEST

RAW TEST DATA

Data Type	Quantity
FLIR Video Tapes (20 minute)	33
Closed Circuit Television Video Tapes (20 minute)	33
70mm Airborne Photos (negatives)	100
Mini-ranger Data (tapes)	9
Mini-ranger Data (computer printouts)	many
Weather Data Sheets (two per day)	26
In-flight Log Data Sheets	25
Ground Log Data Sheets	35
IR Radiometer Measurements	many
35mm of Equipment Installation (negatives)	6

TABLE 2.3.2-2

SUMMARY

MICOM DATA COLLECTION TIME/FLIGHT SUMMARY

	Date (day)	Time	Tape Numbers	Remarks
1.	3-26-82 (Fri)	11:00 a.m.	P1, P2, P3	*Preceded by a 20-minute
	Flight 1	12:20 p.m.	T1, T2, T3	flight test of system
	3-29-82 (Mon)			
2.	Flight 1	10:28 a.m.	P4, P5, P6, P7	
		12:27 p.m.	T4, T5, T6, T7	
3.	Flight 2	2:15 p.m.	P8, P9, P10, P11	
		4:06 p.m.	T8, T9, T10, T11	
	3-30-82 (Tues)			
4.	Flight 1	1:45 p.m.	P12, P13, P14	*Tues (a.m.) - provided
		3:21 p.m.	T12, T13, T14	pilot with T.V. video
	3-31-82 (Wed)			
5.	Flight 1	5:10 a.m.	P15, P16	rain
		5:49 a.m.	T15, T16	
6.	Flight 2	9:28 a.m.	P17, P18, P19	*internal IRIG not synced
		10:42 a.m.	T17, T18, T19	w/range IRIG
				*mini-ranger printout data not available this flight (range tied up)
	4-1-82 (Thurs)			
7.	Flight 1	5:01 p.m.	P20, P21, P22	
		6:30 p.m.	T20, T21, T22	
	4-2-82			
8.	Flight 1	5:52 a.m.	P23, P24, P25	rain
		7:20 a.m.	T23, T24, T25	
	4-6-82 (Tues)			
9.	Flight 1	8:15 a.m.	P26, P27, P28, P29	*range IRIG not available
		11:50 a.m. (Approx)	T26, T27, T28, T29	after 10:30
10.	Flight 2	1:01 p.m.	P30, P31, P32	
		2:57 p.m.	T30, T31, T32	range IRIG not available

TOTAL FLIGHT HOURS (from Helicopter Group) = 18.3

NOTE: FM receiver antenna broke causing the loss of range IRIG signal on 4-6-82 a.m. and p.m.

TABLE 2.3.2-3 EXAMPLE OF DATA COLLECTION TEST SEQUENCE

4-1-82 (EARLY EVENING)

RESULTS:	P20, P21, P22 T20, T21, T22 MINI-RANGER TAPE 5
AREAS:	TA-1
ITEMS:	3 TANKS WARMING UP ON DIRT ROAD
	1 TANK RUNNING NEXT TO ROAD
	FORWARD LOOKING PASS AT 400' AGL
	3 TANKS RUNNING ON DIRT ROAD
	3 TANKS MOVING ON DODD ROAD
	3 TANKS CIRCLING BUILDING
	3 TANKS MOVING TO SLED AREA
	3 TANKS MOVING AROUND 2 TREES
	3 TANKS MOVING ON DIRT ROAD
	3 TANKS MOVING IN OPEN, NEAR GRAVEL ROAD
	3 TANKS MOVING AROUND LARGE BUILDING
	TANK BREAKDOWN
	2 TANKS ON DODD ROAD
	FORWARD LOOKING AT 2 TANKS ON DODD ROAD AT 400' AGL (SEVERAL PASSES)
	SUPPRESSED HOT TANK (PLYWOOD OVER RADIATOR)

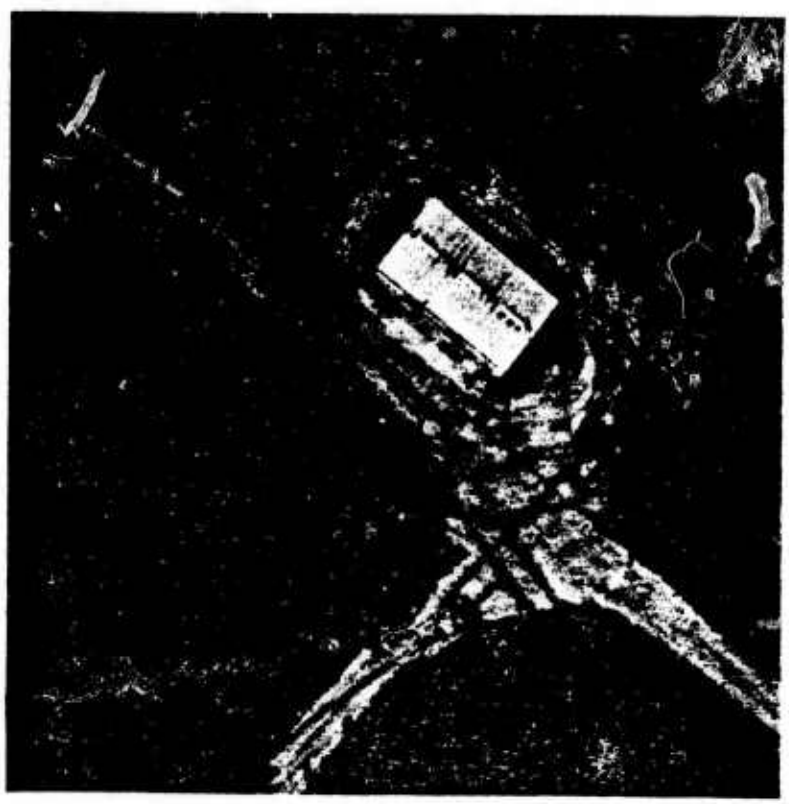


TABLE 2.3.2-4 EXAMPLE OF DATA COLLECTION TEST SEQUENCE

4-8-82 (MORNING)

RESULTS: P26, P27, P28, P29
T26, T27, T28, T29
MINI-RANGER TAPE 7 AND 8

AREAS: TA-1

ITEMS: 1 TANK (WARMING UP) PARKED WITH COLD TANKS
BURNING OIL
COLD TANKS, PARKED IN LINE
FORWARD LOOKING AT 2 HOT, 2 COLD TANKS AT 200' AGL
FORWARD LOOKING AT 3 MOVING TANKS, 2 COLD TANKS AT 100' AGL
3 TANKS MOVING NEAR BURNING OIL
2 TANKS MOVING ON DIRT ROAD
3 TANKS MOVING ON AND OFF DODD ROAD, STAGGERED
3 TANKS CIRCLING BUILDING
2 TANKS MOVING NEXT TO DODD ROAD
3 TANKS MOVING ON SIDE ROAD
3 TANKS MOVING NEAR SMALL BUILDING AND 2 TREES
3 TANKS MOVING THROUGH FIRE BREAK AND FOREST
3 TANKS IN SLED AREA, OUT IN OPEN
3 TANKS MOVING PAST COLD TANK
3 TANKS MOVING ON DODD ROAD
3 TANKS PARKING
FORWARD LOOKING TO REDSTONE AIRPORT
3 COOLING TANKS, TA-1
FORWARD LOOKING AT COOLING DOWN TANKS AT 400' AGL (2 TIMES)
FORWARD LOOKING AT COOLING DOWN TANKS AT 75' AGL (2 TIMES)

Airborne data sheets were generated in the helicopter cabin during testing. These data sheets include the video data tape identification numbers, continuous time of day (range IRIG time), continuous test remarks and 70 mm photographs remote center numbers. See Table 2.3.2-5.

Ground log data sheets were generated by the Northrop team member at the test areas. These data include vehicle start up/turn off time and general remarks during test. Infrared radiometer measurements were taken of M-48/M-60 tanks and background of interest during the test.

The resulting data base included extensive high-quality infrared and visual spectrum images for both tactical vehicle targets and general background. Vehicle targets included tanks, APCs, trucks and jeeps with various thermal states, moving and stationary, top down and forward looking and in various backgrounds. Background data includes trees, open fields, roads and rivers as were available at the MICOM test area. Burning oil drum, flares, plywood array and tank suppression countermeasure data, runways, ramps, buildings, fixed and rotary wing aircraft data was also obtained.
The above data was reviewed and 935 frames of analog video tape data was digitized and ground truthed to form a Northrop data base. This data base was used to train the signal processor algorithms described in section 2.1.

IN-FLIC TEST LOG SHEET

DATE: 4-6-82 PNVS OPERATOR: TERRY

RECORDER OPERATOR: JOE / TED

TUESDAY (MORNING)

TAPE NO. Px Tx	TEST NO.	TARGET PARAMETERS	TIME OF DAY	REMARKS	PLANNED TEST	ACTUAL RESULTS
P26 T26			8:15 AM 8:32 "	START UP DOD ROAD		0-15 - FILM ADVANCING
	1		96 1438	COLD TANKS, PACKED W/ 1 WARMING UP		17,
	2		961444	BURNING OIL		18, 19, 20, 21, 22, 23
	3		961448	COLD TANKS, LINE		
	4		961451	F.L.C ≅ 200' 2 HOT, 2 COLD TANKS		24, 25
	5		961453	F.L.C 100' 2 MOVING TANKS, 2 COLD TANKS		24
	6		961454	3 MOVING TANKS NEAR DRUMS (SMOKE)		27, 28, 29, 30, 31, 32, 33 39
P27 T27	7		961502	3 TANKS ON ROAD, THEN 2 TANKS TURNING LEFT ON 2ND ROAD		35
	8		961505	2 TANKS ON DIRT LOAD		36, 37, 38, 39, 40, 41
	9		961507	3 TANKS, STAGGERED ON/OFF ROAD, MOVING		42, 43, 44

WARNING: 5 SECOND RECORDER STOP PRIOR TO TAPES EJECT

TABLE 2.3.2-5 AIRBORNE TEST LOG SHEETS

2.3.3 Aircraft Integration Certifications

Aircraft integration certification was obtained for each Northrop helicopter equipment installation from the U. S. Army Aviation Research and Development Command (AVRADCOM). This certification was required prior to test flights with the equipment installed. Airworthiness documents were generated for each helicopter equipment installation and submitted to AVRADCOM via MICOM. Each Airworthiness Document included system description, weight and centers of gravity analysis, crash safety analysis, electrical loads analysis, flight scenario, and a list of contractor personnel required to fly on the MICOM-furnished helicopter. All Airworthiness Documents generated during this program are listed below.

1. Northrop Document No. 342401, 29 January 1982, "Description of a Northrop Test System to be Installed on a MICOM UH-1H Helicopter"

   This Airworthiness Document covers the data collection system as flown at MICOM in March/April, 1982.

2. Northrop Document No. 83Y035, 10 March 1983, "Description of Northrop Captive Flight Test Seeker Systems to be Installed on a MICOM UH-1H Helicopter"

   This Airworthiness document covers the Captive Flight Test System as flown at MICOM,November, 1983

3. Northrop Document No. 82Y132, 24 November 1982, "Northrop System Description: Standard UH-1H Helicopter Standby DC Generator as Power Source for Test Equipment"

   This Airworthiness Document covers the use of the UH-1H helicopter standby DC generator as a power source for test equipment during the captive flight tests.

2.3.4 Captive Flight Tests

Captive flight testing with the Advanced Sensor Signal Processor brassboard was conducted to demonstrate the processor's ability to acquire, classify and track targets in various backgrounds in near real time.

2.3.4.1 Captive Flight Test Equipment

The captive flight test equipment consisted of the processor system integrated with the data collection system previously described in Section 2.3.1. (Figures 2.3.4-1 and 2.3.4-2). The integrated systems were fully laboratory checked prior to flight test. The captive flight test helicopter installation was certified as safe and an Airworthiness Report was submitted to AVRADCOM. Flight certification was granted by AVRADCOM prior to any captive flight tests. The additional equipment required a larger cabin rack and the addition of two floor pallets located in the gunner's well areas of the helicopter. Both the rack and floor pallets were designed and fabricated by Northrop per MICOM specifications. The processor equipment was installed in a special frame assembly that was shock mounted to the right gunner's well pallet (Figure 2.3.4-3). The equipment rack and processor is shown installed on a UH-1H helicopter in Figure 2.3.4-4.

The total electrical power required by the Captive Flight Test System exceeded the 100 amp limit of the helicopter non-essential bus. An auxiliary electrical power system was designed and fabricated by Northrop and flight certified by AVRADCOM for use on any standard UH-1H helicopter (Figure 2.3.4-5). This system can provide up to 200 amps of 28 vdc electrical power for the test equipment

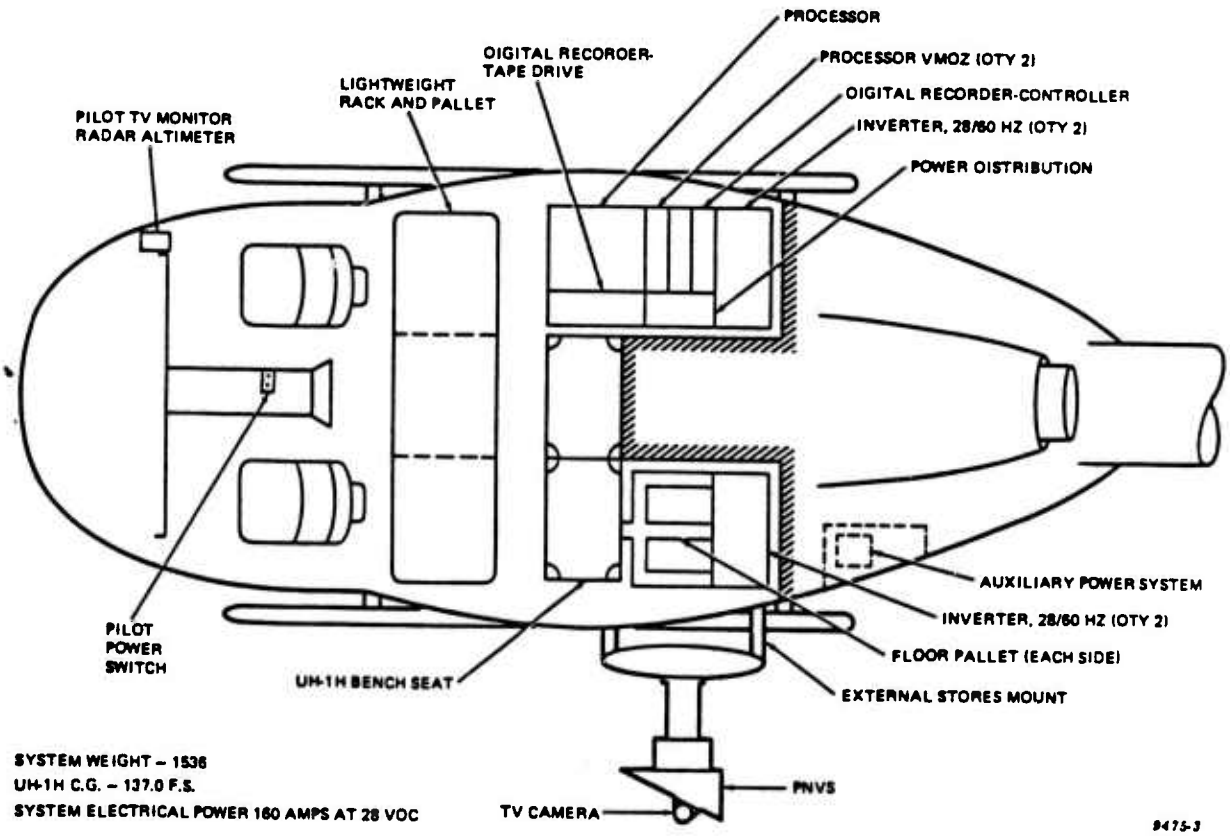


Figure 2.3.4-1 Captive Flight Test System Installed on Standard UH-1H Helicopter

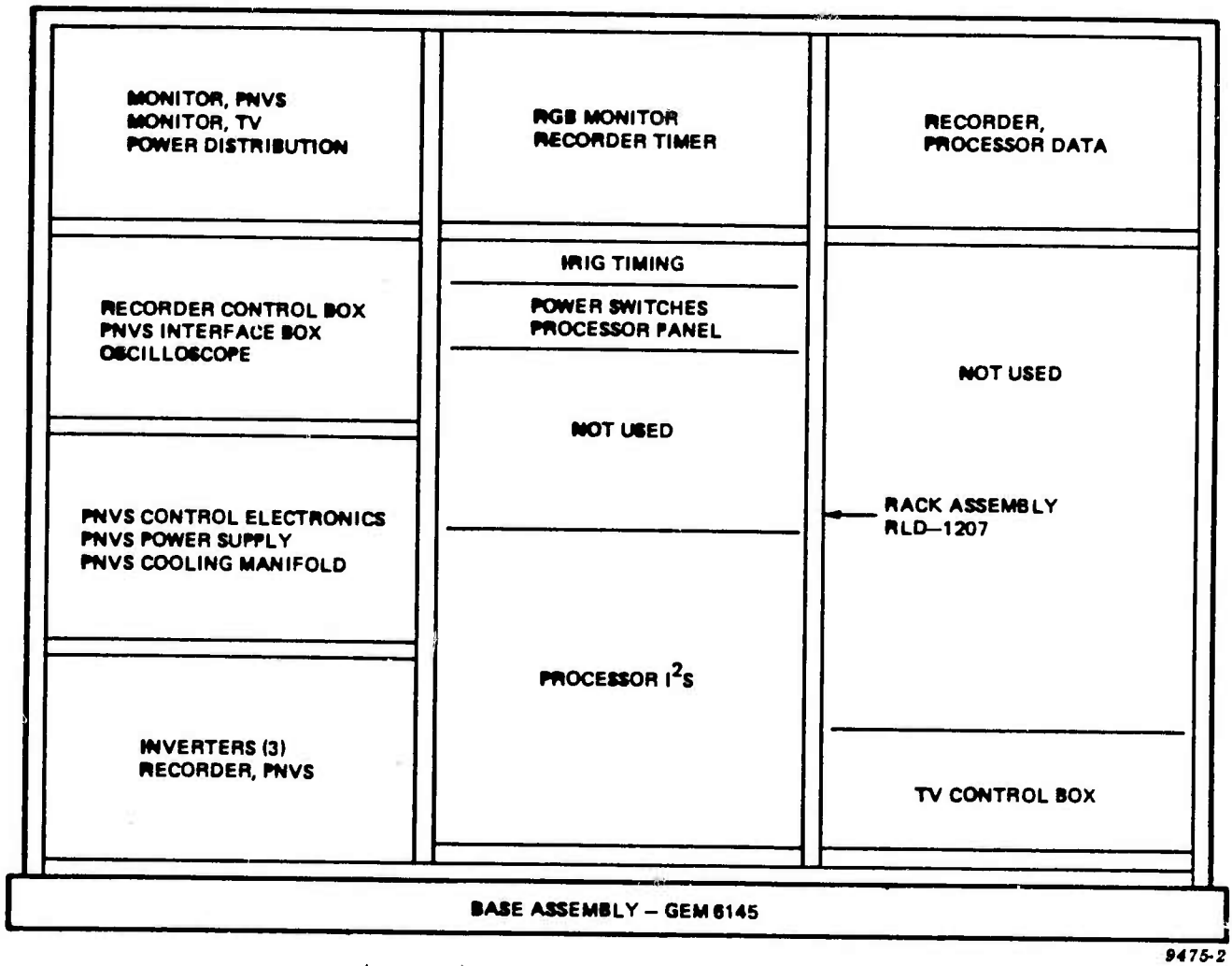


Figure 2.3.4-2 Captive Flight Test System Rack Equipment

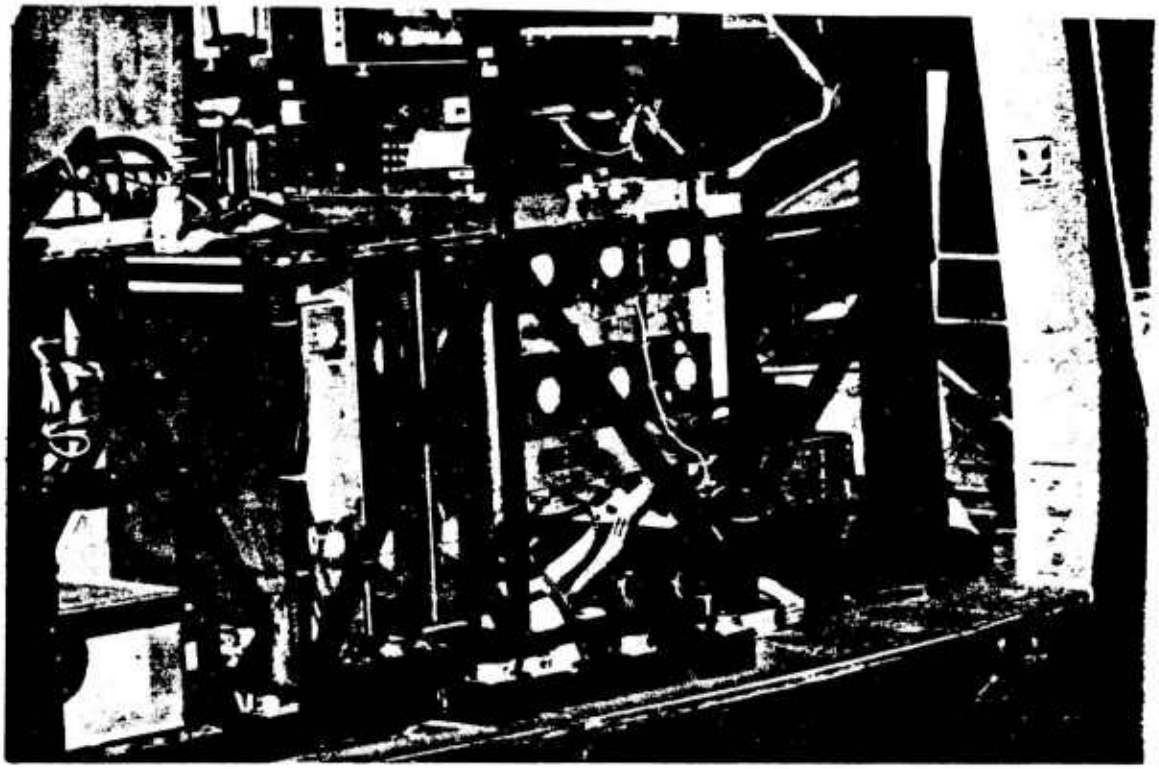


Figure 2.3.4-3 ASSP Processor Installed in Shock Mounted Frame

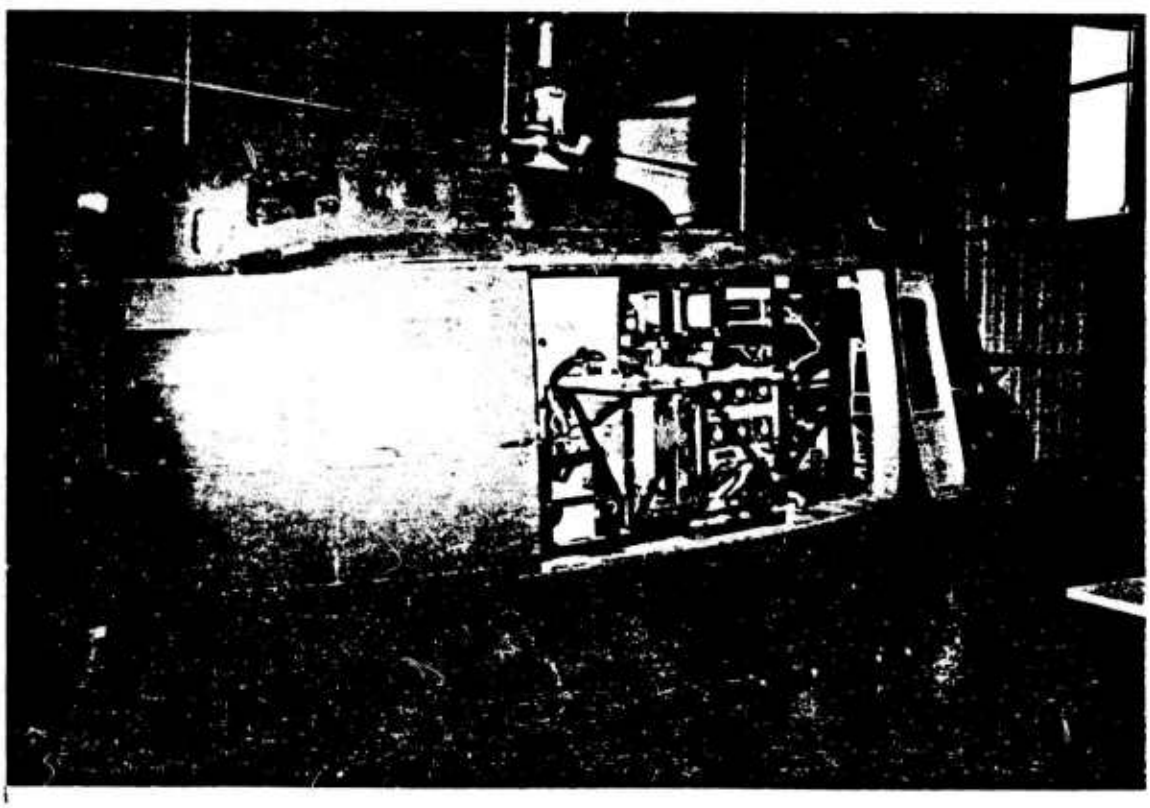


Figure 2.3.4-4 Captive Flight Test System Installed on UH-1H Helicopter

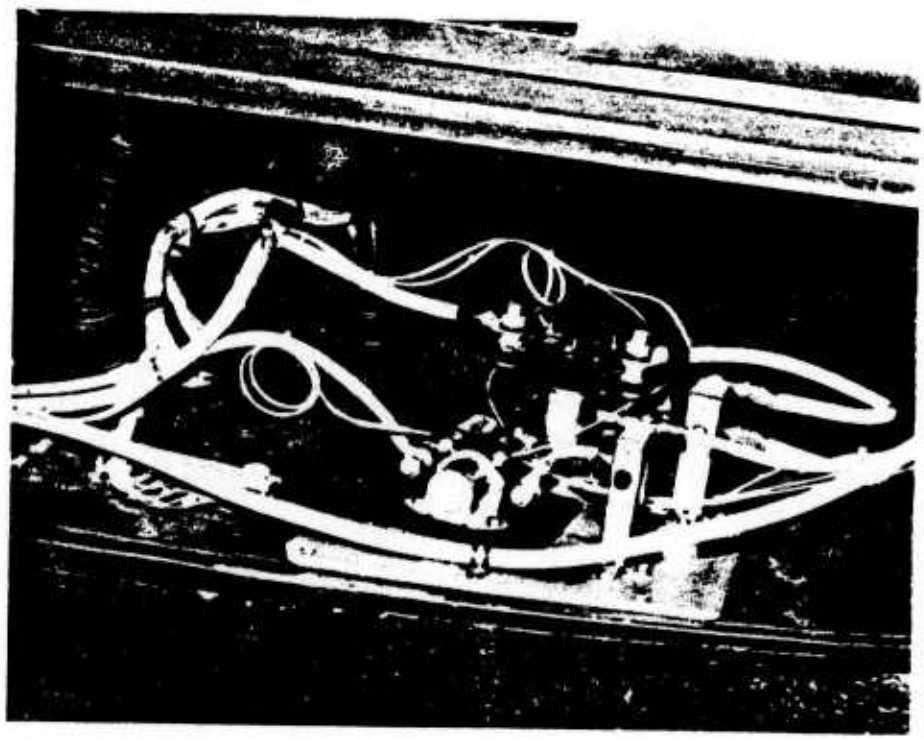


Figure 2.3.4-5 UH-1H Helicopter Auxiliary Power System

from the standby DC generator. The auxiliary electrical power system has the safety features of circuit breakers; and in the event of failure of either the helicopter main or standby DC generator, the test equipment electrical load is automatically taken off line. The processor and other test equipment required a large amount of laboratory type electrical power (115 VAC @ 60 Hz). Three 1000-watt, 28 VDC-to-115VAC, 60 Hz inverters were slaved together to provide 3000 watts of laboratory type electrical power. The entire captive flight test system was capable of being operated on the ground using standard aircraft type ground power supplies (Figure 2.3.4-6).

The captive flight testing required the determination of the true altitude above ground. MICOM furnished a radar altimeter system to Northrop for the captive flight test. The altimeter was integrated into the processor system and installed on the helicopter (Figure 2.3.4-7). The helicopter pilot was provided with the altimeter digital altitude.

Northrop designed and fabricated an auxiliary pilot's panel. Instrumentation was installed in this panel that allowed the pilot to fly the helicopter as required by the captive flight test (Figure 2.3.4-8). This instrumentation included a digital radar altimeter display and a monitor that provided the pilot with the externally mounted closed circuit television camera video.

2.3.4.2 Captive Flight Testing

The captive flight test system was installed on a 63rd Army Reserve Command UH-1H at Northrop Electro-Mechanical Division hanger facility. A check-out flight test was conducted over Camp Pendleton, California, on 4 November 1983, using M-60 tank targets. The system was verified to perform to specification and to have no adverse effect on the operation of the helicopter. The system was packaged and shipped to MICOM for the contract Captive Flight Tests on 7 November 1983. The Captive Flight Test activity required by the contract was completed successfully 11-22 November 1983. These tests were

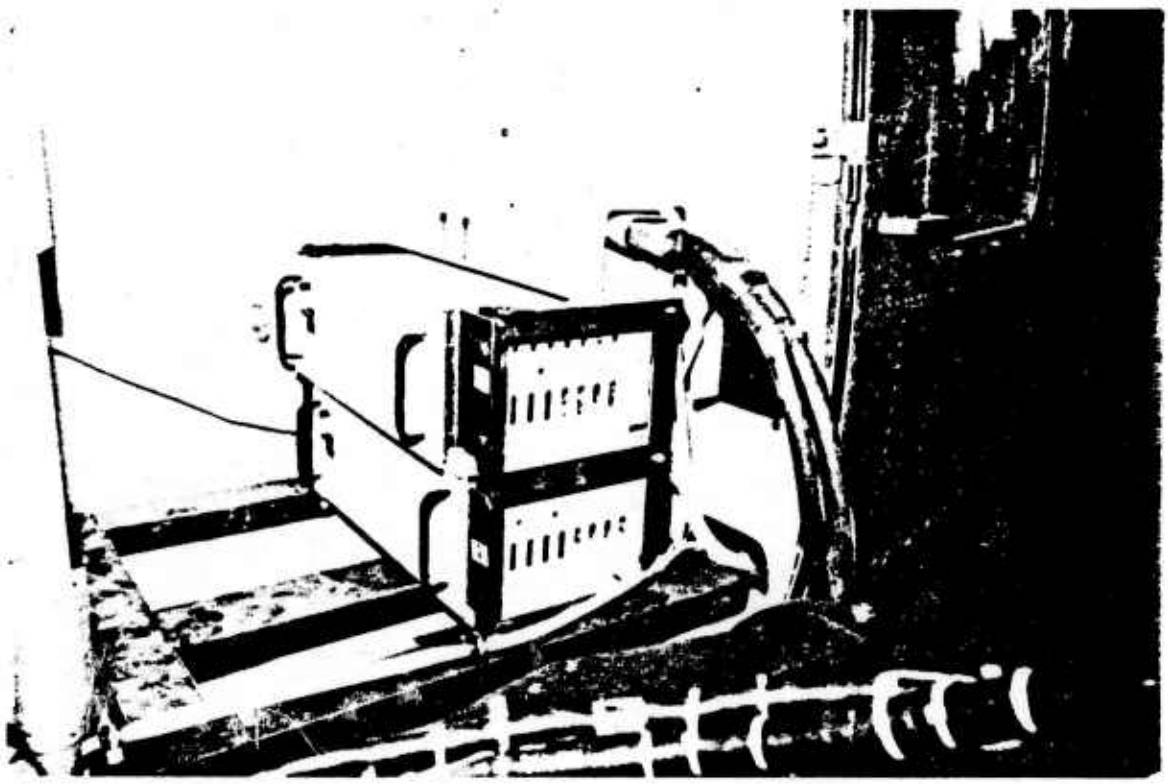


Figure 2.3.4-6 Captive Flight Test System Electrical Power Inverters and Ground Power

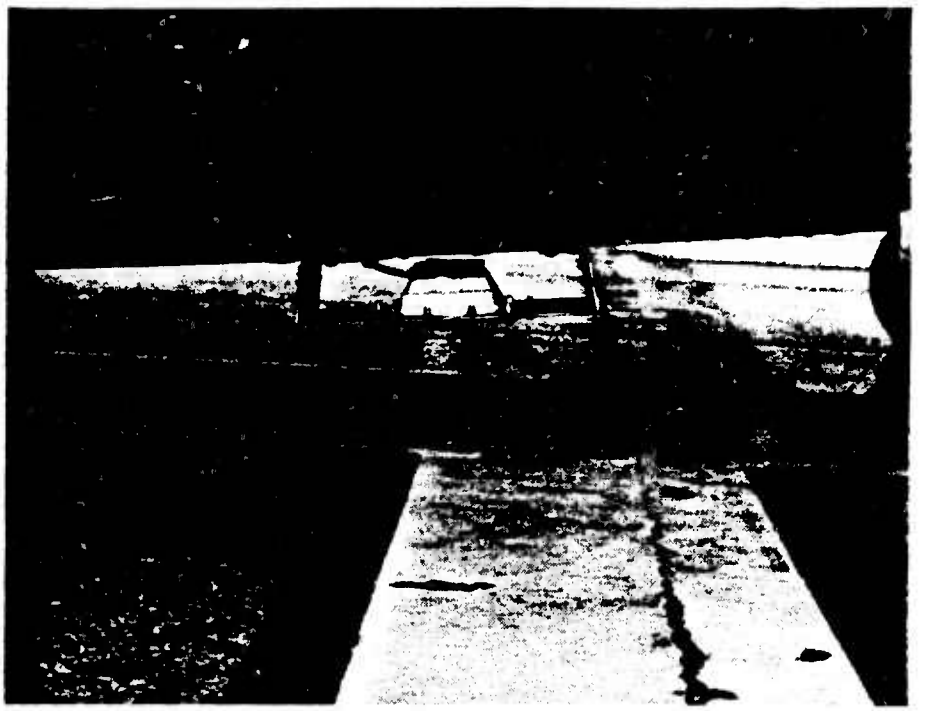


Figure 2.3.4-7 Radar Altimeter Antenna Installed on UH-1H Helicopter

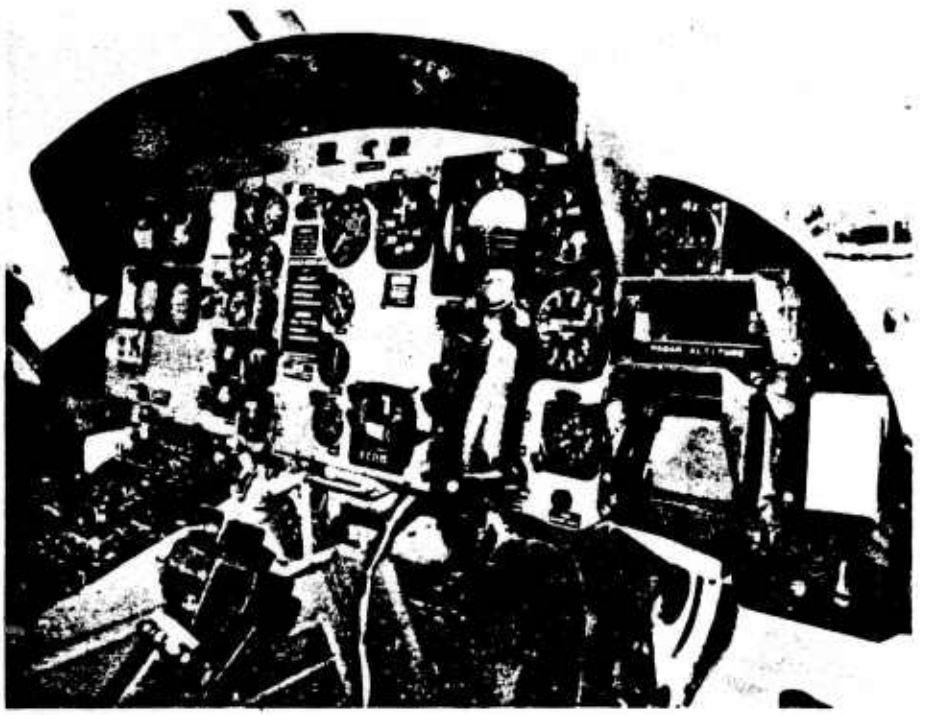


Figure 2.3.4-8 Pilot's Auxiliary Instrument Panel with Altimeter Readout and TV Monitor

controlled by a Captive Flight Test Plan, Northrop Document No. 342402. The Captive Flight Test System was installed in a MICOM UH-1H helicopter and the aircraft weighed by MICOM. Flight testing was conducted over MICOM's test area 6. A total of ten (10) flight hours of captive flight testing were conducted. Vehicle targets included tanks, APCs, jeeps, and 2 1/2 ton trucks, both stationary and moving. The vehicle targets were driven in a race track pattern to obtain a maximum variety of target backgrounds. Signal processor performance was evaluated at various altitudes and sensor line-of-sight pointing angles including both top down and forward looking scenarios.

2.3.4.3 Test Procedure

A typical captive flight test consisted of the following sequence.

1. Coordinate testing with the Helicopter Group and Test Area Group.
2. Discuss test with pilot(s).
3. Helicopter liftoff, system powered on, departure from high bay and establish communication with test area control.
4. Proceed to test site.
5. Start up vehicle targets and conduct test sequence while maintaining close air-to-ground coordination.
6. Maintain helicopter position over target(s) using the closed circuit television display and radar altimeter.
7. Processor operator initiates the signal processor operations.

2.3.4.4 Test Data

Analog video tapes of both the infrared video, closed circuit television video and the displayed signal processor results were generated. The displayed signal processor and preprocessed digitized infrared video frames were digitally recorded. The displayed signal processor results included target acquisition, ranking, tracking and algorithm performance data. The results of the signal processor are provided in section 2.1.4. IRIG timing signals are available on audio track 2 and all helicopter intercom and radio communication is available on audio track 1 of the analog video tapes. Airborne 70 mm photographs were taken with an externally mounted camera during the captive flight testing.

Table 2.3.4-1 provides examples of details measured on the events that occurred during each test sequence during the captive flight test.

TABLE 2.3.4-1 EXAMPLE OF CAPTIVE FLIGHT TEST SEQUENCE

11-22-83 (MORNING)

RESULTS: C1, C2, C3, C4
P1, P2, P3, P4
T1, T2, T3, T4

AREA: TA-6

ITEMS: JEEP, TANK, TRUCK MOVING ON DIRT ROAD TO STAGING AREA
JEEP, TANK, TRUCK MOVING PAST TWO PARKED, COLD APC'S
JEEP, TANK, TRUCK MOVING ON DIRT ROAD THROUGH FOREST
JEEP, TANK, TRUCK TURNING IN OPEN FIELD
JEEP, TANK, TRUCK ACROSS OPEN FIELD
JEEP, TANK, TRUCK ON DIRT ROAD MOVING PAST COWS
JEEP, TANK, TRUCK ON DIRT ROAD MOVING PAST SMALL BUILDING
JEEP, TANK, TRUCK MOVING PAST STAGING AREA
JEEP, TANK, TRUCK MOVING PAST LARGE TREE
JEEP, TANK, TRUCK MOVING IN OPEN FIELD

APPENDIX A

A STREAK-REMOVAL ALGORITHM FOR COMMON MODULE FLIR's

by

Dr. David D. Garber

APPENDIX A

A STREAK-REMOVAL ALGORITHM FOR COMMON MODULE FLIR's

ABSTRACT

The performance of a target acquisition system is dependent on the quality of the imagery input to the system. The multi-detector, parallel, scanning design of many infrared sensors, including common module FLIR's, induces a "streaking" noise into imagery. These sensor artifacts degrade the performance of target acquisition algorithms which use edge and contrast features for target detection and identification. As individual detector response varies with time and is frequently dependent on the content of each given image as well as the numerous contrast and gain controls in the sensor system, the image-dependent, time-variant, adaptive Garber filter was developed. The Garber filter was applied as a preprocessor in the Advanced Sensor Signal Processor (ASSP) program which used the Northrop Pilot Night Vision Sight (PNVS) common module FLIR. The ASSP hardware produced filtered imagery from raw PNVS video input in near-real time. The results of simulation and test flights indicate that the filter is robust under a variety of conditions and the image correction produced improved target acquisition results. The Garber filter also aids human interpretation of imagery by removing distracting sensor-induced streaks. Examples of unfiltered and filtered images are shown.

Introduction

The Garber filter works on each digital image frame entirely independent of any previous frames. It does not require information about the sensor nor does it require knowledge of sensor characteristics. It is designed to remove horizontal streaking image artifacts such as those found in common module FLIR imagery but it is not limited to that specific application. Imagery which exhibits vertical streaking noise could be transformed before application of the filter or an equivalent transposed form of the filter could be used.

As the filter is dependent only on the data in the particular image being processed, the filter is robust to temporal changes in sensor characteristics and image content. The global gains and offsets of the sensor system may be changed without affecting filter performance.

The response characteristics of the individual sensor elements within the FLIR change over time. This is very apparent when the sensor is "warming up". But it is also true that the individual elements appear to change over shorter periods of time during normal use of the sensor system.

Most importantly, the response characteristics of the individual sensor elements are dependent on scene content. Most dramatic is the effect of a hot spot in the scene on sensor response. An example is shown in Figure 1. Such streaking artifacts are caused by burning oil drums and hot tanks.

These two types of artifacts, temporal sensor element changes and hot spot induced, are very difficult to remove using filters which rely on measuring sensor response or either a large set of natural images or uniform test images. They also would degrade the effectiveness of time-averaged filters which would use a large sequence of images, over time, to produce the optimal streak-removal filter. The computational requirements of such filters could, in many cases, be greater than that required for the Garber filter.

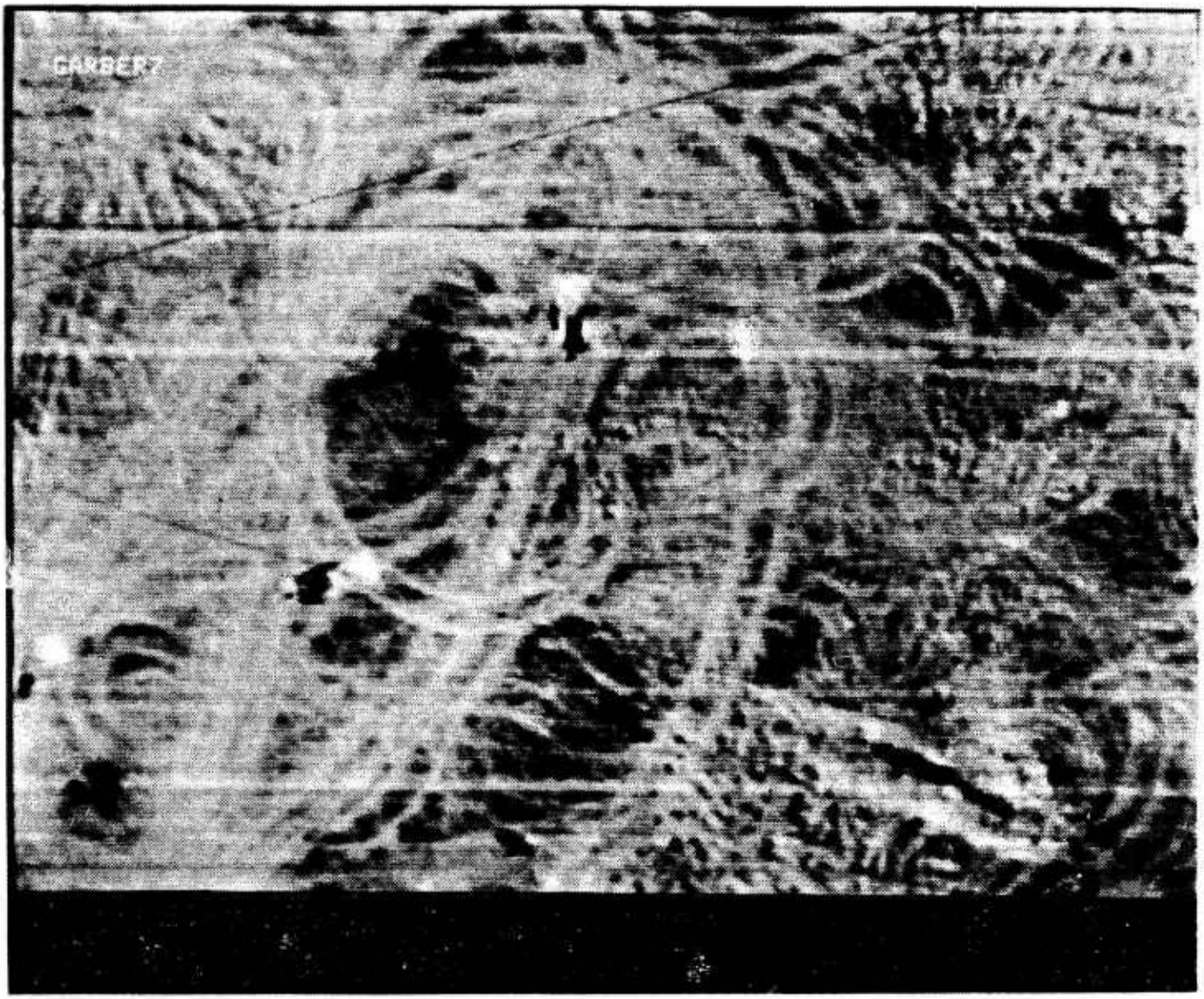


Figure 1. Hot Spot Induced Streaking
(Burning Oil Drum).

The Image Model

Let P_{ij} be the digital value of the pixel in the ith row and jth column of the image. Without loss of generality we will assume that P_{ij} is integer-valued and $0 \leq P_{ij} \leq 255$. P_{ij} is an element of our "ideal", "perfect" image, the value which we would obtain if we had no sensor noise, no sensor streaking.

Unfortunately, in most systems we are not given P_{ij}. We are given a distorted observed image element $\dot{P}_{ij}$. In the model for this filter, we assume that

$$\dot{P}_{ij} = a_i P_{ij} + b_i + \epsilon_{ij}$$

where a_i is the gain distortion induced into row i of the image, b_i is the offset distortion induced into row i of the image, and ϵ_{ij} is the rest of the unexplained variance or noise induced into p_{ij}. Simply stated, each row of the ideal image is distorted by some linear gain and offset and each pixel element is distorted by noise to produce the observed image. From the observed image we will attempt to reconstruct the original ideal image using the model.

Clearly the effect of the noise elements ϵ_{ij} are the most difficult to remove. There is an attempt to reduce their effect on the restoration of the observed image to its ideal state. We would hope that the ϵ_{ij} elements are small and preferably they would be random and uncorrelated but no assumptions to this effect used to be made for the analytical filter derivation. If one can determine that this noise is related to some known element, then the model should be modified to take advantage of that fact. But for purposes of our discussion we will assume that our primary concern is the distortion of the image caused by the gain and offsets a_i and b_i.

If we knew the value of these elements a_i and b_i, then restoring our observed image to its original state would be trivial, ignoring the effects of the

noise term ϵ_{ij}, as

$$P_{ij} = a_i^{-1} [\dot{P}_{ij} - b_i]. \quad (2)$$

Considering the error term we would guess that the magnitude of the error in the restored image P_{ij} would be proportional to the original error ϵ_{ij} and a_i^{-1}. We might simply state that the magnitude of the error in the restored image P_{ij} would be proportional to the original error ϵ_{ij} and a_i^{-1}. We might simply state that

$$\hat{P}_{ij} = a_i^{-1} [\dot{P}_{ij} - b_i] \quad (3)$$

and that $\hat{P}_{ij}$ is only an estimate of our ideal image P_{ij} in this case. In fact, any non-zero error ϵ_{ij} forces us to determine $\hat{P}_{ij}$ which is a mere estimate of P_{ij}. The quality or goodness of that estimate depends on the magnitude of the ϵ_{ij} and our ability to estimate the a_i and b_i.

Thus, our problem is reduced to one of estimating these a_i and b_i and then using these estimates to restore the observed image.

Given an observed image, $\dot{P}_{ij}$, it is not possible to determine a_i and b_i without knowledge of the ideal image, P_{ij}. However, we can estimate a_i and b_i by using a simple principle true in most naturally-occurring digital imagery. That is that neighboring pixel elements in an image are strongly related to each other. This is illustrated by the high correlation values found between adjacent pixels in most imagery. This same high relationship allows image coding compression to be successful - random pixels in images cannot be coded/compressed successfully.

Looking at the images in the PNVS data base we find a high degree of relationship between adjacent pixels. Fields tend to be very homogeneous. Roads are often locally homogeneous. Wooded regions can be locally homogeneous. By taking advantage of this homogeneity, we can estimate a_i and b_i and thus restore the image.

Let us for a moment consider two adjacent lines of an nxm image, P_{ij} and P_{i+1j}, $j=0_{ij}m$. In most cases we would expect the pixel P_{ij} to be a good estimate of the pixel P_{i+1j}. This would be true if the image lines come from a homogeneous image. It would not be true if pixel P_{ij} and pixel P_{i+1j} a) came from different objects in the image (i.e., they are on a boundary between image regions), b) came from a non-homogeneous region or c) were distorted by high noise content. Pixel P_{ij} is a poor estimate of P_{i+1j}, also if row i and row (i+1) are distorted from their ideal values by the linear gain and offset values a_i, a_{i+1}, b_i, and b_{i+1}.

We must note that if pixel P_{ij} and pixel P_{i+1j} are from homogeneous regions (which is often the case) that

$$a_i P_{ij} + b_i \doteq a_{i+1} Pi_{+1j} + b_{i+1}. \tag{4}$$

Expanding our set to include each pixel in the i and i + 1st row, we can see that we have an over-determined set of equations which would allow us to obtain estimates of a_{i+1}/a_i and $(b_{i+1}-b_i)/a_i$ by some method such as least-squares. It is similar to a linear regression problem as

$$P_{ij} \doteq \frac{a_{i+1}}{a_i} P_{i+1j} + \frac{1}{a_i} (b_{i+1}-b_i). \tag{5}$$

Applying this same principle to rows i+1 and i+2 yields a similar equation,

$$P_{i+1j} \doteq \frac{a_{i+2}}{a_{i+1}} P_{i+2j} + \frac{1}{a_{i+1}} (b_{i+2}-b_1). \tag{6}$$

One concept more or less implicit in these equations is the interdependence of the linear gain factors, a_i, and the offsets, b_j. Also, the true a_i's and b_i's cannot be directly determined, but important linear and offset factors (which are indeed the desired correction factors) can be calculated.

The first of the above two equations implies that the correction factor for row i+1 pixels <u>relative</u> to row i pixels is a linear multiplicitive factor of $\frac{a_{i+1}}{a_i}$ and an additive factor of $\frac{1}{a_i} (b_{i+1}-b_i)$. The second of the above two

equations implies the correction factors for row (i+2) relative to row (i+1) are in a linear multiplicative factor of $\frac{a_{i+2}}{a_{i+1}}$ and an additive factor of

$$\frac{1}{a_{i+1}} (b_{i+1} - b_{i+1}).$$

It is necessary to combine these equations to determine the correction factor of row (i+2) relative to row i. This is clearly desirable as a single reference is required to produce uniform correction over an entire image.

$$P_{ij} \doteq \frac{a_{i+1}}{a_i} \left[\frac{a_{i+2}}{a_{i+1}} P_{i+2j} + a_{i+1} (b_{i+2} - b_{i+1})\right] + \frac{1}{a_i} (b_{i+1} - b_i)$$

$$\doteq \frac{a_{i+1}}{a_i} \cdot \frac{a_{i+2}}{a_{2+1}} P_{i+2j} + \frac{a_{i+1}}{a_i} \left[\frac{1}{a_{i+1}} (b_{i+2} - b_{i+1})\right] + \frac{1}{a_i} (b_{i+1} - b_i) \quad (7)$$

Doing this, the correction factors for row (i+2) relative to row i are a linear multiplicative factor of $\frac{a_{i+1}}{a_i} \frac{a_{i+2}}{a_{i+1}}$, the product of factors derived using adjacent rows, and an additive factor of

$$\frac{a_{i+1}}{a_i} \left[\frac{1}{a_{i+1}} (b_{i+2} - b_{i+1})\right] + \frac{1}{a_i} (b_{i+1} - b_i)$$

which involves individual additive factors, one multiplied by a linear multiplicative factor.

Continuing the process, we see that the linear adjustment for row (i+k) relative to row i is a product of k linear factors, the factors determined by a comparison of adjacent line pairs.

The successful and accurate calculation of each factor and their product is important. We cannot merely compare the ith and (i+k)th row for arbitrary k

to determine the correction factors for row (i+k) relative to row i. This violates our assumption (and the requirement) of homogeneity between lines. Non-adjacent rows i and (i+k) are likely to contain non-homogeneous data than are adjacent rows for large k.

Algorithm Implementation

The final implementation of the Garber filter combines the theory expressed above with some ad-hoc rules designed to make application of the filter more robust. The factors to be considered include 1) the noise in the imagery, 2) ill-conditioning problems, and 3) violation of non-homogeneous assumptions.

A special mask shown in Figure 2 was chosen to make the algorithm robust. It is not necessarily the best but our studies verify its robustness in a variety of circumstances. Passing the mask over the observed input image we calculate

$$d_{ij} = \sum_{k=j-1}^{j+1} P_{ik} - \frac{1}{2} \sum_{k=j-1}^{j+1} (P_{i-3k} + P_{i-2k}) \qquad (8)$$

at each point ij in the image. The d_{ij} represents the average difference of the top two lines and the bottom line over a 3-pixel-wide sweep along the line.

Averaging the six pixel values on top and the three on the bottom helps to reduce the effect of noise in the imagery. Eliminating the row (i-1) in the calculations might help to reduce the effects of errors, e_{ik}, which can be related between immediately adjacent rows. The true effect of this has not been studied fully.

Differences between pixel values in rows can also be caused by neighboring homogeneous but unequal regions. A simple illustrative example is shown in Figure 3. In this figure, pixels from object B are not sampled in the (i-3) and (i-2) rows which would cause d_{ij} to be large in absolute value at those points. Inclusion of these pixel values can adversely affect the correct calculation of correction factors for each row. The historgram of d_{ij} along row i would look like that in Figure 4.

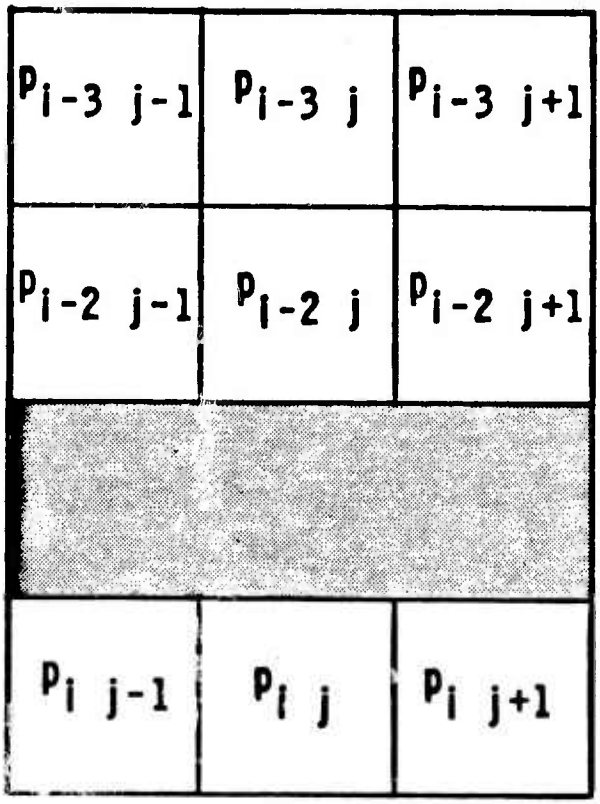


Figure 2. Pre-filter Mask.

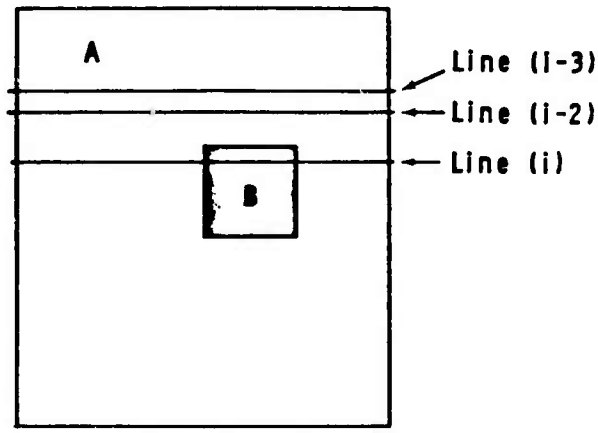


Figure 3. Homogeneous but Unequal Regions.

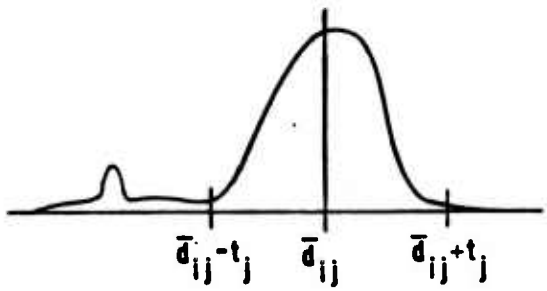


Figure 4. Histogram of Differences Along a Line.

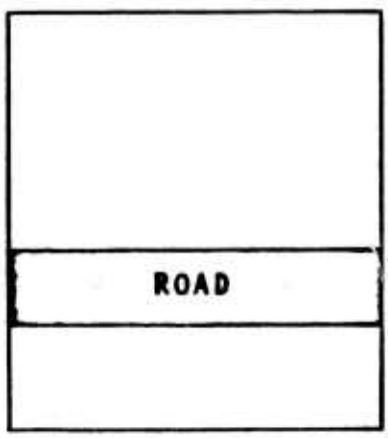


Figure 5. Perfect Homogeneous Horizontal Road Passing Through Homogeneous Field.

Our studies indicate that the majority of streaking noise can be eliminated by offset, b_i, compensation/adjustment only. We might expect, therefore, that the histogram in Figure 5 would be centered around this offset with new object d_{ij} sample values and non-homogeneous area d_{ij} samples occupying the tails of the histogram. By thresholding this histogram of a certain percentage on both sides of the mean, these tails can be eliminated. Let us denote this threshold as t_i. From this point on we wish to only use the pixels which have a $d_{ij} < t_i$ in our correction factor calculation.

For these pixels where $|d_{ij}| < t_i$ we calculate the means and variances,

$$\mu_{k_i} \text{ and } \sigma^2_{k_i},$$

of $(P_{i-kj-1} + P_{i-kj} + P_{i-kj+1})/3$ for k = 0,2,3.

That is we compute the mean and variance of the average of the three pixels for each of the three lines for those points where $|d_{ij}| < t_i$. The final linear adjustment to row (i) is computed as

$$a_i = \left\{ \frac{a_{i-3} \cdot \sigma^2_{3_i} + a_{i-2} \cdot \sigma^2_{2_i}}{2 \cdot \sigma^2_{o_i}} \right\}. \tag{9}$$

Note that this equation requires the previous linear adjustments a_{i-3} and a_{i-2} are required for this process.

Following the calculation of the linear gain adjustments a_i the offset biases b_i are calculated as

$$\hat{b}_i = \frac{1}{2}\left[\mu_{3_i} \cdot \hat{a}_{i-3} + \mu_{3_i} + \mu_{2_i} \cdot \hat{a}_{i-2} + \mu_{i-2}\right] - \mu_{o_i} \cdot \hat{a}_i . \tag{10}$$

It is very important to notice that Equation (9) indicates that the adjustment is a function of the variances within the lines being processed and not a linear, least-squares fit as implied by equation (5). The linear adjustment

of equation (9) attempts to set the variance of the pixels in one row equal to that of the previous row. It is not a pairwise linear fit of adjacent pixels. This concept helps to reduce the effect of non-homogeneous, textured regions on the quality of the image restoration.

Similarly, the adjustment indicated by equation (10) forces equality of means between lines. In practice, the non-linear nature of the selective sampling process which selected samples according to the threshold criteria t_i can produce a set of adjustments a_i and b_i which distort the global contrast and gain of the image. Our studies indicated that contrast change was usually very small, less than 2% in most cases. Therefore, for single image filtering the a_i do not need to be adjusted. In the event that adjustment is desired, setting

$$\prod_{i=1}^{n} \hat{a}_i = 1 \tag{10}$$

by adjusting each $\hat{a}_i$ by the appropriate linear factor will solve the problem. Adjustments to the $\hat{b}_i$, however, is required. In this case setting

$$\sum_{i=1}^{n} \hat{b}_i = 0 \tag{11}$$

will cause the overall image brightness to remain approximately the same between the observed $\dot{P}_{ij}$ and the adjusted output image, $\hat{P}_{ij}$.

The final image adjustment is computed on a row by row basis as

$$\hat{P}_{ij} = \hat{a}_i \dot{P}_{ij} + \hat{b}_j. \tag{12}$$

Implementation Details

The form of the equation used to calculate the a_i requires a large degree of accuracy in the multiplication, division and square root operations. A time-efficient hardware implementation of this process requires careful examination of the propagation errors that can occur.

In our hardware implementation, this expression was efficiently calculated by doing a Chebyshev polynomial approximation for the reciprocal and another for the square root function. The approximation was accurate to 30 bits.

There are a number of image boundary problems that must be considered when doing algorithm implementation. First, the right and left boundaries of the images can be (and should be) ignored if the pixels in these regions can contain noisy or invalid image data. This helps to reduce the effect of the ϵ_{ij} on coefficient calculation. Secondly, equation (9) requires a starting point, which is allowed by setting $a_1 = a_2 = a_3 = 1$. This implies that the top lines of the input image will be used as the global reference. By inducing a "border" covering these lines and beginning the filtering process a few rows from the top of the image, effects of bad data in these rows can be eliminated. Unfortunately, the streak filtering is also eliminated. By implementing the requirement of equation (10) which helps to insure image contrast remains equal between observed and corrected images, the effects of bad data in the border areas can also be reduced. Finally, a top-down filtering could be combined with a bottom-up approach to insure preservation and filtering of both bottom and top lines of the image. In filtering PNVS data we found that setting $a_1 = a_2 = a_3 = 1$ alone produced adequate filtering results.

The success of the method requires a reasonable fit of the model to the degradation process and is aided by low noise, ϵ_{ij}. High noise or corrupted data insures loss of image information and degrades the performance of this (and any other) filtering process. Lost data cannot be recovered.

As with any time-constrained hardware implementation of an algorithm, one is forced to make important tradeoffs of quality performance for time-efficiency. A non-linear model, or higher-order model might better explain

the degradation process and improve the quality of the restored image but such a model introduces complexity into the coefficient and restoration process. As was stated earlier, we dropped the equation (10) constraint to improve execution speed. Histogram thresholding would best be done by computing the optimal linear correction relative to two lines before calculating the histogram differences. The final hardware implementation required that all calculations be done using fixed-point arithmetic to speed execution. As a result of this, the required accuracy of each of intermediate calculations needed to be determined. Of particular concern is the accuracy required by equations (9) and (10).

Finally, it is indeed true that this filtering process could remove a large homogeneous object if that object had a very straight edge and that edge fell along a scan line. An example of such a situation would be a straight road which passes through a field as shown in Figure 5. In this invented situation, the road would be removed as an image streak (which may not necessarily be bad). We find, in natural scenes, that roads are generally not so perfectly straight with perfect borders and the probability of such a road falling adequately close to a horizontal line of an image is very low. No examples of such an occurrence were found in the natural images processed during our study.

Algorithm Extensions

A variety of modifications to this basic algorithm could be tried. Each could yield improved results or could be used to tune the streak-removal algorithm to a particular set of imagery.

The mask shown in Figure 2 could be perturbed a number of ways. The threshold percentage used to determine t_i on each row can be adjusted. The method of thresholding could be modified.

Non-linear models of degredataion could be proposed and appropriate restoration equations developed.

Application of the filter is not limited to horizontally streaked images. Images with vertical streaks can be transposed before filtering. Images which exhibit both horizontal and vertical streaking could be processed

serially in each dimension and for best results. This filtering might be applied a second (or more) time as shown in Figure 6.

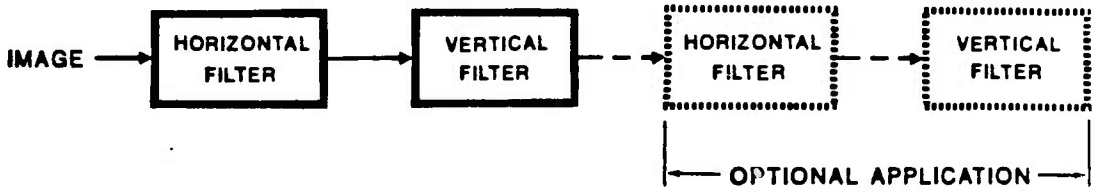


Figure 6. Multiple-pass Image Filtering.

Conclusions

The Garber filter was applied to many thousands of images during our study. The filter appeared to sensor element changes and image content changes. The image correction produced improved target acquisition results.

Two sample results are shown in Figures 7 through 10. These digital images are raw PNVS data, with 512 columns and 380 rows. The images are improved dramatically by applying the filter.

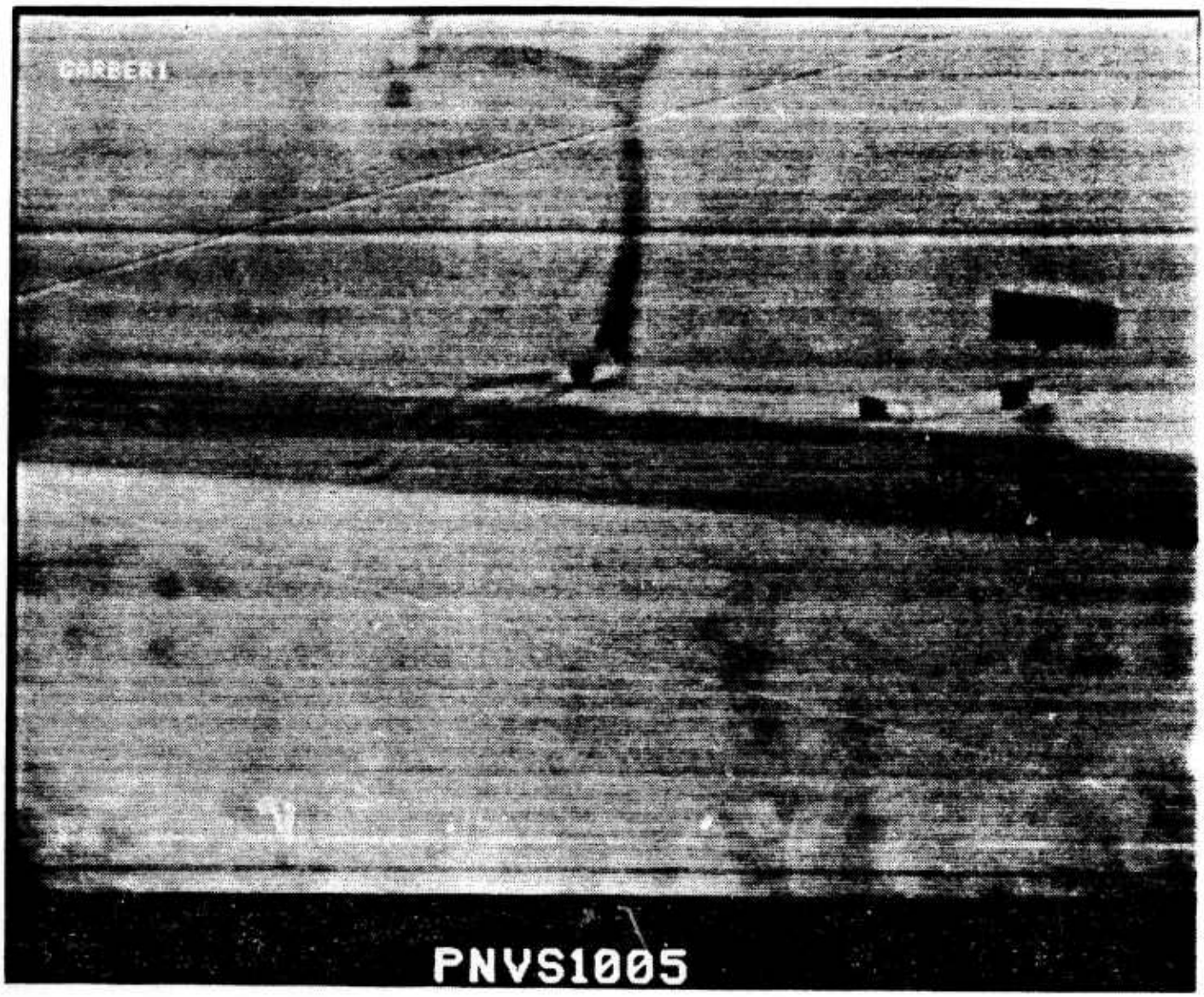


Figure 7. Raw PNVS Image.

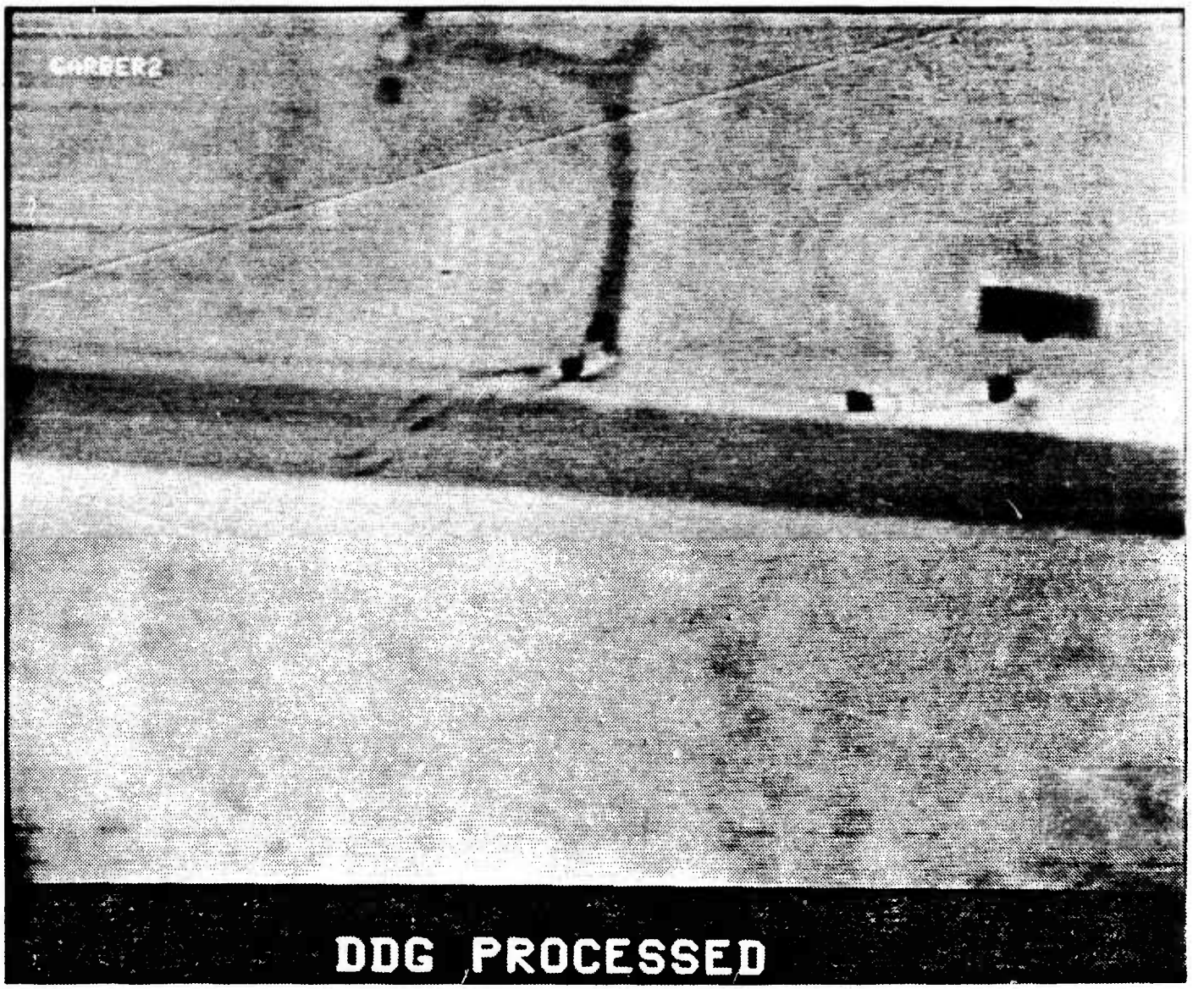


Figure 8. Filtered Image.

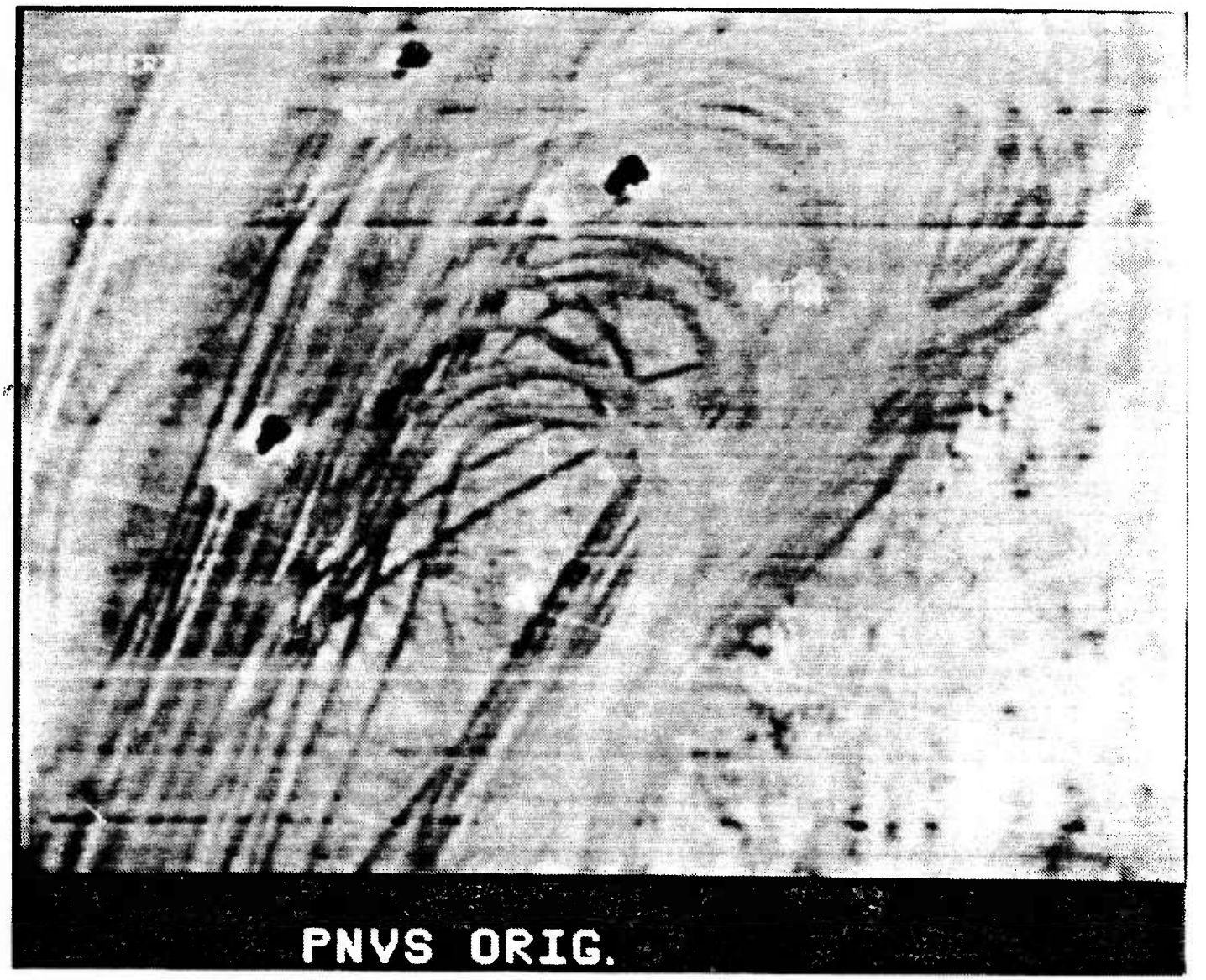


Figure 9. Raw PNVS Image.

Figure 10. Filtered Image.

APPENDIX B

NATS/ASSP REAL-TIME PROCESSOR

by

S. J. Ohlsen
and
G. E. Martin

NATS/ASSP REAL-TIME PROCESSOR

Development of the NATS/ASSP Processor consisted of designing and building a Real-Time hardware system with extensive software and firmware (microcode) to perform the data manipulations specified in the data flow diagrams (see figure B.32). A significant effort was spent in implementing firmware development software, a formalized firmware language processor and a specialized set of tests diagnostics, and checkout procedures for each section of the hardware as-well-as each logically isolated algorithm implementation.

The Real-Time hardware system consists of two sub-systems plus peripheral devices. The first sub-system is a VERSAmodule 68000 mono-board computer system with added memory (RAM and ROM) boards. The second sub-system is the Northrop developed Pipeline Processors. Peripheral devices consist of a Keypad, Graphics Display Unit, Digital Cassette Tape Recorder, IRIG Time Code Generator, and Radar Altimeter (See Figure B.1).

Appendix B is divided into 5 sections. The sections and subject each addresses are as follows:

Section	Description
1	VERSAmodule 68000 COMPUTER SUB-SYSTEM
2	PIPLINE PROCESSOR SUB-SYSTEM
3	PERIPHERAL DEVICES
4	MICROCODE and FIRMWARE DEVELOPMENT SOFTWARE
5	PROCESSOR CHECKOUT, SYSTEM INTEGRATION, and TEST

1 VERSAmodule 68000 COMPUTER SUB-SYSTEM

The first sub-system within the ASSP is the VERSAmodule system. VERSAmodules are a product line manufactured by Motorola. This product line includes chassis, power supplies, mono-board computers, and memory boards. The VERSAmodule board products used by the ASSP are M68KVM02 mono-board computers and M68KVM80 memory boards. The ASSP uses four M68KVM02's, two M68KVM80's, and one Northrop developed memory board. All boards are mounted in a VERSAmodule double chassis. The chassis provides a mounting fixture, provides power to all boards, and interconnection between boards with VERSAbus. See Figure B.1.

1.1 68000 MONO-BOARD COMPUTER, M68KVM02 (VM02)

There are four M68KVM02 mono-board computers in the ASSP system. The M68KVM02 is referred to as VM02 throughout this document. The VM02 specifications pertinent to the ASSP are:

- 8 MHZ, MC68000, 16 bit microprocessor
- Full VERSAbus interface with arbitration logic
- Local on-board bus
- 128K byte, dual port dynamic ram, accessable through VERSAbus from local bus
- 64K byte user PROM
- two RS-232C Ports

See Figure B.6 for a block diagram of the VM02. For a full product description see the manufacturers user manual. (Motorola Publication M68KVM02).

1.2 128K BYTE RAM BOARD (VM80)

There are two M68KVM80 Memory boards in the ASSP system. The M68KVM80 is referred to as VM80 throughout this document. The VM80

specifications pertinent to the ASSP are:

- 128K byte dynamic RAM
- VERSAbus interface
- 256K bytes user PROM

See Figure B.7 for a block diagram of the VM80. For a full product description see the manufacturers user manual. (Motorola Publication M68KVM80).

1.3 512K BYTE EPROM BOARD

This Northrop developed expansion memory board provides a large non-alterable memory addressable through a VERSAbus interface. The pertinent specifications of this board are:

- 512K byte EPROM (32 sockets for 28 pin EPROMs)
- VERSAbus interface

2 PIPELINE PROCESSOR SUBSYSTEM

Six Northrop developed pipline processors were used to collect data, compensate for sensor flaws, and perform data analysis and reduction. The bipolar, bit slice logic used in the processors was layed out on ten 15 by 16 inch wire wrapped boards. Four processors use 2 boards each and two processors use one board each. Each processsor is logically independent and supports its own microcode driven architecture. The processors interface to each other, to the video sensor, to the peripherals, and to the VM02's (i.e. 68000 mono-board computers). The Pipeline Processors are interconnected in a loosely-coupled fashion. This means that each Processor is independent and self contained with respect to memory requirements and computational needs. The Processors communicate with each other only to transfer data and commands. When a Processor rece the required data (image) and/or command, that data will be operated on

autonomously. The results are passed on to another Processor or to the VMD2. The firmware and the hardware architecture of the processors is modular to a great extent. In hardware, this means that a functional circuit on one processor is duplicated and exists as a part of another processor. For example, the micro-engine (the CPU group, address sequencer, and microcode memory) is (nearly) identical for all six of the processors (see section 2.2.1.4). In firmware, this means that (because the hardware is modular) a firmware module which works on one processor will work on other processors. For example, the I/O firmware is almost identical on all processors. See section 2.2.1 for a description of the major hardware circuit modules. The Pipeline Processor discussion is in three major sections. The titles of these sections and their numbers are:

2.1 PIPELINE PROCESSOR CHASSIS

The chassis used to hold the Pipeline Processors is a Mupac 347 series chassis. It can hold 13 wirewrap boards of the kind used in the ASSP (see section 2.2.1.1). The ASSP consists of 10 boards. Each board has five 108 pin connectors. (see Figure B.2). Three connectors plug into the chassis. The power for each board comes in through the chassis connectors. The other two 108 pin connectors are on the other side of the board and are used for the WCS interface (see section 2.2.1.8). The chassis connectors are also used for interconnection of the system data busses, I/O handshake signals, and command signals between Pipeline Processors. Mounted on the side of the chassis are the power supplies, the video analog-to-digital (A/D) converter, the D/A converter, the

ON/OFF switch and the cooling fans (see Figure B.3). Each of the chassis mounted items is discussed below.

2.1.1 POWER SUPPLIES

There are two power supplies mounted on the Pipeline Processor chassis (see Figure B.3). One power supply provides 5 Volt DC power to the Pipeline Processors. The second power supply provides +/- 15 Volt DC power for the A/D converter. Both power supplies use 110 Volt 60 Hz. power. The 5 Volt power supply is an LH Research P/N 847547-002. It is rated at 300 Amps. The 5 Volt and Ground terminals of the 5V power supply are wired directly to the backplane VCC and GND posts using 14, 12 ga wire for 5V and 14, 12 ga wires for GND. Current measurements taken on the 5V line, using Hall effect probes indicate that less than 200 Amps are being used. The +/- 15 Volt DC power supply is an Acopian model VTD15-160. It is connected directly to the A/D converter. Note that there is no +/- 15 Volt power on the Pipeline Processor backplane or in the Pipeline Processors.

2.1.2 VIDEO ANALOG-TO-DIGITAL (A/D) CONVERTER AND SYNC BOX

The A/D converter/Sync box is mounted on the side of the Pipeline Processor chassis (see Figure B.3). The BNC inputs to this box are the 875 line composite video and a TTL level composite sync signal. Other inputs are 5V and +/- 15 Volt power. The video input is conditioned by an (optional) 5 pole anti-aliasing filter and DC restored with a key clamp. After conditioning, the video goes to the A/D converter. The A/D converter used is a TRW TDC-1007J. It is an 8 bit, 'flash' A/D converter. A 17.3 Mhz clock generated on the SFV board is used for digitizing. The 8 bit output from the A/D goes through a clocked output driver to the SFV board. The TTL level

composite sync input signal goes through a sync stripper which creates a separate TTL level signal for vertical blank and horizontal blank. The two blank signals are sent to the SFV to synchronize the SFV Operational program to the incoming video. The A/D-Sync box also wraps the input video (after the anti-aliasing filter) back through a separately buffered BNC output for use by a video recorder or monitor. A small box 'piggy-backed' to the A/D-Sync box contains a digital-to-analog (D/A) converter and output buffer. The cable to this box is connected at the interface between the A/D-Sync box and the SFV board. The D/A box is connected to the clock, horizontal and vertical blank, composite sync, and 8 bit video. The BNC output from this small box provides a (buffered) composite video signal which is used to monitor the output of the A/D. During flight test, this output was connected to an oscilloscope. The PNVS operator watched the video level on the oscilloscope and adjusted the gain of the PNVS to maximize the video level.

2.1.3 COOLING FANS

Six cooling fans are mounted on the side of the Pipeline Processor chassis (see Figure B.3). The fans use 120V at 60 Hz and are protected by a 3 amp fuse. Between the fans and the chassis is a plenum which concentrates the air flow across the boards and away from unused slots.

2.1.4 CHASSIS ON/OFF SWITCH, INDICATOR LIGHT, FUSE, AUX. 5V POWER PLUG

Mounted on the cooling fan assembly is an ON/OFF switch, an Indicator Light, fuse holder, and an auxiliary 5V power plug (see Figure B.3). The ON/OFF switch controls the 60 HZ input power to all power supplies and the cooling fans. Placing the switch in the OFF position will

remove 60 HZ power from the chassis mounted items. When the switch is in the ON position the Indicator Light will be lit. When the switch is in the OFF position the Indicator Light will be off. The fuse is a 3 Amp fuse which protects only the cooling fans. The auxiliary 5V power plug is a Mulex connector. The two leads from the Mulex are connected to VCC and GND on the chassis backplane. The red lead is VCC, and the black lead is GND.

2.2 DESCRIPTION OF PIPELINE PROCESSORS

In the following sections the Processors will be described individually. For simplicity, the common hardware modules will be described in a separate section. These names of these Processors, their abbreviations, the number of wirewrap boards per Processor, and the section describing each Processor is given below.

ABBR	NAME	Bd/Proc.	SECTION
-	COMMON FUNCTIONAL MODULES	-	2.2.1
SFV	SENSOR FORMATTER/VIDEO	1	2.2.2
SFN	SENSOR FORMATTER/NUMERIC	1	2.2.3
GP	GLOBAL PROCESSOR	2	2.2.4
MLC	MAXIMUM LIKELIHOOD CLASSIFIER	2	2.2.5
SL/TT	SCENE LOCK/TARGET TRACKER	2	2.2.6
CD	CONTROL AND DISPLAY	2	2.2.7

2.2.1 COMMON CIRCUIT MODULES

The Sections below discuss the 'Common Circuit Modules'. These Modules are used on more than one Processor.

2.2.1.1 WIREWRAP BOARD - The particular wirewrap board used in all of the Processors is a Mupac P/N 9191001-01 (see Figure B.2). This

board is a 4 level (2-VCC,2-GND) board with 136 rows and 69 columns of wirewrap pins. The arrangement of the rows is with a universal pattern in the center of the board and a high density pattern on the outer parts of the board. This board has three connectors on one end and two connectors on the other. Each connector has three rows by 36 pins. The three connector side of the board connects to the backplane. All system signals and power come through the backplane connectors. The two remaining connectors are used for interface to the Writeable Control Store (WCS-see section 2.2.1.8).

2.2.1.2 BLOCK ACCESS MEMORY (BAM) - This functional module is a 64K word memory with special addressing capabilities. Conceptually, this memory looks like a 256 X 256 array of picture elements (pels), with two 8 bit bytes of data contained in each pel. The two bytes and can be written independently. The BAM can be read with the bytes in normal position or in swapped position. There are three parts to the BAM. The three parts are: the Memory array, the DMA address chips, and the sub-sample register. The following sections will discuss these parts individually.

2.2.1.2.1 BAM (DMA) ADDRESS GENERATOR - The addressing for the BAM is provided by two separately controled DMA address generator chips (Am2940). Each of these DMA chips create an 8 bit address. One DMA chip generates addresses for one side of the 256 x 256 array, and the other DMA chip generates addresses for the other side of the 256 x 256 array. The DMA chips are independent so either one can be designated the X-Axis (fast axis), or Y-Axis (slow axis). The hardware is connected in a fashion which divides the 256 X 256 array into four quadrants of 128 X 128 pels. Using the DMA chips

in this fashion relieves the Programmer of a great deal of computational overhead. Typically, the data in a BAM is a picture, and the algorithms operate sequentially by row or by column. With DMA chips providing the addresses, the Programmer loads only the origin point and length of the window being operated on into the DMA chips. The Program accesses data in the window by incrementing the fast axis DMA chip until a signal from the DMA chip indicates that the end of the window has occured. The Program then increments the slow axis and loops back. When the end of window occurs in both axis, the routine is complete. A subsample register (SSR) is associated with each DMA address chip. The SSR controls when the DMA chips increment (see section 2.2.1.2.2) and, therefore, enables noncontiguous BAM data access.

2.2.1.2.2 BAM SUB-SAMPLE REGISTER - Each BAM DMA address generator chip (see section 2.2.1.2.1) has a 4 bit Sub-Sample Register (SSR). The SSR controls when the DMA address can increment. Both SSR's are loaded prior to using the BAM. The values loaded into the SSR's can range from 1 to 15. The SSR allows the DMA chips to increment only when the SSR has counted through the number loaded into it. The effect of the SSR on BAM operations depend on wheather the data (image) is being loaded into BAM or is being read from BAM. For example, consider an image being loaded into a BAM with both X and Y axis SSR's loaded with three. In this instance, the input digital video is sent to the BAM (at one pel per clock) and the BAM address will increment every third clock. This means that 2/3rd's of the image is ignored. Every third pel (horizontally) from each third line (vertically) is stored into

BAM. Therefor, loading a BAM with both SSR's=3 causes a 3:1 image reduction vertically and horizontally. As a second example, consider an image being read from BAM with both X and Y axis SSR's loaded with three. In this case, when the image is read, the BAM addresses will increment only every third clock. This means that each pel (horizontally) is duplicated three times and each line (vertically) is duplicated three times. Therefor reading a BAM with both SSR's=3 causes a 1:3 enlargement vertically and horizontally. The range of this image scaling is from 1:1 to 15:1 and works on the entire picture or on a window. Image scaling can be different in each axis.

2.2.1.2.3 BAM MEMORY ARRAY - The BAM Memory Array is 256 x 256 x 16 bits (64K of 16 bit words). The memory chip used for the BAMs is an IMS1400. This component is a 16K x 1 bit static RAM. To completely fabricate a BAM, 64 IMS1400's are needed. Special techniques were used in the wiring of the BAM address lines. Serial resistors of individually selected values were used at the address buffer chips. Each address line is broken up into 'stubs' and distributed through the memory array. To eliminate reflection problems in the 'stubs,' the ends of two adjacent 'stubs' are tied together. See Figure B.5 for a diagram of the addressing techniques in the BAM's.

2.2.1.3 I/O - All I/O is done through Am2950's. This component is an 8 bit bi-directional, I/O port chip which generates handshake signals. Each Processor has two 16 bit I/O ports for interprocessor communication and one 8 bit I/O port for interface to its VMO2. The inter-Processor I/O ports can be operated in either a DMA fashion,

or in a full handshake fashion. The I/O ports between a Processor and a VMD2 operate only in a full handshake mode. Each Processor has dedicated handshake signal lines connected to the other Processors and its VMD2. Communication by another Processor starts by loading of an ID code into a register. The ID code selects the correct dedicated handshake signals to be set and/or monitored during the data transaction.

2.2.1.4 MICRO-ENGINE - The Micro-Engine consists of the CPU group, the address generator, and the microcode memory. Each of these three topics will be discussed separately below.

2.2.1.4.1 CPU GROUP - The CPU group consists of the Am29116 16 bit bipolar microprocessor, a separate shift count register, and a separate double ram address multiplexer. For a full description of the Am29116 see the manufacturers data sheet.

2.2.1.4.2 ADDRESS GENERATOR - The Address generator consists of an Am2910 sequencer with a separate Vector prom, Map register, and Pipeline register. Also in this circuit is the Condition Code multiplexer (CC-Mux). The inputs to the CC-Mux are different for each Processor. Generally, the CC-Mux inputs are signals which are needed by the Program for jumping or looping decisions. An example of a CC-Mux input is an I/O port signal called Next-Data-Available (NDA). This signal indicates when the selected I/O port contains new data (all Processors have the NDA signal to their CC-Mux). For a full description of the Am2910 see the manufacturers data sheet.

2.2.1.4.3 MICROCODE MEMORY - The PROM (Programmable Read-Only Memory) device used for Microcode Memory is the Am27S35. This device is a

1K x 8 bit memory with both synchronous and asynchronous output enables. This device also has an output latch and an initialize register. The size and location of the Microcode Memory varies from Processor to Processor. The lower 2K of Microcode memory always exists on the same board as the Micro-Engine. Additional Microcode memory was added to some of the Processors. The size of the Microcode memory, and the loction of the added Microcode memory is given in the table below. For a full description of the Am27S35 see the manufacturers data sheet.

Proc.	Microcode size	Location of added Microcode
SFV	2K x 88 bits	
SFN	4K x 96 bits	added 2K on extender card
GP	3K x 96 bits	added 2K located on GP-A Board
MLC	3K x 96 bits	added 2K on extender card
SL/TT	2K x 112 bits	
CD	4K x 96 bits	added 2K located on CD-B Board

2.2.1.5 Multiplier Accumulator (MAC) - The MAC used is a TRW TDC-1010J. This is a 16 bit x 16 bit, parallel, 4 quadrant multiplier/accumulator. Additional input/output multiplexing buffers were added to control the data in and out of the MAC. The Programmer loads the MAC X and Y input registers separately, and can read the LSP,MSP,XTP separately. Full control of the MAC control register is also provided. For a full description of the TDC-1010J, see the manufacturers data sheet.

2.2.1.6 INDEX REGISTER - The Index register is a circuit composed of

an Register/Incrementer (S161), an ALU (S181), and output drivers. The Register/Incrementer is loaded from the data bus (DBUS). The Register/Incrementer is one input to the ALU. The second input to the ALU comes directly from the Microcode Literal Field. The ALU is constrained to do only a few of the possible S181 functions. Control for the Index Register comes from three bits in the Microcode word. They select the ALU function and allow the Register/Incrementer to post-increment. The Index register (typically) provides the address for the Scratch Pad RAM, and Look Up Tables. Below is a table showing the available addressing modes. In the table below, 'Literal Field,' is refering to a particular part of the Microword where the Programmer specifies a value (i.e. address). For a description of the various fields used in a Microword see section 4.3.

ADDRESSING MODE	ADDRESS SOURCE
DIRECT	LITERAL FIELD
REGISTER	INCREMENTER
INDEXED	LITERAL FIELD + INCREMENTER
REGISTER POST-INCREMENT	INCREMENTER (POST INC)
INDEXED POST-INCREMENT	LITERAL FIELD + INC (POST INC)
FORCE A ZERO	ZERO CREATED BY ALU

2.2.1.7 CLOCKS - Clock generation and distribution is treated very carefully within the Pipeline Processor Chassis. The main system

clocks are created by the CD Processor. They must be distributed throughout the Chassis with a minimal clock skew between Processors. Also, if a WCS is connected to a Processor, the WCS has the ability to enable/disable the system clocks. The clock generator is fully discussed in section 2.2.7.2. See Figure B.4 for a diagram showing clock distribution. The clock signals are wired between the connectors on the backplane as twisted pairs. The CD is the source of the clocks and is on one end of the Chassis. The SFV is the farthest Processor away from the CD and has terminating resistors on the clock lines to improve signal quality. All boards receive system clocks with an S240 component (Schottky buffer with input hysteresis). The S240 is located near the connector where the clock signals enter the board. The wiring stub between the connector pins and S240 is a twisted pair. Minimizing the length of the stub avoids introducing excessive ringing on the clock lines. The ground wire in the twisted pair is connected to a ground pin near the S240. The boards used in the ASSP are large and many clocked components exist on each board so a second level of clock buffering is provided. The second level of buffering replicates each system clock as many times as needed. The clocks at this level are the clocks that the operating circuits use. All circuits, on all boards, use clocks that are buffered twice from the backplane. The WCS interface for each Processor has a signal which enables/disables the system clocks. Each Processor sends this signal through an open-collector S38 to the backplane. The S38 outputs from all the Processors are wired together and connected to the CD clock generator circuit. See section 2.2.7.2. for a description of the

system clock generator.

2.2.1.8 WCS INTERFACE - Each of the Processors has an interface for a Writeable Control Stores (WCS). The P4, and P5 connectors are used for this interface (See Figure B.2). During checkout, the WCS's were used for Microcode memory and tracing data (i.e. used as a logic analyzer). The interface to the WCS is shown in Figure B.8. (see section 2.2)

2.2.1.9 TEST IMAGE PROMS - One of the images from the algorithm training set was placed into EPROM. During checkout this prom was used in place of the normal video source to provide a known image. The Real-Time hardware/firmware was validated using this image. The SFV Processor has a test image prom. The SFV test image represents the image before streak removal. The CD and SFN Processors have test image proms also. The images in the CD and SFN proms have a representation of the image after the streak removal process.

2.2.1.10 STATUS REGISTER - Each Processor has a Status Register with four (active) bar LED's. The use of the Status Register varies somewhat from Processor to Processor but some things can be said in general. Two of the LED's are used to indicate specific, errors. One is 'stack overflow' in the 2910 Address generator. An LED will light if the 2910 stack is full and an instruction is executed which could push another address onto the stack. The second error indicates a Microcode problem in the BAM. An LED lights if the BAM DMA chip instruction directs the chip to source onto the DBUS and the Microcode DBUS source field does not select the DMA chip. The BAM circuit detects this error and the DMA chip will not source onto the DBUS if this occurs. When either of these errors is detected

the LED will light and a bit is set in a register. This bit remains set until cleared by firmware or a board reset. There is a 4 bit Status Register which the Programmer can access. The Status Register has 4 bits. The first two bits are connected to LED's which can be turned on/off by writing the Register. The third bit resets the Error Register and turns off the error indicating LED's. The fourth bit resets the associated Processor's 'B' board. The Programmer can read the Status Register. During a read the 4 bits of the Status Register are present along with the bits from the error Register. When a reset occurs the Error Register and Status Register are reset and the error LED's will light (lamp test).

2.2.1.11 DATA BUS SOURCE/DESTINATION DECODE - Each Processor has a general purpose, 16 bit, internal data bus (DBUS) used to route data and control parameters on the board. The Microword has a field to select the circuit which sources data onto the DBUS and another field to select the circuit which reads data from the DBUS. (See section 4.2) Generally, the two Microword fields are five bits wide and contain an encoded value (0-31) to select DBUS data source and destination. By having the source/destination encoded, multiple sources onto the DBUS are avoided. The Am29116 Microword control field has a bit which allows it to read the DBUS regardless of what value is in the DBUS destination field. Letting the Am29116 read the DBUS at the same time that another circuit is reading the DBUS, allows the Am29116 to calculate a checksum while passing an image in/out of BAM.

2.2.1.12 HEXADECIMAL DISPLAYS - Each Processor has a Hexadecimal display. This display has two digits for the MLC, SL/TT, GP, and 4

digits for the SFV, SFN, and CD.

2.2.1.13 SCRATCH PAD RAM - Each board (except SFV) has some Scratch Pad RAM. This RAM is used for holding run-time parameters, for temporary buffering of data, and for holding intermediate values during calculations. The size of the Scratch Pad RAM varies from Processor to Processor. Below is a table showing the amount of Scratch Pad RAM per Processor.

PROCESSOR	SCRATCH PAD RAM SIZE
CD	16 K WORDS
SL/TT	256 WORDS
MLC	256 WORDS
GP	256 WORDS
SFN	32 K WORDS
SFV	(0)

2.2.2 SENSOR FORMATTER/VIDEO (SFV)

The Sensor Formatter/Video Processor is referred to throughout this document as the SFV. The SFV is a single board Processor which has it's own clock. This clock operates at a frequency different from the other Processors.The SFV is the ASSP sensor interface. The SFV has no algorithmic duties other than video filtering. The SFV is the source of the digital video picture for the ASSP. The SFV is functionally slaved to the SFN. More specifically, the purpose of the SFV is to receive the real-time digitized picture, extract the 384 x 384 pel 'picture of interest' from the input video, do a 3:1 reduction horizontally, do a line by line average of the current field with the previous field, and respond to commands from the SFN. The final

picture formed by the SFV is a 128 x 384 image. (The streak removal filter in the SFN reduces the image to 128 x 128.) The sections below will discuss the SFV. Refer to the block diagram shown in Figure B.14. A picture of the SFV wirewrap board can be found in Figure B.20.

2.2.2.1 SFV COMMON CIRCUIT MODULES

Refer to section 2.2.1 for a description of the Pipeline Processor Common circuit modules. The SFV uses fewer of the common circuit modules than any other Processor. The SFV does use the following common circuit modules:

WIREWRAP BOARD	(see section 2.2.1.1)
MICRO-ENGINE	(see section 2.2.1.4)
CLOCKS	(see section 2.2.1.7)
WCS INTERFACE	(see section 2.2.1.8)
TEST IMAGE PROMS	(see section 2.2.1.9)
STATUS REGISTER	(see section 2.2.1.10)
HEXADECIMAL DISPLAYS	(see section 2.2.1.12)

The Micro-Engine in the SFV does not support a shift count register or double register addressing for the CPU. This is because of the special clock generated on the SFV. The Test Image Prom on the SFV contains a 128 x 384 image that has not had the streaks removed. This Prom contains the same image data that would have come out of the SFV if the sensor had been looking at the 'test image'. The SFV has two I/O ports which are unique enough to be described separately below.

2.2.2.2 SFV CLOCK GENERATOR

The SFV is the only processor using both system and 'on board'

clocks. A clock generator oscillating at 17.3 Mhz is used by the digitizer (A/D), Blanking Synchronizer, and Data Synchronizer circuits. The significance of this clock period is that the digitizer will digitize 'square' pixels from the 875 line video. To digitize 'square' pixels, the aspect ratio of the sampling period is 4:3. This is the inverse of the 3:4 aspect ratio of the incoming 875 line video. The remainder of the SFV uses clocks which are three times slower than the 17.3 MHZ oscillator. The slower clocks are referred to as Processor clocks.

2.2.2.3 SFV BLANKING SYNCHRONIZER

The A/D Converter and Sync Box (see section 2.1.2) strips horizontal and vertical blanking from the incoming video and sends these signals to the SFV Blanking Synchronizer. These signals are asynchronous to the SFV. The Blanking Synchronizer detects where the leading edge of the active video line is (i.e. the end of the horizontal blank), with respect to the Processor clock. The outcome of this detection process is sent to the Data Synchronizer (see section 2.2.2.4). The horizontal and vertical blank signals are also sent to the 2910 CC-Mux for synchronizing the SFV Microcode program to the incoming video field.

2.2.2.4 SFV DATA SYNCHRONIZER

The Data Synchronizer receives the 8 bit digitized video from the A/D Converter and Sync Box (see section 2.1.2). The digitized data is sent through three successive delaying registers clocked by the 17.3 MHZ clock. The Blanking Synchronizer (see section 2.2.2.3) determines how the incoming video line relates to the Processor clock. The data will be selected from the delaying register having

the correct number of delays to sync the video with the Processor clock.

2.2.2.5 SFV 3:1 HORIZONTAL REDUCTION

The digitized data from the Data Synchronizer (see section 2.2.2.4) goes directly to the 3:1 Horizontal Reduction circuit. This circuit adds three successive 17.3 MHZ 8 bit pixels in real time to form a 10 bit sum. The sum is passed, in real time, to the Field Store Memory (see section 2.2.2.7).

2.2.2.6 SFV FIELD ADDRESS COUNTER

The Field Address Counter is a 16 bit counter used to address the Field Store Memory (see section 2.2.7). This counter can be loaded from DBUS and can be read from DBUS. Control for this counter is a separate bit in the Microword which enables the counter to increment.

2.2.2.7 SFV FIELD STORE MEMORY (FSM)

The Field Store Memory (FSM) is a 48K x 10 bit static ram memory. The data for the FSM is the digital video stream from the 3:1 Horizontal Reduction Circuit (see section 2.2.2.5). The address for the FSM is provided by the Field Address Counter (see section 2.2.2.6). The SFV Microcode controls the FSM address counter and write pulse so that the central 128 x 384 pixels are stored in the FSM. The Programmer can write and read the FSM from the data bus.

2.2.2.8 SFV TWO FIELD ADDER

The Two Field Adder circuit is a simple adder with two 10 bit inputs and an 11 bit output. One input is the digital video stream from the 3:1 Horizontal Reduction Circuit (see section 2.2.2.5). The other input is from the FSM (see section 2.2.2.7). The purpose of

the Two Field Adder is to add, in real time, the pixels from the current field with the corresponding pixels in the previous field.

2.2.2.9 SFV DIVIDE BY SIX

The data source for the Divide By Six is the Two Field Adder circuit (see section 2.2.2.8). The pixels coming into the Divide By Six circuit are six times the value they need to be. The 3:1 Horizontal Reduction Circuit (see section 2.2.2.5) adds three successive pels together, and the Two Field Adder (see section 2.2.2.8) sums together two pixels from the 3:1 Horizontal Reduction Circuit. The Divide By Six circuit divides each pixel by six in real time. Functionally, the Divide By Six circuit is a PROM (Programmable Read Only Memory). The data at each PROM address is 1/6th the value of its address.

2.2.2.10 SFV IMAGE ADDRESS COUNTER

The Image Address Counter (IAC) is a 16 bit counter for addressing the Image Store Memory (see section 2.2.11), and the Test Image Table (see section 2.2.2.12). This counter can be loaded (pre-set) from DBUS, and can be read from DBUS. The increment for this counter can come from either the SFV Microcode, or from the SFN Microcode. The SFV controls the IAC when the Image Store Memory (ISM - see section 2.2.2.11) is being written to. The SFN controls the IAC when the ISM is being read. The SFN commands the SFV to place the increment control of the IAC with the SFN or the SFV. The SFN can only increment the IAC, and cannot load it or read it.

2.2.2.11 SFV IMAGE STORE MEMORY (ISM)

The Image Store Memory (ISM) is a 48K x 8 bit static ram. Data for the ISM comes from the Two Field Average Circuit (see section

2.2.2.8). The address for the ISM is provided by the Image Address Counter (see section 2.2.2.10). To load the ISM, the SFV Microcode program controls the IAC address increment and ISM write pulse. As the digital video stream comes in, the central 128 x 384 pixels are stored into the ISM. The Programmer can write and read the FSM from DBUS.

2.2.2.12 TEST IMAGE TABLE

The Test Image Table is a 64K by 8 bit PROM addressed by the IAC (Image Address Counter; see section 2.2.2.10). Part (48K) of this table is a Test Image PROM (see section 2.2.1.9)

2.2.2.13 SFV I/O PORT

The SFV I/O port is similar to the I/O Common Module described in section 2.2.1.3. The difference is that there is only one port. The I/O port is a 16 bit bi-directional port which is used both synchronously and asynchronously with the SFN. Because the SFV has only one port, there is no port select circuitry. The port handshake signals are connected directly to the CC-MUX. The handshake signals to the SFN go through synchronizing flip-flops to eliminate problems caused by the different clocks on the SFN.

2.2.3 SENSOR FORMATTER/NUMERICAL (SFN)

The Sensor Formatter/Numeric Processor is referred to throughout this document as the SFN. The SFN is a single board processor which completes the sensor image processing which was started by the SFV, and sends that picture to the GP. During acquisition, the SFN performs the VSC algorithm (see section 2.1.3.3 of the main report). The sections below will discuss the SFN hardware. Refer to the block diagram shown in Figure B.15. A picture of the SFV wirewrap boards

can be found in Figure B.21.

2.2.3.1 SFN COMMON CIRCUIT MODULES

Refer to section 2.2.1 for a description of the Pipeline Processor Common circuit modules. The following common circuit modules are used in the SFN.

WIREWRAP BOARD	(see section 2.2.1.1)
BLOCK ACCESS MEMORY (BAM)	(see section 2.2.1.2)
I/O	(see section 2.2.1.3)
MICRO-ENGINE	(see section 2.2.1.4)
MAC	(see section 2.2.1.5)
INDEX REGISTER	(see section 2.2.1.6)
CLOCKS	(see section 2.2.1.7)
WCS INTERFACE	(see section 2.2.1.8)
TEST IMAGE PROMS	(see section 2.2.1.9)
STATUS REGISTER	(see section 2.2.1.10)
DATA BUS SOURCE/DESTINATION	(see section 2.2.1.11)
HEXADECIMAL DISPLAYS	(see section 2.2.1.12)
SCRATCH PAD RAM	(see section 2.2.1.13)

The I/O circuitry in the SFN is similar to the I/O ports described in section 2.2.1.3, except that there is an additional 16 bit port for communicating with the SFV. The address for the Test Image Prom on the SFN comes from the BAM DMA address chips.

2.2.3.2 SFN RECIPROCAL TABLE

A Reciprocal Table is contained in a 2K x 16 bit prom. The table contains the three coefficients of the quadratic equation defining reciprocals. These coefficients are piece-wise linear with respect to the true curve. Obtaining the correct coefficients is achieved

by correctly indexing into the table. The address for the table is generated by the Index register (see section 2.2.1.6).

2.2.3.3 SFN SQUARE ROOT TABLE

A square-root table is contained in a 2K x 16 bit prom. The table contains the three coefficients of the quadratic equation defining square-roots. These coefficients are piece-wise linear with respect to the true curve. Obtaining the correct coefficients is achieved by correctly indexing into the table. The address for the table is generated by the Index register (see section 2.2.1.6).

2.2.3.4 SFN ROW PROCESSOR

The Row Processor is a group of circuits in the SFN which perform the streak removal filter on the SFV image in as little time as possible. The streak removal filter involves taking each pixel in a line and doing a MX+B calculation on it, followed by a power of two scaling, and then adding that pixel to the pixel directly above it in the image. This goes on for three lines. When these three lines have been processed into one line they are returned to the BAM and the next three lines are processed. The Row Processor is where the 128 x 384 image from the SFV is reduced (vertically) to 128 x 128. The sections below discuss the individual circuits in the Row Processor.

2.2.3.4.1 MAC X-REGISTER INPUT MUX

The MAC circuit in the Row Processor is similar to the Common Module discussed in section 2.2.1.5. The difference here is that the MAC X-register has an additional input mux. This mux brings data directly from the Scratch Pad RAM to the MAC X-register without using DBUS. The Programmer can load 16 bit data from the

Scratch Pad RAM into the Mac X-register. The Programmer may also load only the upper byte from Scratch Pad RAM with zero in the lower byte. A third option for the Programmer is to load the lower byte from Scratch Pad RAM into the upper byte of the MAC X-register with zero into the MAC X-register lower byte.

2.2.3.4.2 MAC

The MAC circuit is similar to the Common Module described in section 2.21.5. The difference here is in the additional X-register input mux described in section 2.2.3.4.1, and in the added output data path to the adder which is described in section 2.2.3.4.4. In the Row Processor the MAC Y-register is loaded with the 'M' coefficient in the MX+B calculation and the 'X' values come from the Scratch Pad RAM.

2.2.3.4.3 B-REGISTER

The B-Register is a 16 bit register which is loaded from DBUS and can be read back from DBUS. This register provides the 'B' coefficient in the MX+B calculation.

2.2.3.4.4 ADDER

The Adder in the Row Processor performs the '+' function in the MX+B calculation. One input is the 16 bit output from the MAC MSP register (see section 2.2.3.4.2). The other input to the Adder is the value loaded into the B-register (See section 2.2.3.4.3). The output from the Adder is a 17 bit sum. The sum goes to the scaling headers (see section 2.2.3.4.5) onto DBUS.

2.2.3.4.5 SCALING HEADERS

The Scaling Headers provide a method of scaling the output of the Adder (see section 2.2.3.4.4) by powers of two (i.e shift right or

shift left). The scaling is accomplished in hardware by wiring jumpers to the Header pins. The input to the Scaling Headers is the 17 bit sum from the Adder, and the output of the Scaling Headers is a 16 bit value. The 17 bit input value can be shifted up by five places with zero fill or with one fill. The 17 bit input value can be shifted down by four places with zero fill, one fill, or sign fill. The current mechanization is with no scaling and the sign bit is ignored. The 16 bit output from the Scaling Headers goes to the ALU (see section 2.2.3.4.6)

2.2.3.4.6 ALU

The ALU is the Row Processor is fabricated using S181 ALU's. There are two inputs to the ALU. One input is the 16 bit value from the Scaling Headers (see section 2.2.3.4.5), and the other input is the 16 bit value from the One Line Memory (see section 2.2.3.4.7). The control lines into the ALU are constrained such that only two functions are possible. One function is to pass the Scaling Header value without modification, and the second function is to add both inputs together. The ALU function is determined by a bit in the Microcode word. The output of the ALU goes to the One Line Memory and to the Clip and Divide By Three Prom Circuit (see section 2.2.3.4.9).

2.2.3.4.7 ONE LINE MEMORY

The One Line Memory is made up of four, 256 x 4 RAM memory chips which have separate input and output pins (93422). Together with the One Line Memory Address Generator (see section 2.2.3.4.8) these RAMs function as a 128 word memory for temporary storage of one line of video which is partially processed. The One Line

Memory is inter-connected with the ALU (see section 2.2.3.4.6) in a manner that allows a read-modify-write operation to occur. During one clock cycle, data is read from the Memory into the ALU, modified by the ALU, and written back to Memory at the same address. One bit in the Microcode word enables the write pulse in the One Line Memory. This is the same bit which allows the One Line Memory Address Generator to post-increment.

2.2.3.4.8 ONE LINE MEMORY ADDRESS GENERATOR

The One Line Memory Address Generator is an 8 bit register/counter which is loadable and readable from DBUS. This counter provides the address for the One Line Memory (see section 2.2.3.4.7). One bit in the Microcode word enables the One Line Memory Address Generator to post-increment, this is the same bit which enables the write pulse in the One Line Memory.

2.2.3.4.9 CLIP AND DIVIDE BY THREE PROM CIRCUIT

The Row Processor does an MX+B calculation on each pixel in a line while adding three lines pixel by pixel. At the end of the third line of processing, the final values of each pixel are three times what they should be. As each pixel value in the third video line comes out of the ALU (see section 2.2.3.4.6) these pixels go through the Clip and Divide By Three Prom Circuit. The Clip and Divide By Three Prom is 2K x 8 bits. The input (address) is the upper 12 bits from the ALU. The lower 4 bits from the ALU are ignored. The output is an 8 bit value. The Clip and Divide By Three Prom Circuit does the following function:

- If the ALU data is negative (i.e bit 15 is 1), the output is zero.

- If the ALU data is greater than 765 (decimal), the output is 255.
- If the ALU data is between 765 (decimal) and zero, the output is the input data divided by three.

2.2.4 GLOBAL PROCESSOR (GP) Figure B.16

The Global Processor is referred throughout this document as the GP. The GP is a double board processor with special data reducing hardware for pre-processing data. Several of the algorithms require various convolutional kernals to be convolved over the data acquired and filtered by SFV and SFN. GP performs the General Target to Interference Ratio algorithms (see section 2.1.3 in the main report). The sections which follow describe the GP hardware (see block diagram in figure B.16). A picture of the GP boards can be found in Figure B.22.

2.2.4.1 GP COMMON CIRCUIT MODULES

Refer to section 2.2.1 for a description of the Pipline Processor Common Circuit modules. The following common circuit modules are used in the GP.

WIREWRAP BOARD	(see section 2.2.1.1)
BLOCK ACCESS MEMORY (BAM)	(see section 2.2.1.2)
I/O	(see section 2.2.1.3)
MICRO-ENGINE	(see section 2.2.1.4)
CLOCKS	(see section 2.2.1.7)
WCS INTERFACE	(see section 2.2.1.8)
STATUS REGISTER	(see section 2.2.1.10)
DATA BUS SOURCE/DESTINATION	(see section 2.2.1.11)
HEXADECIMAL DISPLAYS	(see section 2.2.1.12)

The GP has two BAM memories with a mapped addressing scheme. The programmer configures the source and destination of data flow between the memories along with the fast (X) and slow (Y) axis increments through an address mapping register. Data from a BAM can be routed through the Generalized Convolutional Operator (GCO), Look Up Table (LUT), or Merge Table (MT) logic on its way to the other BAM or I/O port. The GCO, LUT, and MT functions are described in sections 2.2.4.2 through 2.2.4.4 respectivly.

2.2.4.2 GENERALIZED CONVOLUTIONAL OPERATOR (GCO)

The GCO is a one or two dimensional filter capable of convolving a BAM source with a selected kernal at BAM access rates. A high speed, bit sliced, carry-save-adder network is used to convolve a 9 element kernal (either 3 by 3 or 1 by 9) in parallel with 9 elements in the data pipline.

2.2.4.3 LOOK UP TABLE (LUT)

DBUS data are used as an address into the selected LUT prom and the contents of the address passed on as the result in a single cycle.

2.2.4.4 MERGE TABLE (MT)

Six bits in the upper and six bits of the lower DBUS data bytes are used as a 12 bit address into a selected Merge Table prom. The 8 bit data found at that address is passed on as the result in a single cycle.

2.2.5 MAXIMUM LIKELIHOOD CLASSIFIER (MLC)

The Maximum Likelihood Classifier Processor is referred to throughout this document as the MLC. The MLC consists of two wirewrap boards. The MLC algorithm is performed in this Processor during Acquisition. The sections below will discuss the MLC hardware. Refer to the block

diagram shown in Figure B.17. A picture of the MLC wirewrap boards can be found in Figure B.23 .

2.2.5.1 MLC COMMON CIRCUIT MODULES

Refer to section 2.2.1 for a description of the Pipeline Processor Common circuit modules. The following common circuit modules are used in the MLC.

WIREWRAP BOARD	(see section 2.2.1.1)
BLOCK ACCESS MEMORY (BAM)	(see section 2.2.1.2)
I/O	(see section 2.2.1.3)
MICRO-ENGINE	(see section 2.2.1.4)
MAC	(see section 2.2.1.5)
INDEX REGISTER	(see section 2.2.1.6)
CLOCKS	(see section 2.2.1.7)
WCS INTERFACE	(see section 2.2.1.8)
TEST IMAGE PROMS	(see section 2.2.1.9)
STATUS REGISTER	(see section 2.2.1.10)
DATA BUS SOURCE/DESTINATION	(see section 2.2.1.11)
HEXADECIMAL DISPLAYS	(see section 2.2.1.12)
SCRATCH PAD RAM	(see section 2.2.1.13)

The MLC has two internal data busses named DBUS and EBUS. The EBUS is used to provide a second data path during the MLC algorithm. Some of The Common Modules listed above have a second output mux for interface to the EBUS. The MLC Microword contains a second Data Bus source for the EBUS source definition. There is one data bus destination field in the MLC microword for both the DBUS and EBUS. The EBUS destinations are the Histogrammer and MAC.

2.2.5.2 MLC RECIPROCAL TABLE

A Reciprocal Table is contained in a 2K x 16 bit prom. The table contains the three coefficients of the quadratic equation defining reciprocals. These coefficients are piece-wise linear with respect to the true curve. Obtaining the correct coefficients is achieved by correctly indexing into the table. The address for the table is generated by the Index register (see section 2.2.1.6). The Reciprocal Table can be read by the DBUS or EBUS.

2.2.5.3 MLC HISTOGRAM

The purpose of the MLC Histogram is to provide statistical information on the Joint-Edge features, occuring in selected two dimensional gradient regions, within the image or sub-image. The sections below will describe the hardware implementation of the Histograms.

2.2.5.3.1 HISTOGRAM CONTROL REGISTER

The Histogram Control Register is a 16 bit register accessable as a source and destination on the DBUS. Five bits from this register are the Histogram ALU control bits (see section 2.2.5.3.4). Another bit disables the two Mapping proms (see section 2.2.5.3.2 and 2.2.5.3.3). If the Mapping proms are disabled, Histogram increment control is selected by a bit in the Histogram Control Register and the Histogram address is in the Index Register. Two other bits in the Histogram Control Register control the most significant address bits into each Mapping prom. With these two bits the Programmer selects between two data sets contained in the Mapping proms.

2.2.5.3.2 ABSOLUTE GRADIENT MAPPING PROM

The Absolute Gradient Mapping Prom is a 512 x 8 bit prom. The

lower seven bits of the Prom address come from the lower seven bits of the EBUS. The most significant address bit comes from the Histogram Control Register (see section 2.2.5.3.1). The purpose of this Prom is to translate the Absolute Gradient information of a pixel into a Histogram address. The Histogram address formed by this Prom is used by all Histograms simultaneously (see section 2.2.5.3.4).

2.2.5.3.3 JOINT-EDGE MAPPING PROM

The Joint-Edge Mapping Prom is a 512 x 8 bit prom. The lower seven bits of the Prom address come from EBUS bits 8-14. The most significant address bit for the Joint-Edge Mapping Prom comes from the Histogram Control Register (see section 2.2.5.3.1). The purpose of this Prom is to decode the Joint-Edge information of a pixel into six discrete output signals. Each signal is connected to a separate Histogram (see section 2.2.5.3.4).

2.2.5.3.4 SIX HISTOGRAMS

The MLC has six Histograms with 12 bit data paths using three 256 x 8 bit RAMS (93422), three ALU's (S181), and a 12 bit output buffer. The ALU function defined by the Histogram Control Register (see section 2.2.5.3.1) is the same for all six ALU's. The address into the Histogram RAMs comes from the Absolute Gradient Mapping Prom (see section 2.2.5.3.2). A separate ALU increment pulse for each Histogram is derived from the Joint-Edge Mapping Prom (see section 2.2.5.3.3). Each of the six Histogram output buffers can be read through the DBUS using its address in DBUS source field (see section 2.2.1.11)

2.2.5.3.5 SUM HISTOGRAM 1 AND 2

The Sum Histogram 1 and 2 circuit is a 12 bit adder. Inputs are from Histogram 1 and 2 RAM outputs. The output of this circuit can be read on the DBUS and goes to the Compare circuit (see section 2.2.5.3.6)

2.2.5.3.6 COMPARE

The Compare circuit is a 12 bit comparator. One input to the comparator is the output of the Sum Histogram 1 and 2 circuit (see section 2.2.5.3.5). The other input to the comes from the EBUS. The MLC algorithm uses the MAC output as EBUS source when the Compare circuit is being used. The Compare circuit output is two signals. which go to the CC-MUX (see section 2.2). These two signals indicate:

(SUM HISTOGRAM 1 AND 2) <= (MAC PRODUCT)

AND (SUM HISTOGRAM 1 AND 2) >= (MAC PRODUCT)

2.2.5.4 MLC PARZEN TABLE

The MLC Parzen table is a 256K WORD look-up table using 32, 2764 EPROMS (8K x 8). There is a 16 bit latch for holding the lower address bits and a 4 bit latch for holding the upper address bits. The circuit wiring supports using 27128 EPROMS (16K x 8) by changing the jumpers on a header. If 27128 Proms are used, the maximum size of the table would be 512K words.

2.2.6 SCENE LOCK - TARGET TRACK (SL/TT)

The Scene Lock Target / Track Processor is referred to throughout this document as the SL/TT. The SL/TT consists of two wirewrap boards. The SL/TT Processor performs the Normalized Product Correllation (NPC) during Track mode. The sections below will discuss the SL/TT hardware. Refer to the block diagram shown in Figure B.18. A picture

of the SL/TT wirewrap boards can be found in Figure B.24.

2.2.6.1 SL/TT COMMON CIRCUIT MODULES

Refer to section 2.2.1 for a description of the Pipeline Processor Common circuit modules. The following common circuit modules are used in the SL/TT.

WIREWRAP BOARD	(see section 2.2.1.1)
BLOCK ACCESS MEMORY (BAM)	(see section 2.2.1.2)
I/O	(see section 2.2.1.3)
MICRO-ENGINE	(see section 2.2.1.4)
MAC	(see section 2.2.1.5)
INDEX REGISTER	(see section 2.2.1.6)
CLOCKS	(see section 2.2.1.7)
WCS INTERFACE	(see section 2.2.1.8)
STATUS REGISTER	(see section 2.2.1.10)
DATA BUS SOURCE/DESTINATION	(see section 2.2.1.11)
HEXADECIMAL DISPLAYS	(see section 2.2.1.12)
SCRATCH PAD RAM	(see section 2.2.1.13)

The SL/TT has two BAM and two MAC Common Circuit Modules (see section 2.2.1.5). One MAC is on board A. The BAMs and second MAC are on board B and have special control circuits. The SL/TT Microcode word has two MAC and two BAM control fields. The SL/TT Microcode proms are distributed on both boards. Board A has Microcode bits 0-79, and Board B has Microcode bits 64-111. Note that Boards A and B both have Microcode bits 64-79 (Data Bus Source/Destination fields).

2.2.6.2 SL/TT RECIPROCAL TABLE (COEFFICIENT)

A Reciprocal Table is contained in a 2K x 16 bit prom. The table

contains the three coefficients of the quadratic equation defining reciprocals. These coefficients are piece-wise linear with respect to the true curve. Obtaining the correct coefficients is achieved by correctly indexing into the table. The address for the table is generated by the Index Register (see section 2.2.1.6). The Reciprocal Table can be read to the DBUS or EBUS.

2.2.6.3 SL/TT SQUARE ROOT TABLE (COEFFICIENT)

A Square-Root table is contained in a 2K x 16 bit prom. The table contains the three coefficients of the quadratic equation defining square-roots. These coefficients are piece-wise linear with respect to the true curve. Obtaining the correct coefficients is achieved by correctly indexing into the table. The address for the table is generated by the Index Register (see section 2.2.1.6).

2.2.6.4 SL/TT LOOK-UP-TABLE

The SL/TT has a 64K Word Look-Up-Table. The address for the Table comes from a 16 bit register which is loaded from DBUS. The Table is fabricated with eight 27128 EPROMS (16K x 8). Part of this table is a combined Reciprocal/Square Root table (see section 2.2.6.5). Also located in the SL/TT Look-Up-Table is a Reciprocal Table (see section 2.2.6.6)

2.2.6.4.1 SL/TT RECIPROCAL/SQUARE ROOT TABLE (STRAIGHT)

Part of the SL/TT Look-Up-Table is a Reciprocal/Square Root Table. This table is a straight look-up-table. If the Programmer desires the Reciprocal/Square Root of a number, that number is used as the address into the table. The data out of the table is the Reciprocal/Square Root.

2.2.6.4.2 SL/TT RECIPROCAL TABLE (STRAIGHT)

Part of the SL/TT Look-Up-Table is a Reciprocal Table. This table is a straight look-up-table. If the Programmer desires the Reciprocal of a number, that number is used as the address into the table. The data out of the table is a 16 bit number. If this 16 bit number were multiplied by the original number the result would be hex 10000.

2.2.6.5 SL/TT BAM CONTROL SELECTOR

The SL/TT BAM Control Selector consists of a six bit BAM Control Register (BCR) and some signal multiplexers. The BCR can be loaded and read from DBUS. Only one bit is active in the BCR. This bit defines which BAM is BAM 1. There are two BAMS on the SL/TT which are labeled, in hardware, BAMA and BAMB. All control signals and flags for both BAMs go through signal multiplexers. The Programming model for SL/TT labels the BAMs as BAM1 and BAM2. During Track mode, context of the BAMs are being swapped continuously. One BAM has the reference image and the other BAM has the current image. When the reference image is obsolete, BAM context is switched and the BAM which was the current image now becomes the reference image. The SL/TT BAM Control Selector allows the Microcode Tracking routines to be written with BAM1 as the reference and BAM2 as the current image. When BAM context needs to be switched, the bit in the BCR is toggled and the track program continues.

2.2.6.6 SL/TT MAC-B CONTROL REGISTER

The MAC-B has a special Control Register circuit. This circuit consists of a 12 bit register followed by a mapping FPLA. The register can be loaded and read from DBUS. The lower 4 bits of the register hold the MAC-B control terms (i.e. Round, Subtract,

Accumulate, 2's Complement). The next 4 bits are MAC-B X-Register Input Select codes. These bits go to the mapping FPLA for decoding into the proper enable signals. There is one additional input to the FPLA. The signal from the BCR (see section 2.2.6.5) which identifies BAM1 is also an input to the FPLA. The FPLA has seven output terms. Four of the output terms are the upper/lower byte enable terms for BAMA/BAMB. The Programmer loads the BAM1/BAM2 choice into the MAC-B Control Register. The FPLA enables either BAMA or BAMB based on the Programmers choice and the state of the BCR bit. The other three terms from the FPLA are select terms which go to the MAC-B, X-Register Input Mux (see section 2.2.6.7).

2.2.6.7 SL/TT MAC-B, X-REGISTER INPUT SELECT MUX

The MAC-B, X-Register Input Select Mux is composed of 4:1 and 2:1 selectors. The 4:1 selectors are for the lower 8 bits, and the 2:1 selectors are for the upper 8 bits. The four choices for the MAC-B, X-Register Inputs are:

UPPER BYTE	LOWER BYTE
ZERO	BAMA
ZERO	BAMB
00	01
DB-UPPER	DB-LOWER

2.2.6.8 SL/TT MASK MEMORY

NOTE: THE SL/TT BOARD WAS WIRED FOR A MASK MEMORY BUT THESE CHIPS WERE NEVER INSTALLED OR CHECKED OUT.

The SL/TT Mask Memory Address generator is identical to the BAM Address Generator (see section 2.2.1.2.1). The SL/TT Mask Memory

Address Generator has a Sub-Sample register Identical to a BAM Sub-Sample Register (see section 2.2.1.2.2). The SL/TT Mask Memory array is quite different from a BAM. The Mask Memory array is configured as a 256 x 256 x 1 bit memory. Data is loaded into the Mask Memory in byte packed format, and read a single bit at a time. This occurs because the lower 3 bits from the X-Address generator are not connected to the Mask Memory chips. The X-Address generator lower 3 bits are connected to an output mux which selects the Mask Memory data one bit at a time.

2.2.6.9 SL/TT ZERO BUFFER

The SL/TT Zero Buffer provides 8 bits of zero to the upper half of DBUS and/or the lower half of DBUS.

2.2.7 CONTROL AND DISPLAY (CD)

The Control and Display is referred to throughout this document as the CD. The CD consists of two wirewrap boards. The CD Processor directs the other Processors through their operational modes. CD interfaces with the peripheral devices described in section 3. The sections below will discuss the CD hardware. Refer to the block diagram shown in Figure B.19. A picture of the CD wirewrap boards can be found in Figure B.25.

2.2.7.1 CD COMMON CIRCUIT MODULES

Refer to section 2.2.1 for a description of the Pipeline Processor Common circuit modules. The following common circuit modules are used in the CD.

WIREWRAP BOARD	(see section 2.2.1.1)
BLOCK ACCESS MEMORY (BAM)	(see section 2.2.1.2)
I/O	(see section 2.2.1.3)

MICRO-ENGINE	(see section 2.2.1.4)
MAC	(see section 2.2.1.5)
CLOCKS	(see section 2.2.1.7)
WCS INTERFACE	(see section 2.2.1.8)
TEST IMAGE PROMS	(see section 2.2.1.9)
STATUS REGISTER	(see section 2.2.1.10)
DATA BUS SOURCE/DESTINATION	(see section 2.2.1.11)
HEXADECIMAL DISPLAYS	(see section 2.2.1.12)
SCRATCH PAD RAM	(see section 2.2.1.13)

The CD Processor has several more interface ports than described in the Common Circuit Module (section 2.2.1.3). All of the CD I/O ports will be described below. CD also is the source of the system clocks. The system clock generator will be described below. CD directs all the Pipeline Processors through their modes of operation. The CD is also the system interface to all peripherals.

2.2.7.2 SYSTEM CLOCK GENERATOR

The system clock generator is located on the CD Processor. The clock generator uses a 20 MHZ crystal oscillator for generation of all system clock signals. There are six phases to the system clocks and all have a period of 200 ns (5 MHZ). Four clocks have a 25% duty cycle and two clocks have a 50% duty cycle. The 25% duty cycle clock phases are non-overlapping. The 50% duty cycle clock phases overlap by 25% (see Figure B.9). The four 25% clock phases are labeled CLOCKA, CLOCKB, CLOCKC, and CLOCKD. The two 50% clock phases are labeled CLOCKE and CLOCKF. All six clock signals can be enabled and disabled by a signal from the backplane. This enable/disable signal originates from a WCS. (see section 2.2.1.8).

There is a seventh clock on the backplane labeled CLOCKW. CLOCKW is a free running clock which is in phase with CLOCKA and is unaffected by the WCS enable/disable signal. CLOCKW is generated for the WCS which needs a free running clock for proper operation. The seven clocks are distributed across the backplane with twisted pairs. Each Processor buffers all clock inputs. The Processor farthest from CD (i.e. the SFV) terminates each clock line (see section 2.2.1.7).

2.2.7.3 CD LOOK-UP-TABLE

The CD Processor has a 64K word look-up-table. It is made up of sixteen (2764) EPROMS. Addressing is through a 16 bit latch. This 64K table has several smaller tables located within it. Each is discussed below.

2.2.7.3.1 TEST IMAGE

There are actually two Test Image tables within the CD look-up-table. One is as described in section 2.2.1.9). The other is a translated version of the original Test Image. The translation is about 10 pixels in the horizontal direction. By switching between the two images a pseudo moving picture was created. This method was used to validate the Tracker mode of operation.

2.2.7.3.2 IRIG TABLES

The CD Processor reads the IRIG Time Code generator and displays it on the IIS graphics unit. To simplify decoding of the IRIG code, a look-up-table is used. This table is part of the CD Look-Up-Table (see section 3.5).

2.2.7.3.3 IIS TABLES

The IIS requires a specific data transfer format containing header and trailer information. The IIS also requires a great deal of initialization. To simplify the IIS data interface the header/trailer formats and initialization parameters are stored in the CD Look-Up Table (see section 3.2).

2.2.7.4 SYSTEM I/O INTERFACES

The CD Processor has interfaces to all of the ASSP system peripherals. Each of the peripheral I/O circuits are described below. The Peripheral devices are described in section 3.

2.2.7.4.1 IIS GRAPHICS DISPLAY UNIT INTERFACE

The IIS Graphics Display Unit interface consists of two uni-directional, 16 bit differential, I/O ports. One port is for CD to IIS, and the other is for IIS to CD. A separate IIS master clear signal to the interface is controled by a bit in the CD Microcode word. There are two handshake signals. One is a command signal from the CD to IIS, and the other is the IIS response. There is a response timer which starts when the CD issues a command to the IIS. If the IIS does not respond within 200 ms, a flag is set in the CC-MUX. The cabling between the CD and the IIS are two 36 strand ribbon cables. In both cables, every other wire is ground (see section 3.2). The IIS displays a multi-image picture array of 256 x 256 pixels. For flight test, four quadrants of 128 x 128 pixels are displayed. The quadrants are used for displaying the incoming video and the subsequent results. The IIS is also displays the IRIG time and the Radar Altitude.

2.2.7.4.2 IRIG TIME CODE GENERATOR INTERFACE

The interface to the IRIG Time Code Generator is a demand/response interface. The CD requests IRIG data, waits for the IRIG data ready flag, then reads the 16 bit encoded data. Cabling to the IRIG is a 36 strand ribbon cable. In this cable every other signal is ground (see section 3.5). The IRIG time is displayed on the IIS and is stored on the Digital Tape Recorder at Acquisition.

2.2.7.4.3 RADAR ALTIMETER INTERFACE

The Radar Altimeter Interface consists of three signals plus three grounds. One signal is a clock and the second signal is a Validity bit. The third signal is an altitude pulse which is true for a specific number of clock periods. The number of clock periods is the altitude. The clock signal is used for incrementing counters. The altitude pulse is used to enable the counters. There are some flip-flops in the interface circuit which look for the trailing edge of the altitude pulse signal. At the trailing edge of the altitude pulse, the count is loaded into a 16 bit register, a flag is set to the CC-MUX, and the counter is cleared. This process is autonomous with regard to the CD Microcode (see sect 3.4). The Validity bit is connected as the MSB in the 16 bit register. The cabling between the Radar Altimeter and the CD is with three twisted pair of wire. The Radar Altitude is displayed on the IIS.

2.2.7.4.4 DIGITAL TAPE RECORDER INTERFACE

The Digital Tape Recorder Interface is a partial implementation of Digital Equipment Corporation's UNIBUS. Cabling between CD and the Digital Tape Recorder consists of two 36 wire ribbon cables soldered to a UNIBUS connector. A standard UNIBUS connects the

UNIBUS connector to the Digital Tape Recorder (see section 3.3). The Digital Tape Recorder is used to store the IRIG time, initial video image, and final results.

2.2.7.4.5 KEYPAD INTERFACE

The KEYPAD interface is on both CD boards. The interface on the CD-A board consists of an 11 bit output latch, a 13 bit input buffer, 5V power, and ground. CD-A reads and writes to the KEYPAD push-buttons. CD-B has an 8 bit input buffer for reading the thumbwheel switches. The interface is through two 36 wire ribbon cables. The Programmer selects either the Keypad or the Thumbwheels for reading. The Keypad is used to reset the Pipeline Processors, set the Pipeline integration level, and to start algorithm processing.

3 PERIPHERAL DEVICES

The sections below briefly discuss each of the ASSP peripherals.

3.1 KEYPAD

The Keypad is a Northrop developed panel which is the operator interface to the ASSP. It consists of 15 pushbuttons, two thumbwheel switches, and 8 DIP switches. Ten of the pushbuttons are the numerals 0-9. Three pushbuttons control keypad enter (ENT), clear (CE), and escape (ESC). The pushbutton labeled RST is the Pipeline Processor reset signal. One pushbutton was not defined. One numeric value is valid at one time. Each pushbutton has a red LED and a yellow LED. The CD Processor can enable any one or all numeric pushbuttons. The yellow LED's light to indicate which numerals the CD Processor has selected as valid. When the operator pushes a valid pushbutton the red LED will light. The selected pushbutton will stay active with the red LED lit until the CD

Processor reads and clears the Keypad. The ENT, CE, ESC and RST pushbuttons are valid at all times. The CE pushbutton will clear the selected numeral. The ENT and ESC are latched signals to the CD Processor CC-Mux. The upper thumbwheel is used to dial in a constant to be used by the ASSP algorithms. The lower thumbwheel is used to indicate the System integration level (see section 5). The 8 bit DIP switches are not used. See Figure B.11

3.2 IIS GRAPHICS DISPLAY UNIT

The IIS Graphics Display Unit is an International Imaging Systems (IIS) Model 70/F Image Computer and Display Terminal. The IIS is a powerful Color Graphics Display computer measuring 26 x 19 x 16 inches. The ASSP uses only a small portion of the IIS capabilities. The ASSP uses the IIS to translate the digital video out of the CD into R-G-B format for display on a R-G-B monitor. The CD also uses the IIS graphics plane to display IRIG time, Radar Altitude, ASSP mode, and Acquisition results. A small box was added to the IIS chassis for summing all three video outputs (i.e RED, GREEN, BLUE) and sync. The sum is a pseudo RS-170 output which can be recorded with a video recorder (see figure B.12).

3.3 DIGITAL TAPE RECORDER

The Digital Tape Recorder consists of two standard 19 inch rack-mount chassis. One chassis is a Wesperline Model WP909 four slot DEC compatible UNIBUS backplane and power supplies. It holds a Western Peripherals Model TC-180 magnetic tape controler/formatter. The TC-180 is a DEC PDP-11 compatible unit which emulates a TM-11/TU-10 tape subsystem. The second chassis is Wesperline Model 2102 S/D Cassette Drive unit. The Cassette Drive unit has two drives which use 3M type, 1/4 inch tape cartridges. The Tape Controller is interfaced to the CD

Processor with a standard UNIBUS cable. See Figure B.10.

3.4 RADAR ALTIMETER

The Radar Altimeter is an APN-209 Radar Altimeter Set which has a range of 1500 feet.

3.5 IRIG TIME CODE GENERATOR

The IRIG Time Code Generator device is a DATUM Model 9300 Time Code Generator/Translator. The 9300 chassis is 1-3/4 inches high and fits in a standard 19 inch equipment rack. In the ASSP flight test the 9300 is used as a Time Code Generator. The standard 1KHZ IRIG-B signal is recorded by the analog video recorders. The CD Processor interfaces to the digital output of the 9300 to read days, hours, minutes, seconds, and milliseconds (See Figure B.13).

4 MICROCODE AND FIRMWARE DEVELOPMENT SOFTWARE

All of the ASSP Pipeline Processors are Microcoded machines. The Microcode for all Pipeline Processors was developed using a Microcode Meta-Assembler (see section 4.1). The Microcode was downloaded into a development station for checkout in each Processor (see section 4.2). The Microword definitions for the Processors are very similar and reflect the usage of Common Circuit Modules in the Processor hardware implementation (see section 2.2.1). The commonality in both hardware and Microword definitions allowed the writing of Macros which were portable to each Processor. These common Microcode Macros are referred to as MACLANG (MACro LANGuage-see section 4.4).

4.1 META-ASSEMBLER

The Microcode Meta-Assembler used for the ASSP Microcode development was written by Microtec Corporation. It allows the Programmer to define the target machine, and create 'Assembly Language' instructions for all

machine operations. The Meta-Assembler also allows creation of multi-instruction Macros with variable input parameters. This feature was used extensibly in all phases of microcode developement. Almost every line of code, in the Assembly Source Code, in every Processor is a Macro call. The selection of process related macro names created source programs which are easily understood and, therefore, readily debugged. A number of the more useful Macros are contained in Maclang (see section 4.4) and are portable between Processors.

4.2 MICROCODE DEVELOPMENT STATION (WCS-WRITEABLE CONTROL STORE)

The Microcode development station used in developing the ASSP Microcode is the HiLevel Technology Model DS270. The DS270 is a Writeable Control Store (WCS), combined with a Logic Analyzer. The WCS is a fast RAM memory which simulates Microcode Proms. The WCS allows the Programmer to alter microcode during checkout as many times as needed until the code is right. The Meta-Assembler object code is downloaded into the WCS at the beginning of the checkout session. The Programmer can then exercise the Processor via commands to the WCS and use the Logic Analyzer to monitor specific test points and data in the Processor. (See section 2.2.1.8)

4.3 MICROWORD DESCRIPTIONS

The following sections give the Microword layout descriptions for each Processor.

4.3.1 SFV MICROWORD FORMAT

The SFV Microword format is shown in Figure B.26

4.3.2 SFN MICROWORD FORMAT

The SFN Microword format is shown in Figure B.27

4.3.3 GP MICROWORD FORMAT

The GP Microword format is shown in Figure B.28

4.3.4 MLC MICROWORD FORMAT

The MLC Microword format is shown in Figure B.29

4.3.5 SL/TT MICROWORD FORMAT

The SL/TT Microword format is shown in Figure B.30

4.3.6 CD MICROWORD FORMAT

The CD Microword format is shown in Figure B.31

4.4 MACLANG (MACRO-LANGUAGE)

MACLANG is a collection of Microcode Macros and routines which are portable between all ASSP Processors. These routines and Macros can be divided into the categories of Program Control, Arithmetic Operations, Program Branching Primitives, and Read/ Write - Import/Export Macros. Each category is described in the sections below.

4.4.1 MACLANG PROGRAM CONTROL

The MACLANG Program Control Macros are intended to provide 'High-Level Language,' program control capabilities. MACLANG contains Macros which can accomplish IF-THEN-ELSE, DO loops, DOWHILE loops, and DOUNTIL loops. MACLANG also defines GOTO, CALL, and RETURN. The memory cells used for loop count and test values can be 29116 registers, Scratch RAM address, or an element of an array in Scratch RAM.

4.4.2 MACLANG ARITHMETIC OPERATIONS

The MACLANG Arithmetic Operations define a variety of useful arithmetic, logical, and data movement operators. The operators can be used for single precision and double precision values. Some of the operators available are: Increment, Decrement, Move, Complement, Negate, Add, Subtract, AND, OR, Multiply, Rotate, and Shift. The

location of the operands and destinations of the result can be 29116 registers, Scratch RAM address, or an element of an array in Scratch RAM. Double Precision Values must be adjacent memory cells.

4.4.3 MACLANG PROGRAM BRANCHING PRIMITIVES

The MACLANG Program Branching Primitives are used to compare two operands and then branch to the indicated label if the specified relation (between operands), is true. The relations available are: Less Than, Less Than or Equal To, Equal, Greater Than or Equal to, and Greater Than. The operands can be double precision values. The operands can be 29116 registers, Scratch RAM address, or an element of an array in Scratch RAM. Double Precision Values must be adjacent memory cells.

4.4.4 MACLANG READ/WRITE AND IMPORT/EXPORT MACROS

The MACLANG Read/Write and Import/Export Macros provide inter-processor data communication. These Macros execute a very sophisticated I/O protocol which does a great deal of error check and retry subroutines. The Read/Write Macros are used for transferring a data list between Processors, and for doing I/O with a VMO2. The Import/Export Macros are used to pass a 128 x 128 image between Processors. All data transfers in Read/Write use full handshake. In the Import/Export Macros, data are sent synchronously in a DMA-like fashion, and I/O control and error check parameters are sent asynchronously with full handshake.

5 PROCESSOR CHECKOUT, SYSTEM INTEGRATION, AND TEST

The task of bringing the ASSP Real-Time hardware up from initial Processor checkout through verification of the algorithms was done in a predetermined manor. First, the circuitry on each Processor was verified

separately. Second, the Operational Microcode Firmware was installed and verified (as much as possible) in a stand alone situation. Third, the Processors were integrated with test programs in the VMD2s. Fourth, the operational VMD2 programs were installed and checked out. To facilitate the first and second step, two Processor work stations were established. Each work station had power supplies, system clock generator, oscilloscopes, logic analyzers, a WCS (see section 4.2), and computer terminal. One workstation was the Pipeline Processor Chassis (see section 2.1), and the second workstation was separate. The third and fourth steps were done using the Pipeline Processor Chassis. Each of these four steps will be discussed in more detail in the sections below.

5.1 PROCESSOR CHECKOUT

The Pipeline Processors were each checked individually before being integrated in with the rest of the ASSP system. The board level checkout was done using oscilloscopes, logic analyzers, and the Writeable Control Stores (WCS-see section 4.2). When a Processor was ready for checkout the circuits (i.e. I.C.'s) were installed one function at a time. Some of these functions are described in 'Common Circuit Modules' (see section 2.2.1), and some functions are unique to one Processor. Each of the modules was checked out electrically and with a Microcode test program. For checkout of first Processor, all Microcode test programs were uniquely written. For the checkout of all subsequent Processors, the test programs written for a 'Common Circuit Module' could be re-used. When a circuit unique to one Processor was added, a new test program was written. As each circuit was installed into a Processor and checked out, the Microcode test programs were concatenated into one program to form a Microcode diagnostic program for

that Processor. There are six Microcode diagnostic programs. These diagnostics are comprehensive and can be used to detect a hardware fault in a Processor. These programs check the 'internal' operations of a Processor and do not verify the I/O ports between the Processors or the VMD2 I/O port. To verify the I/O hardware, the Import/Export macros from MACLANG (see section 4.4.4) were used in a test program which continually passed a data list between two processors. When all circuits had been installed into a Processor and the diagnostic was running successfully, the Processor was ready for Operational Microcode Integration/Checkout (see section 5.2)

5.2 OPERATIONAL MICROCODE INTEGRATION/CHECKOUT

The operational microcode was installed and checked out in two steps. In the first step, a Processor was set up as an independent, stand-alone machine in one of the two workstations. In the second step, the Processors were integrated one-by-one in the Chassis with test programs running in the VMD2s. These steps are discussed in more detail in the following two sections.

5.2.1 STAND-ALONE INTEGRATION

To begin operational microcode integration, each Processor was set up as an independent, stand-alone, machine with the operational program loaded into the WCS memory. While in this configuration as much of the operational program as possible was verified. The operational microcode for the CD to System Peripherals (see section 3) was checked out while the CD was in stand-alone mode. The operational Algorithms were checked out (somewhat) while in stand-alone mode. The Test Image Proms (see section 2.2.1.9) were used in this mode as the source of the video image. The SFV operational Program was almost completely

checked out while the SFV was in stand-alone mode due to the autonomous nature of the SFV. Much of the operational microcode can be checked out in stand-alone mode, but each Processor must be integrated into the System before the operational programs can be fully checked out (see section 5.2.2)

5.2.2 ONE-BY-ONE INTEGRATION

After a Processor has gone through Stand-Alone integration (see section 5.2.1), the Processors are integrated one-by-one. The full operational microcode requires that all Processors be present and functional for everything to work. In order to integrate the Processors one-by-one and still maintain a resemblance of the actual operational Program data flow, the integration must occur in a specific order. Each Processor is told which Processors are present, this causes the I/O routines to select an alternate Processor if the intended Processor is not present. All I/O operations are performed or simulated. The code in each Processor assumes that the Processors integrated previously are present, and that any I/O routine branching concerns only the Processors not yet integrated. The order of integration is with CD first, then GP, then MLC, then SLTT, and then the SFV/SFN. The Operator must indicate which Processors are present, this is the 'System Integration Level'. The Operator indicates the 'System Integration Level' by dialing a numeral on the lower thumbwheel switch located on the Keypad (see section 3.1). The relation between the thumbwheel numeral and the Processors present is given below.

THUMBWHEEL NUMERAL	SYSTEM INTEGRATION LEVEL

5	CD
4	CD-GP
3	CD-GP-MLC
2	CD-GP-MLC-SLTT
0	CD-GP-MLC-SLTT-SFV/SFN (ALL PROCESSORS)

The Test Image Proms (see section 2.2.1.9) are used as the video source during this integration/checkout. The Test Image Proms provide a (numerically) known image for input into the algorithms. This same Test Image was used as an image source in the computer simulation. The Algorithms in the Processor operational microcode were verified against the results of the computer simulation. During the Processor one-by-one integration there are test programs in the VMO2s. These VMO2 test programs provide the correct I/O interaction with each Pipeline Processors. The data being sent to the Processors is provided empirically in the VMO2 test programs and the data being read from the Pipeline Processors is displayed on the VMO2 screens. The VMO2 test programs operated independently of each other and moding is accomplished by manually directing each VMO2 via a terminal.

5.3 OPERATIONAL VMO2 FORTRAN INTEGRATION/CHECKOUT

When the One-By-One Pipeline Processor integration was complete, the Operational VMO2 Fortran programs were loaded into the VMO2s. The Test Image Proms (see section 2.2.1.9) were used again as the video source during this integration/checkout. The Test Image Proms provide a (numerically) known image for input into the Pipeline Processors. The Algorithms in the Operational VMO2 programs were verified against the results of the computer simulation.

5.3 BUILT IN TEST CAPABILITIES

The built in test Capabilities of the ASSP provide a limited Processor level test, an End-To-End operational test, and a Fault Location test. Each of these tests are described below.

5.3.1 PROCESSOR LEVEL (LIMITED) BUILT IN TEST

The Processor level tests are performed by all Processors at reset. The Processor level tests consist of an exhaustive test of all BAMs (see section 2.2.1.2.3). This BAM test is the test developed for checking out the BAM initially (see section 5.1). The BAM test also exercises the Micro-engine (see section 2.2.1.2.4) and Data Bus Source/Destination circuits (see section 2.2.1.2.11). Both BAMs are checked in the SL/TT and GP.

5.3.2 END-TO-END BUILT IN TEST

The End-To-End Operational, built in test consists of selecting the Test Image as the video source, running the Processors, and observing the results on the monitor. If the results are not correct, an error exists.

5.3.3 FAULT LOCATION BUILT IN TEST

The Fault Location built in test consists of selecting the Test Image as a video source, selecting a System Integration Level (see section 5.2.2), running the Process, and interpreting the results displayed on the monitor. The System Integration Level is initially set to 5 (i.e only the CD is in the System). If the results are positive, then the System Integration Level is decreased (i.e another Processor is added to the System) and the test repeated. When the results indicate a failure, the last Processor added is where the problem is. This technique requires some knowledgable interpretation of the results

displayed on the monitor. This test utilizes the I/O branching originally installed to allow the Processor One-By-One integration (see section 5.2.2). The Operational VMO2 programs do not have the comparable branching capabilities. The Operational VMO2 programs assume all Processors and all VMO2s are present. This means that the VMO2 Operational programs will not complete, and the ASSP System will not complete operation, with the System Integration Level set to anything other than zero (all Processors). By careful interpretation of the monitor display, correllated with the System Integration Level, the location of a failed Processor can be determined.

THE END

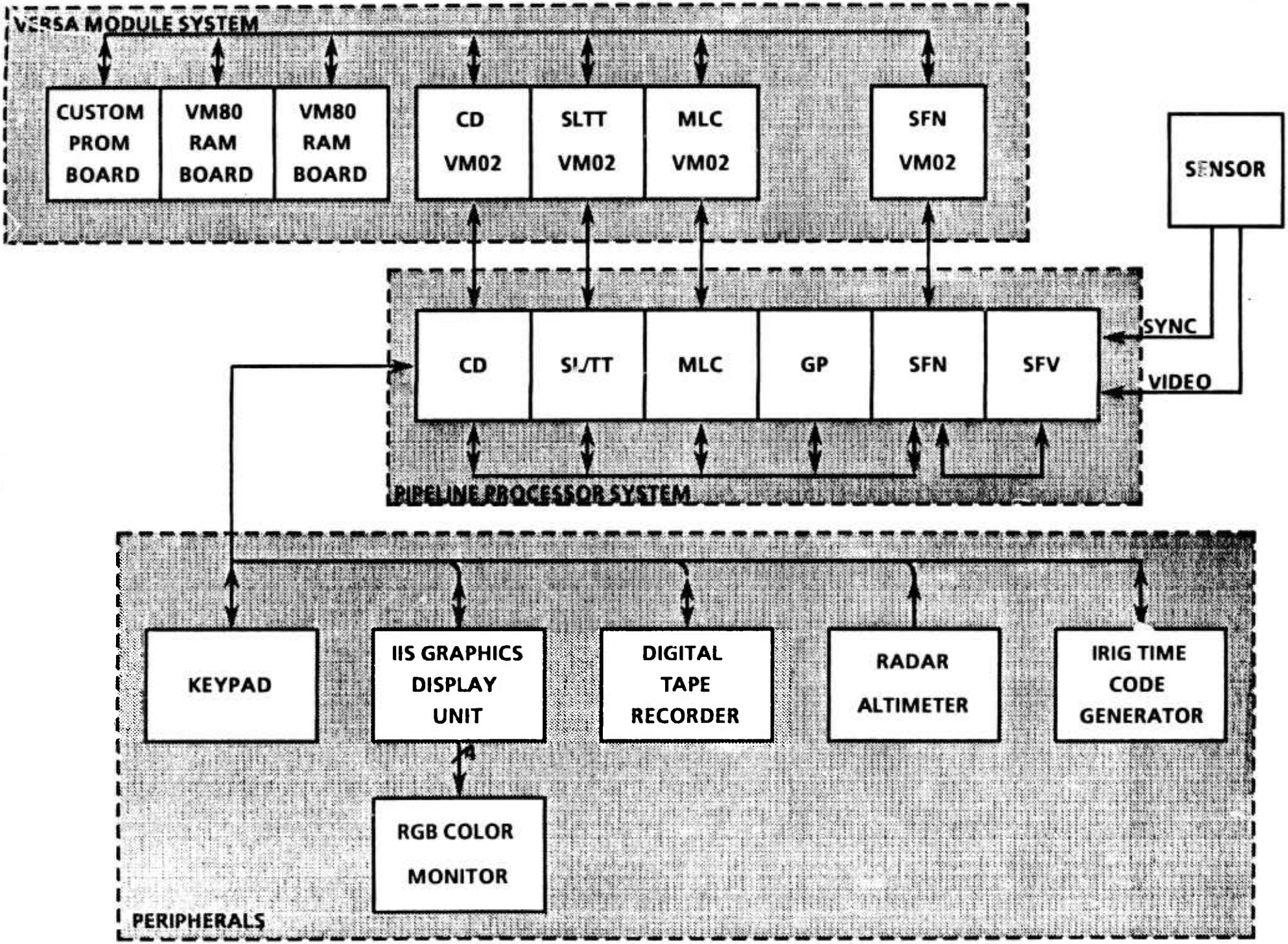


Figure B.1. ASSP Real-Time Hardware, System Block Diagram.

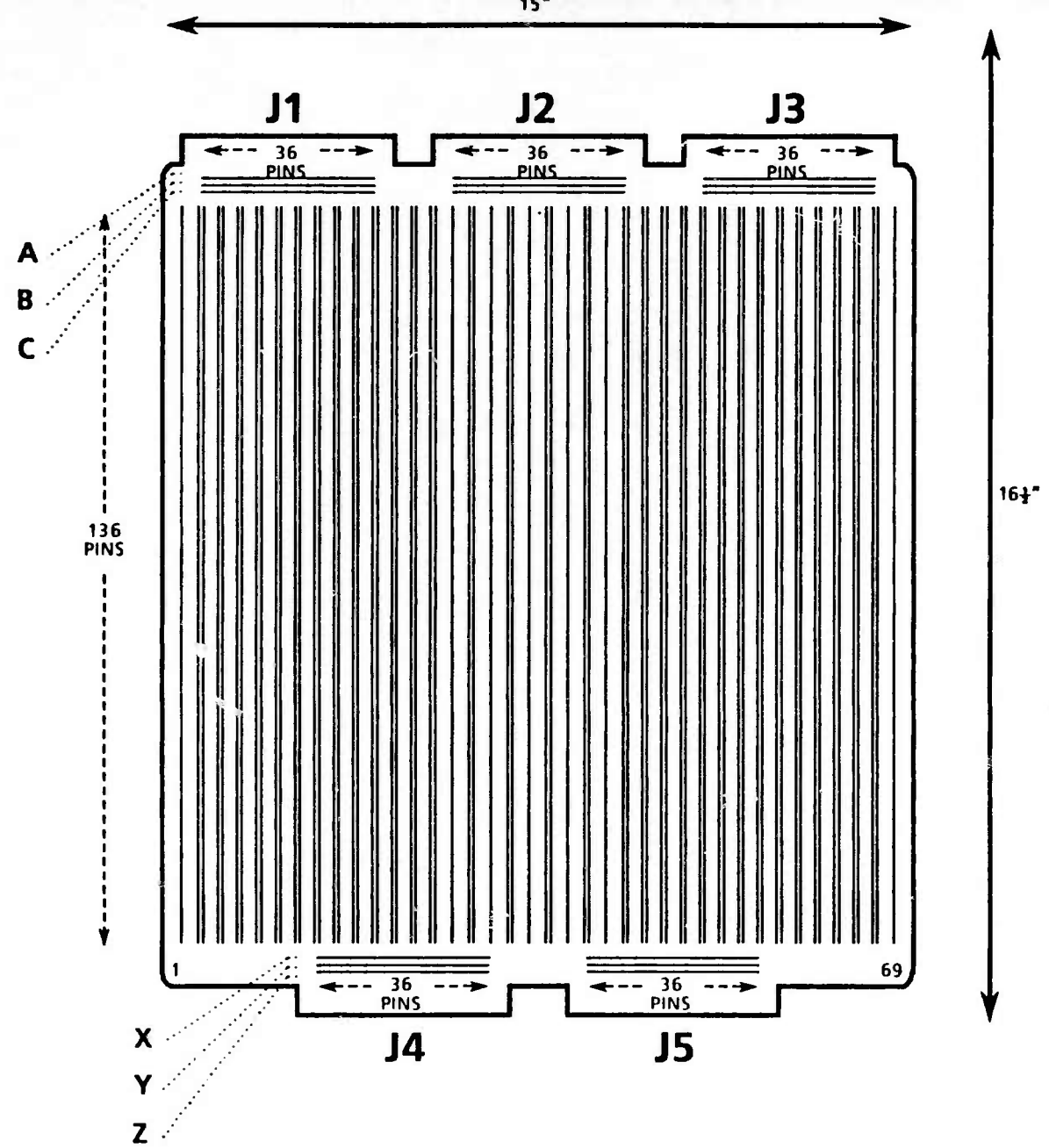


Figure B.2. Wirewrap Board.

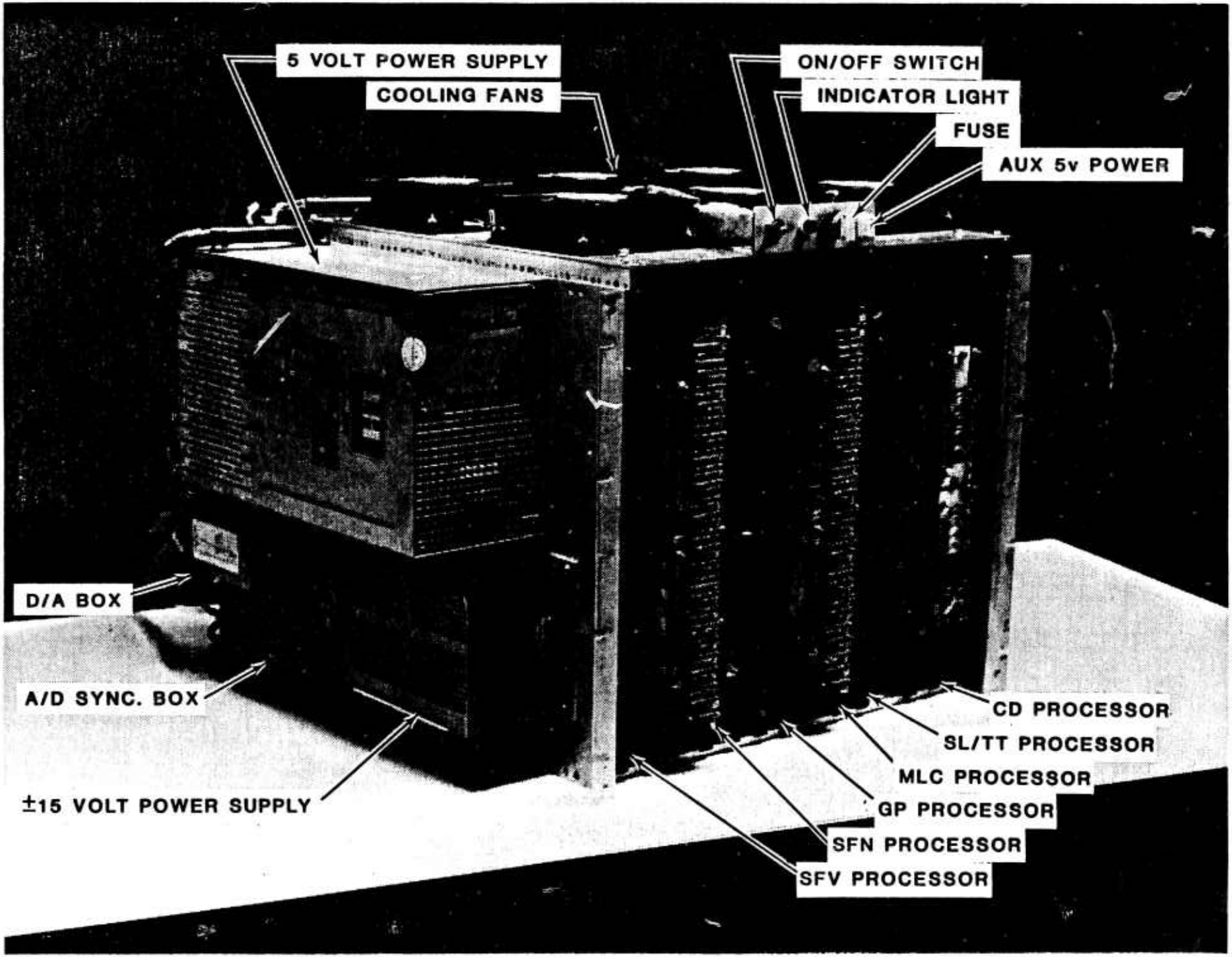
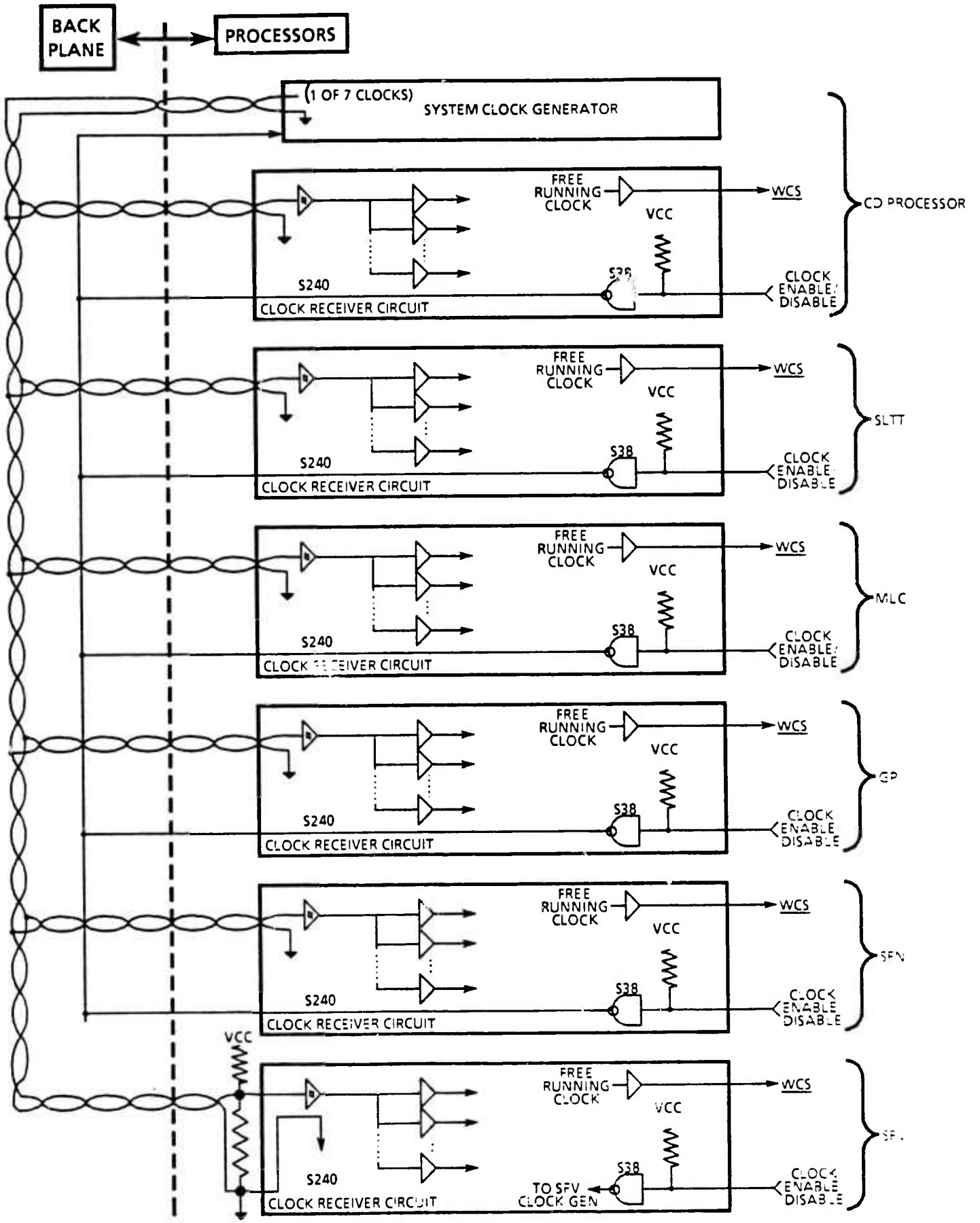


Figure B.3. Pipeline Processor Chassis, Chassis Mounted Items.



NOTE: ONLY ONE CLOCK SIGNAL IS SHOWN. THERE ARE SEVEN CLOCK SIGNALS IN THE PIPELINE PROCESSORS

Figure B.4. System Clock Distribution.

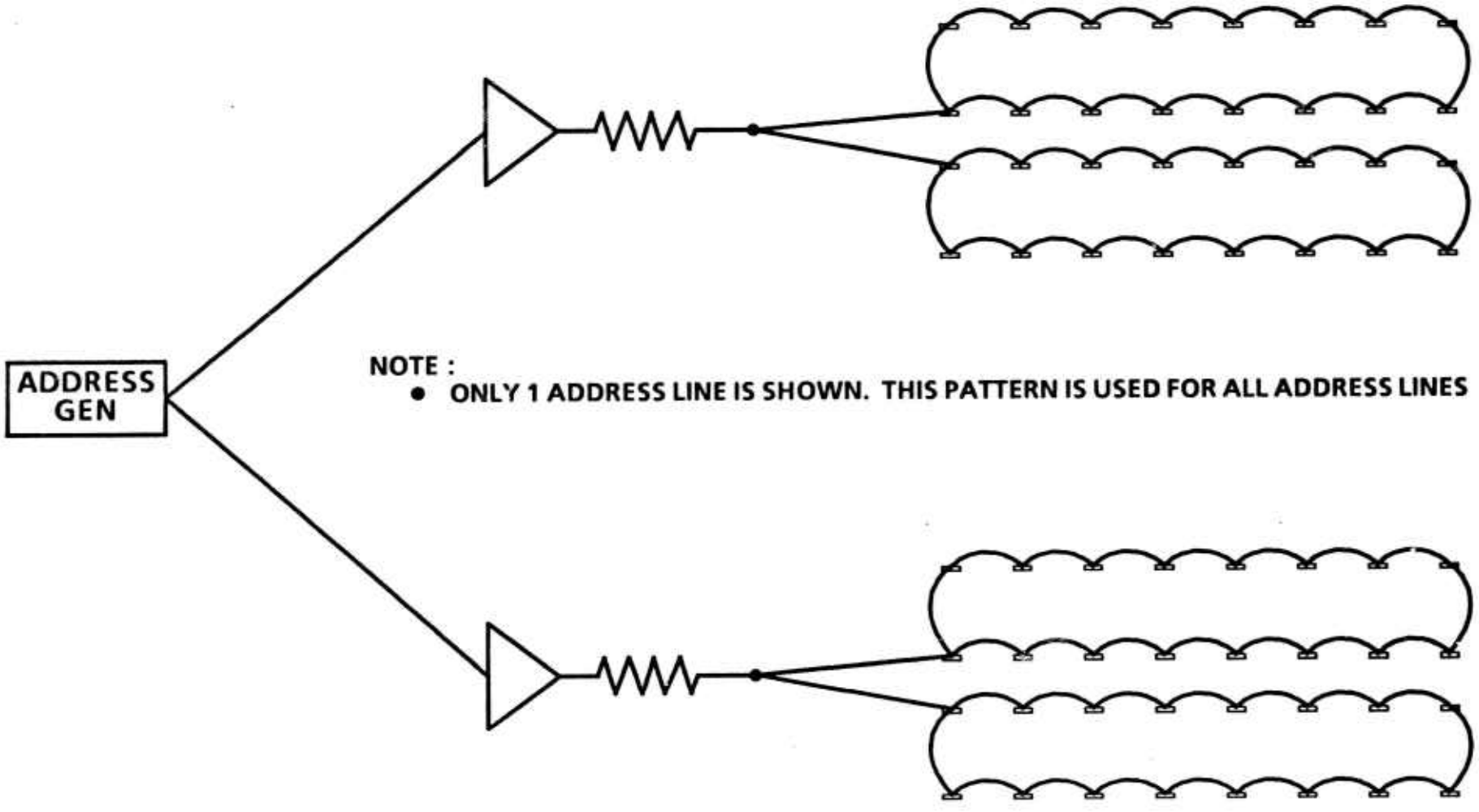


Figure B.5. BAM Block Diagram and Address Wiring Techniques.

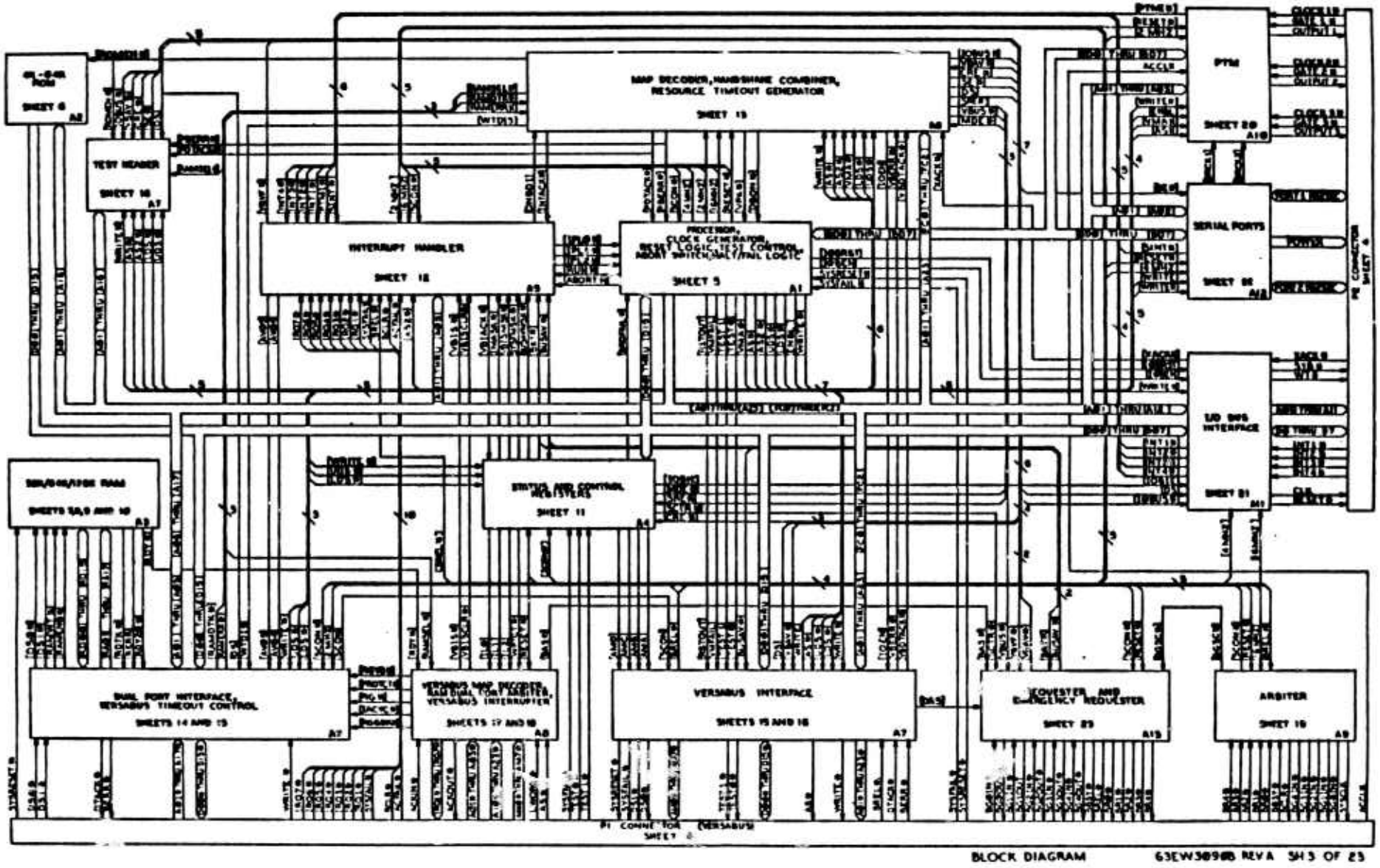


Figure B.6-a. M68KVM02 Mono-Board Computer Block Diagram.

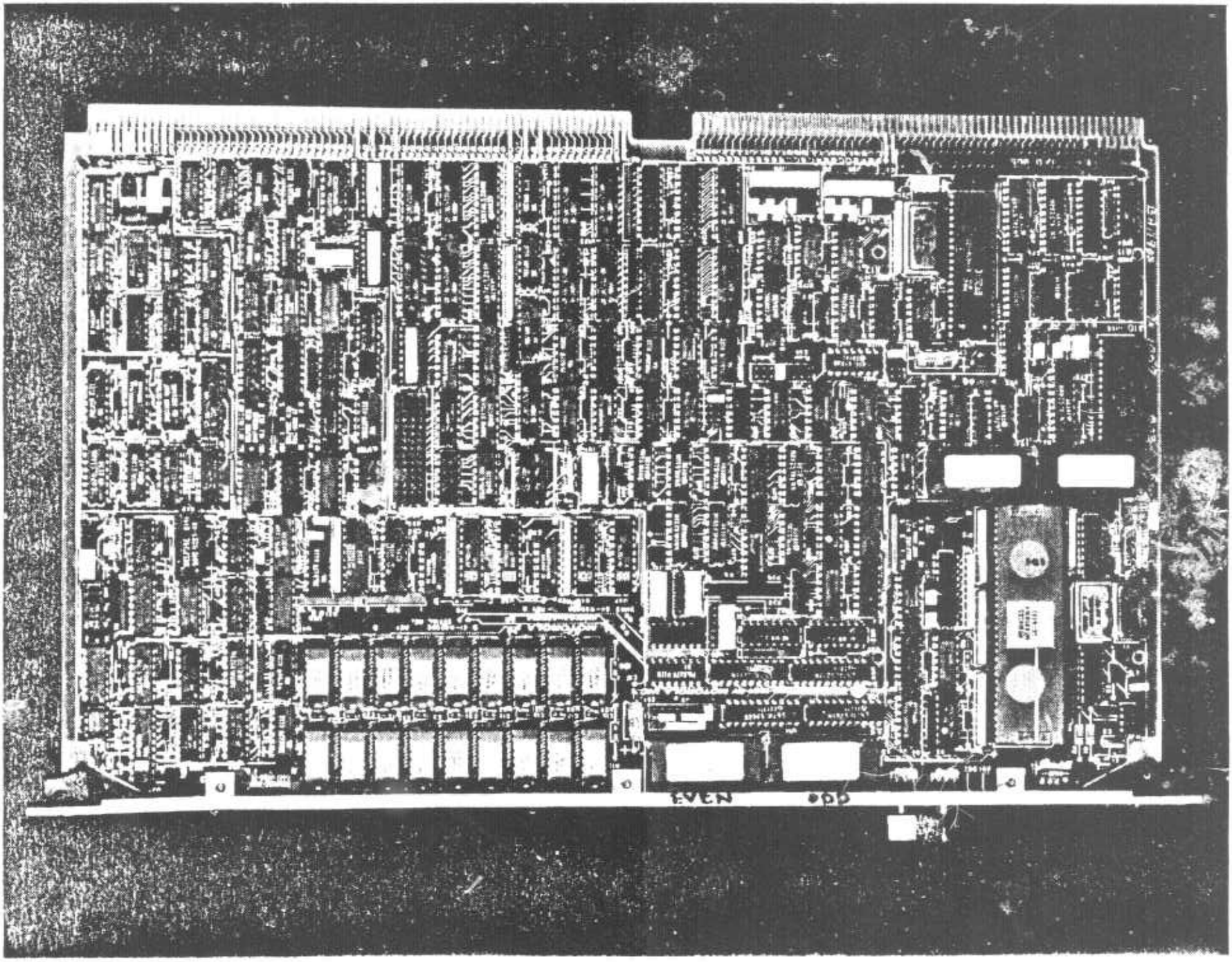


Figure B.6-b. M68KVM02 Mono-Board Computer Block Diagram.

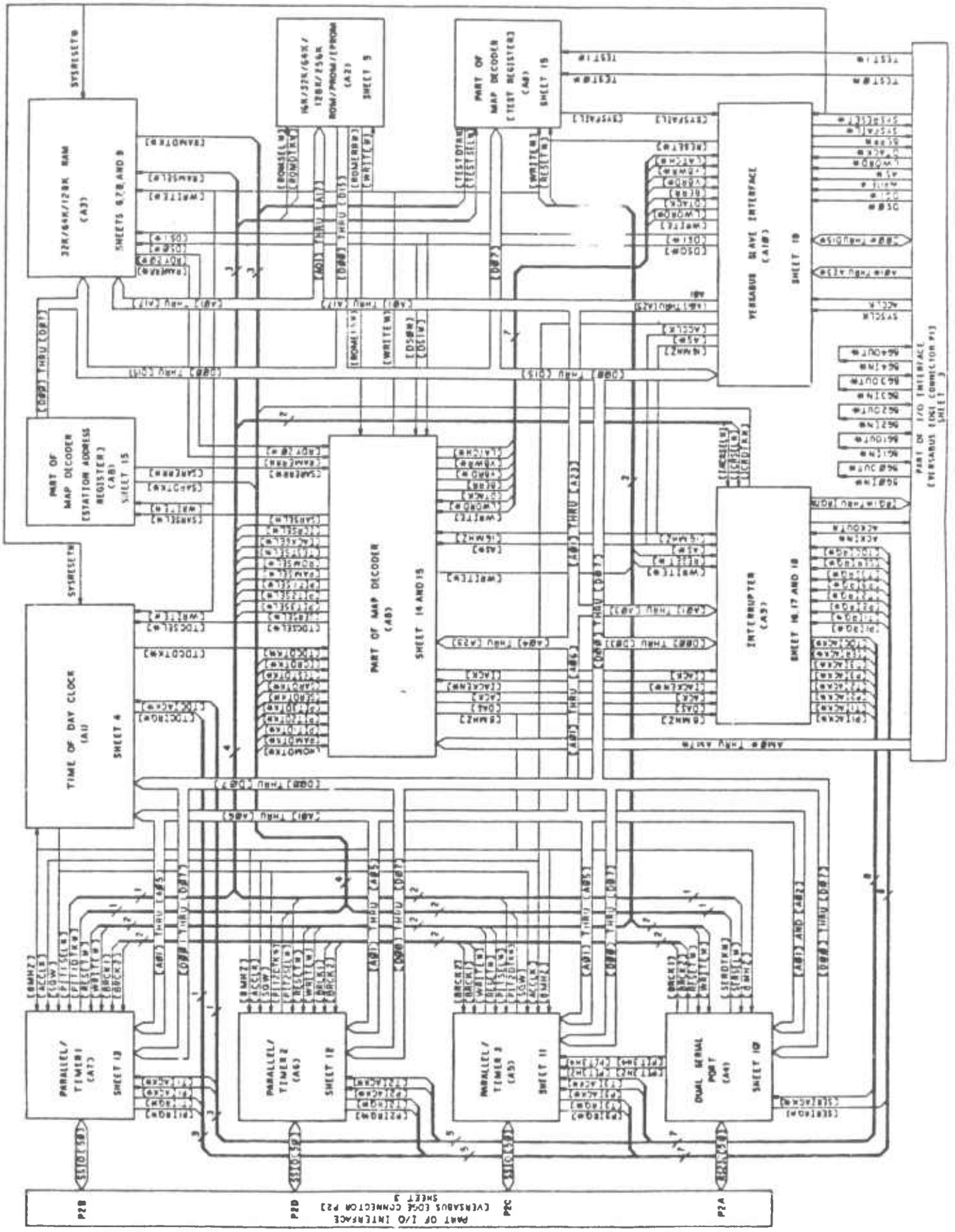


Figure B.7-a. M68KVM80 Memory Board.

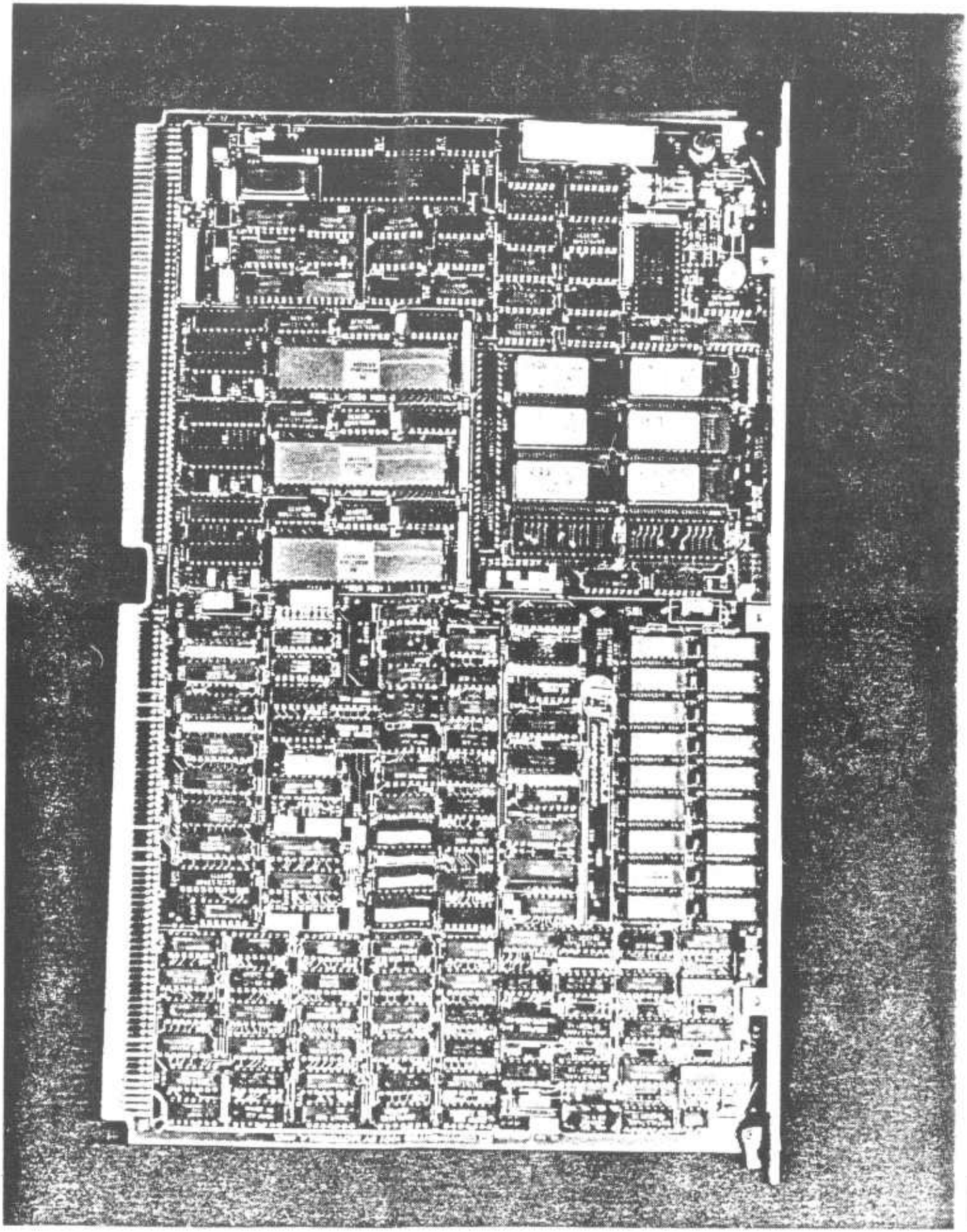


Figure B.7-b MARVMBO Memory Board

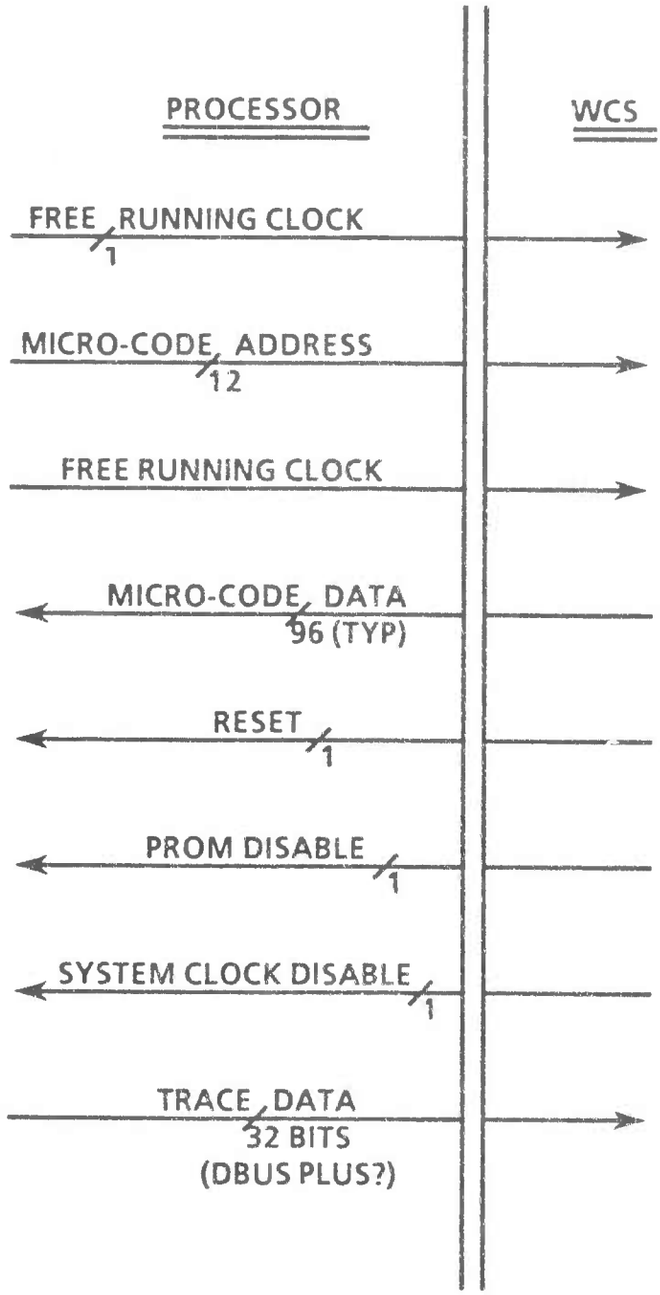


Figure B.8. Writeable Control Store (WCS) Interface.

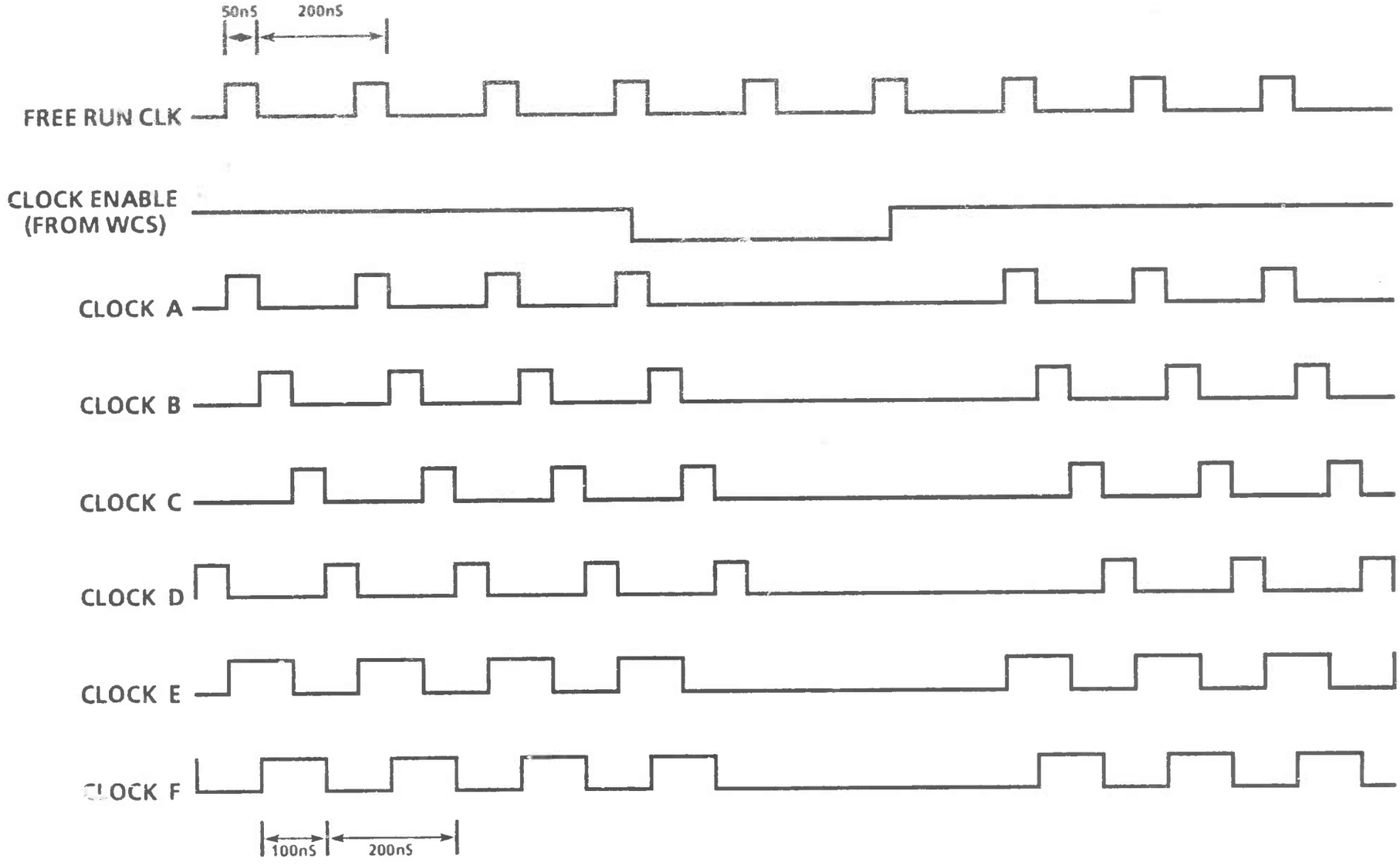


Figure B.9. System Clock Generator.

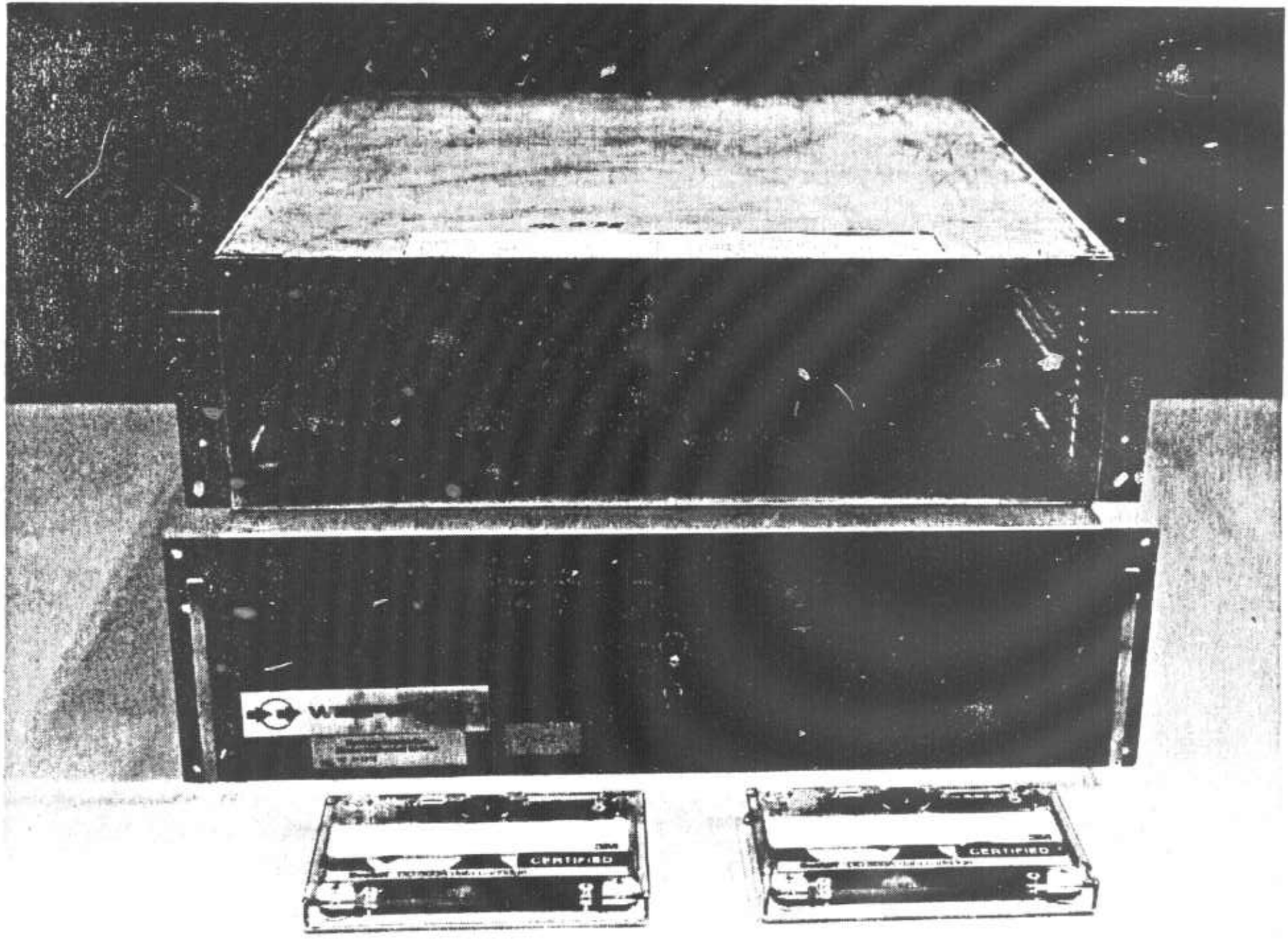


Figure B.10. Digital Tape Recorder.

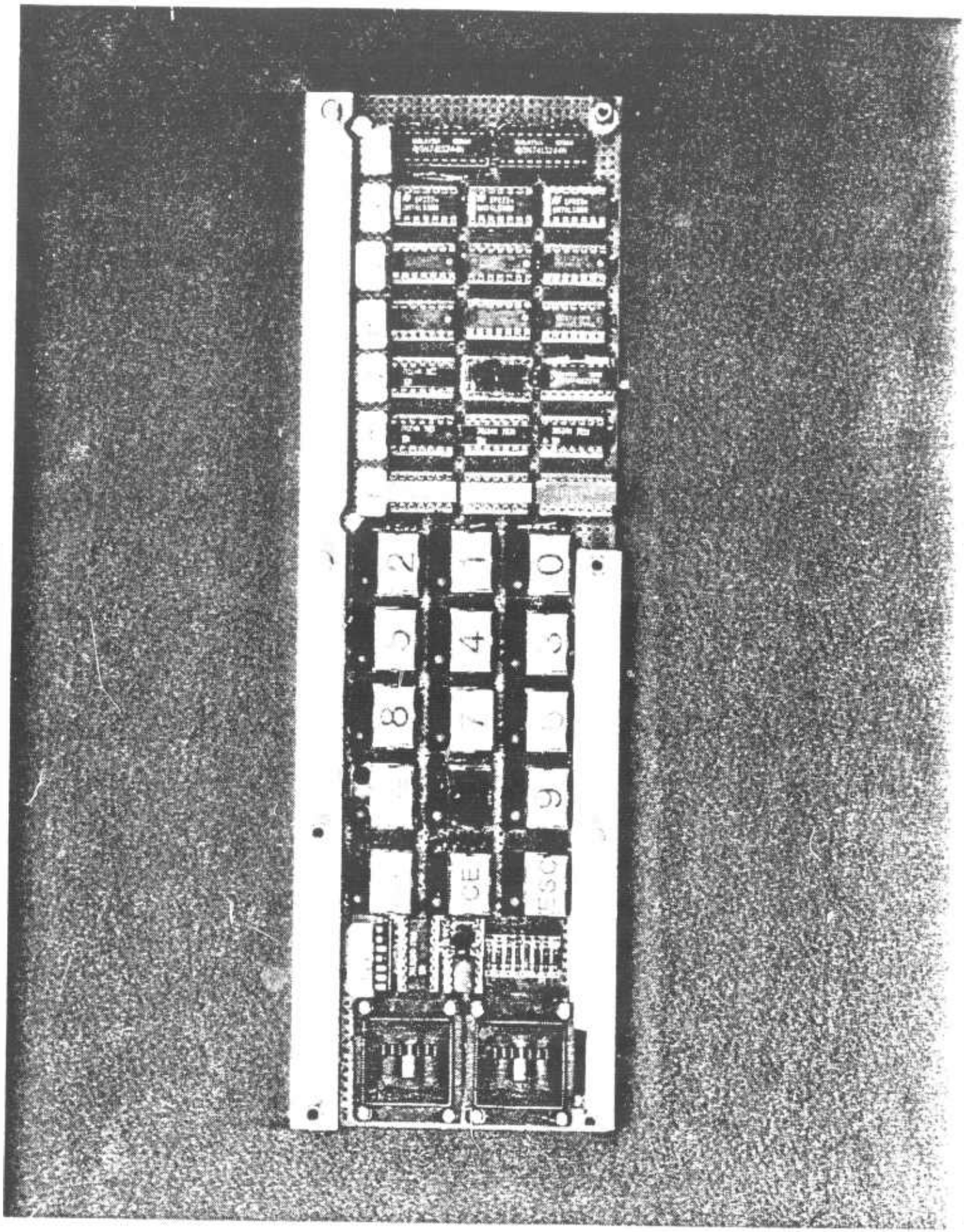


Figure B.11. Keypad.

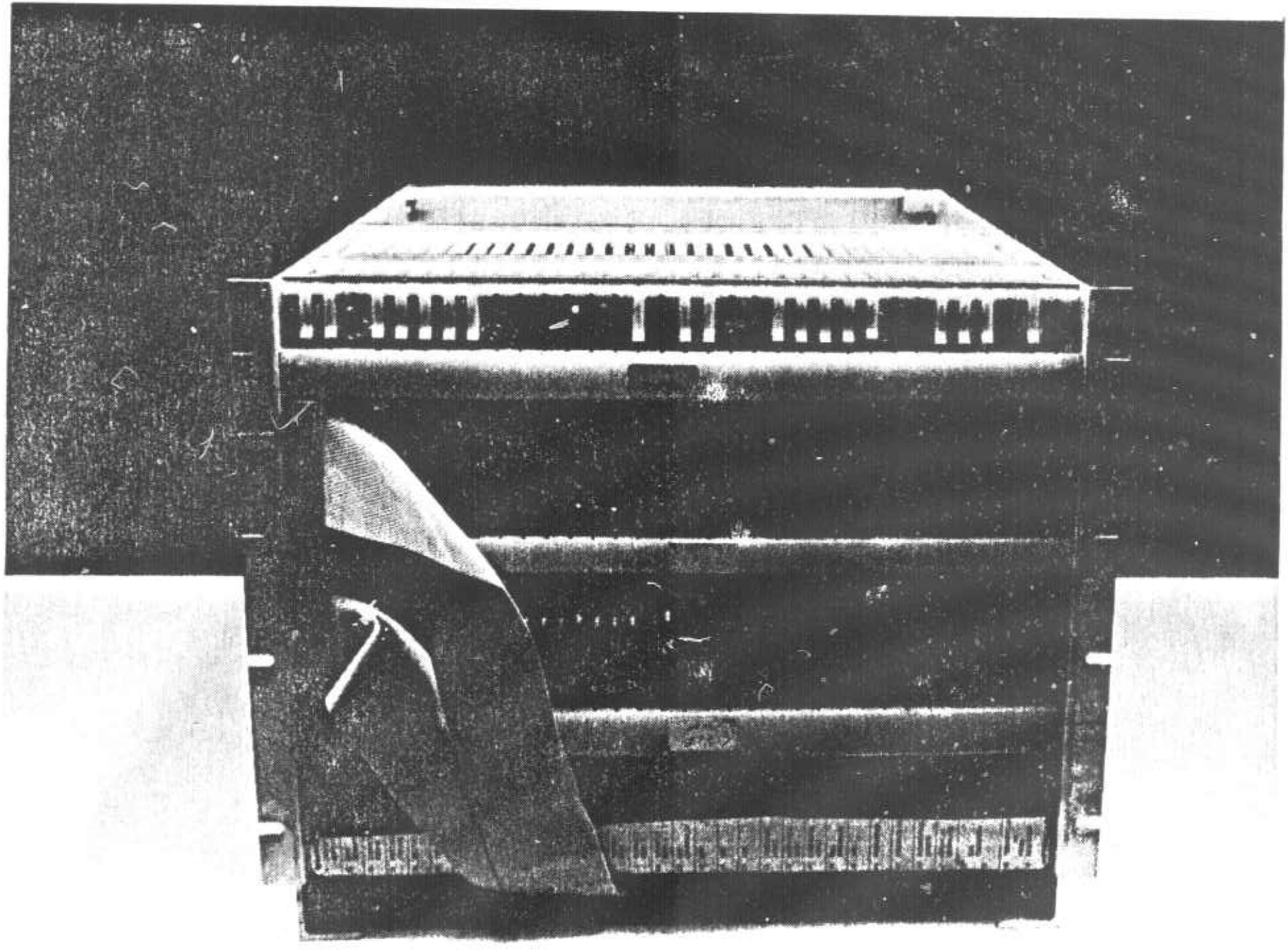


Figure B.12. IIS Graphics Display Unit.

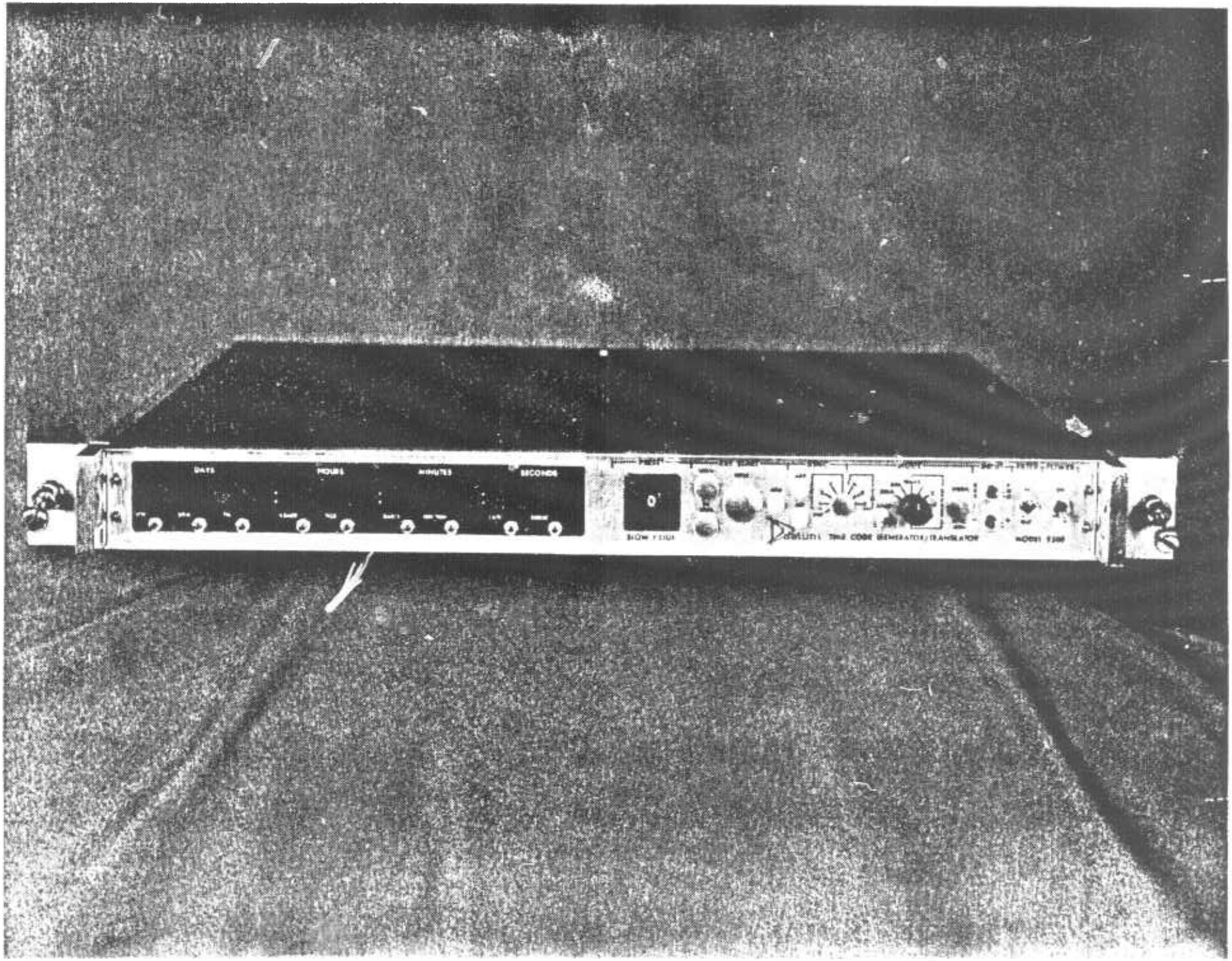


Figure B.13. IRIG Time Code Generator.

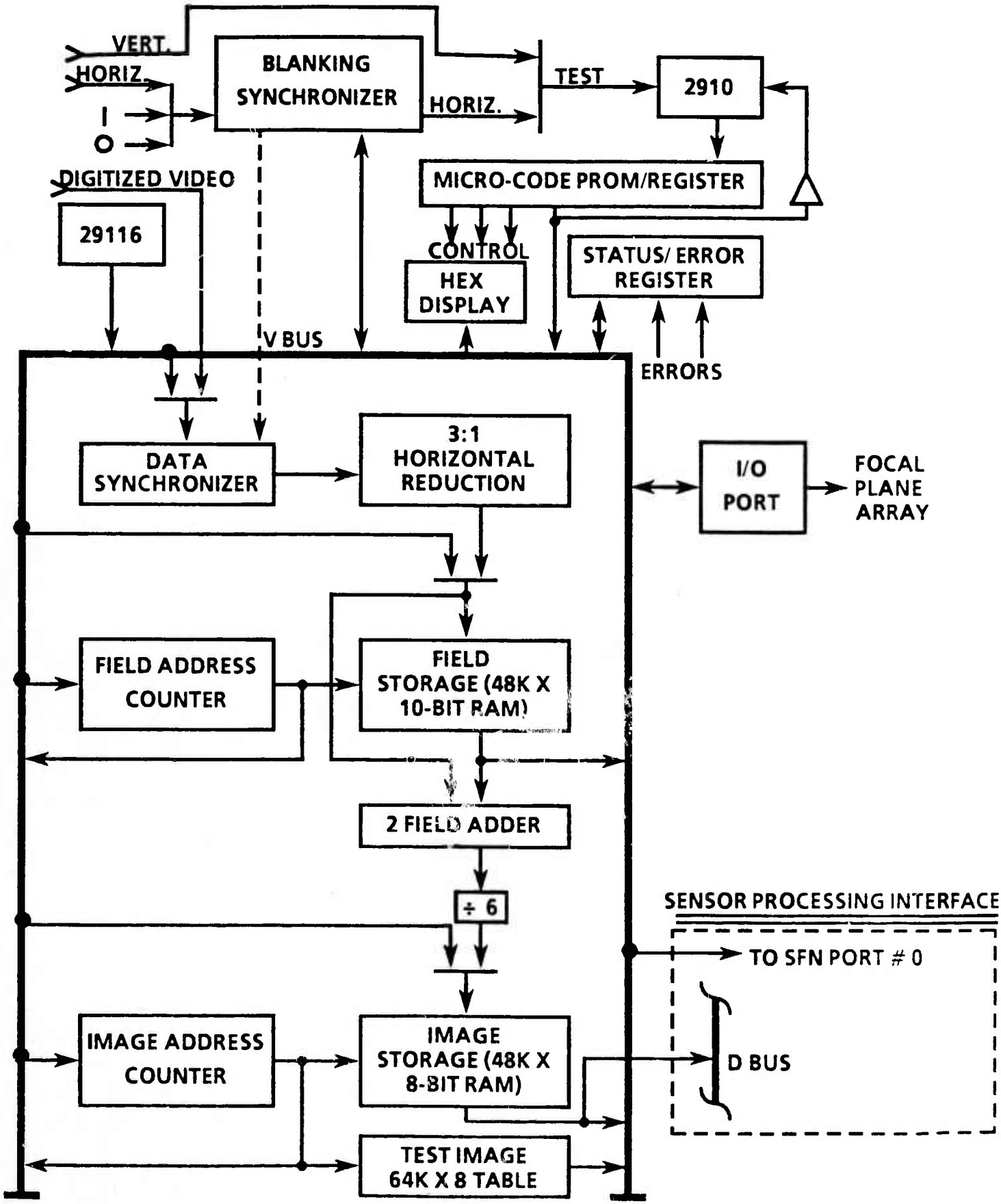


Figure B.14. SFV Processor Block Diagram.

ASSP / NATS PROCESSOR: SENSOR FORMATTER NUMERIC / VSC

FIG. SFNBD

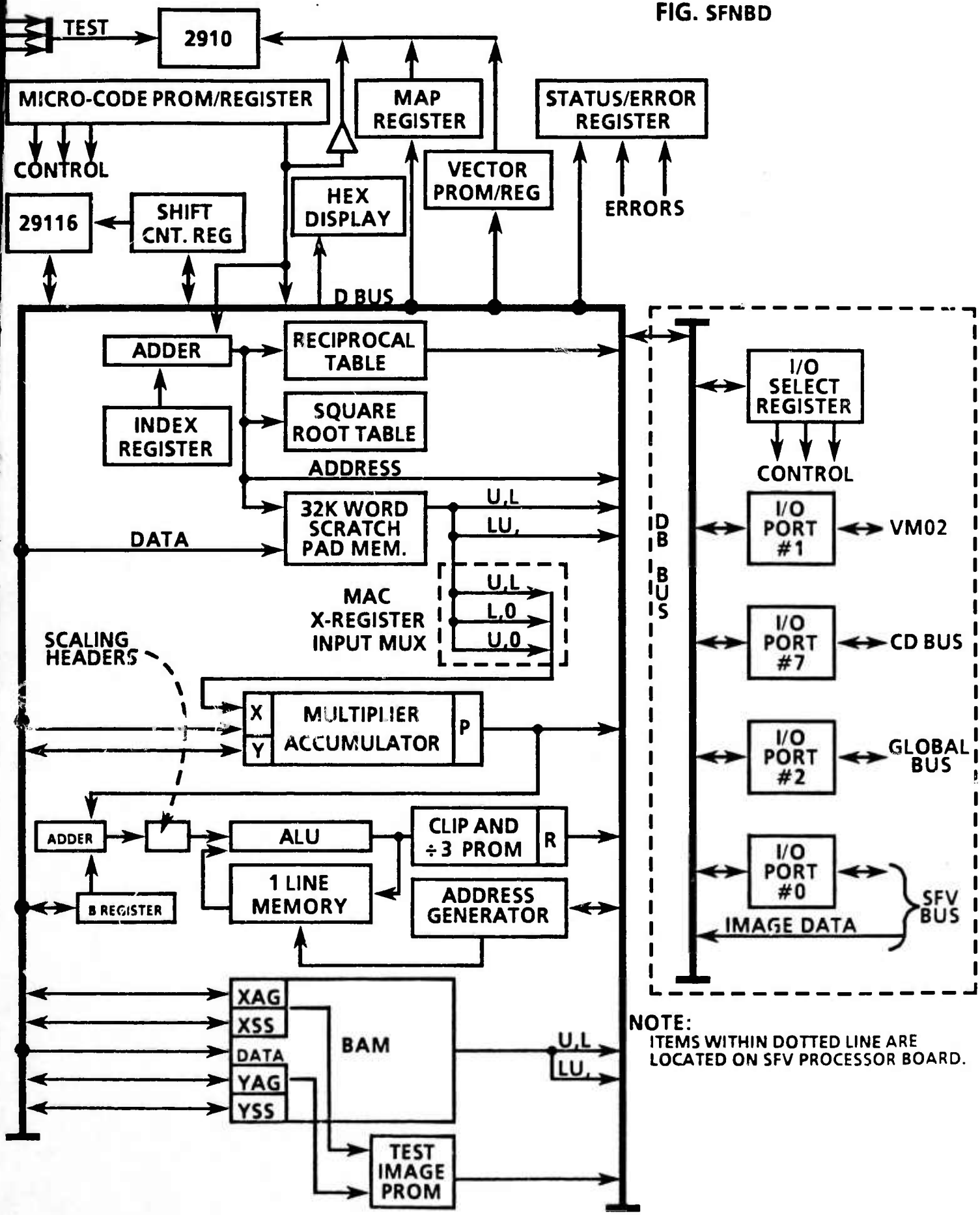


Figure B.15. SFN Processor Block Diagram.

ASSP / NATS PROCESSOR: GLOBAL PROCESSOR

FIG. GPBD

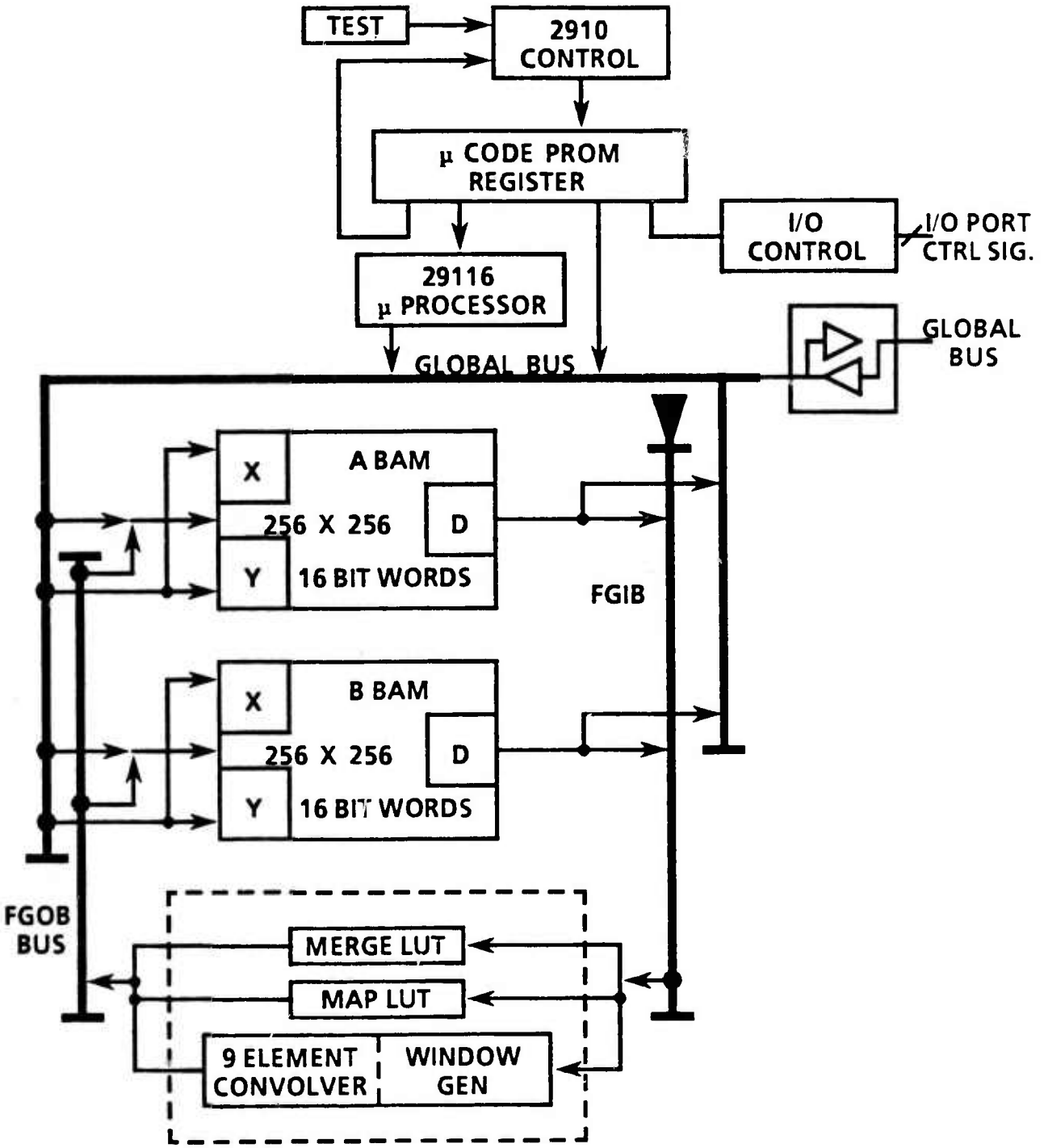


Figure B.16. GP Processor Block Diagram.

ASSP / NATS PROCESSOR: MAXIMUM LIKELIHOOD CLASSIFIER

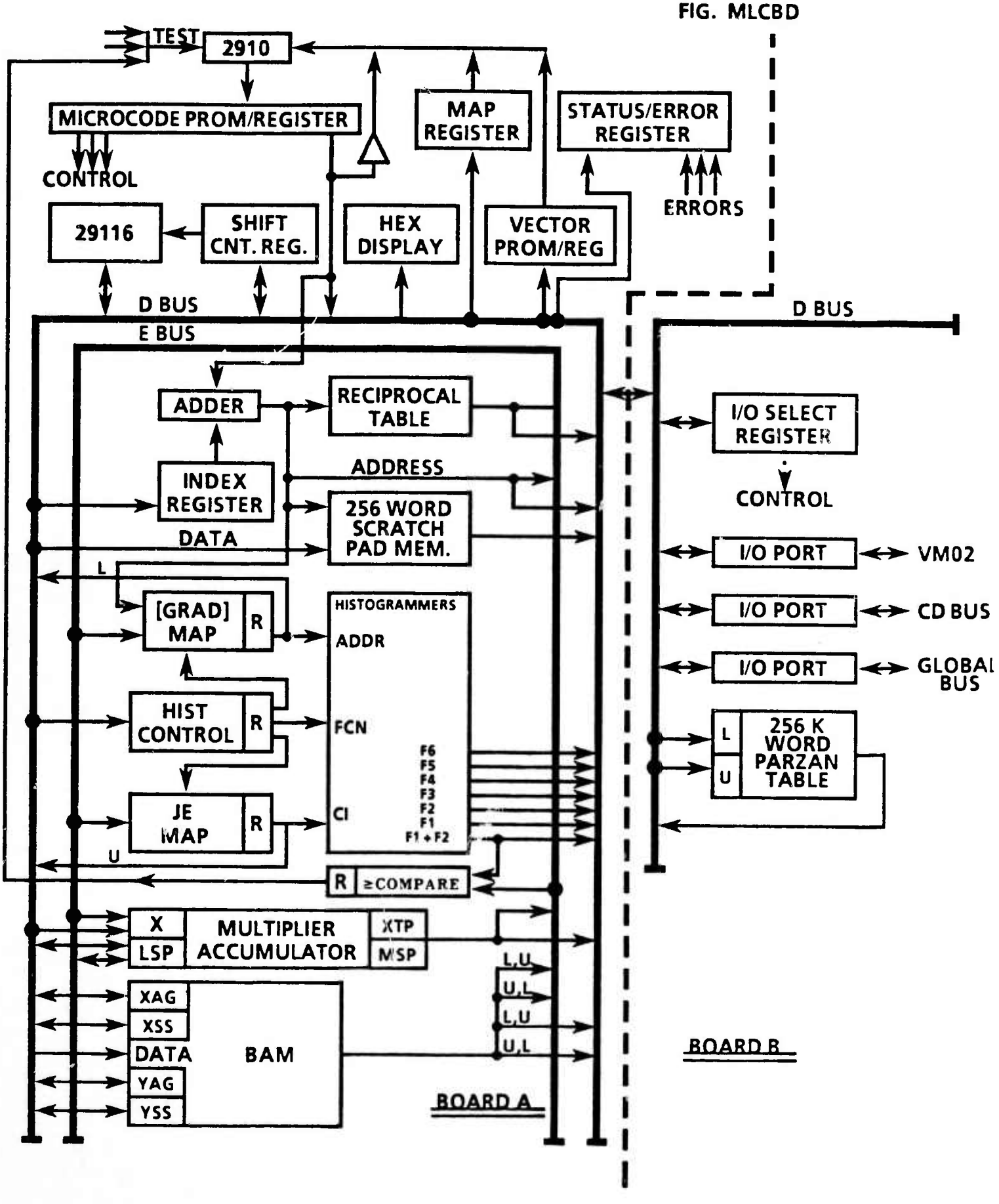


Figure B.17. MLC Processor Block Diagram.

ASSP / NATS PROCESSOR: SCENELOCK / TARGET TRACKER

FIG. SLTBD

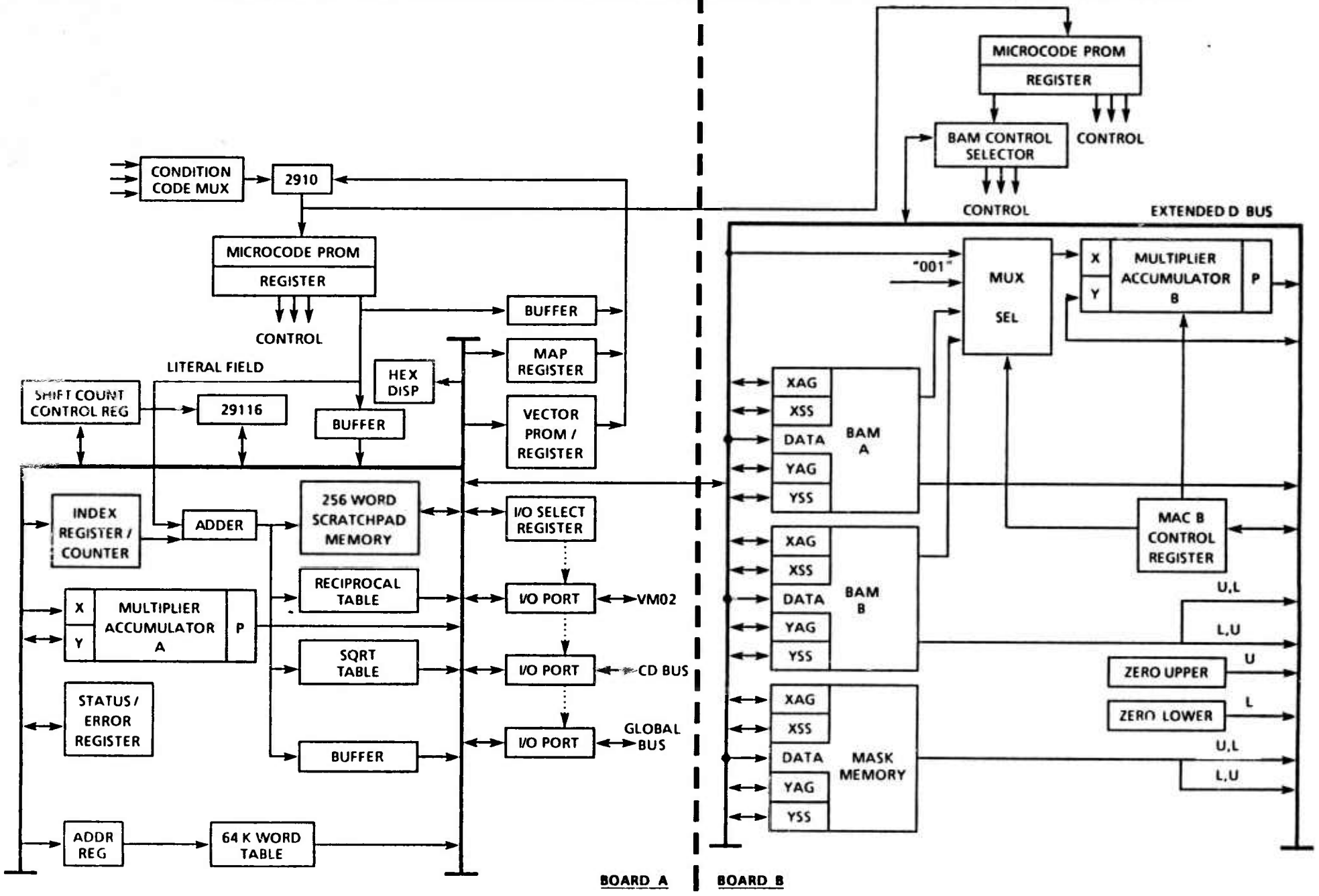


Figure B.18. SL/TT Processor Block Diagram.

ASSP / NATS PROCESSOR: CONTROL AND DISPLAY

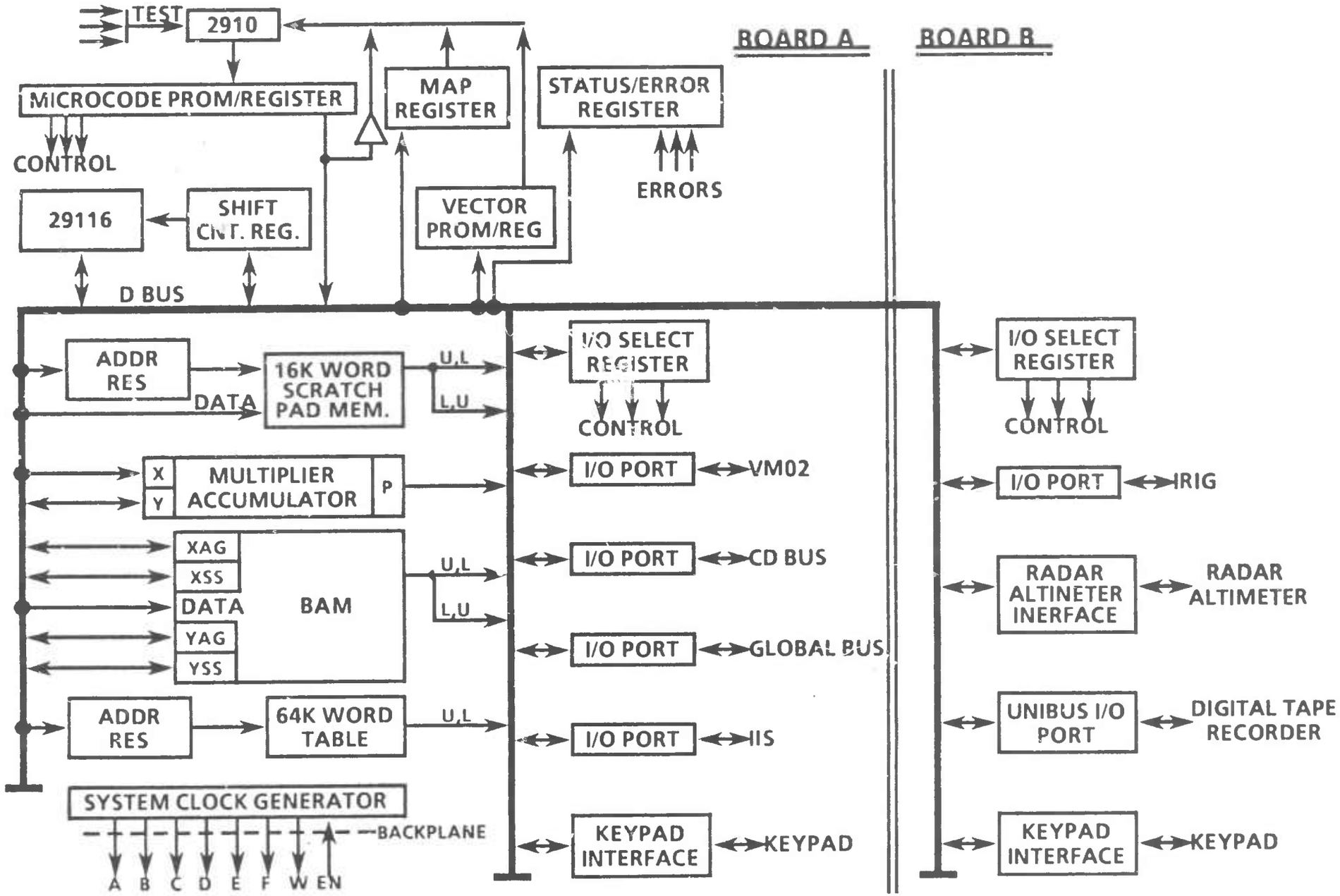


FIG. CDBD

Figure B.19. CD Processor Block Diagram.

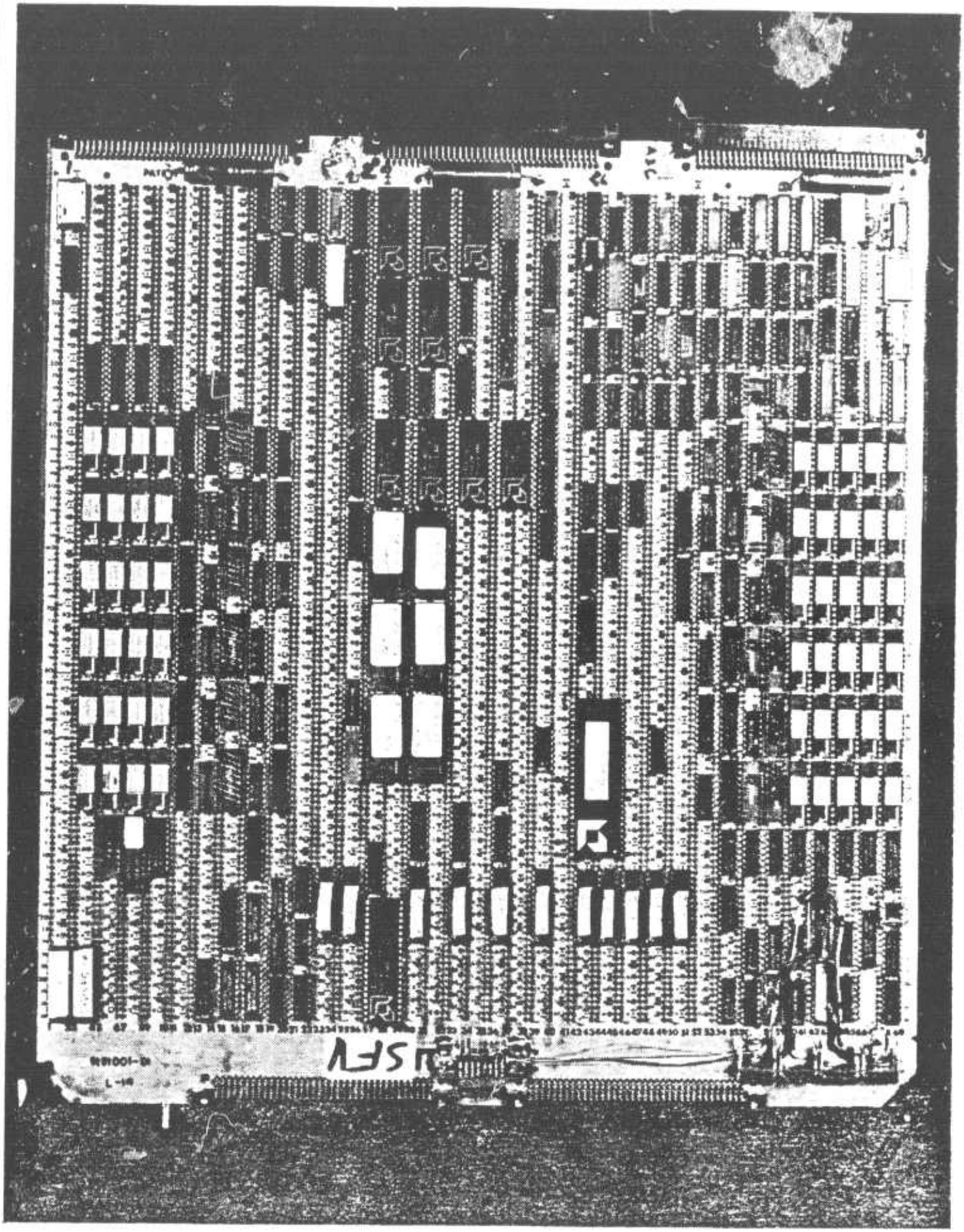


Figure B.20. SFV Processor Picture.

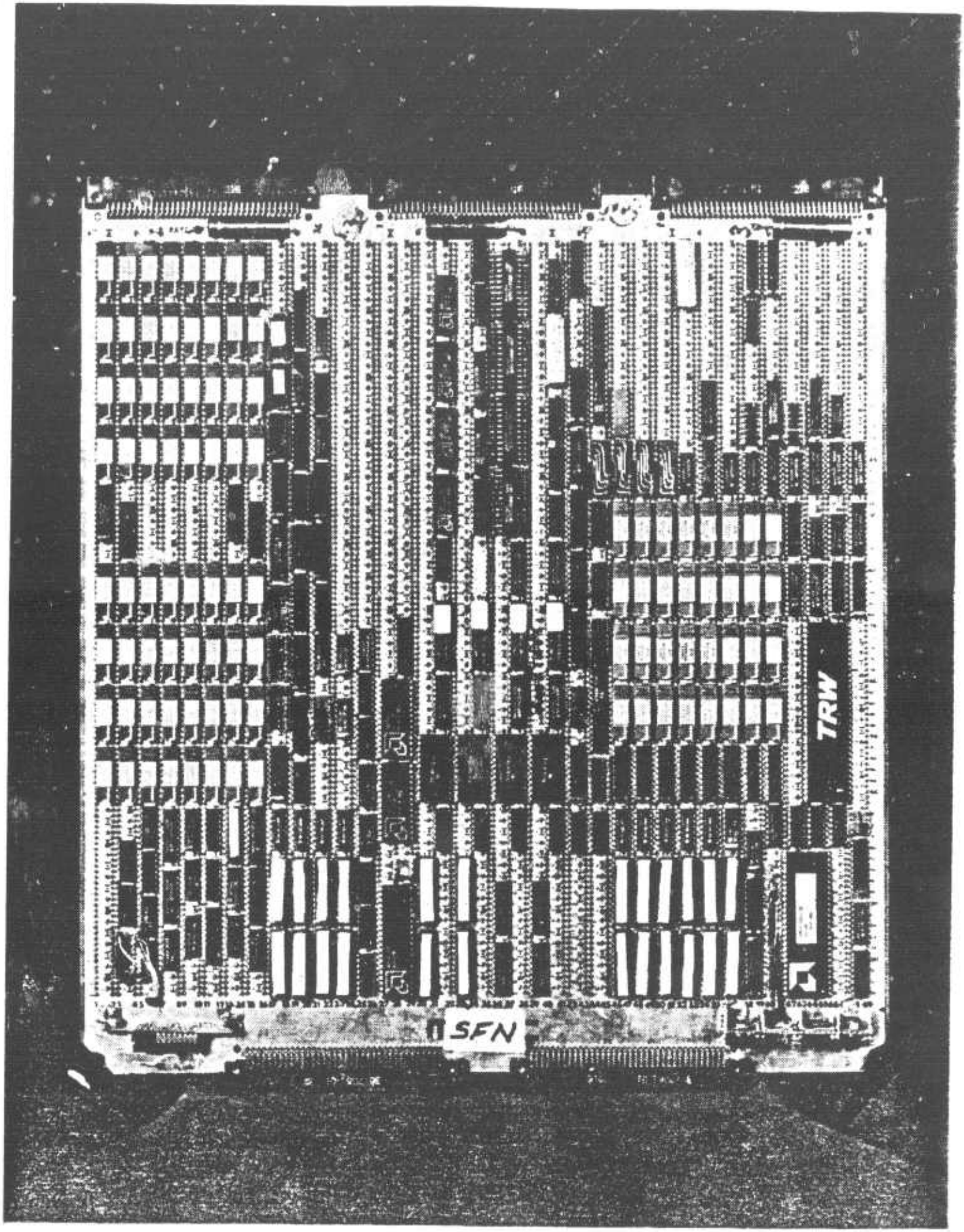


Figure B.21. SFN Processor Picture.

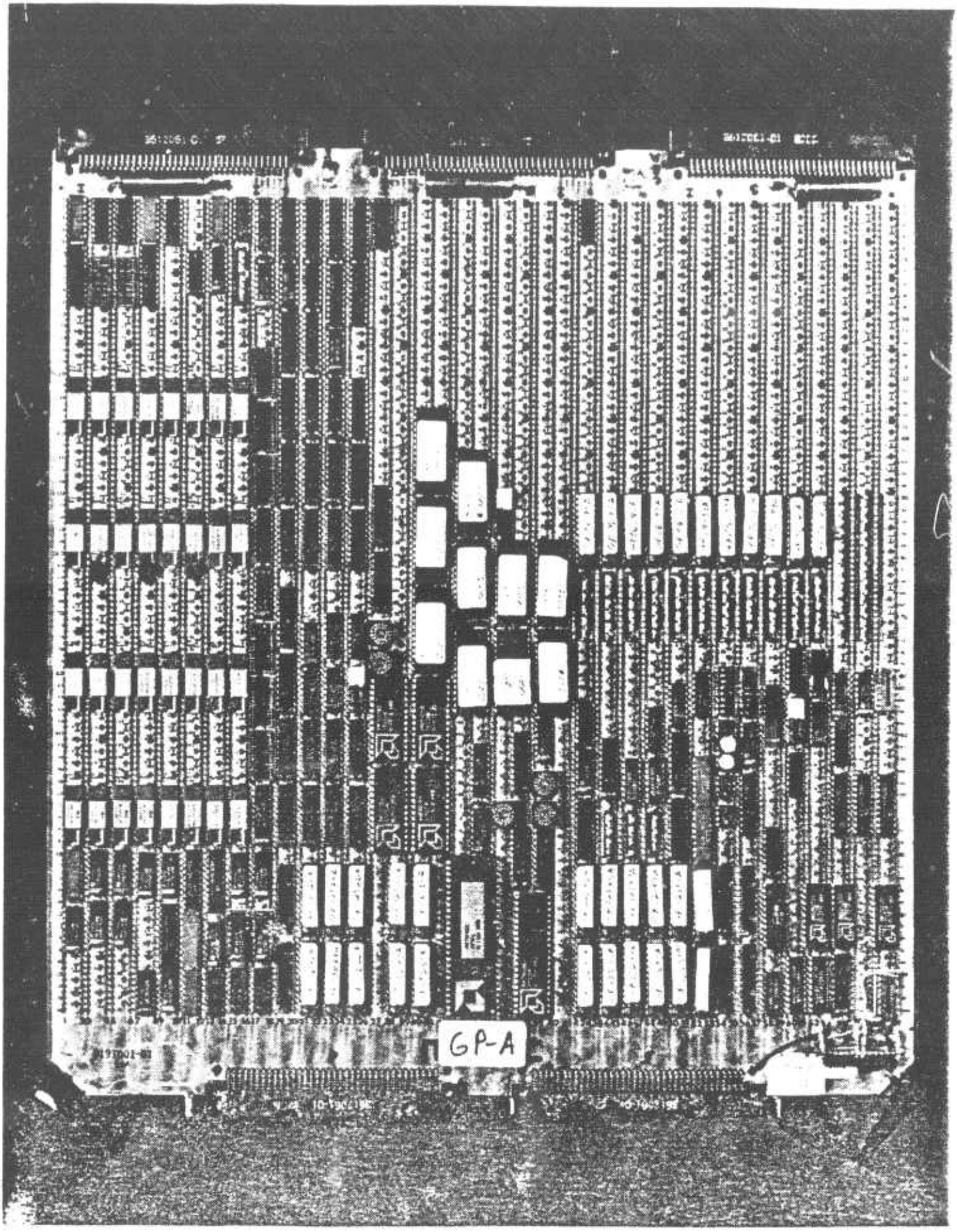


Figure B.22-a. GP Processor Picture.

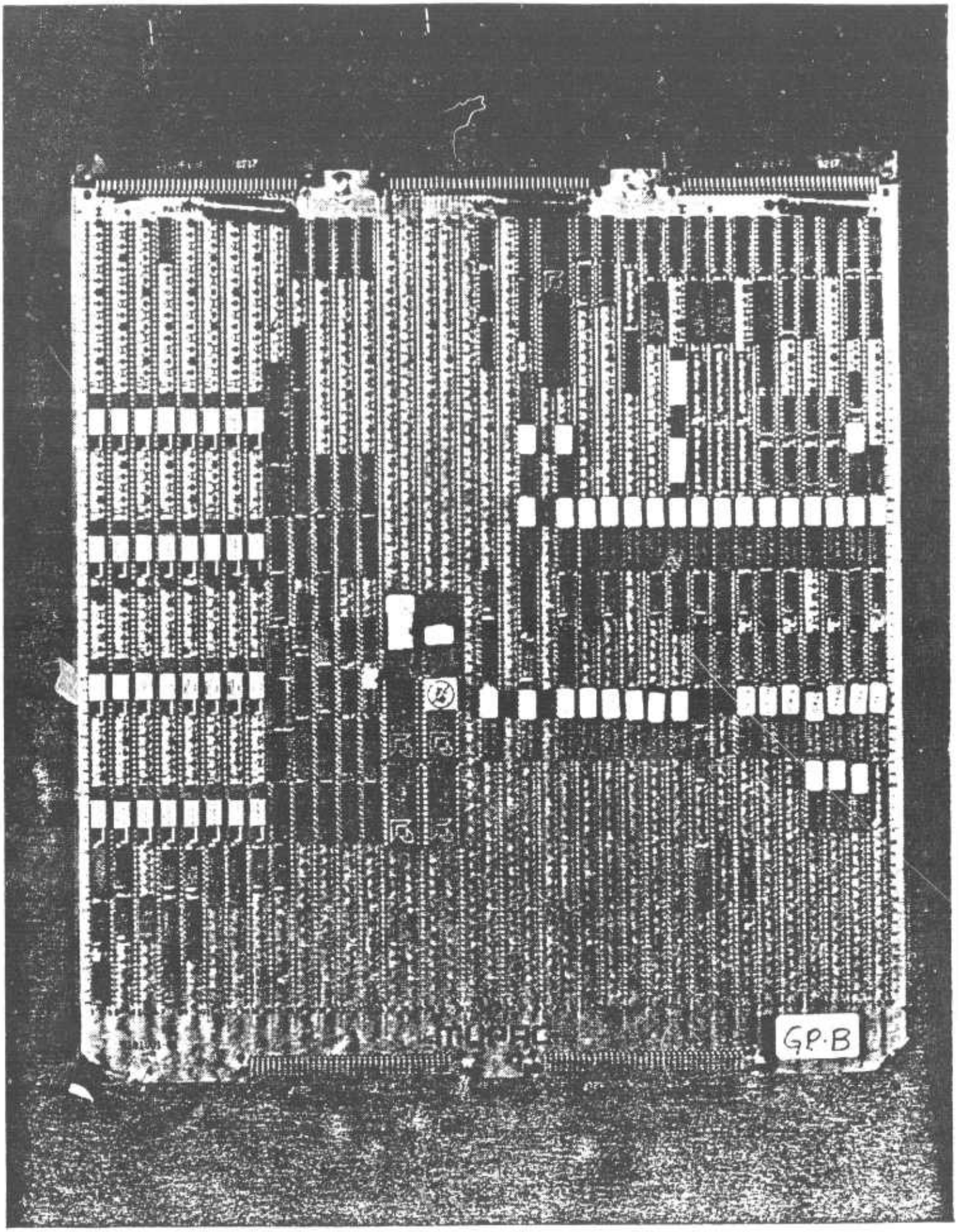


Figure B.22-b. GP Processor Picture.

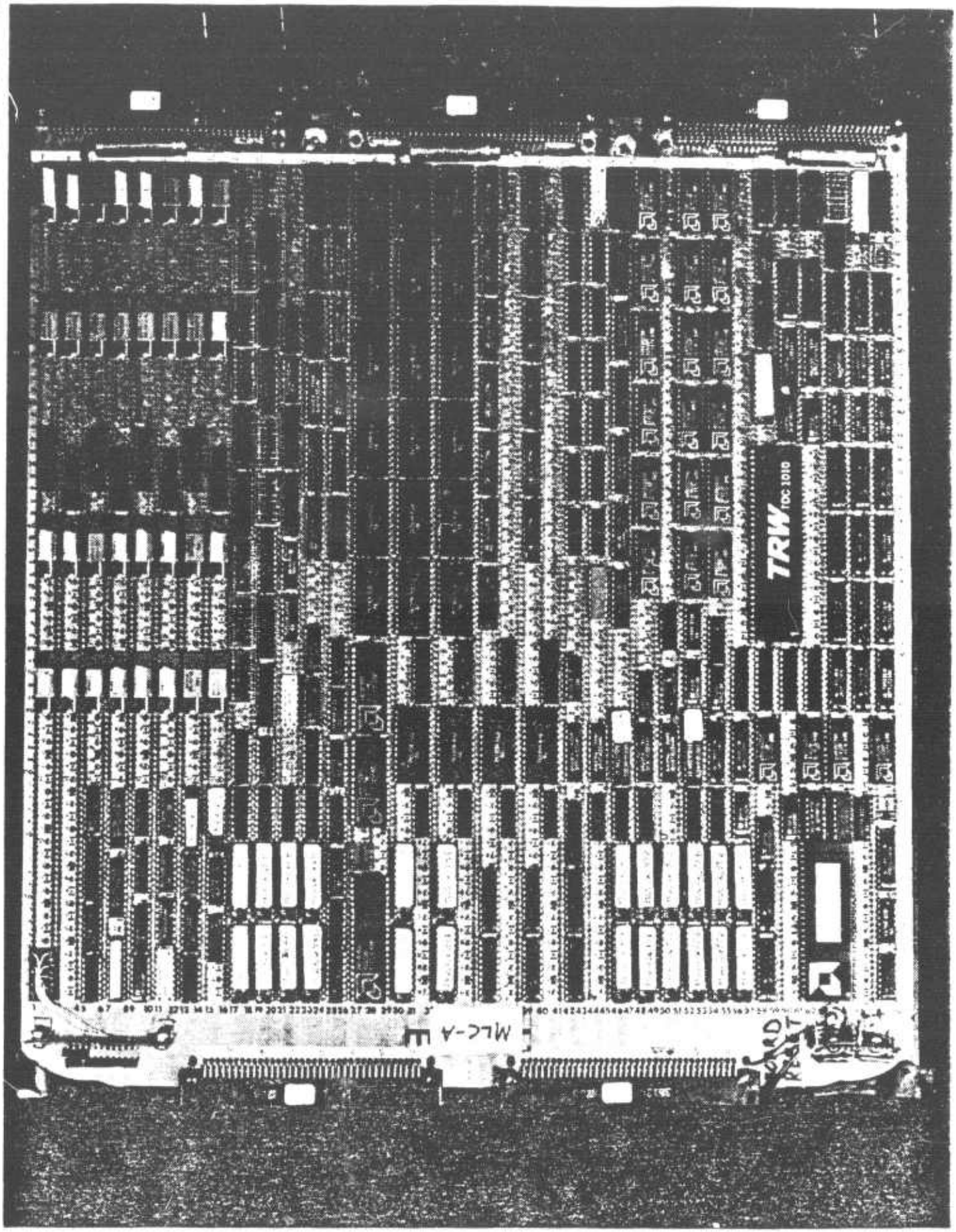


Figure B.23-a. MLC Processor Picture.

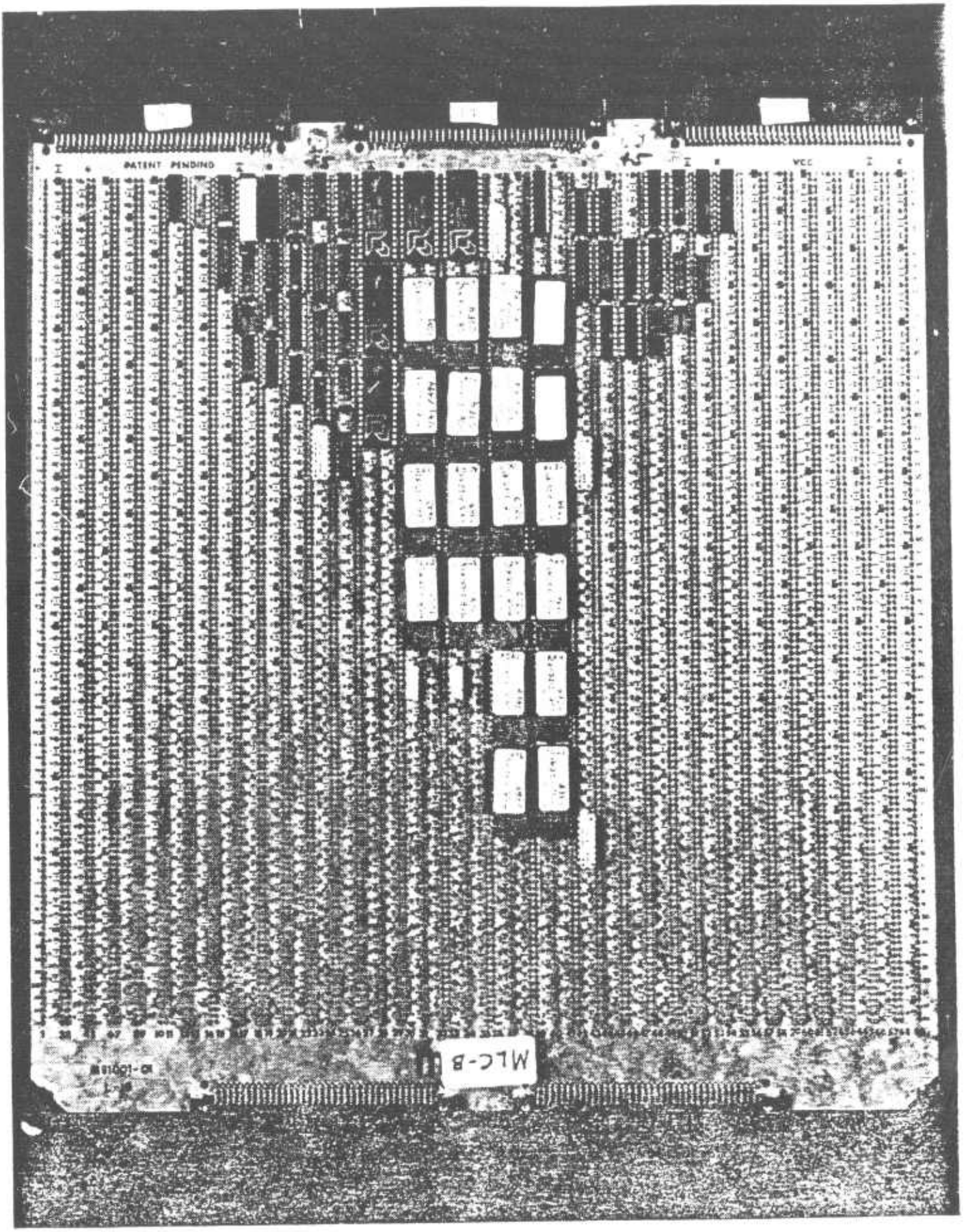


Figure B.23-b. MLC Processor Picture.

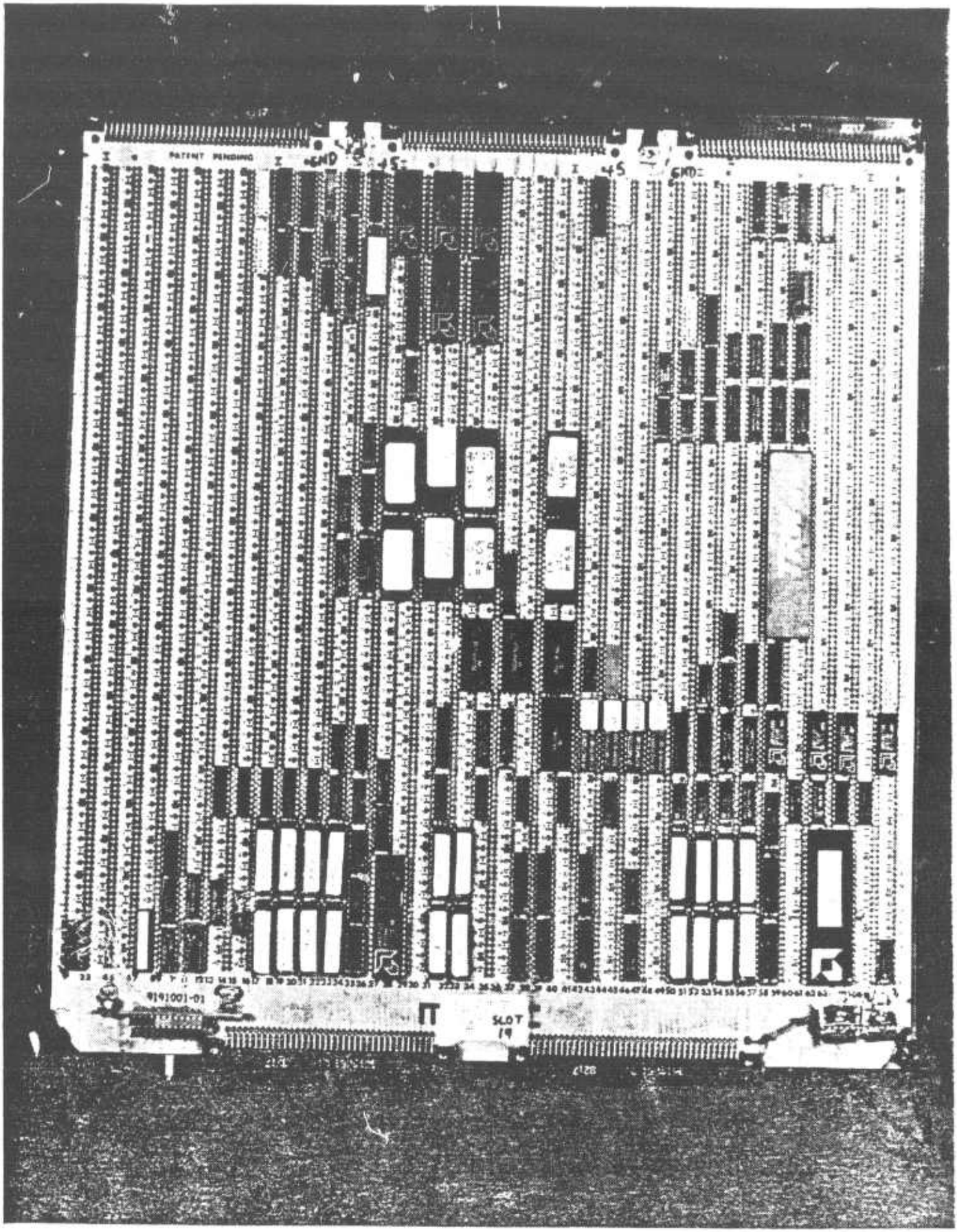


Figure B.24-a. SL/TT Processor Picture.

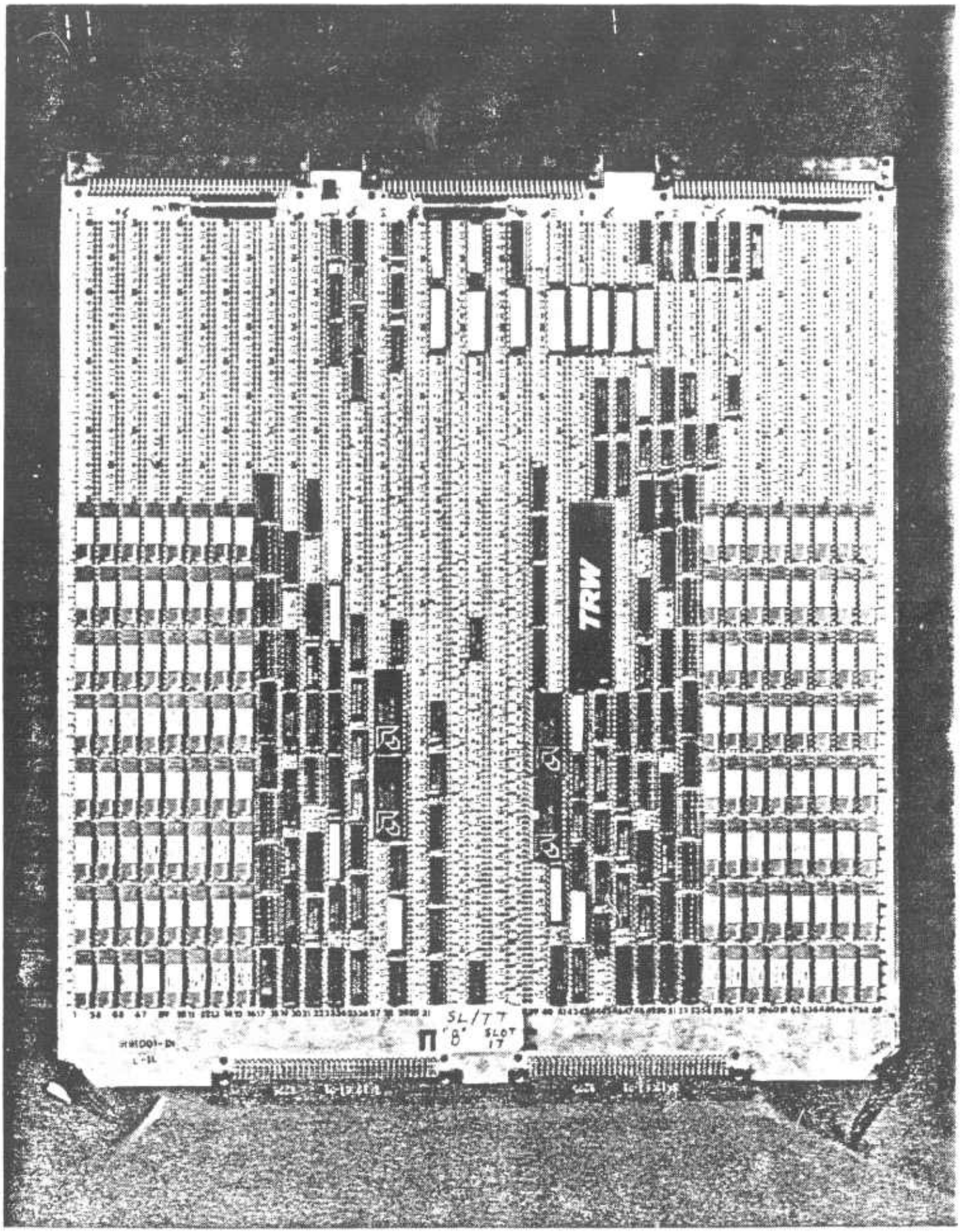


Figure B.24-b. SL/TT Processor Picture.

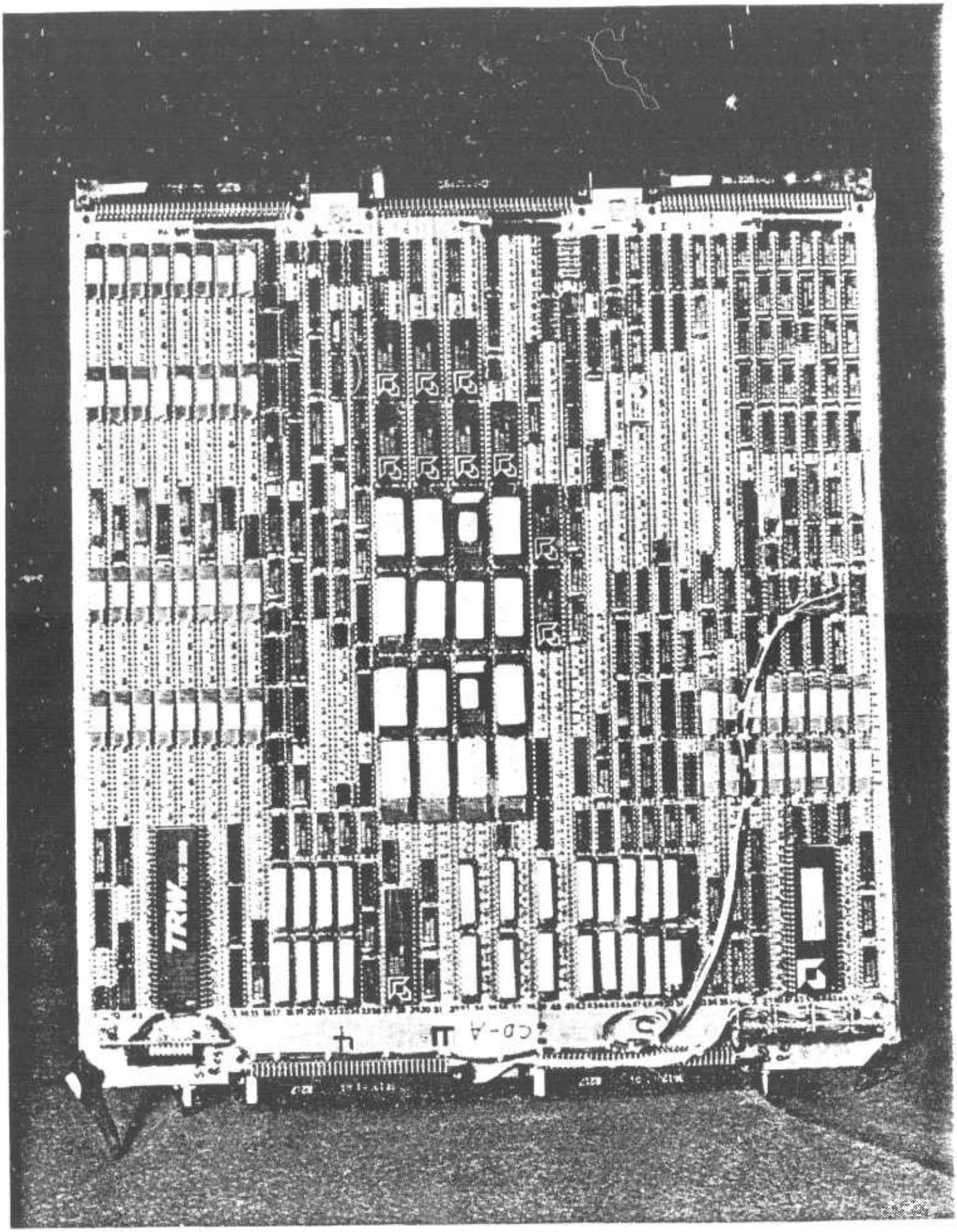


Figure B.25-a. CD Processor Picture.

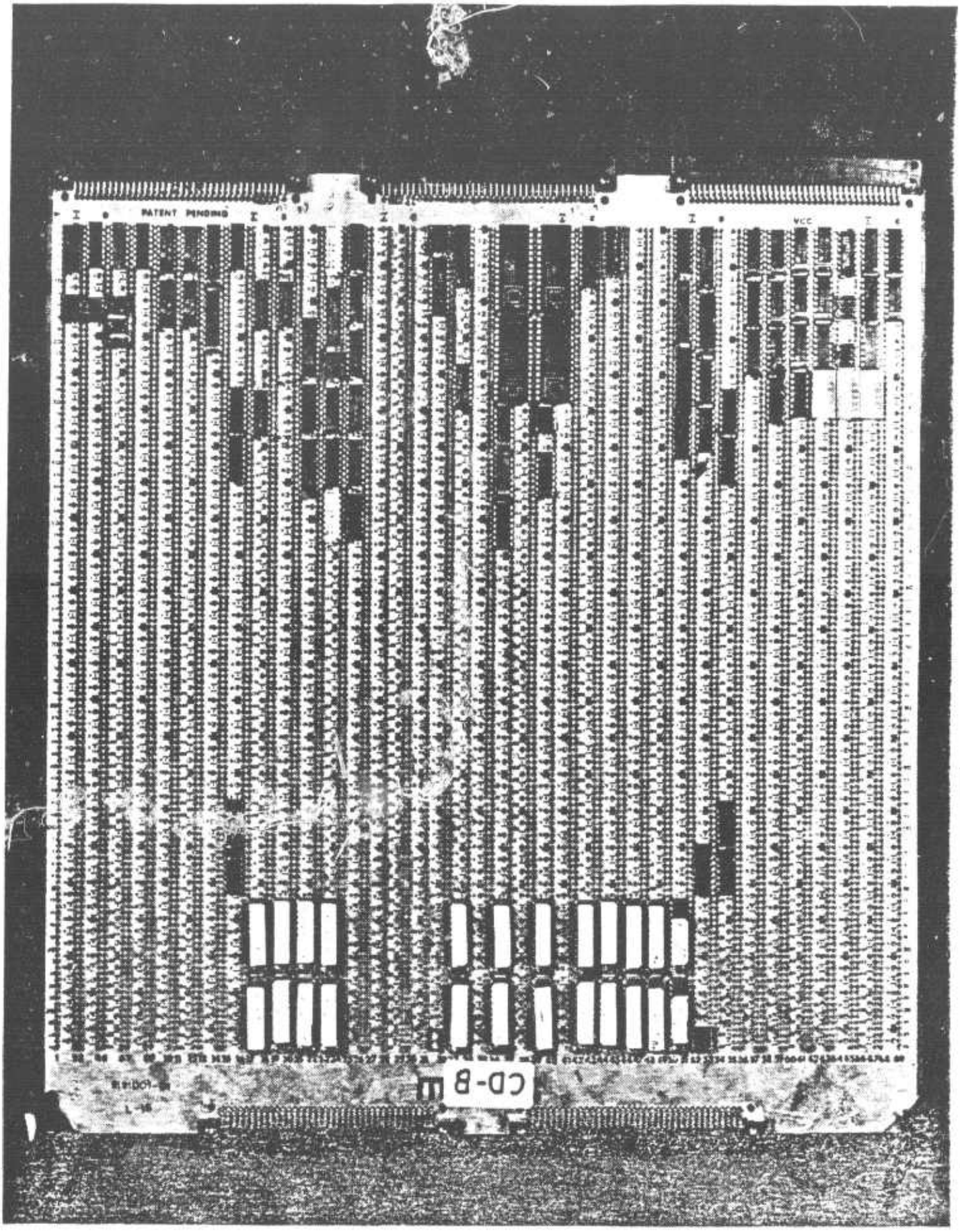


Figure B.25-b. CD Processor Picture.

NATS - SFV MICROWORD DEFN.

FIG SFVUW

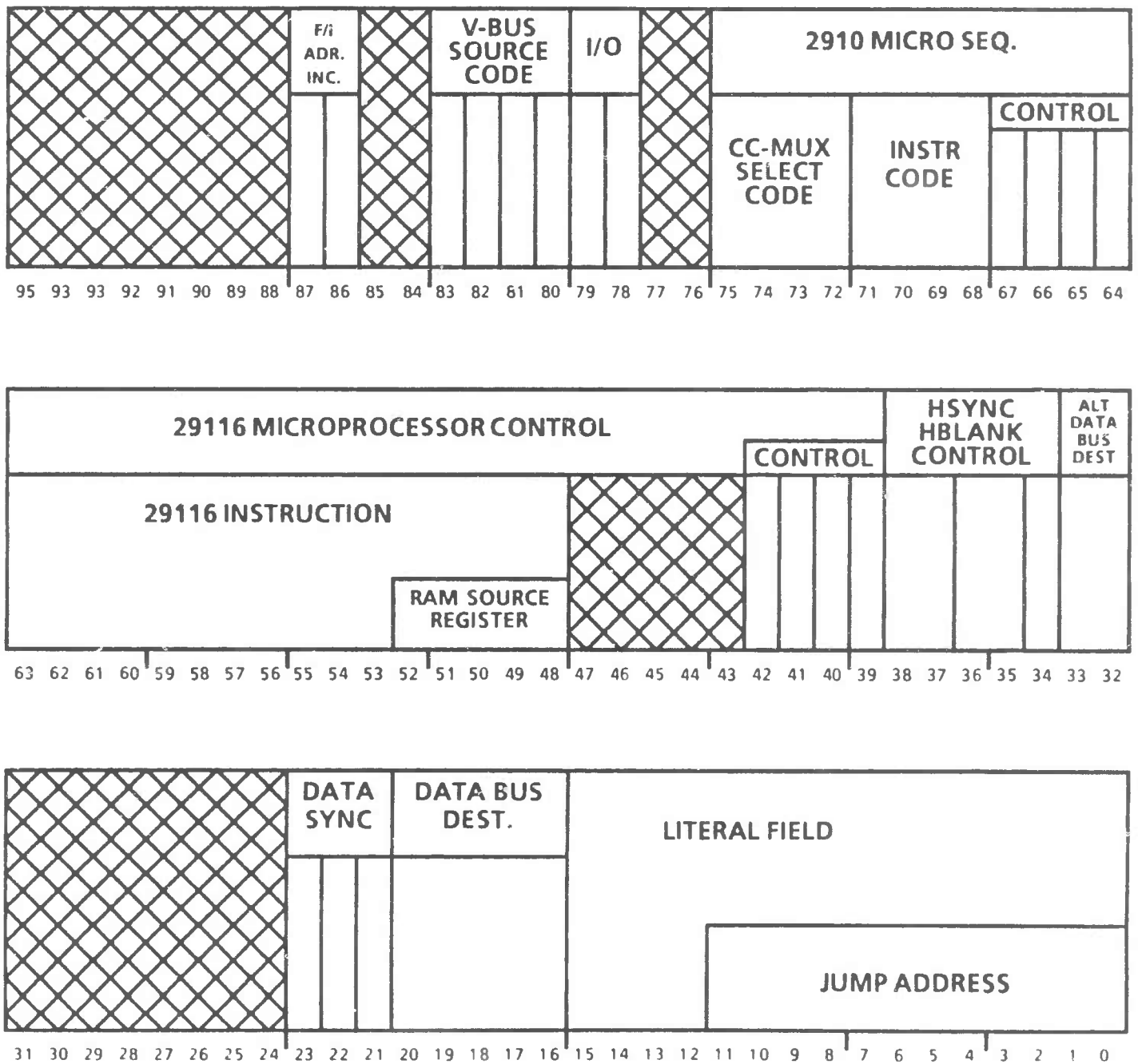
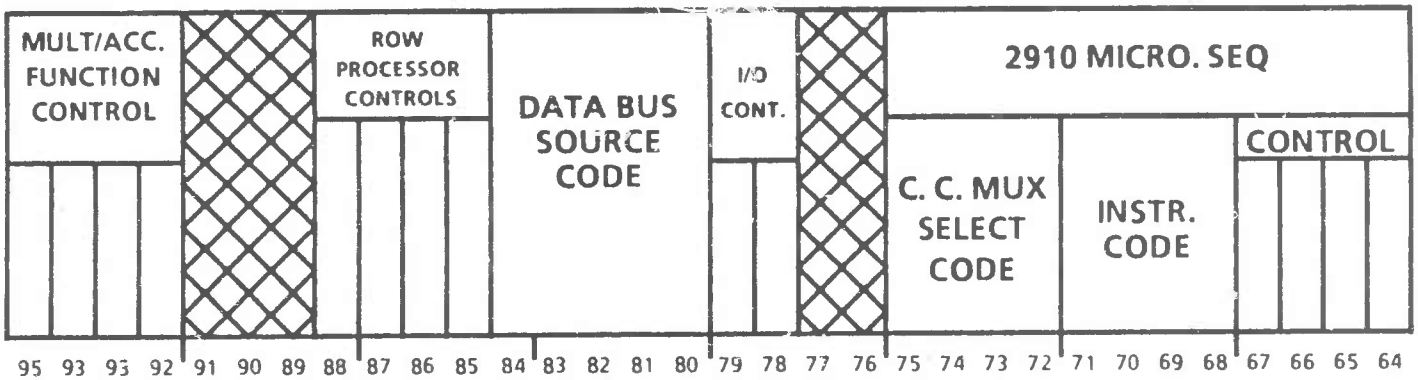
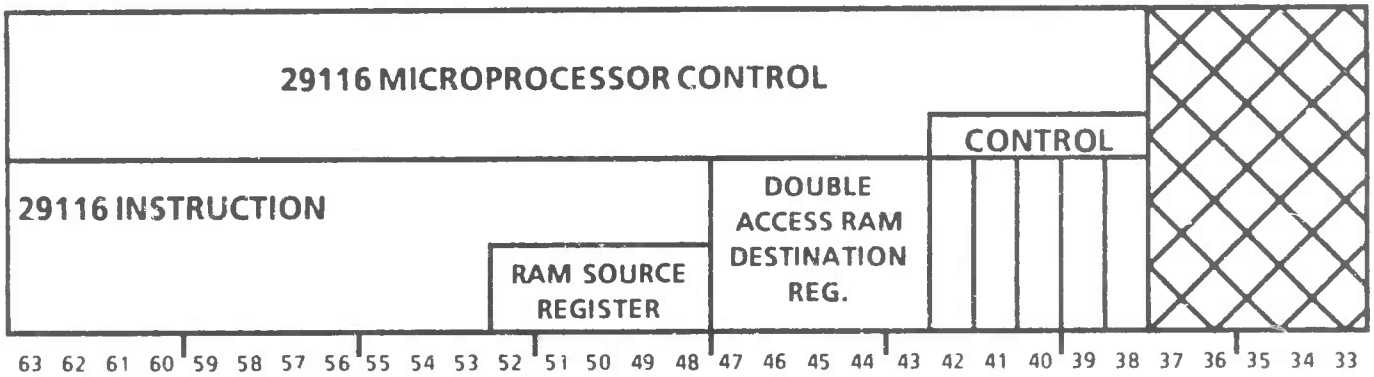
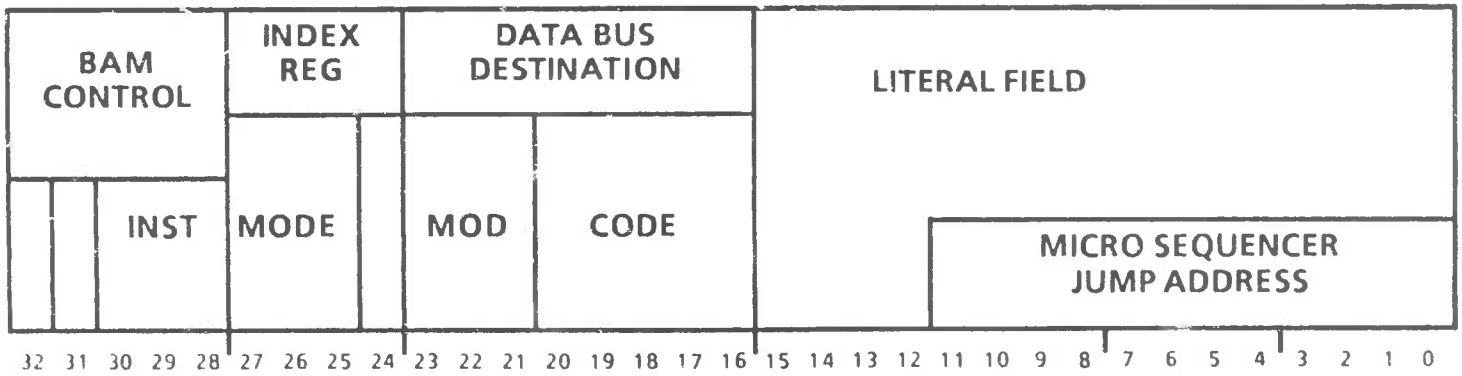


Figure B.26. SFV Microword Format.

SFN PROCESSOR MICROWORD LAYOUT







B.27-a. M68KVM80 Memory Board.

GP PROCESSOR MICROWORD LAYOUT FIG GPUW

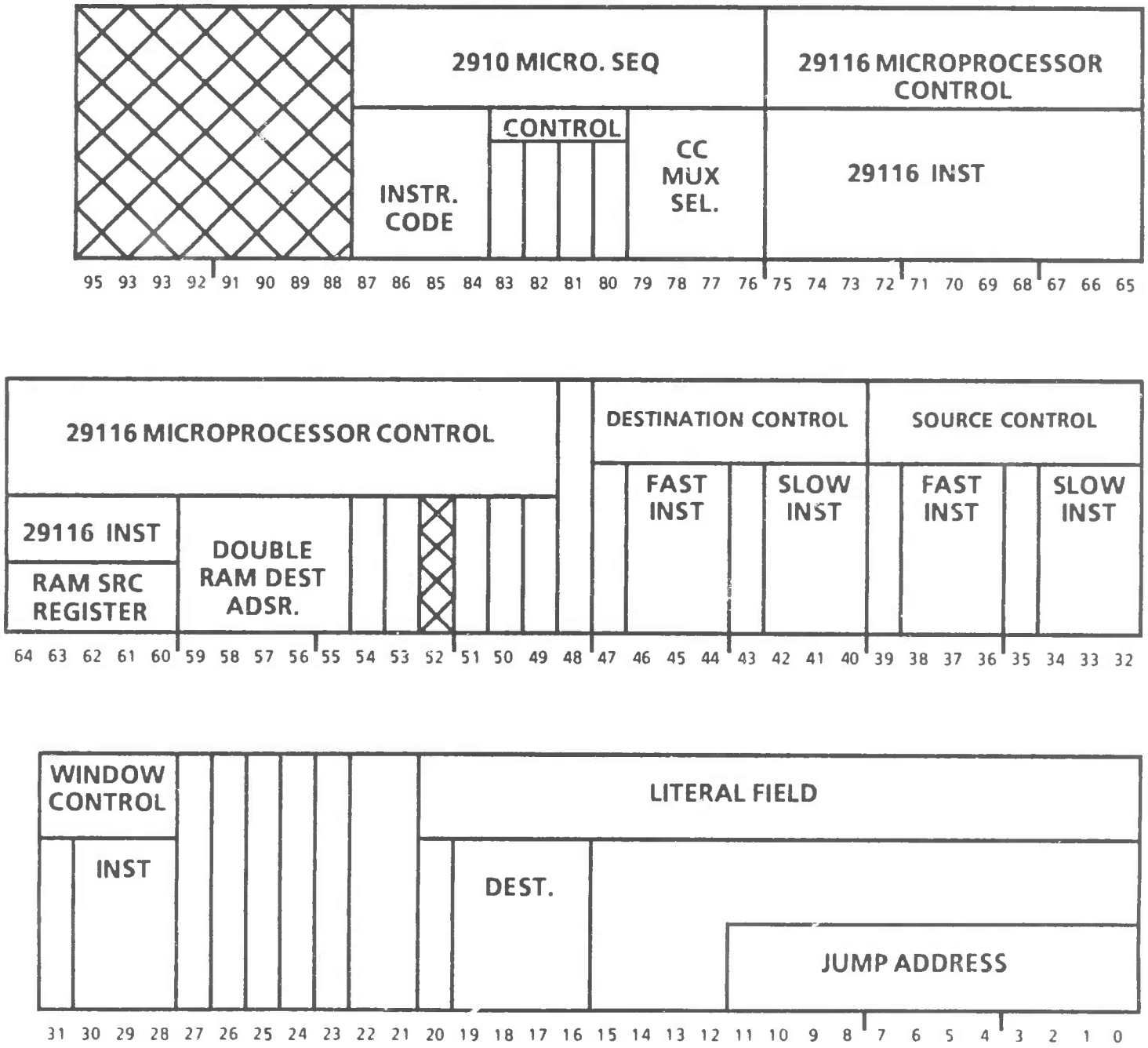


Figure B.28. GP Microword Format.

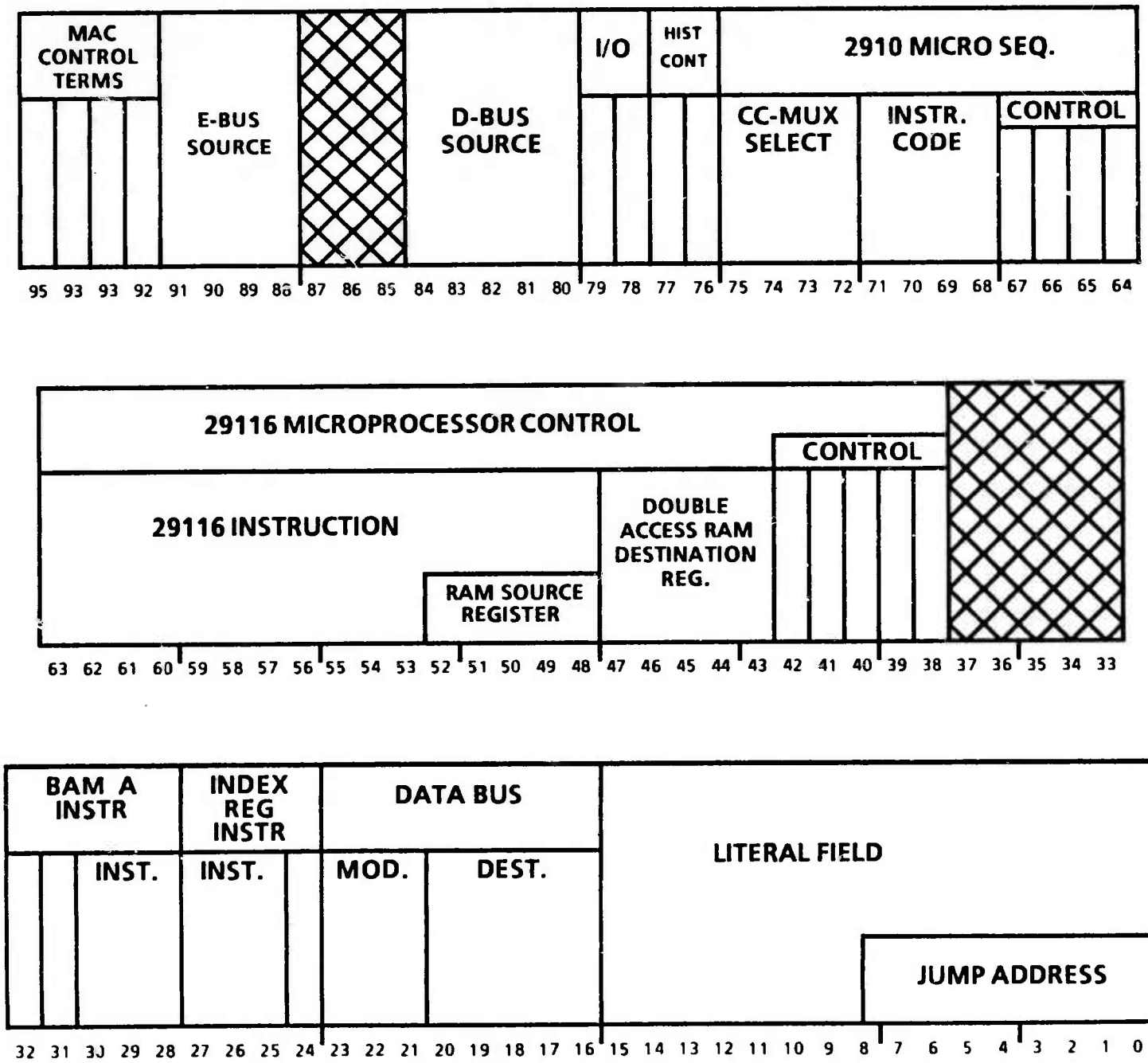


Figure B.29. MLC Microword Format.

NATS - SL/TT MICROWORD DEFN. FIG SLTUW

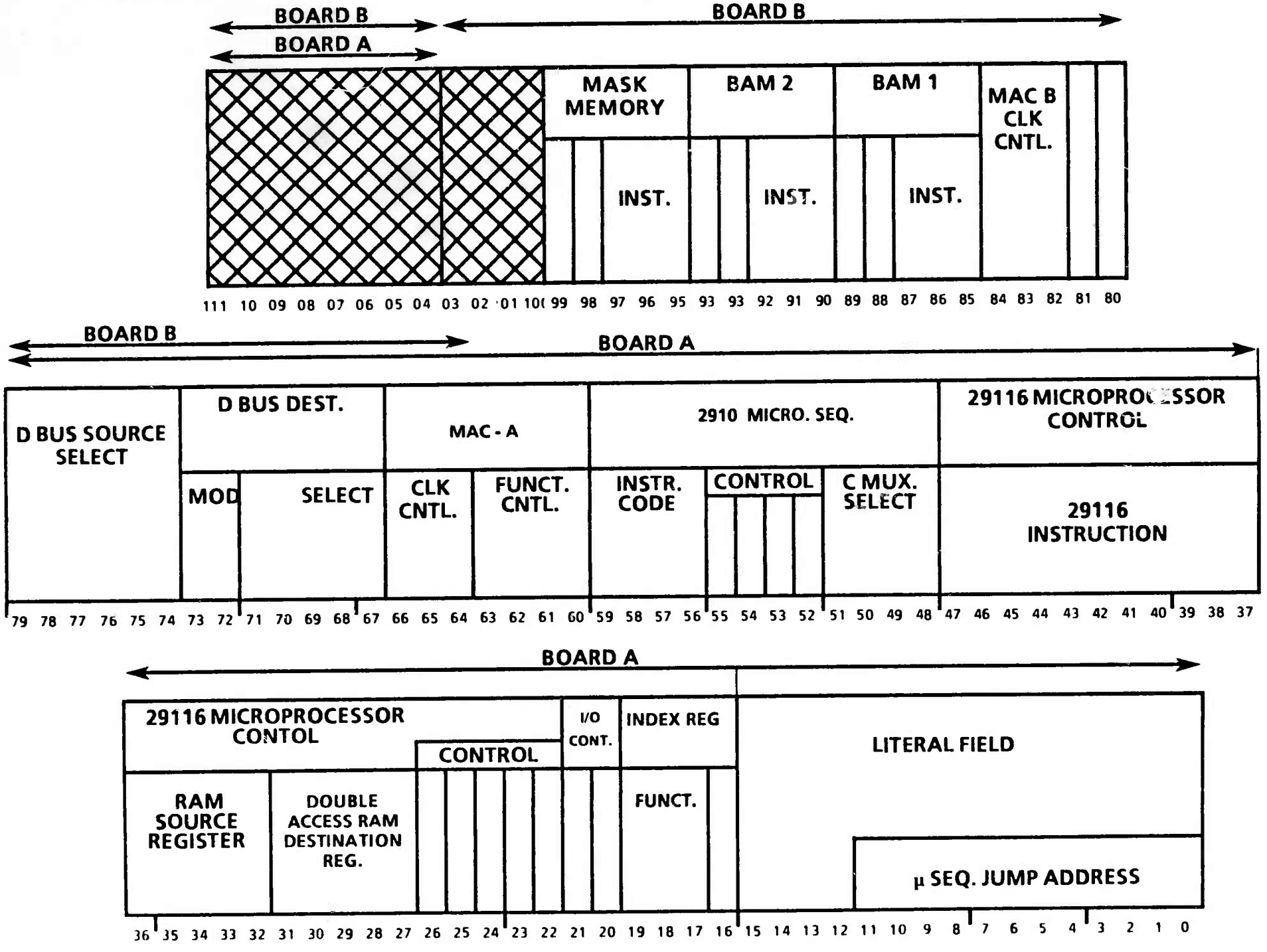


Figure B-30 SL/TT Microword Format

CD PROCESSOR MICROWORD LAYOUT

FIG CDUW

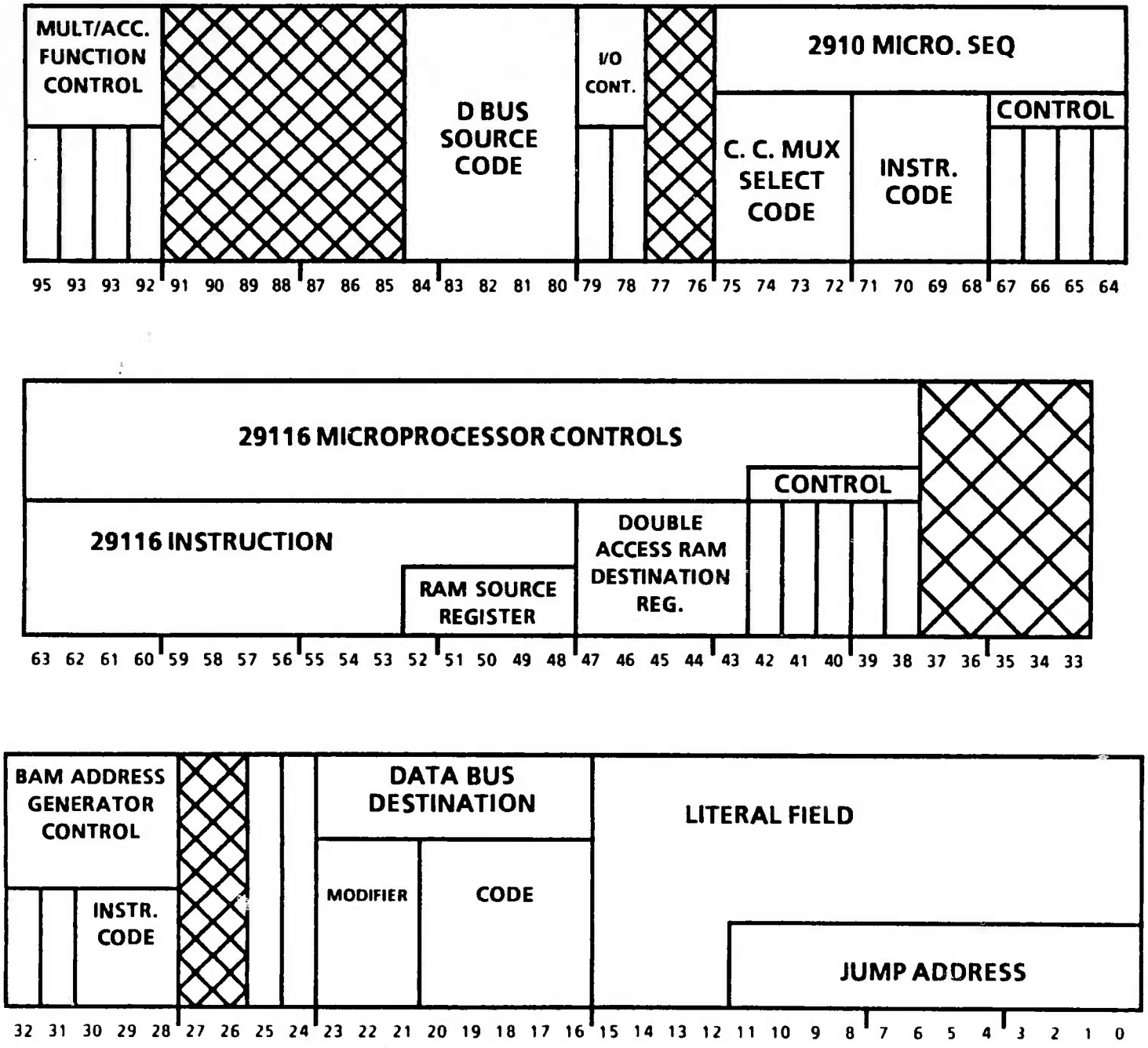


Figure B.31. CD Microword Format.

APPENDIX C

RESULTS FROM ASSP CAPTIVE FLIGHT TEST

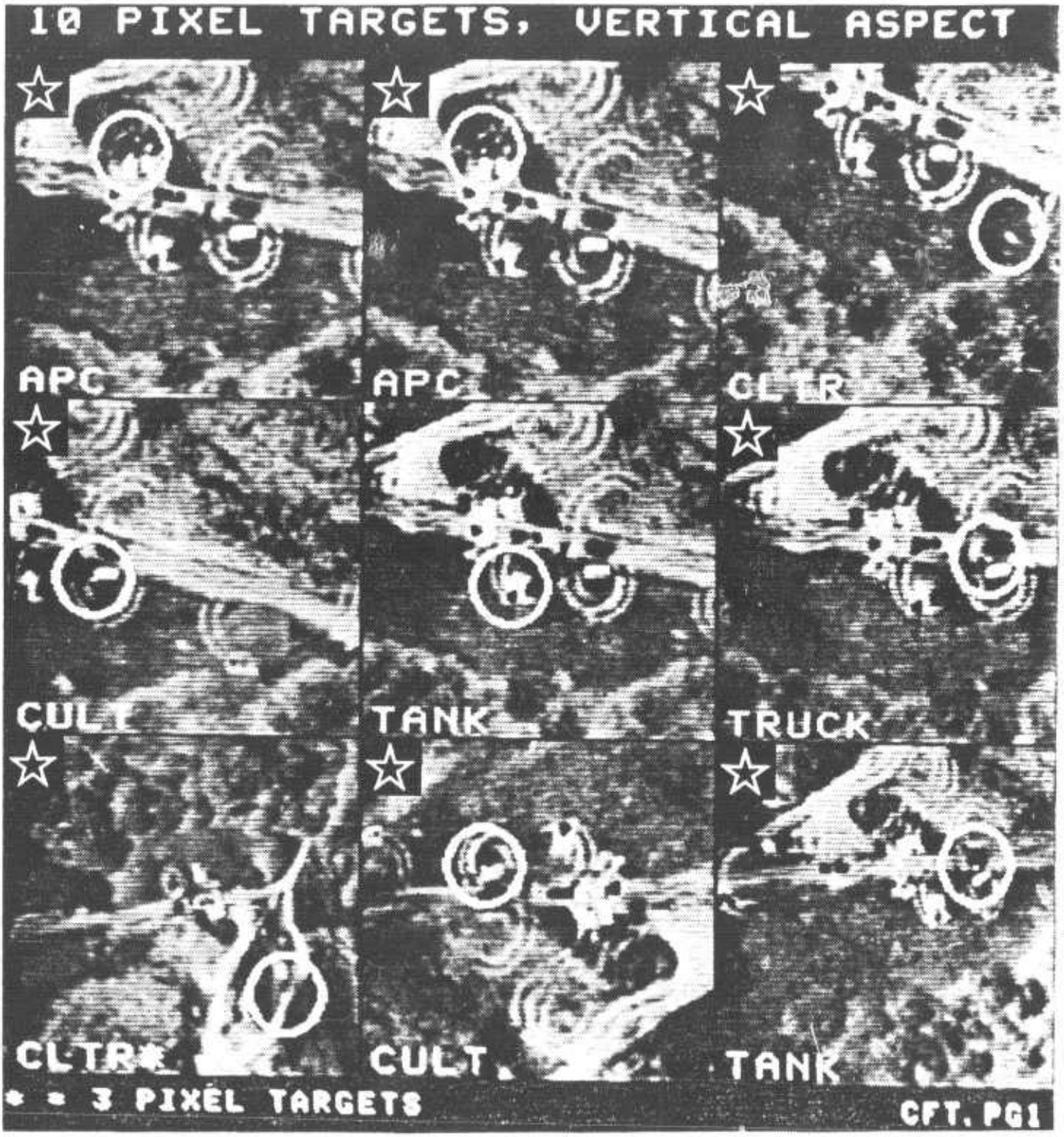
APPENDIX C

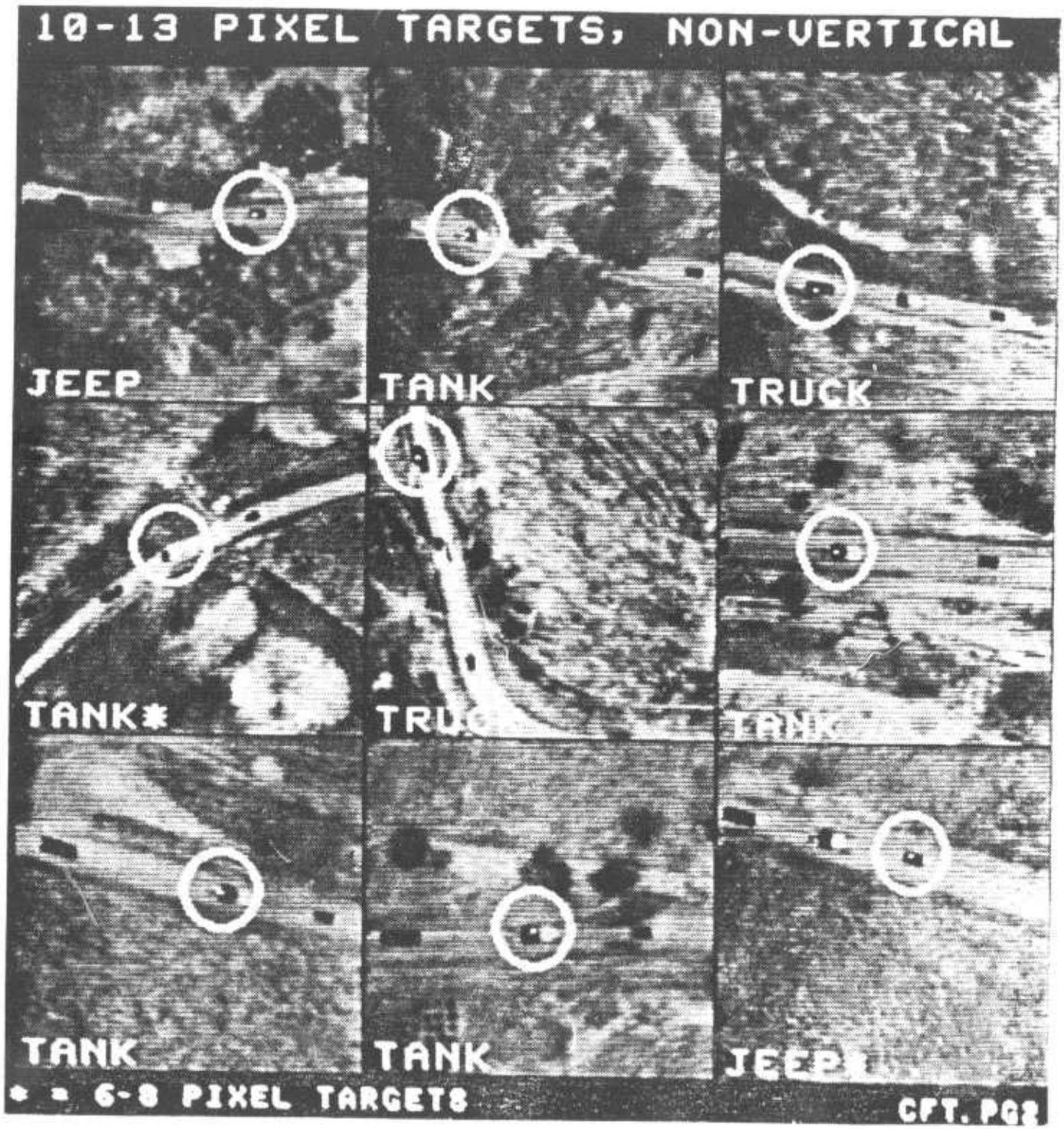
RESULTS FROM ASSP CAPTIVE FLIGHT TEST

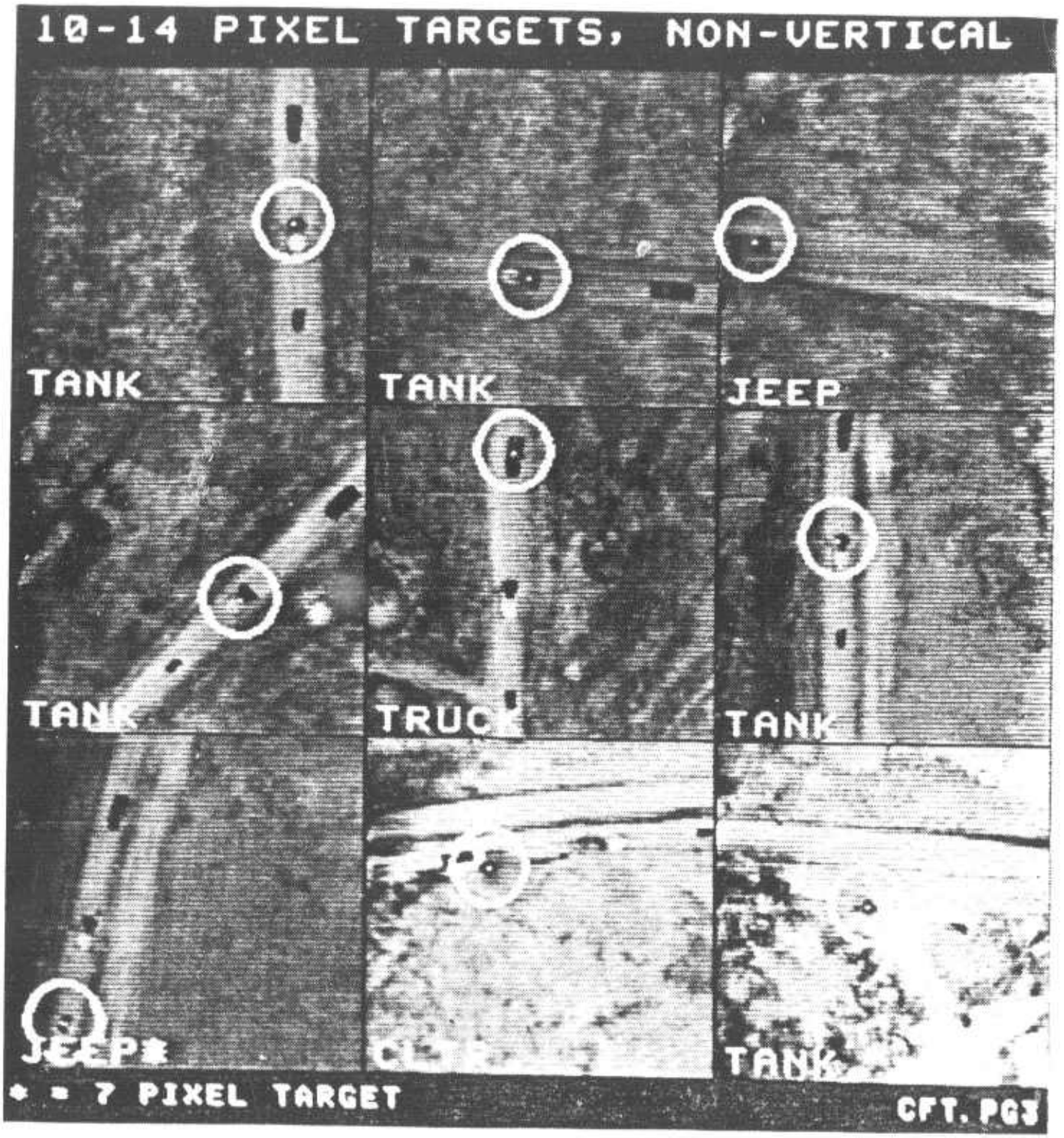
In November 1983, the Advanced Seeker Signal Processor was flown in a series of captive flight tests over Test Range 6 at the Army Missile Command (MICOM), Redstone Arsenal, Huntsville, Alabama. In three separate flights, 249 images were taken with depression angles from 20 degrees to 90 degrees (vertical lookdown) and target sizes from 1 pixel to 33 pixels in length. Target types included tanks, trucks, and APC's in both the cold and hot thermal states.

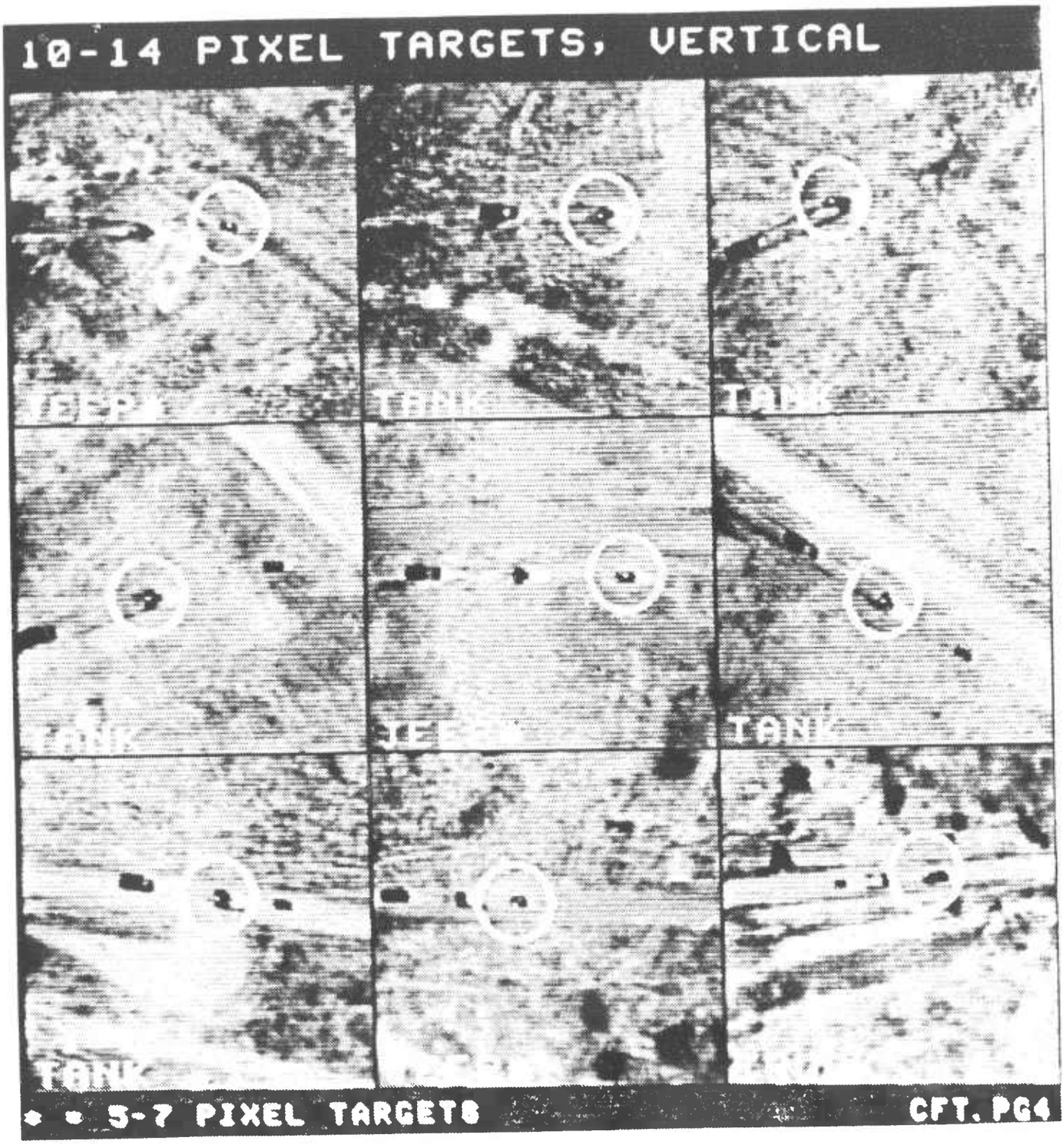
The ASSP was run against all images with target sizes of 6 pixels in length or longer. Some images were not included in the performance measure due to 1) excessive platform vibration or motion resulting in image breakup, blur, or distortion, 2) targets 5 pixels or less in length, 3) no targets in the field of view.

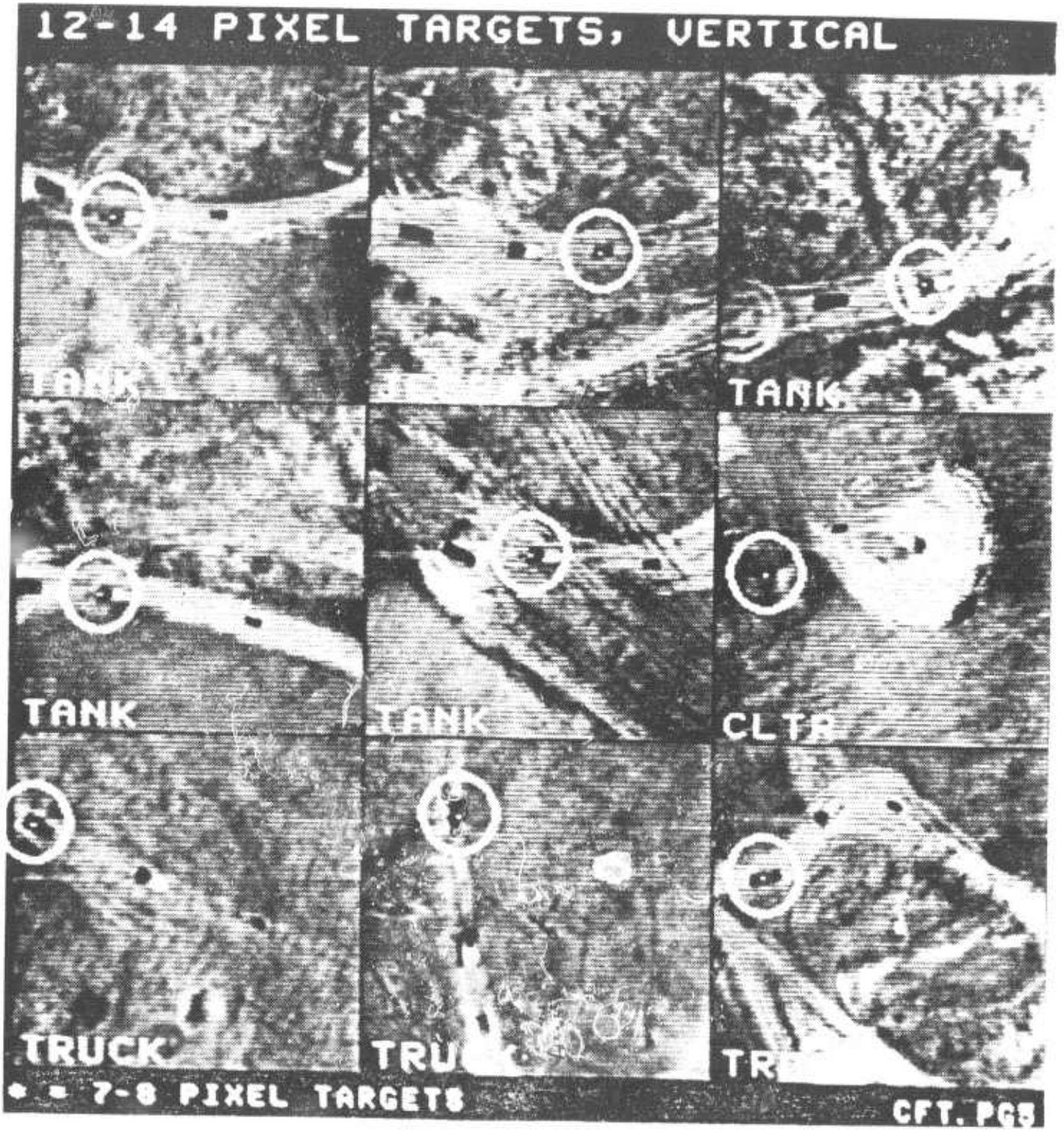
The entire Captive Flight Test data set is included in Figures C.1 to C.29. Where applicable, the number one ranked object has been indicated by a circle 12 pixels in radius and the corresponding ground truth annotated in the lower left hand corner of each 128 x 128 image. Images marked by "  " are considered dependent, and are not included in the performance measure.

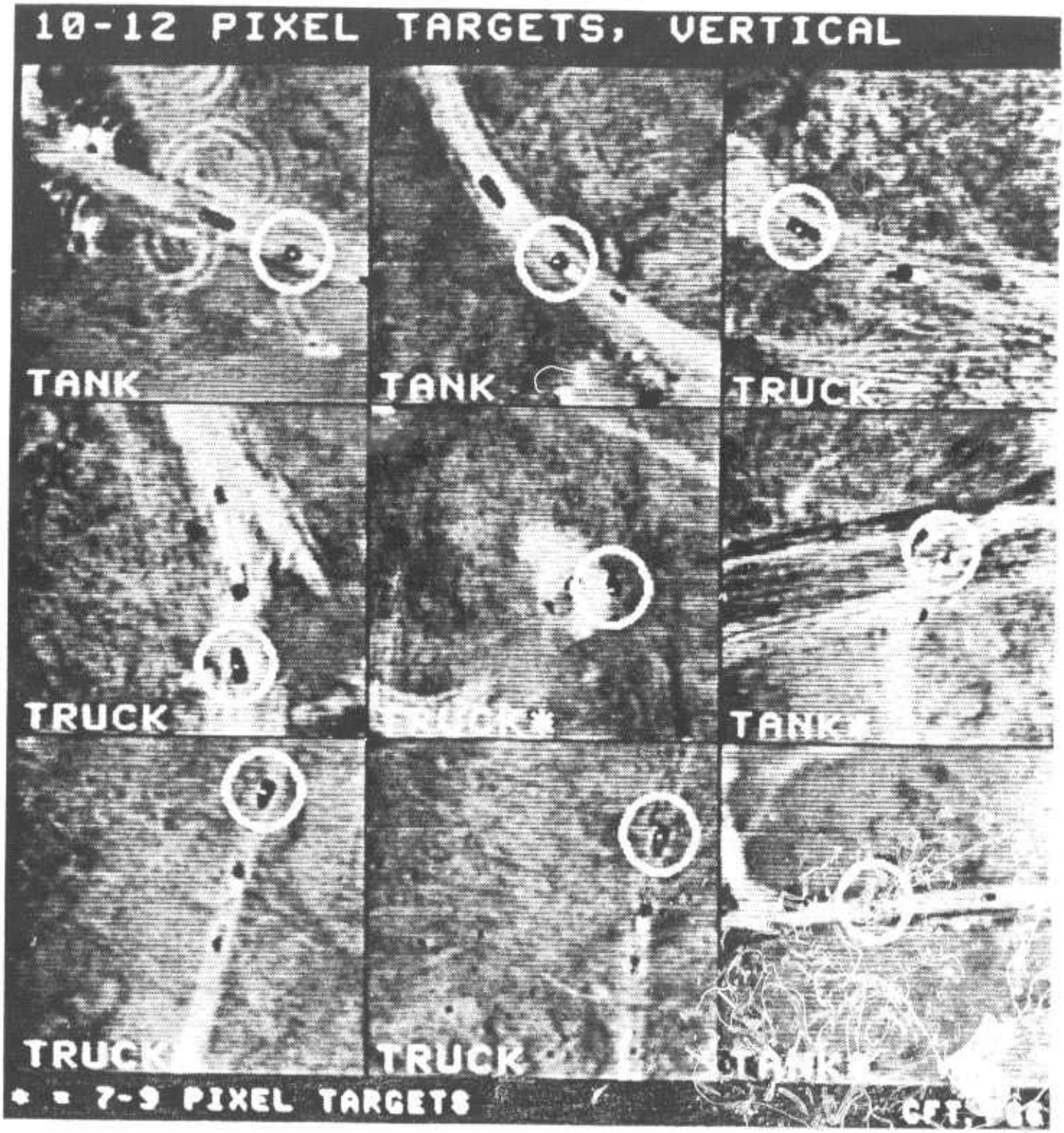

10 PIXEL TARGETS, VERTICAL ASPECT
APC
APC
CLTR
CULT
TANK
TRUCK
CLTR
CULT
TANK
* = 3 PIXEL TARGETS
CFT. PG1

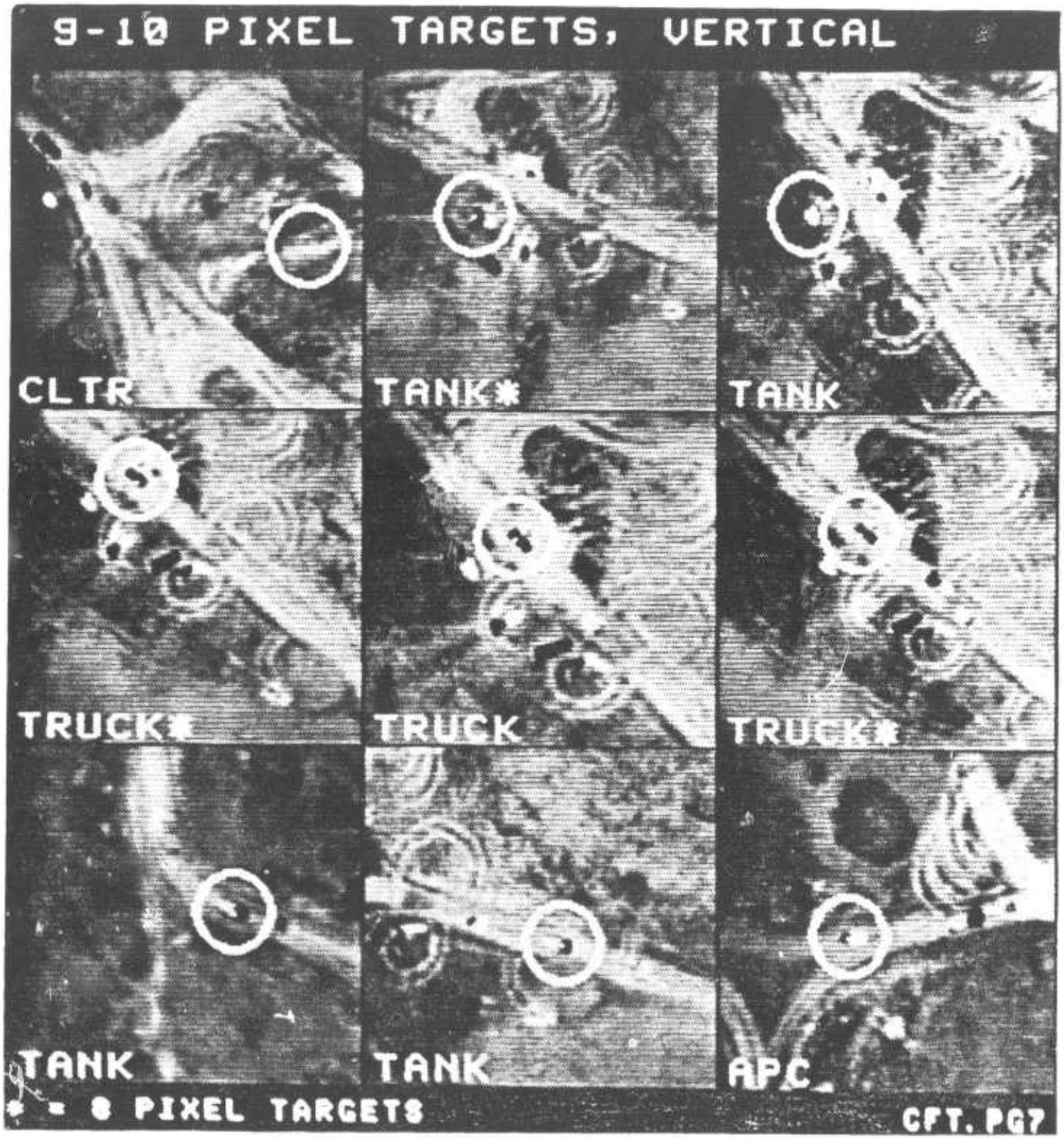

10-13 PIXEL TARGETS, NON-VERTICAL
JEEP
TANK
TRUCK
TANK*
TRU
TANK
TANK
TANK
JEEP
* = 6-8 PIXEL TARGETS
CPT. PGS

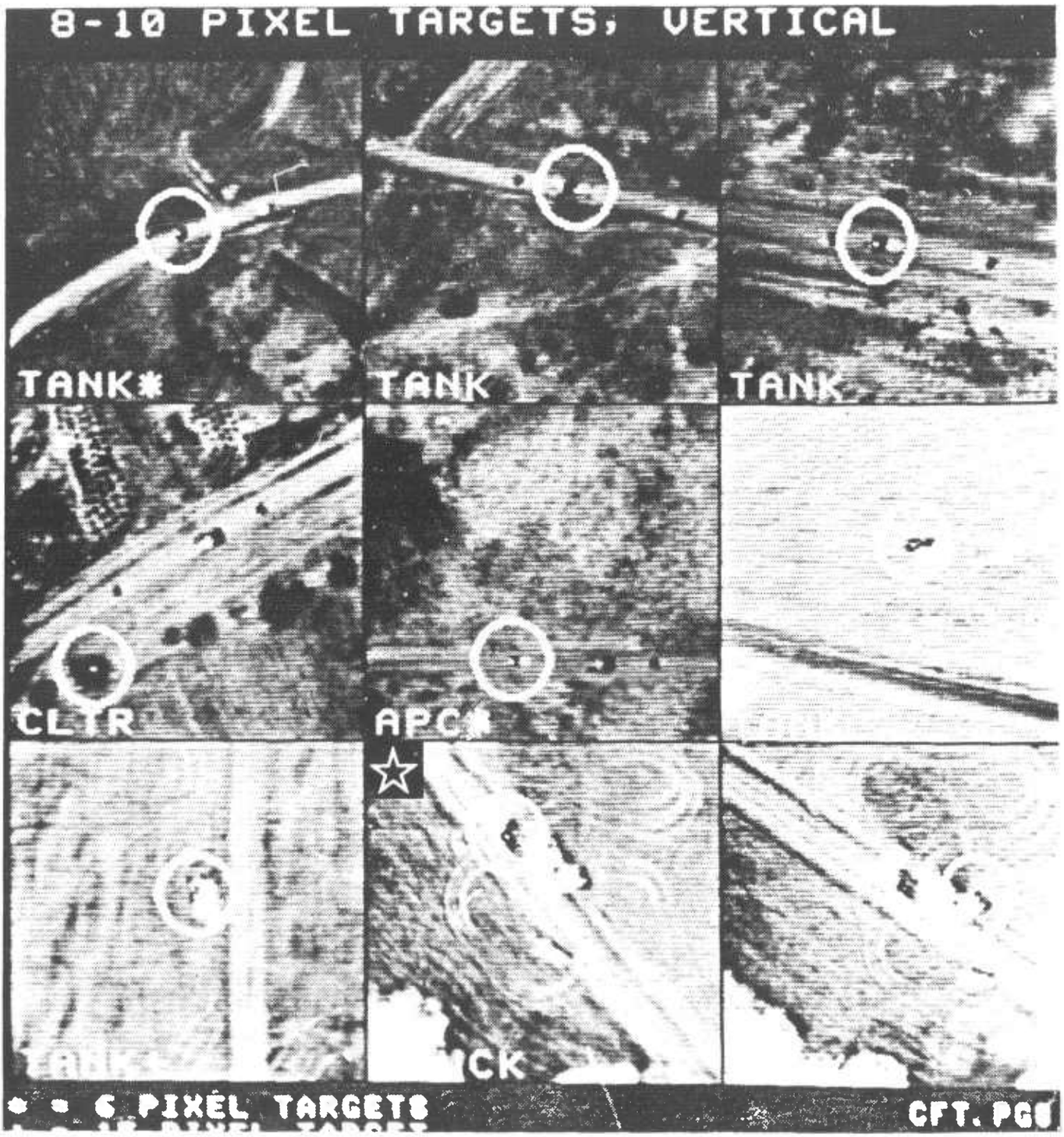

10-14 PIXEL TARGETS, NON-VERTICAL
TANK
TANK
JEEP
TANK
TRUCK
TANK
JEEP*
TANK
* = 7 PIXEL TARGET
CFT. PG3

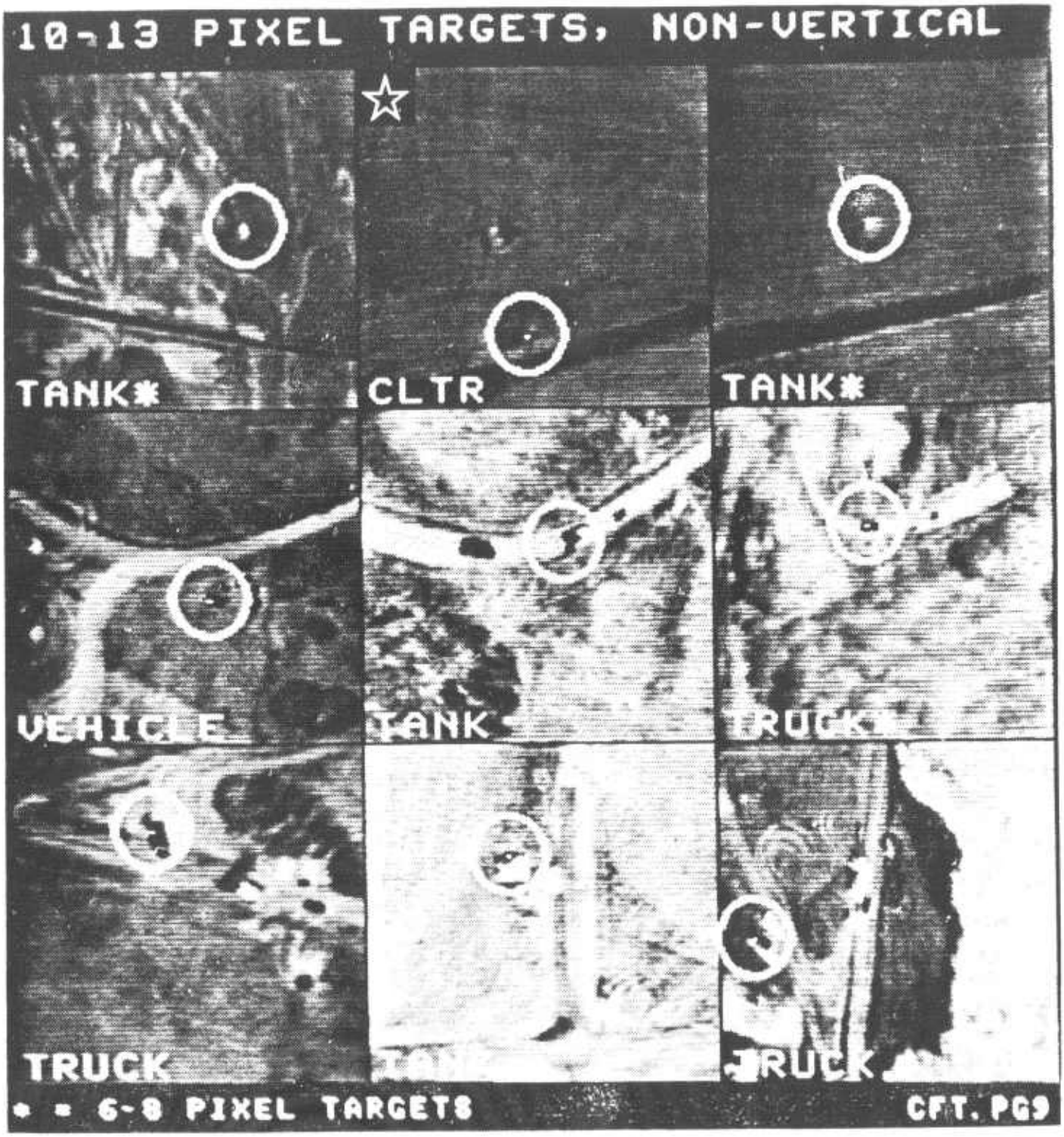

10-14 PIXEL TARGETS, VERTICAL
5-7 PIXEL TARGETS
CFT. PG4

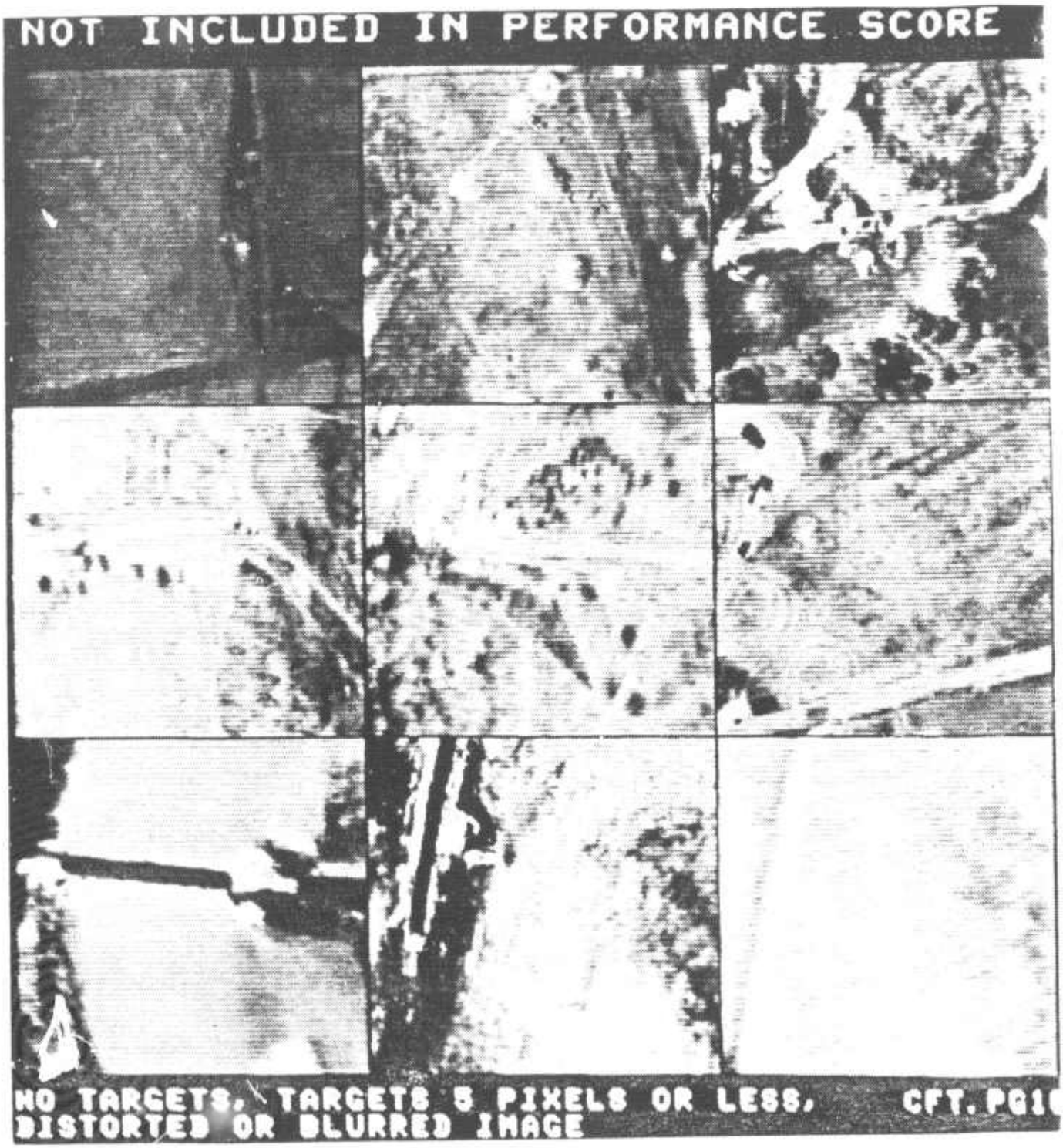

12-14 PIXEL TARGETS, VERTICAL
TANK
TANK
TANK
TANK
CLTR
TRUCK
TRU
TR
* = 7-8 PIXEL TARGETS
CFT. PG8

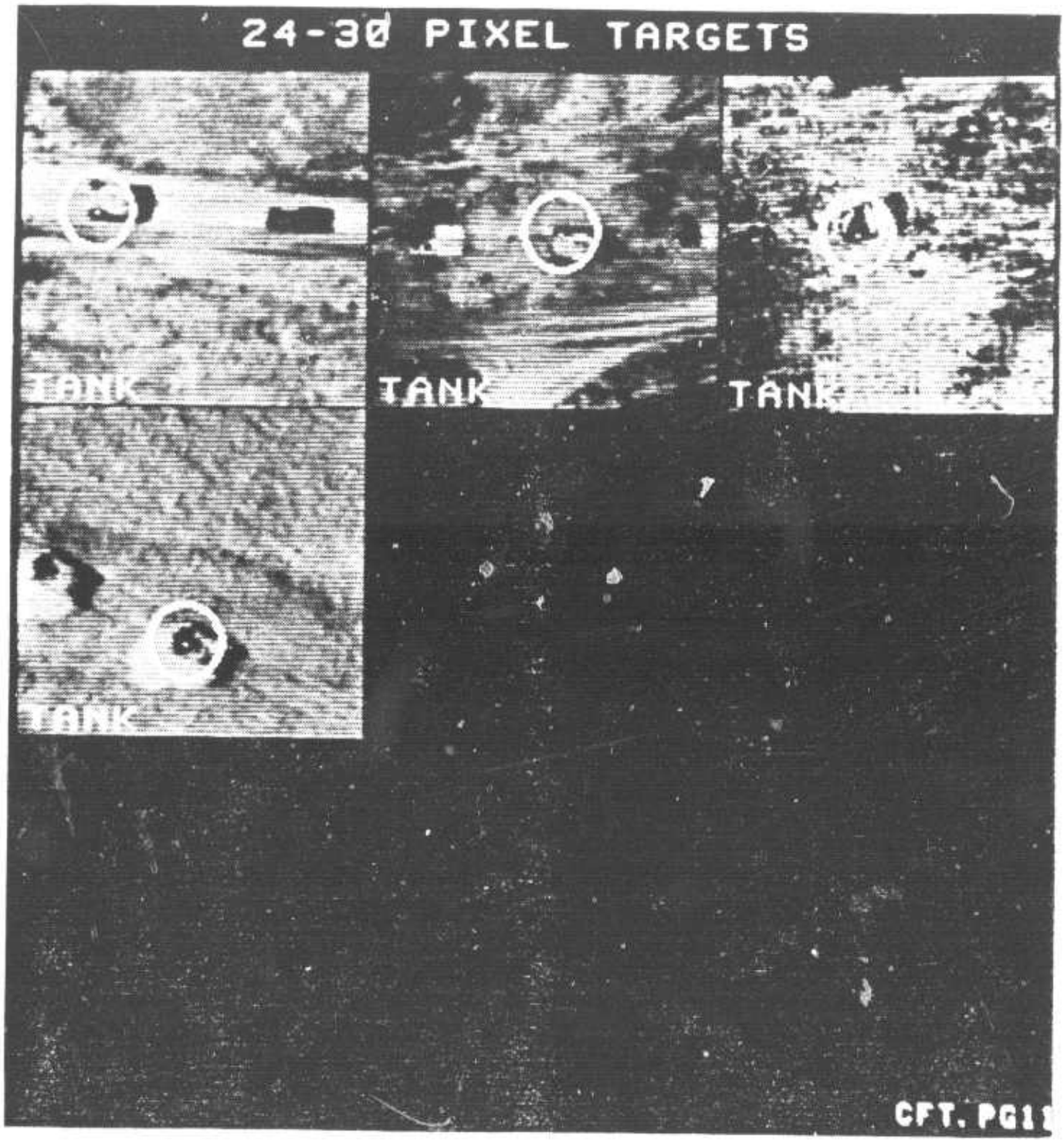

10-12 PIXEL TARGETS, VERTICAL
TANK
TANK
TRUCK
TRUCK
TANK
TRUCK
TRUCK
* = 7-9 PIXEL TARGETS

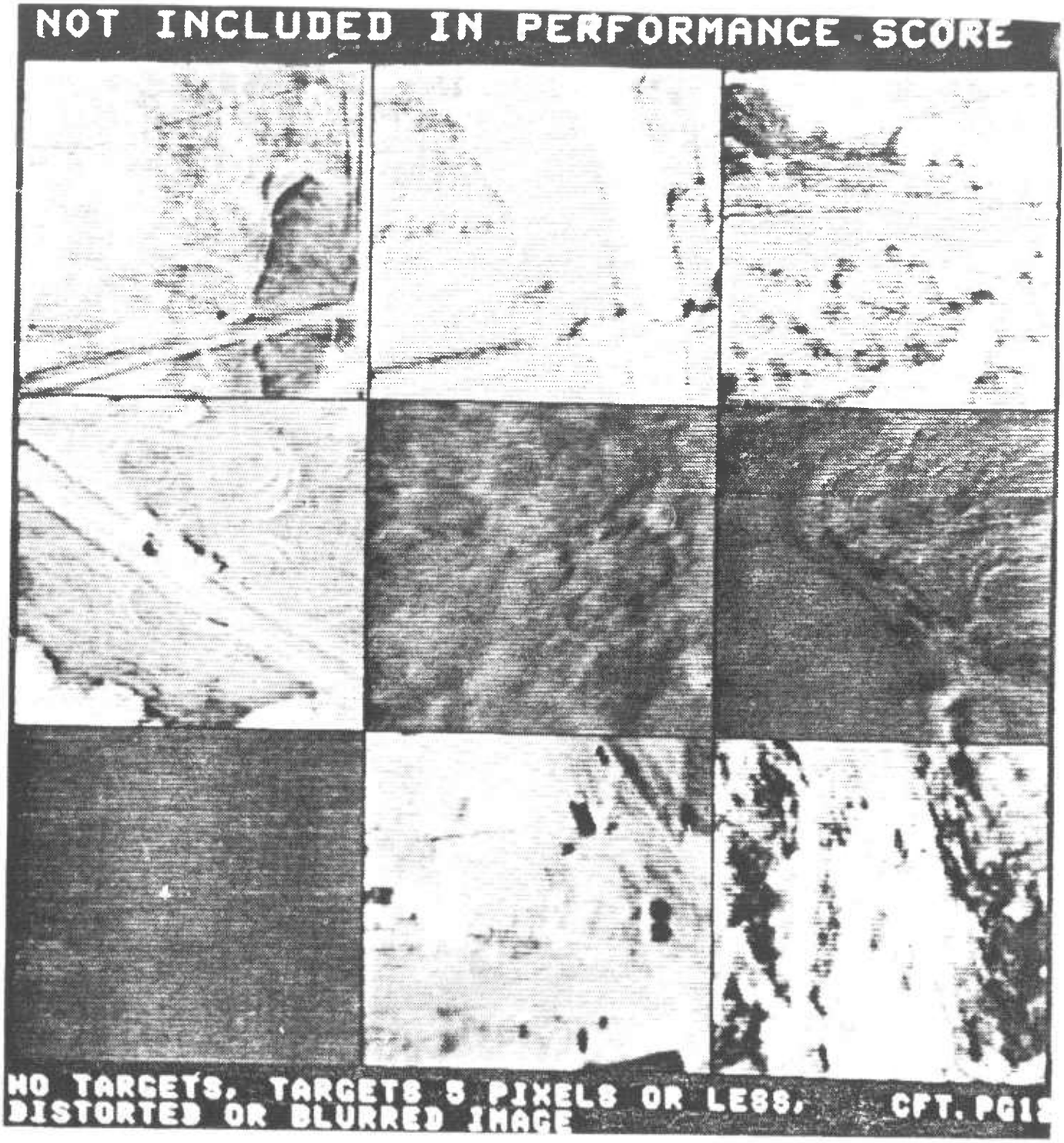

9-10 PIXEL TARGETS, VERTICAL
CLTR
TANK*
TANK
TRUCK*
TRUCK
TRUCK*
TANK
TANK
APC
* = 8 PIXEL TARGETS
CFT. PG7

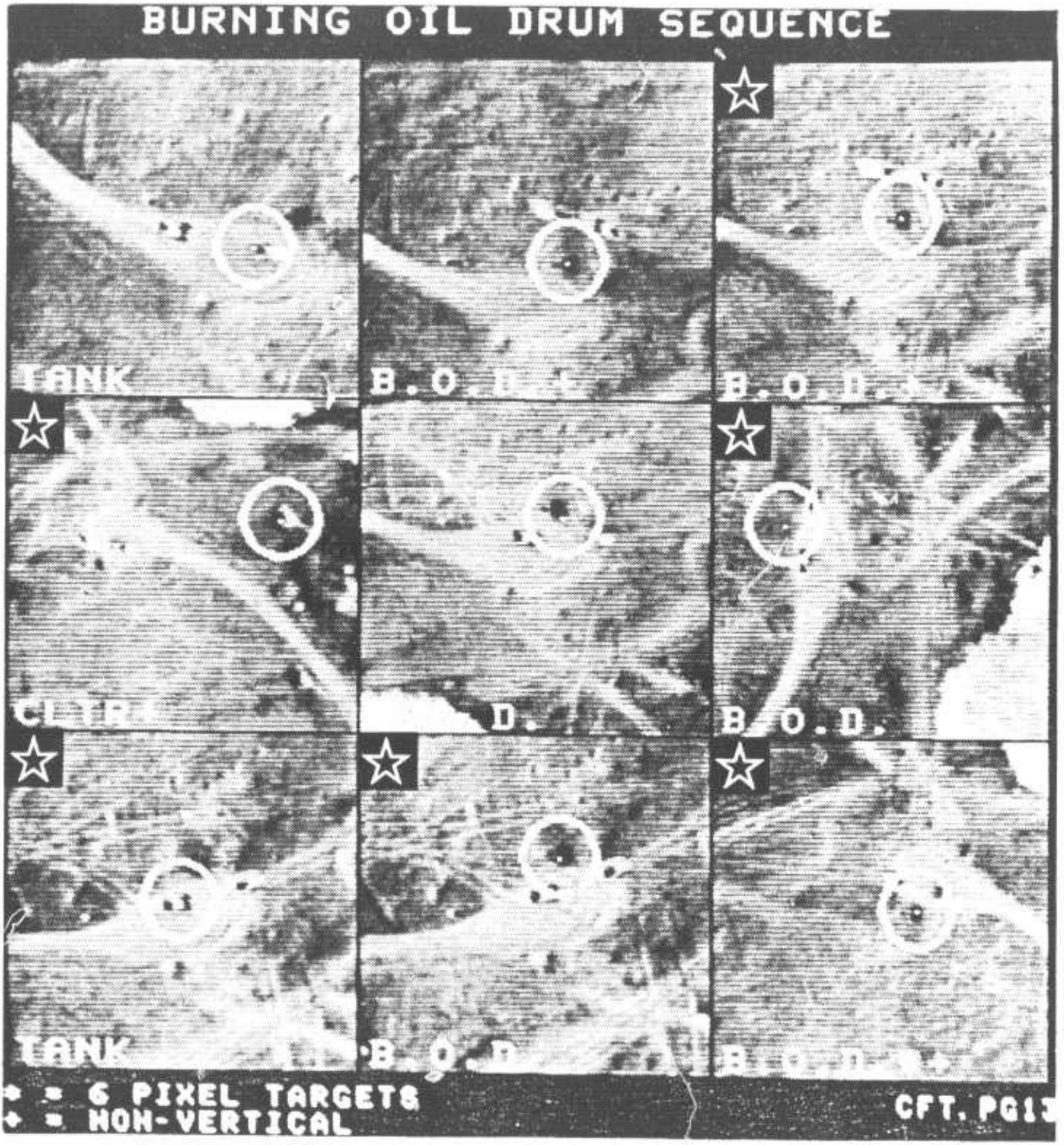

8-10 PIXEL TARGETS; VERTICAL
TANK*
TANK
TANK
CLTR
APC
CK
* = 6 PIXEL TARGETS
CFT. PG

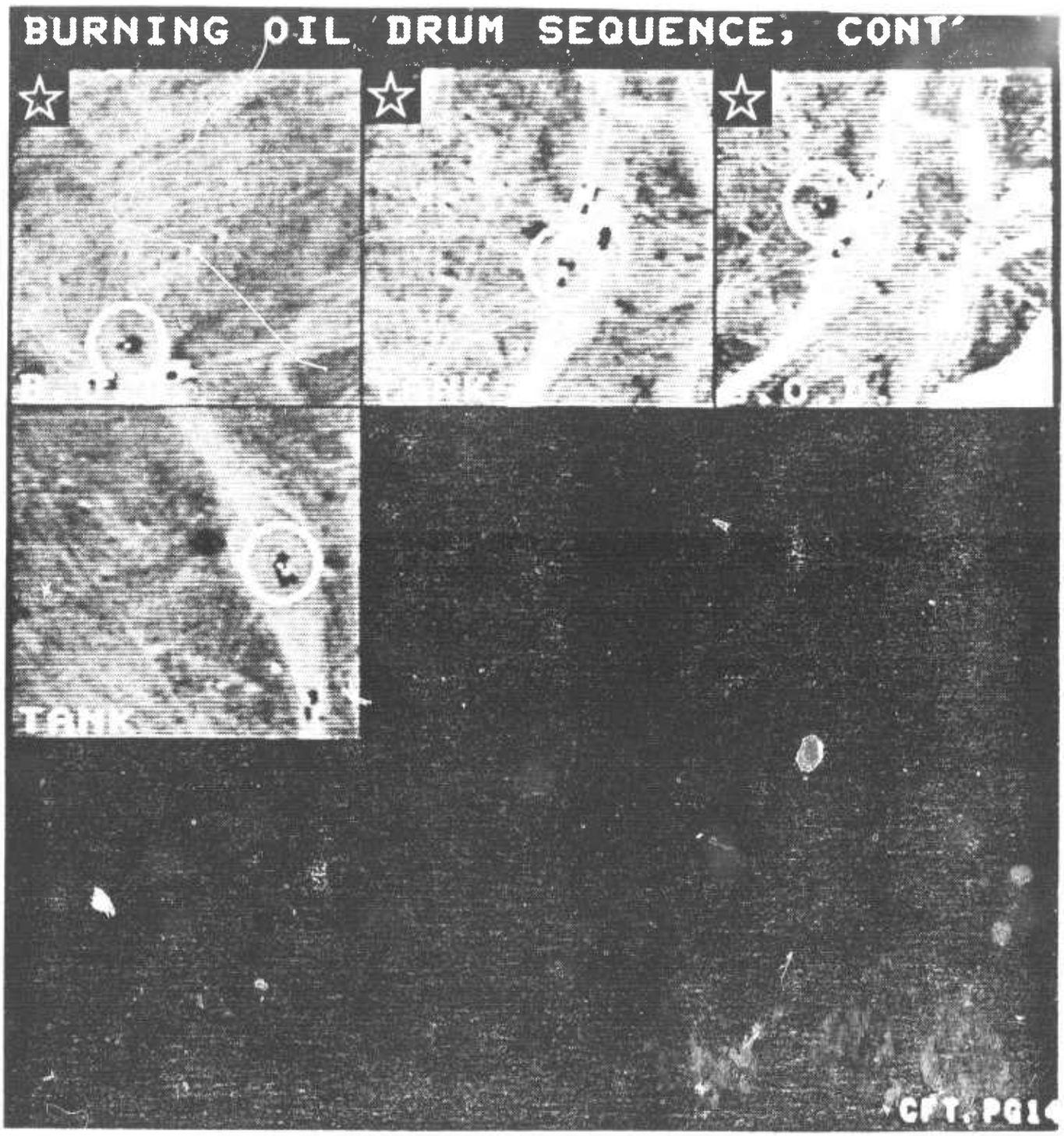

10-13 PIXEL TARGETS, NON-VERTICAL
TANK*
CLTR
TANK*
VEHICLE
TANK
TRUCK
TRUCK
TRUCK
* = 6-8 PIXEL TARGETS
CFT. PG9

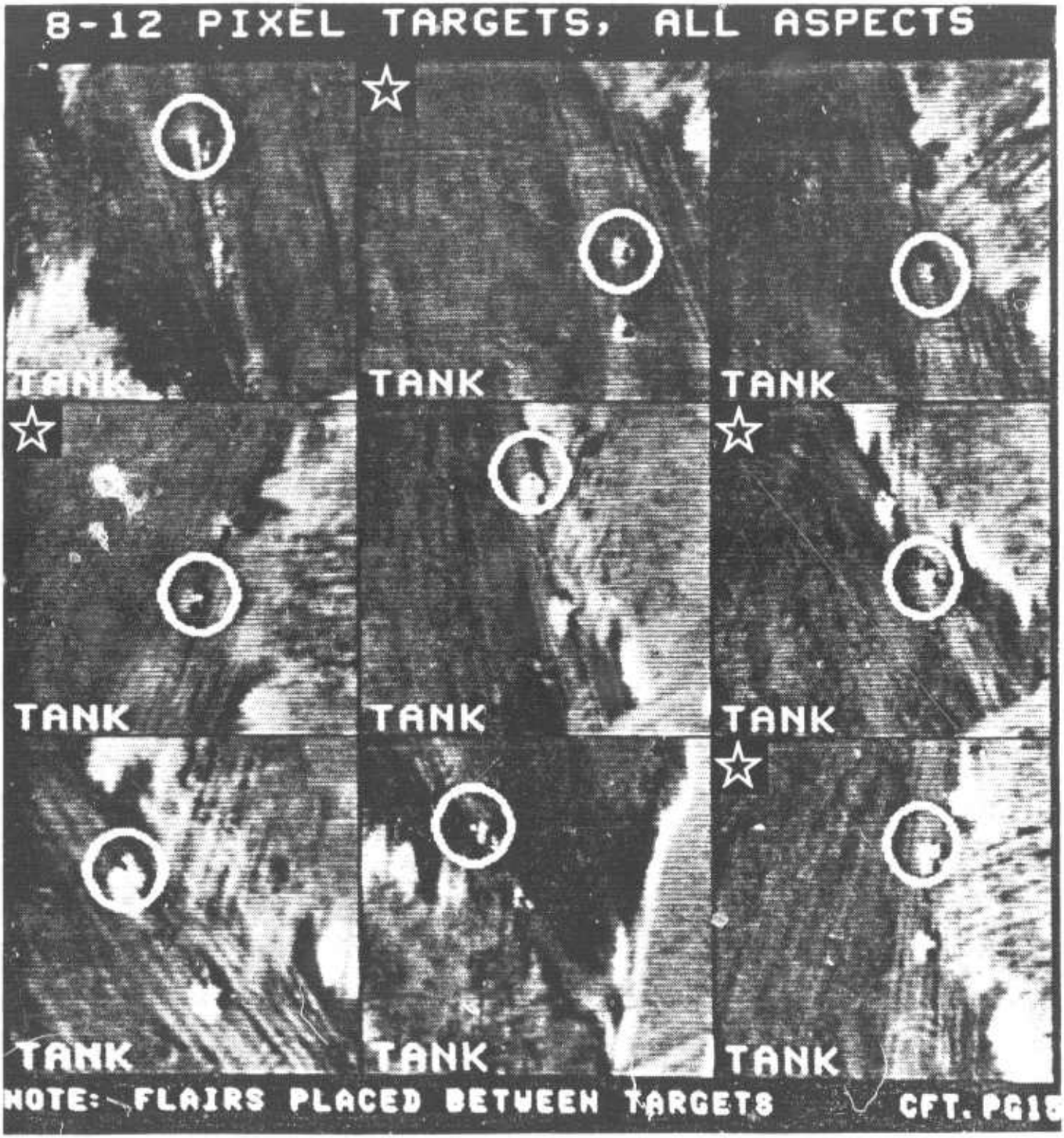

NOT INCLUDED IN PERFORMANCE SCORE
NO TARGETS, TARGETS 5 PIXELS OR LESS, CFT. PG1
DISTORTED OR BLURRED IMAGE

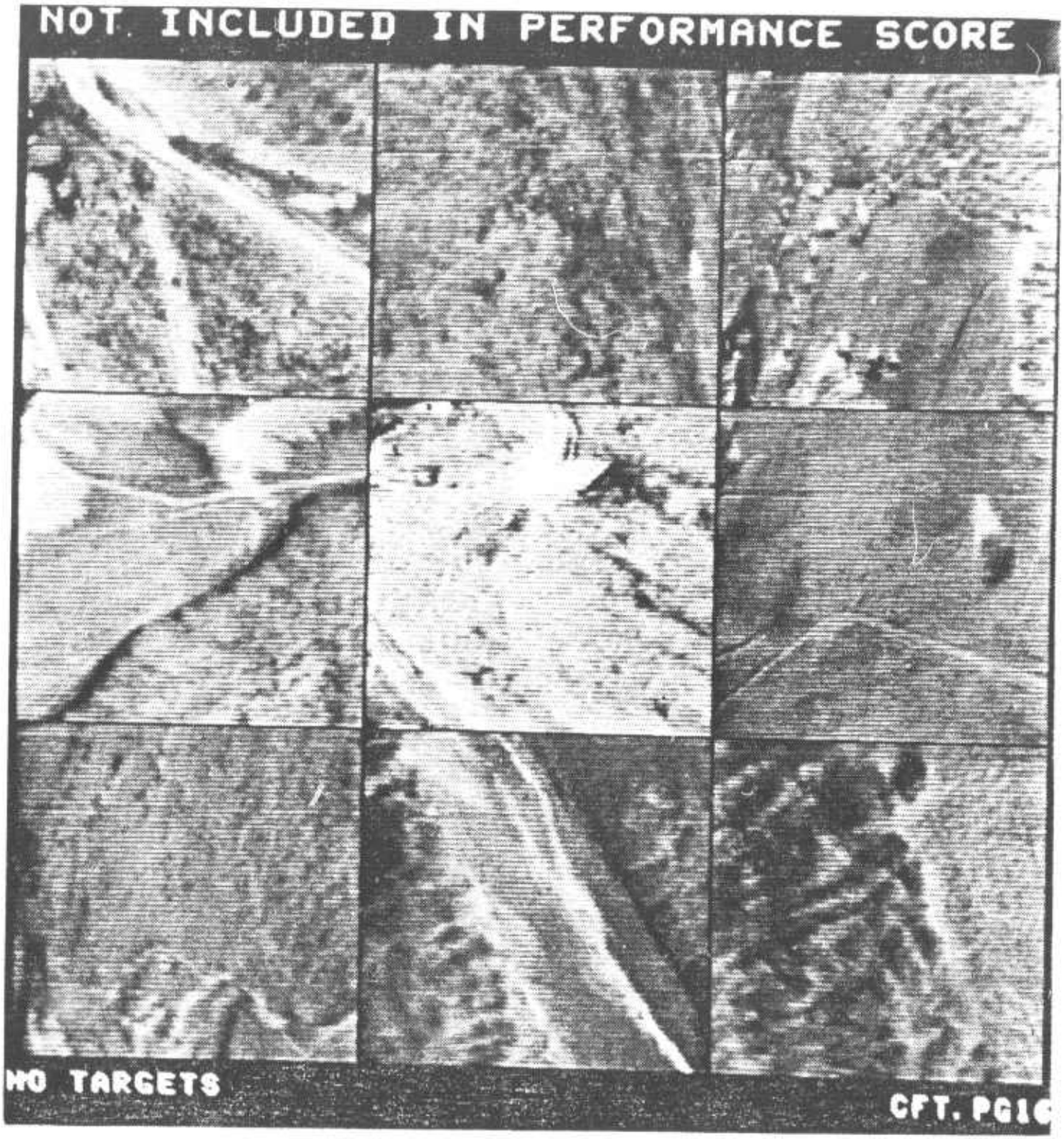

24-30 PIXEL TARGETS
TANK
TANK
TANK
TANK
CFT. PG1

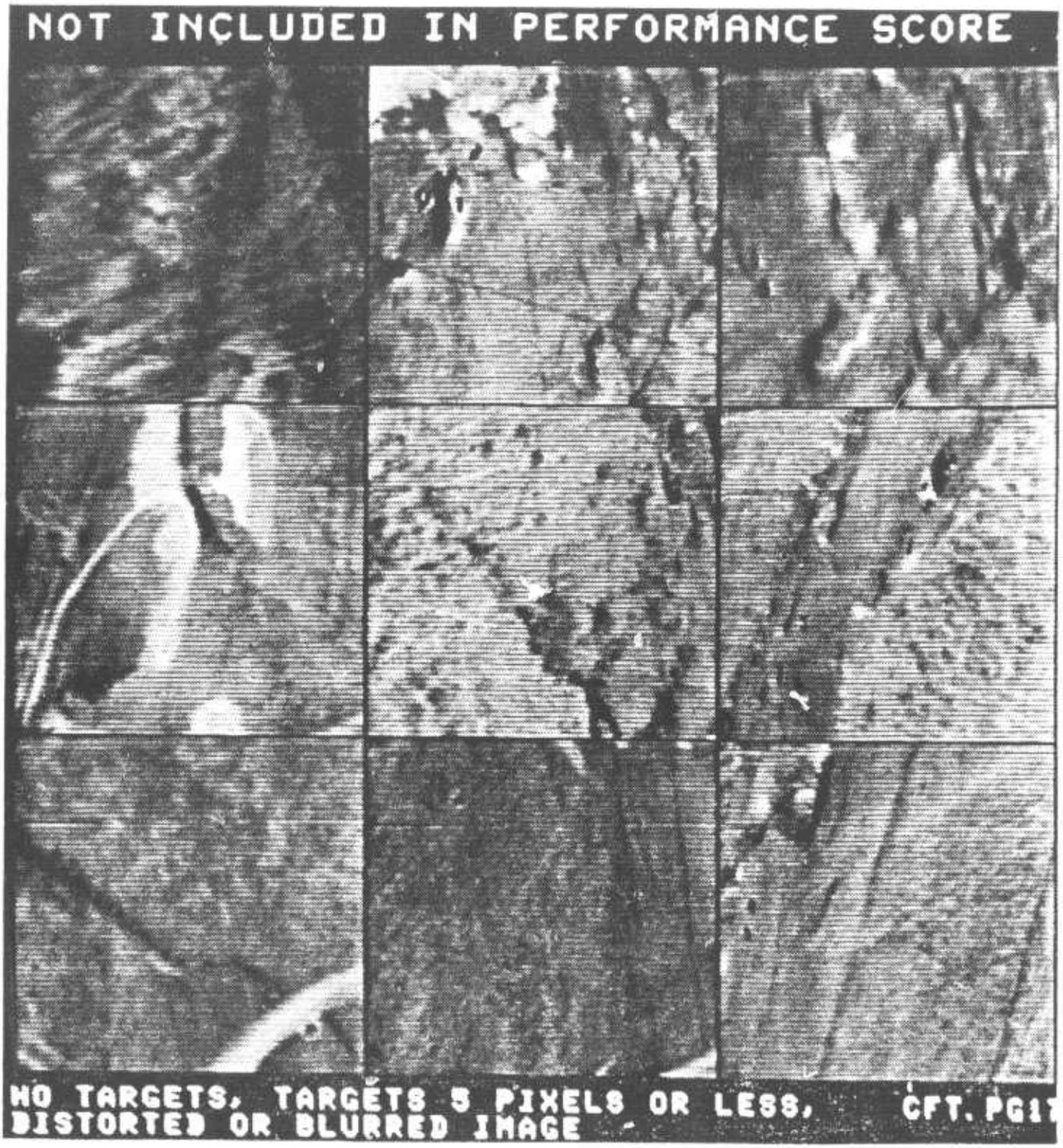

NOT INCLUDED IN PERFORMANCE SCORE
NO TARGETS, TARGETS 5 PIXELS OR LESS, CFT. PG1
DISTORTED OR BLURRED IMAGE

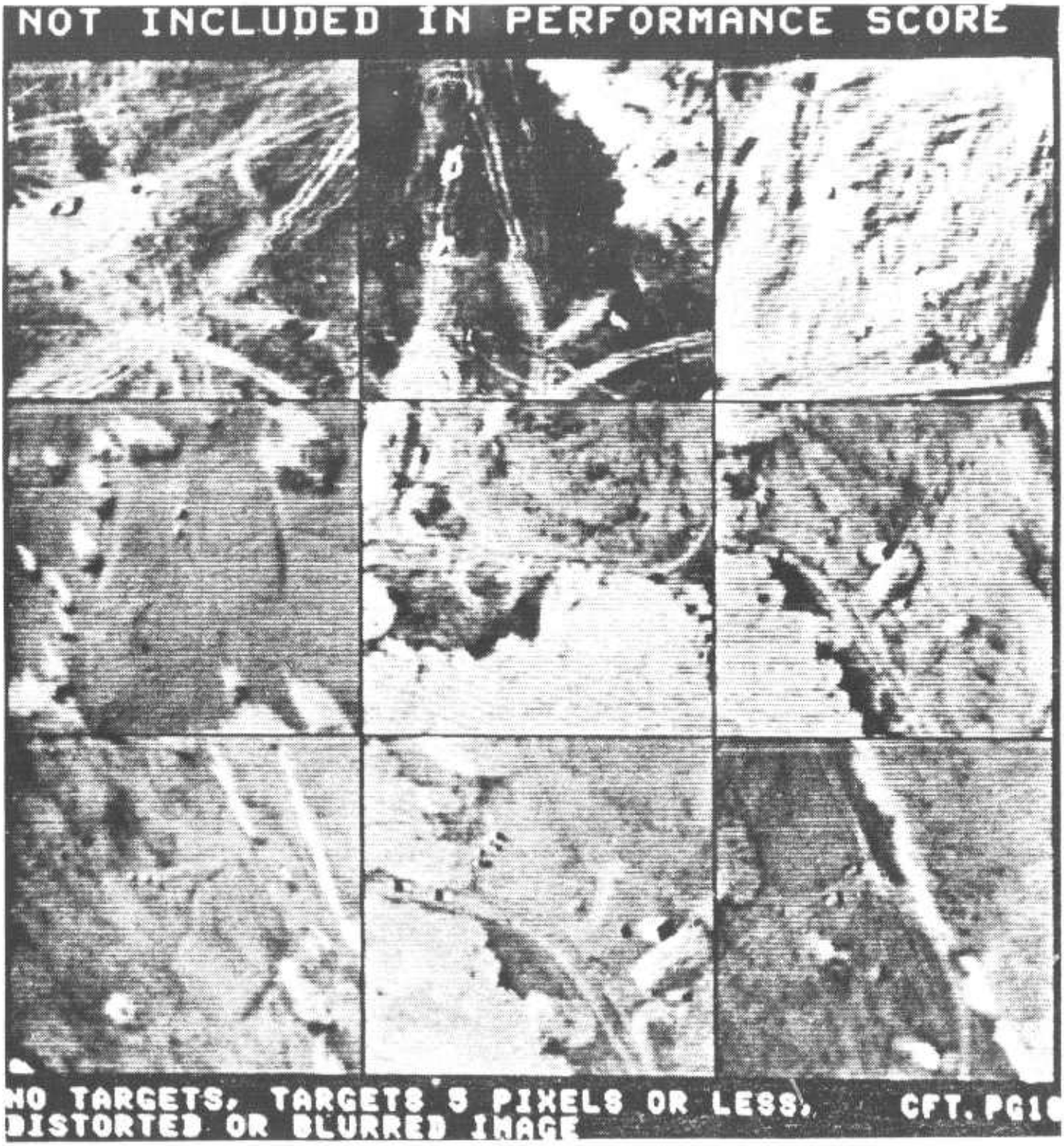

BURNING OIL DRUM SEQUENCE
TANK
B.O.D.
B.O.D.
D.
B.O.D.
TANK
= 6 PIXEL TARGETS
= NON-VERTICAL
CFT. PG1


BURNING OIL DRUM SEQUENCE, CONT'
TANK
CPT. PGI

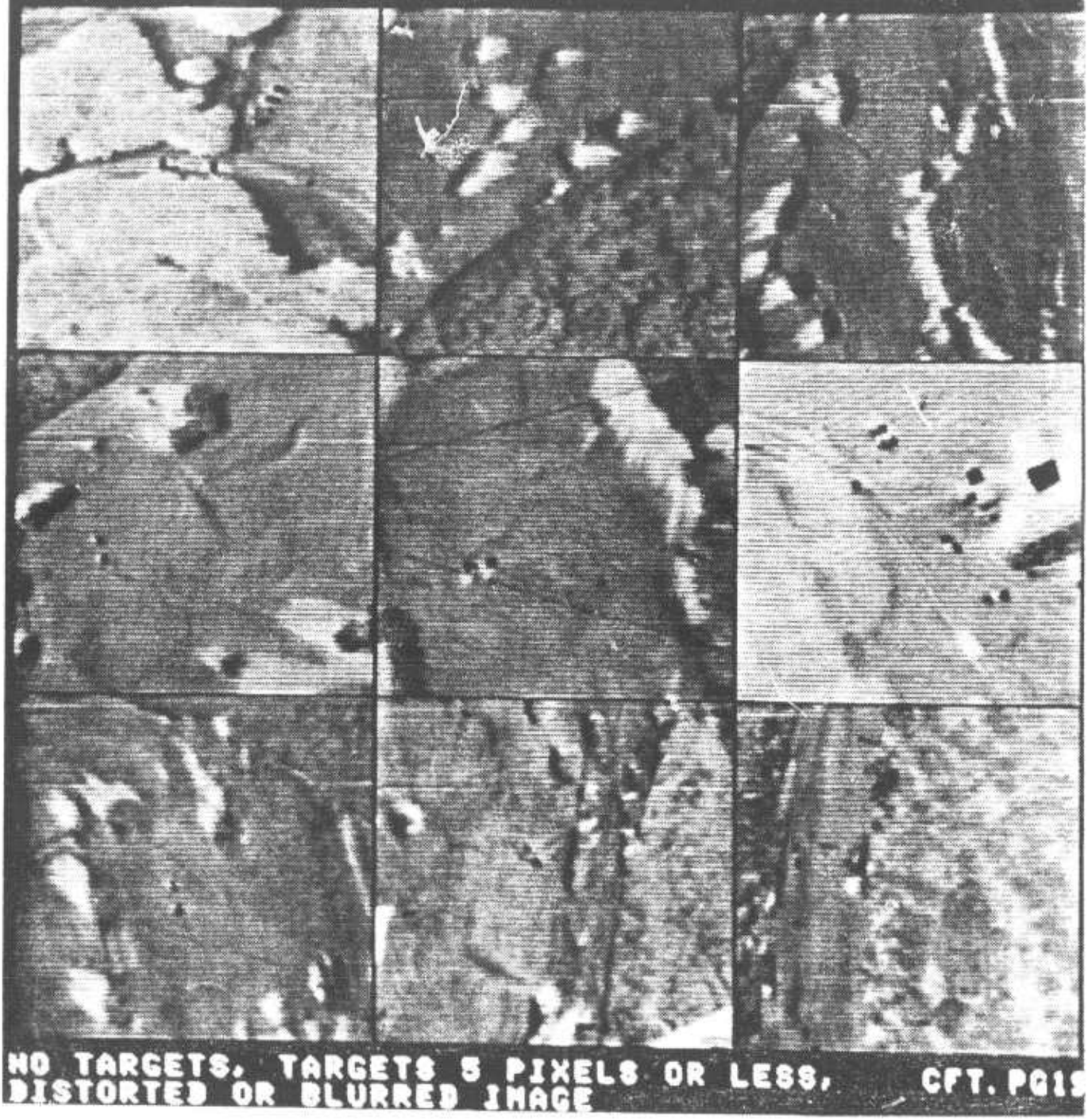

8-12 PIXEL TARGETS, ALL ASPECTS
TANK
TANK
TANK
TANK
TANK
TANK
TANK
TANK
TANK
NOTE: FLAIRS PLACED BETWEEN TARGETS
CFT. PG1


NOT INCLUDED IN PERFORMANCE SCORE
NO TARGETS
CFT. PGIC


NOT INCLUDED IN PERFORMANCE SCORE
NO TARGETS, TARGETS 5 PIXELS OR LESS,
DISTORTED OR BLURRED IMAGE
CFT. PG1

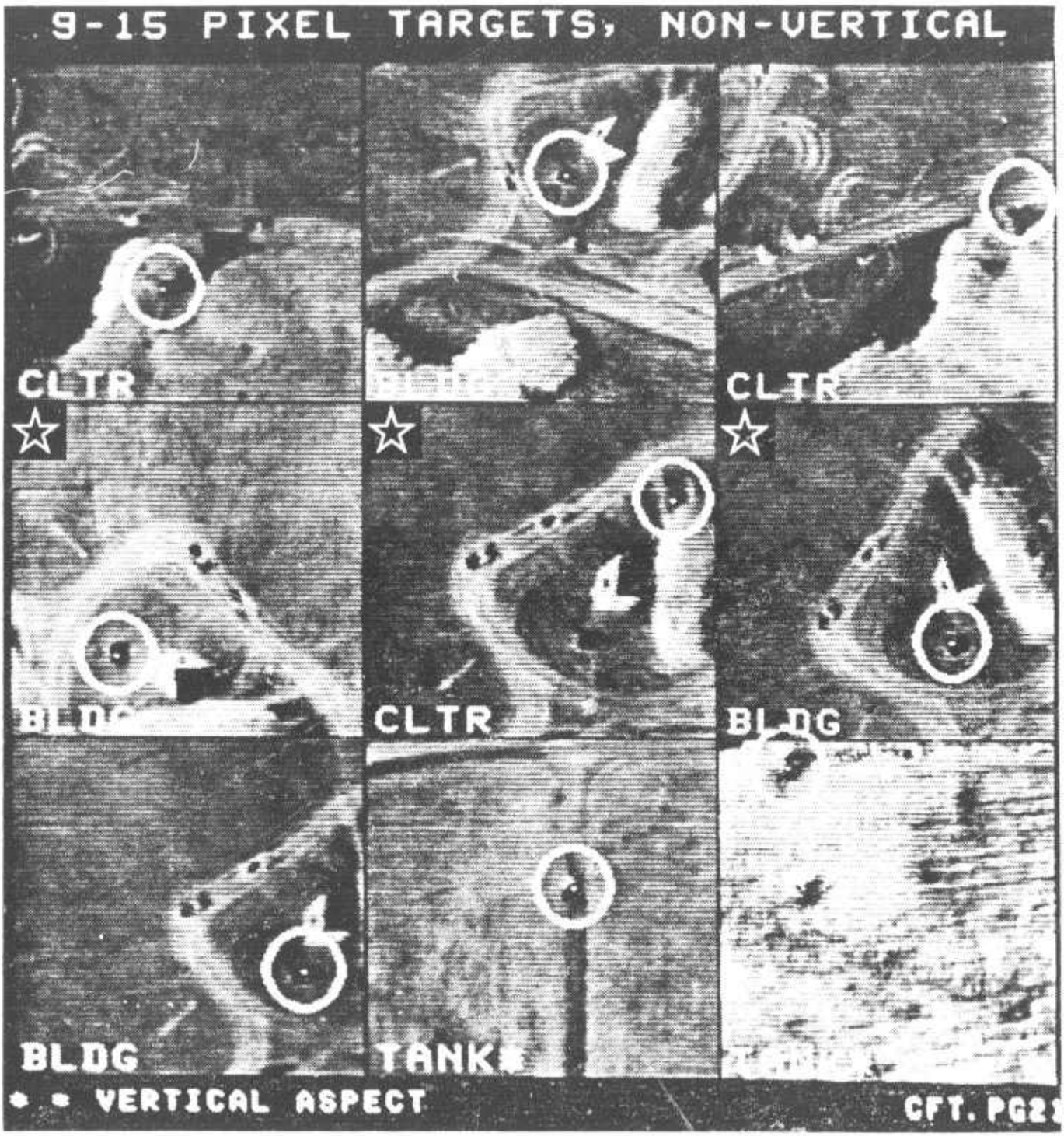

NOT INCLUDED IN PERFORMANCE SCORE
NO TARGETS, TARGETS 5 PIXELS OR LESS,
DISTORTED OR BLURRED IMAGE
CFT. PG1

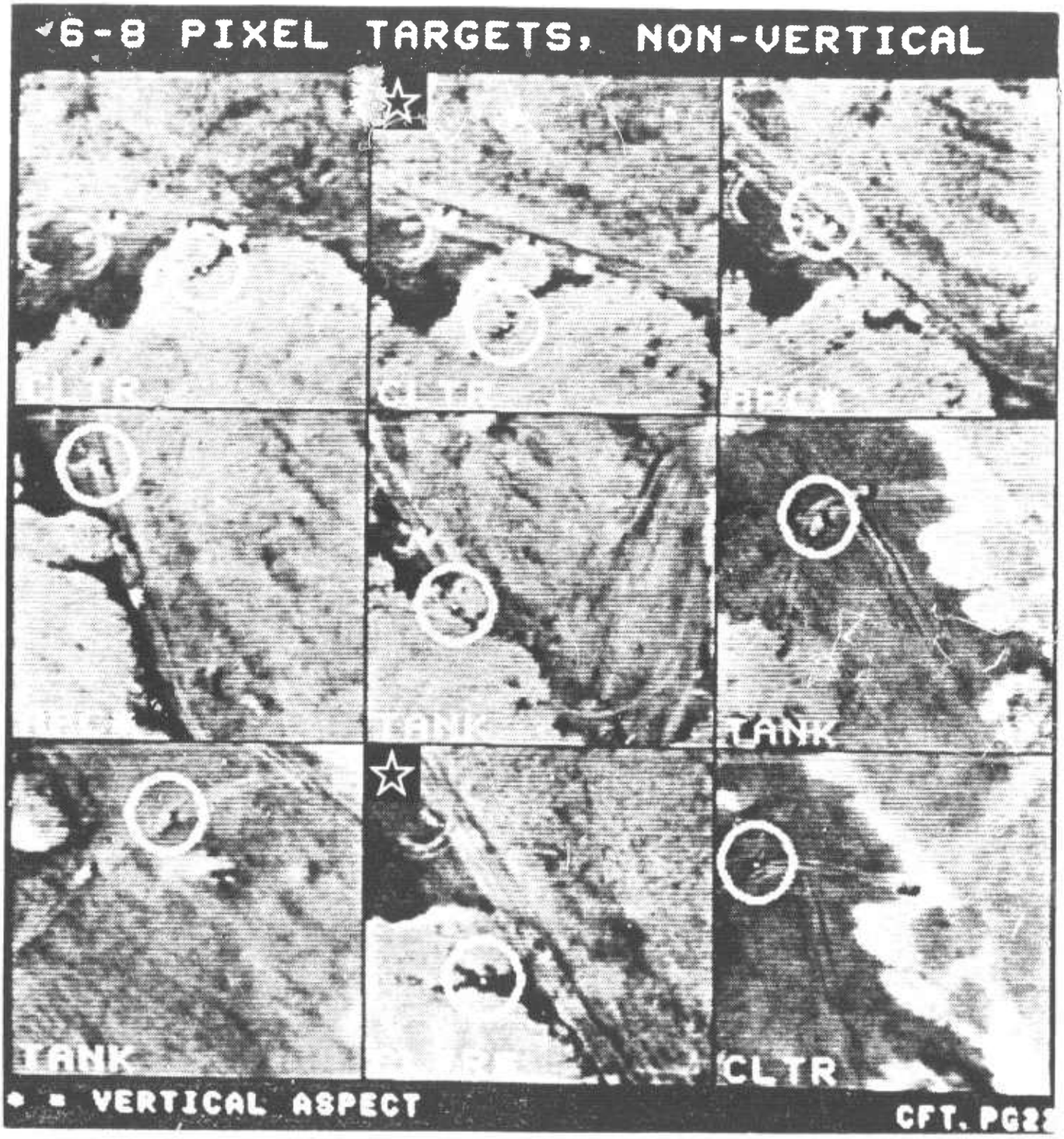
NOT INCLUDED IN PERFORMANCE SCORE

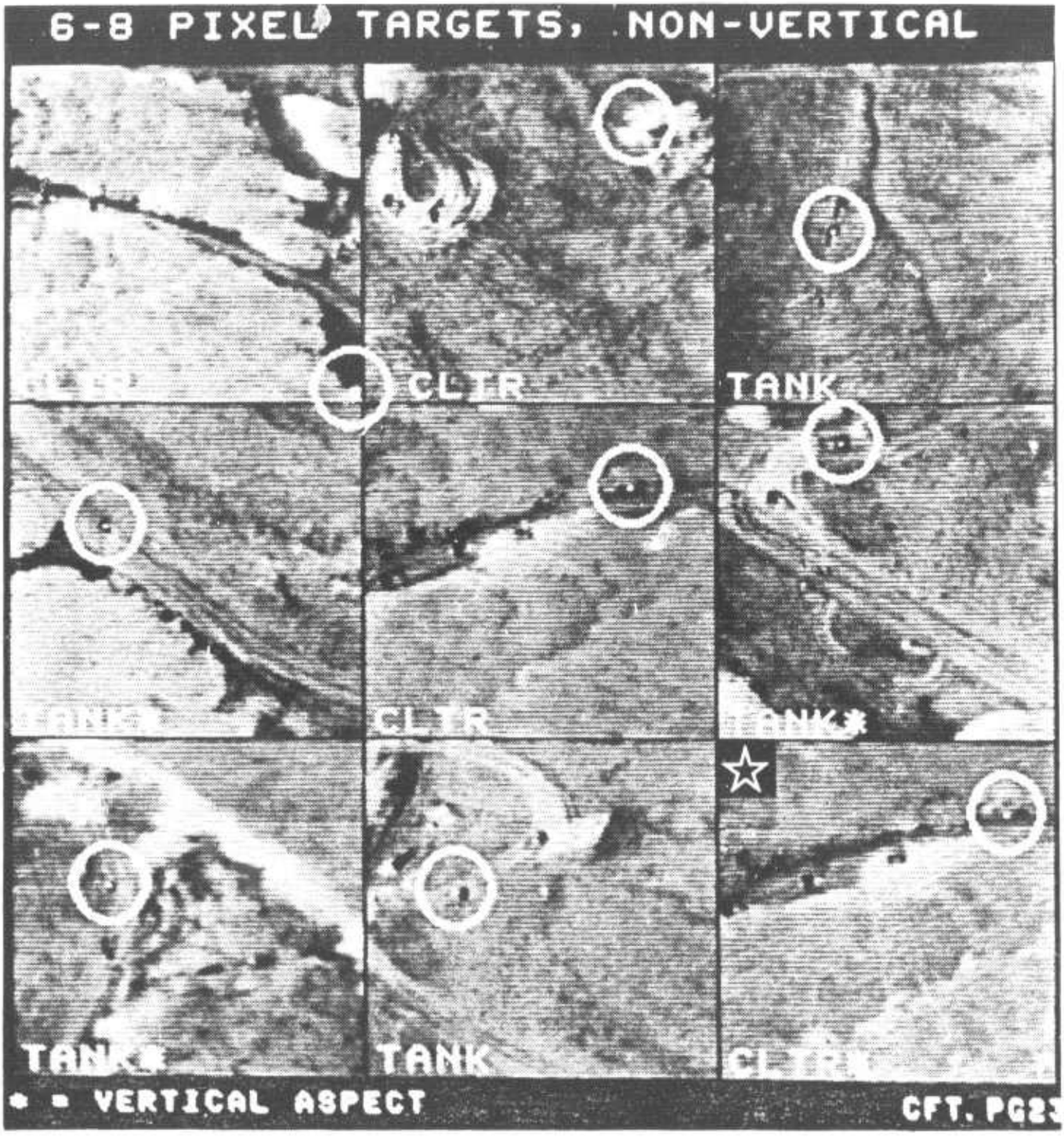


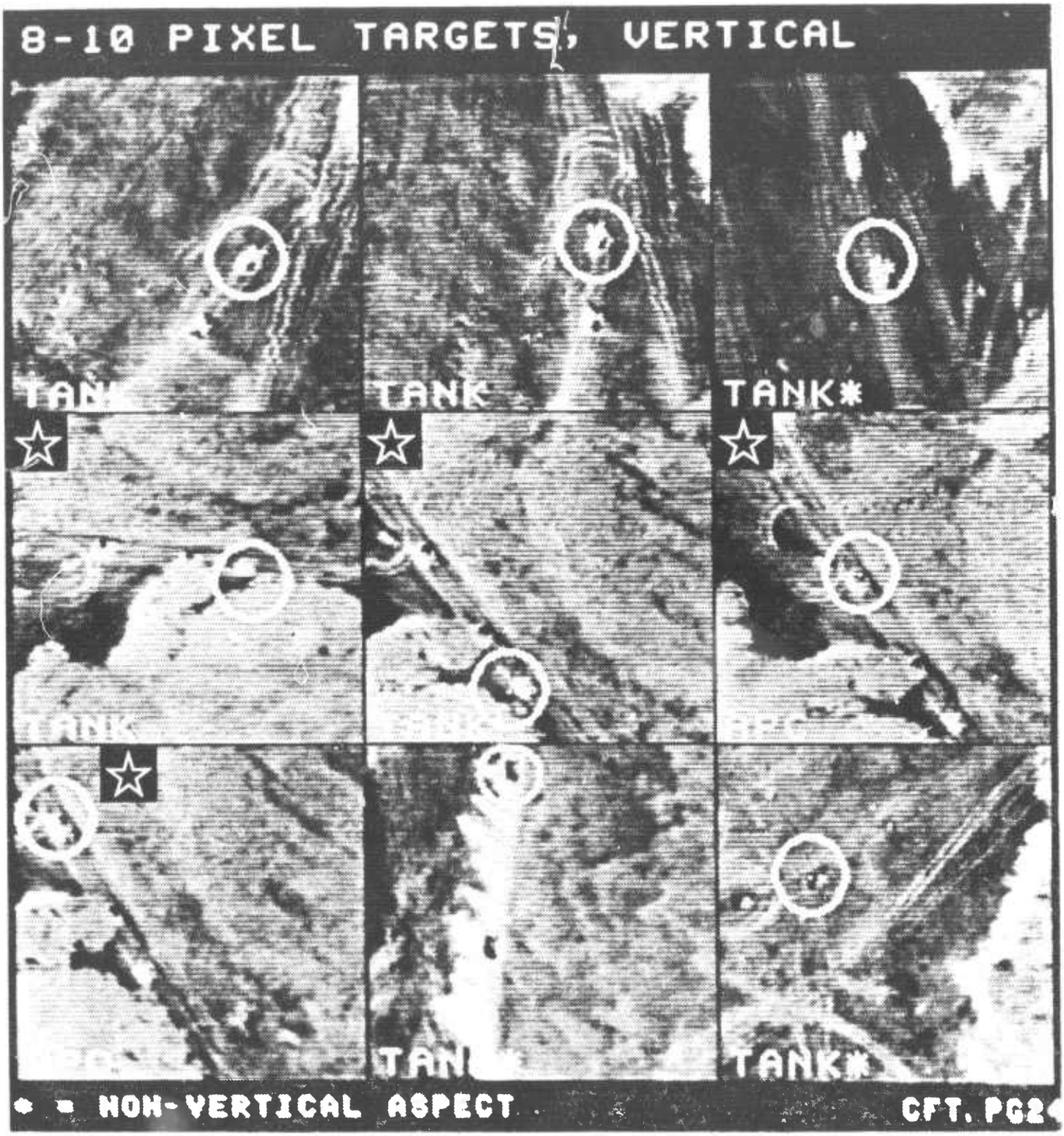
NO TARGETS, TARGETS 5 PIXELS OR LESS, CFT. PG1
DISTORTED OR BLURRED IMAGE

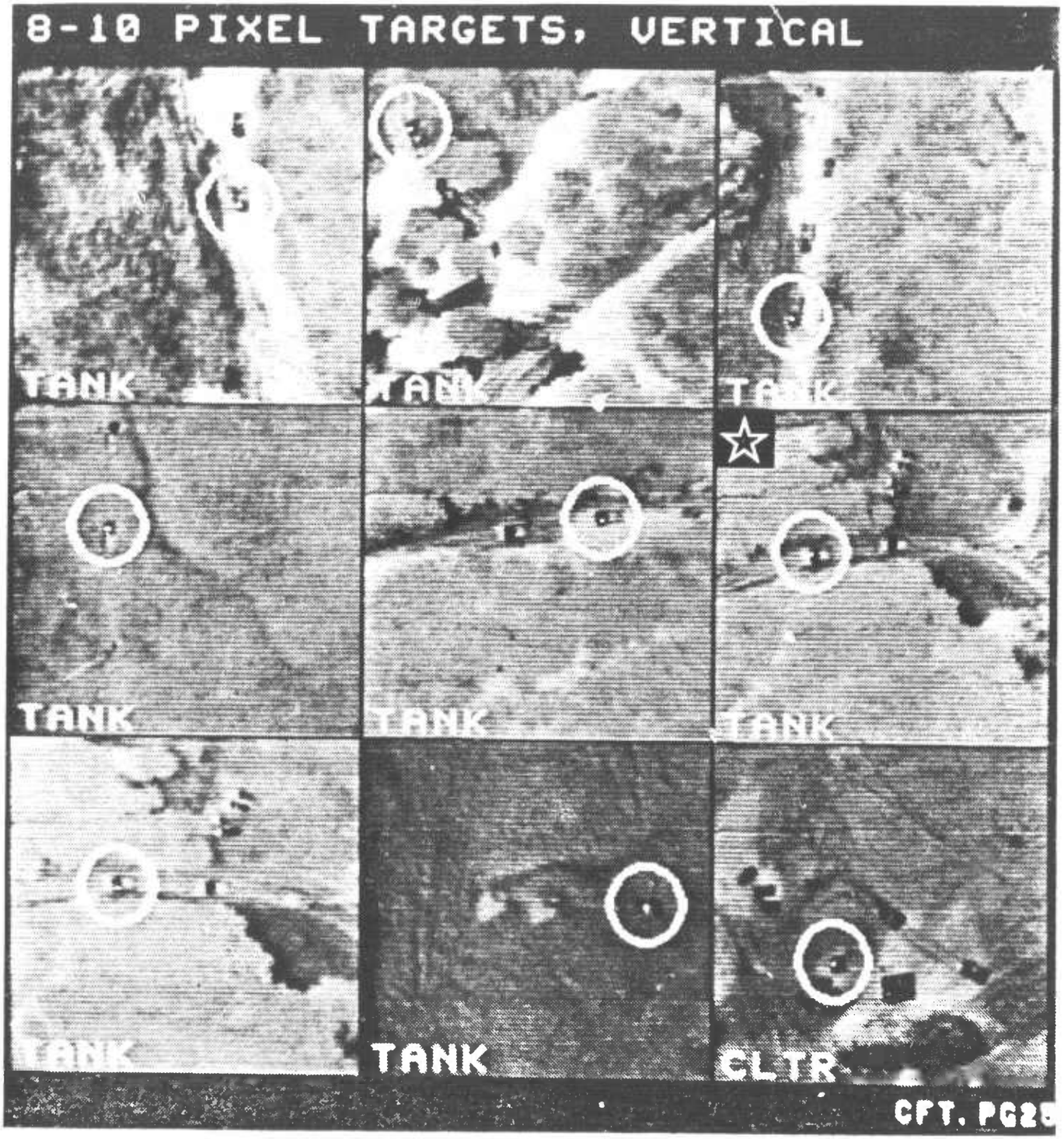

NOT INCLUDED IN PERFORMANCE SCORE
NO TARGETS, TARGETS 5 PIXELS OR LESS,
DISTORTED OR BLURRED IMAGE
CFT. PG2

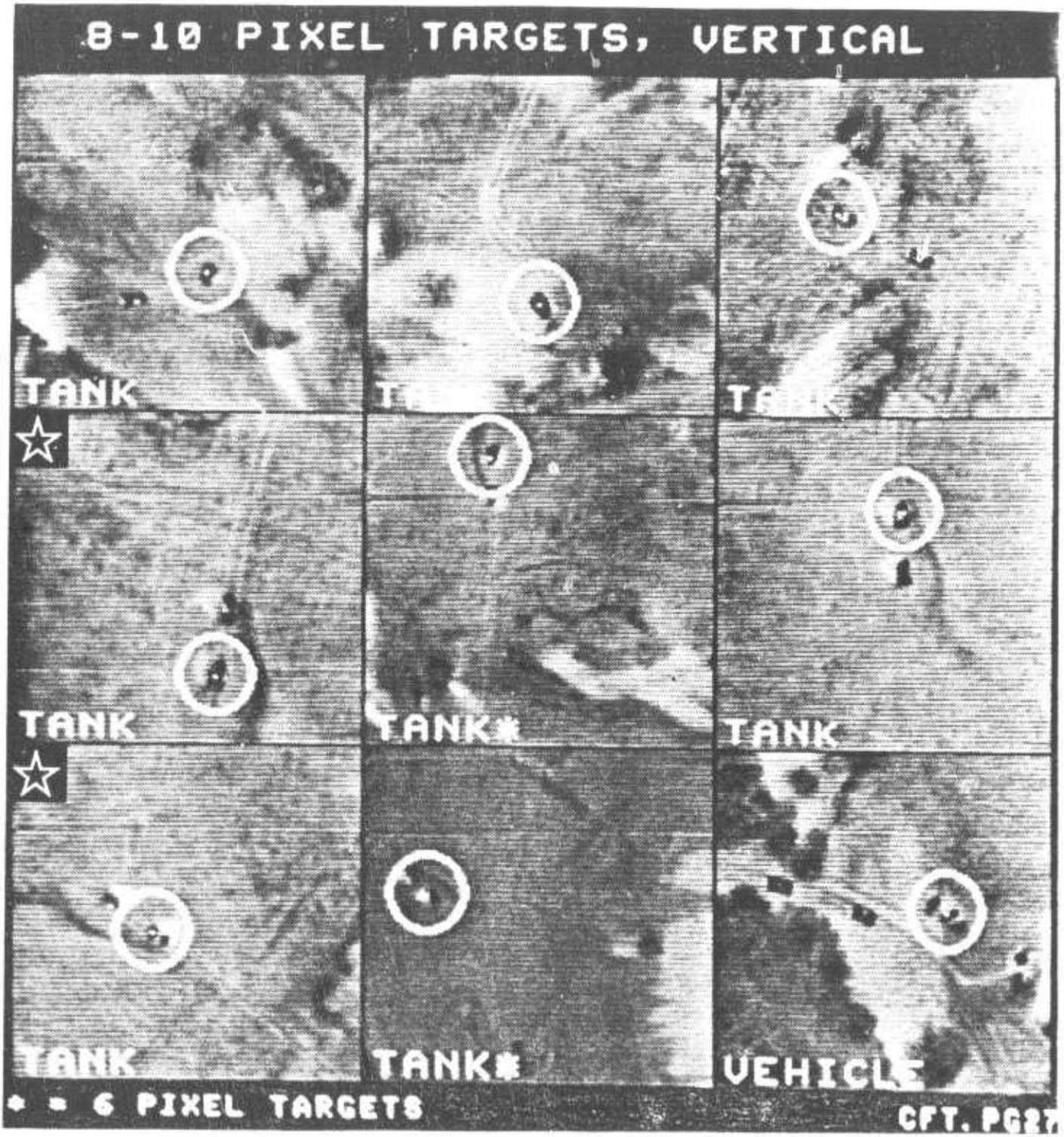

9-15 PIXEL TARGETS, NON-VERTICAL
CLTR
CLTR
BLDG
CLTR
BLDG
BLDG
TANK*
* = VERTICAL ASPECT
CFT. PG2

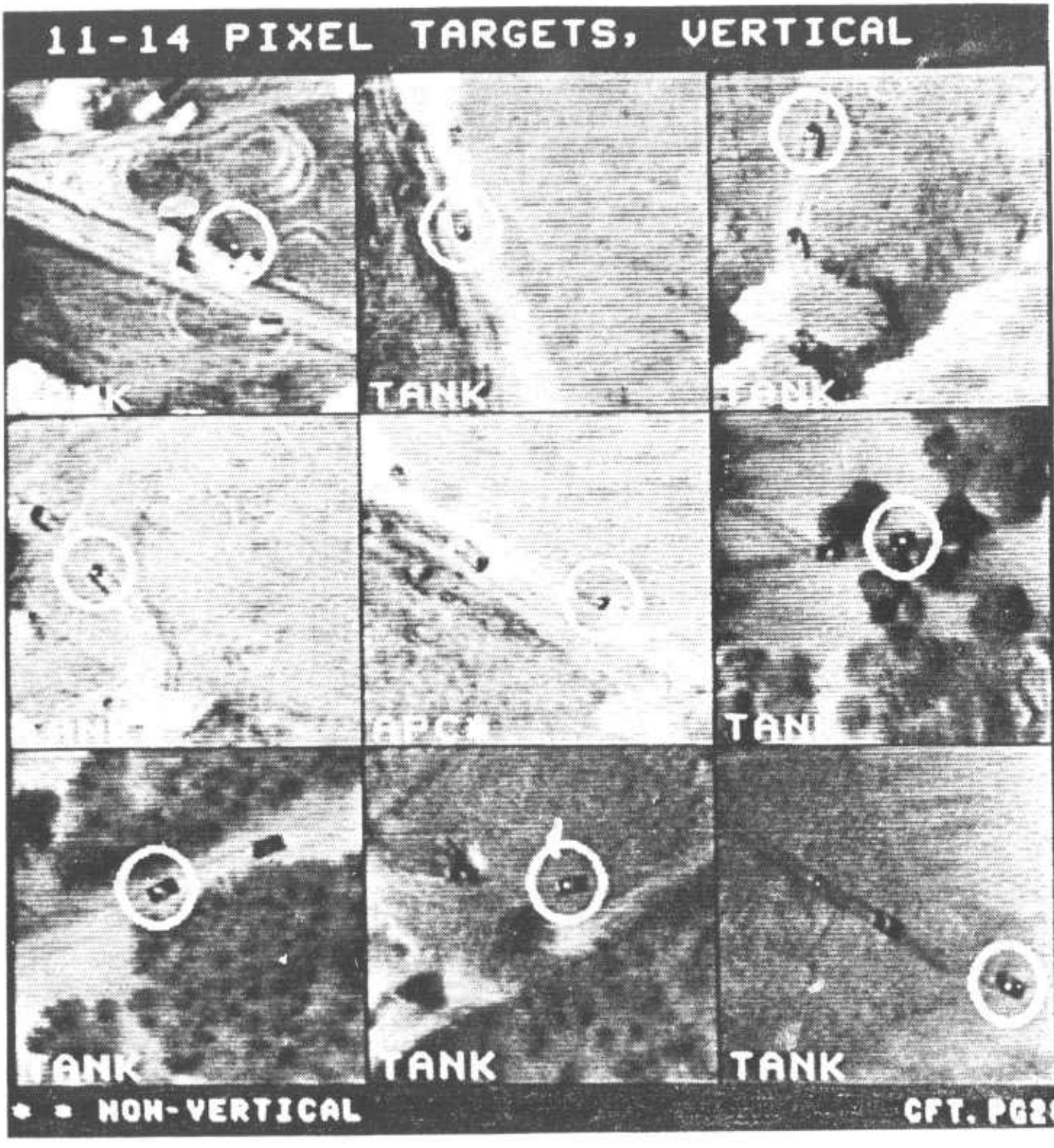

6-8 PIXEL TARGETS, NON-VERTICAL
TANK
TANK
CLTR
* = VERTICAL ASPECT
CFT. PG2

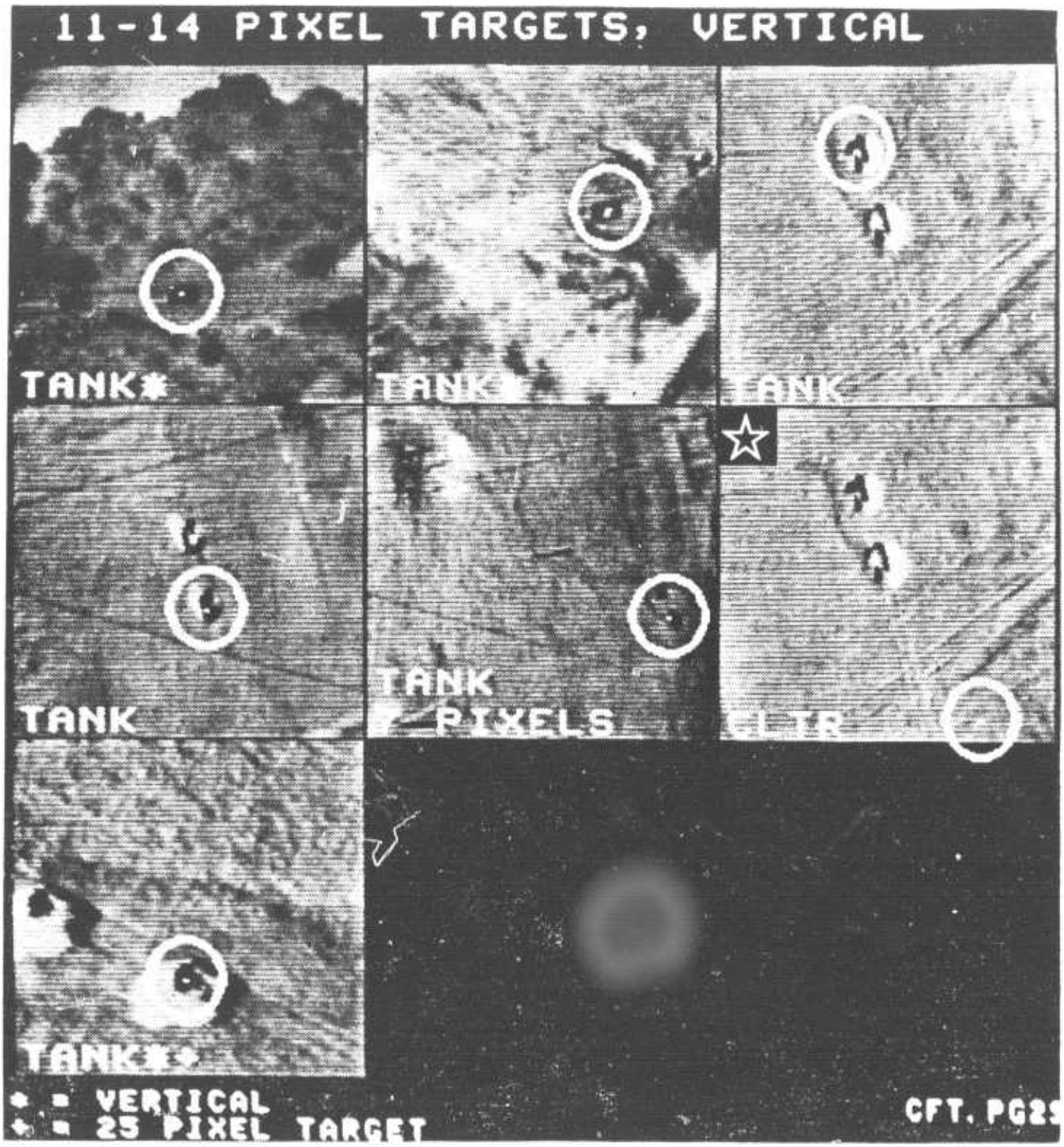

6-8 PIXEL TARGETS, NON-VERTICAL
CLTR
TANK
CLTR
TANK
TANK
* = VERTICAL ASPECT


8-10 PIXEL TARGETS, VERTICAL
TANK
TANK
TANK*
TANK
TANK*
* = NON-VERTICAL ASPECT
CFT. PG2


8-10 PIXEL TARGETS, VERTICAL
TANK
TANK
TANK
TANK
TANK
TANK
TANK
TANK
CLTR
CFT. PG2


8-10 PIXEL TARGETS, VERTICAL
TANK
TANK
TANK*
TANK
TANK
TANK*
VEHICLE
* = 6 PIXEL TARGETS


11-14 PIXEL TARGETS, VERTICAL
TANK
TANK
TANK
TANK
TANK
* * NON-VERTICAL

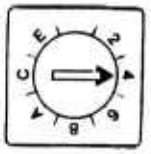

11-14 PIXEL TARGETS, VERTICAL
TANK*
TANK
TANK
TANK
TANK
PIXELS
TANK*
* = VERTICAL
= 25 PIXEL TARGET
CFT. PG2

SELF TEST
THUMB-WHEEL
SINGLE DIGIT

DEBUG THUMWHEEL

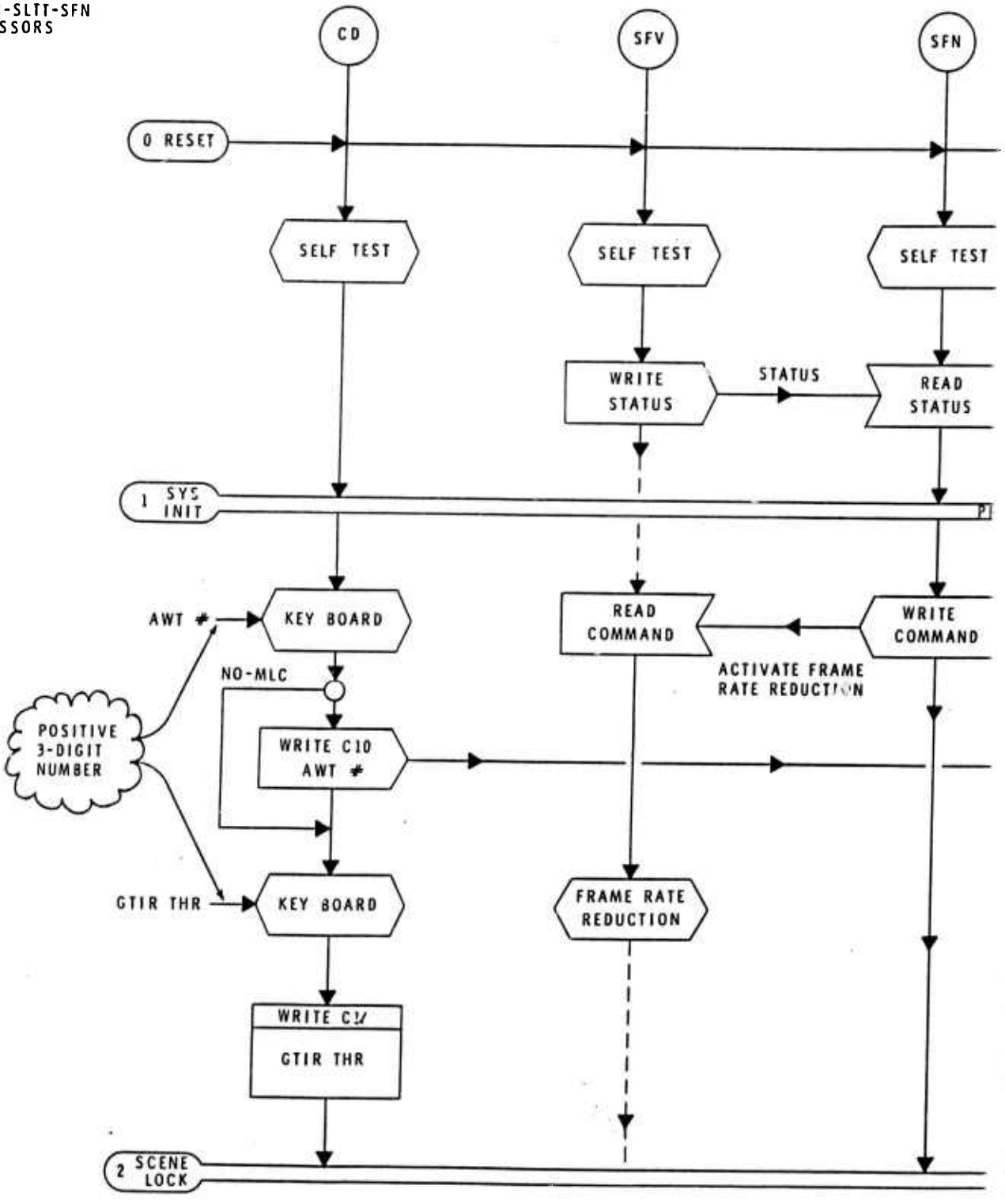
5 = CD ONLY
4 = CD-GP
3 = CD-GP-MLC
2 = CD-GP-MLC-SLTT
6 = CD-GP-MLC-SLTT-SFN
0 = ALL PROCESSORS

IMAGE SOURCE SWITCH
ON CD PROCESSOR

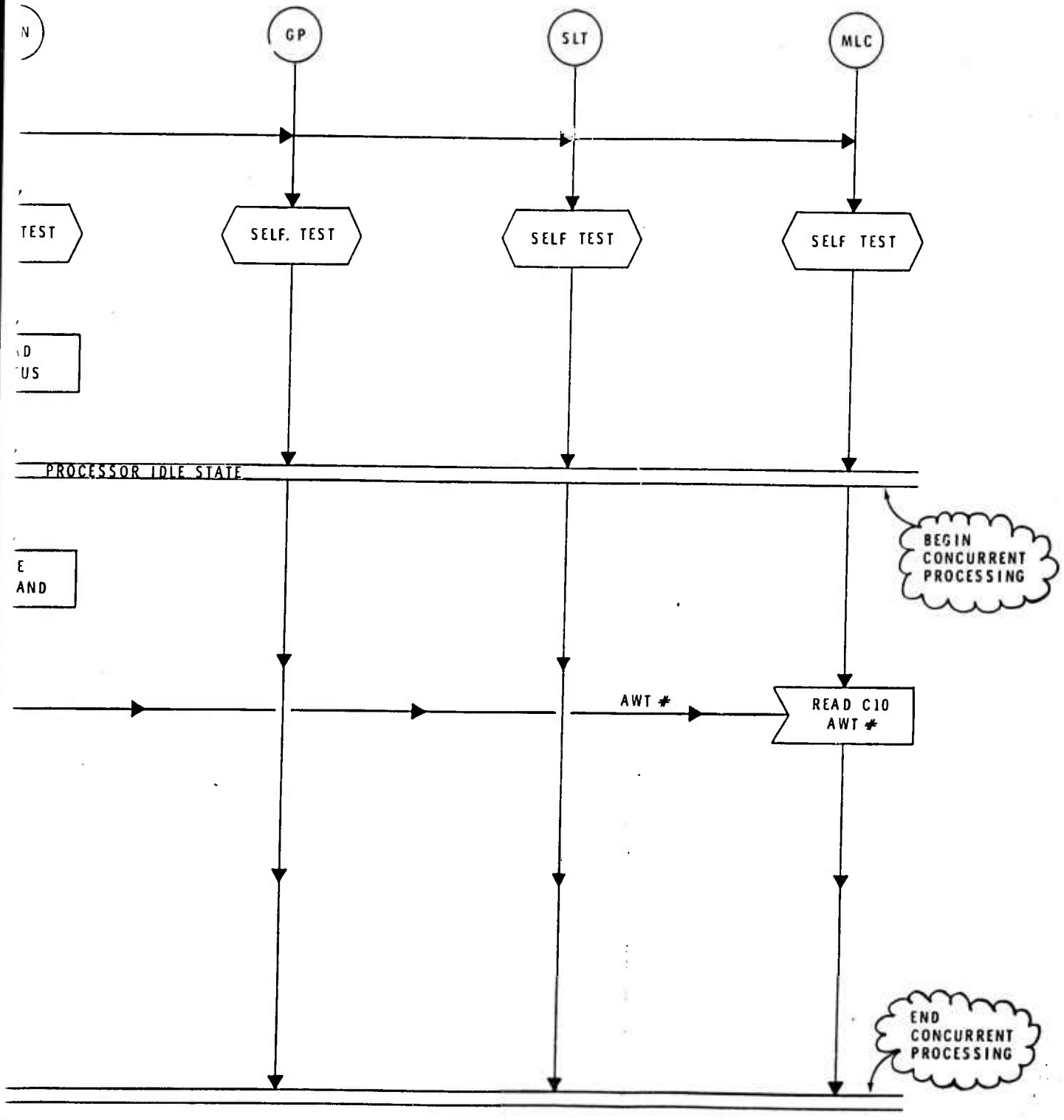


SWITCH

ISW0 = PNVS
ISW8 = PROM
ISW4 = GP-DEBUG ON

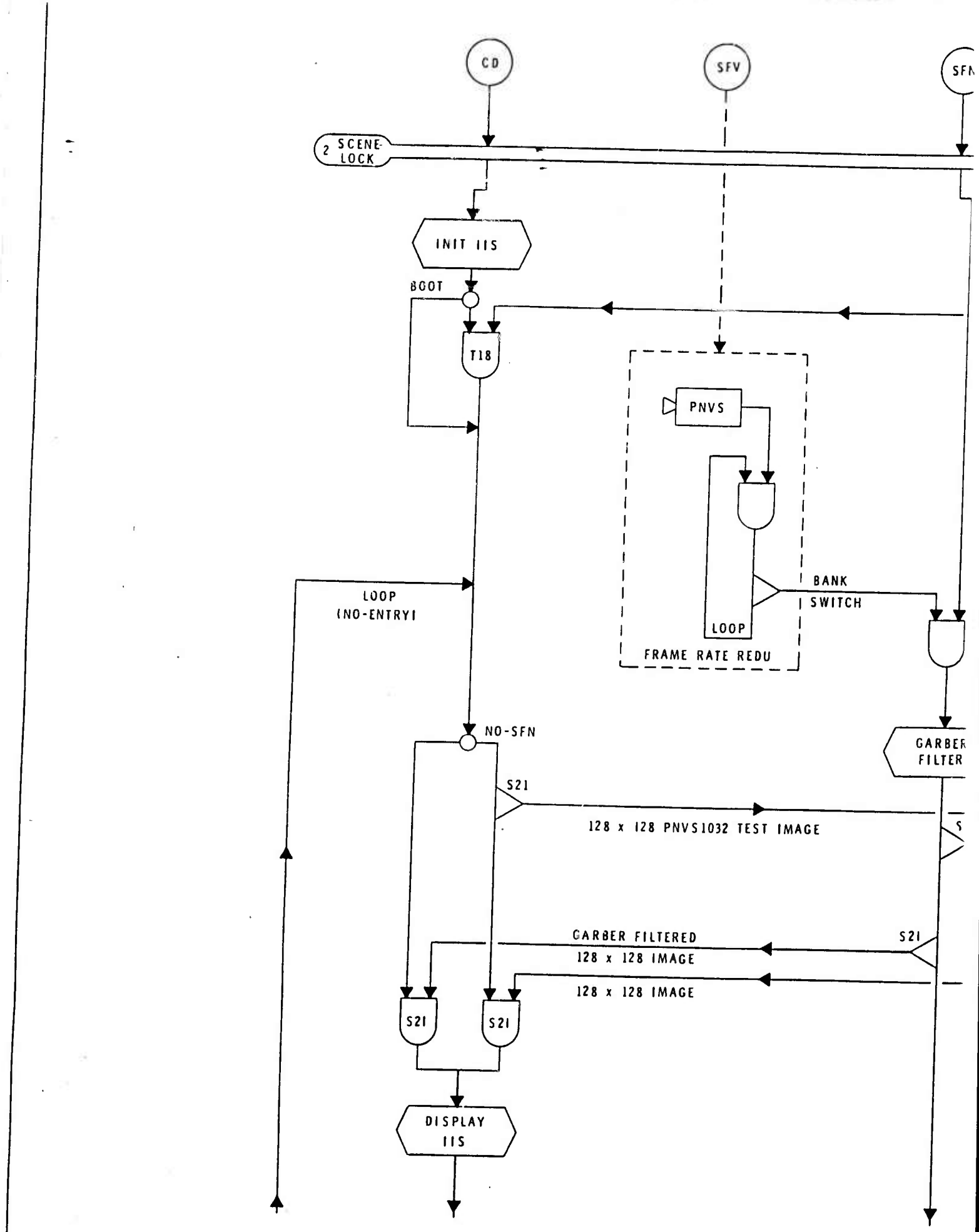


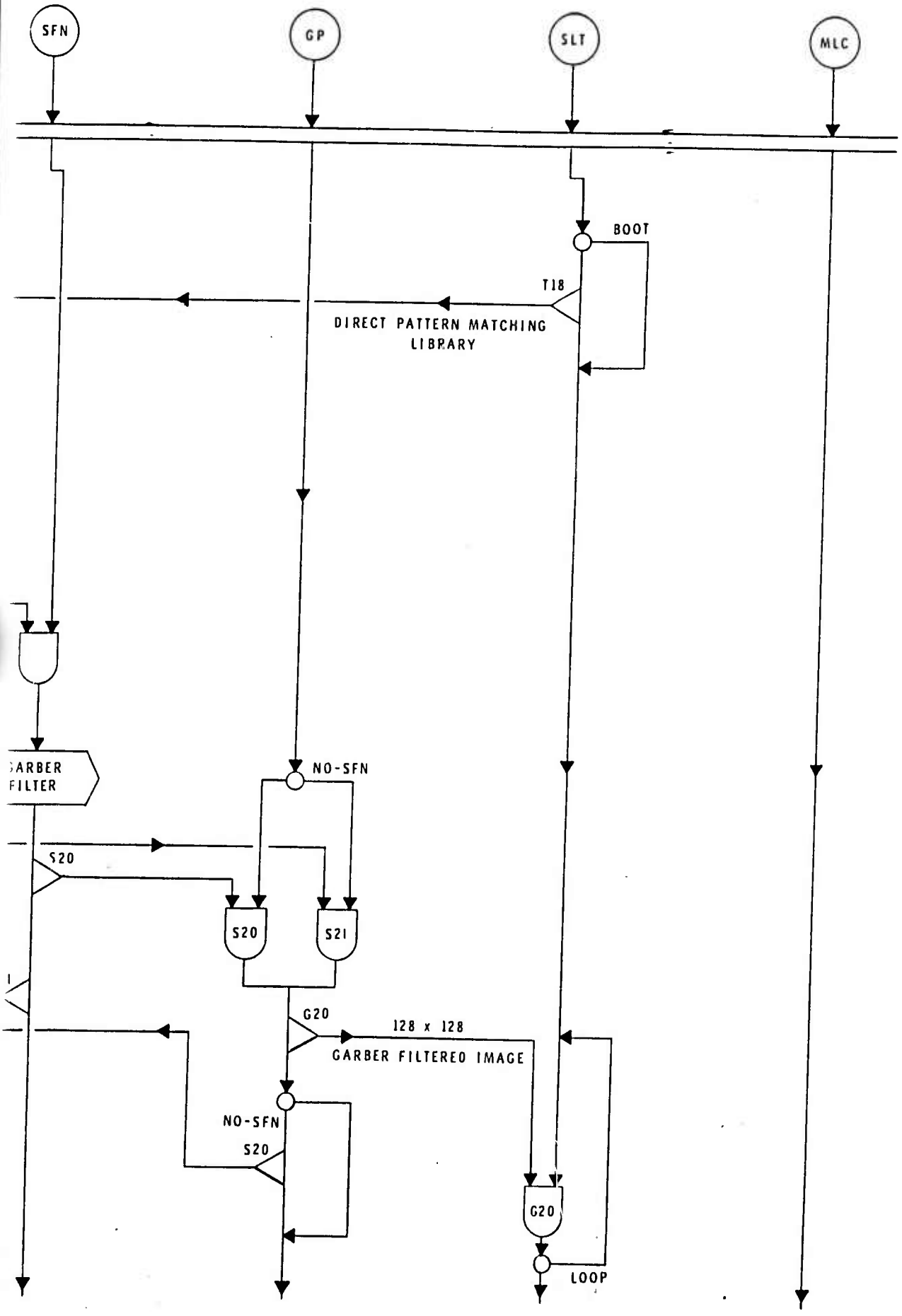
A



NATS SYSTEM DATA FLOW
REV 2.1 - JUNE '83

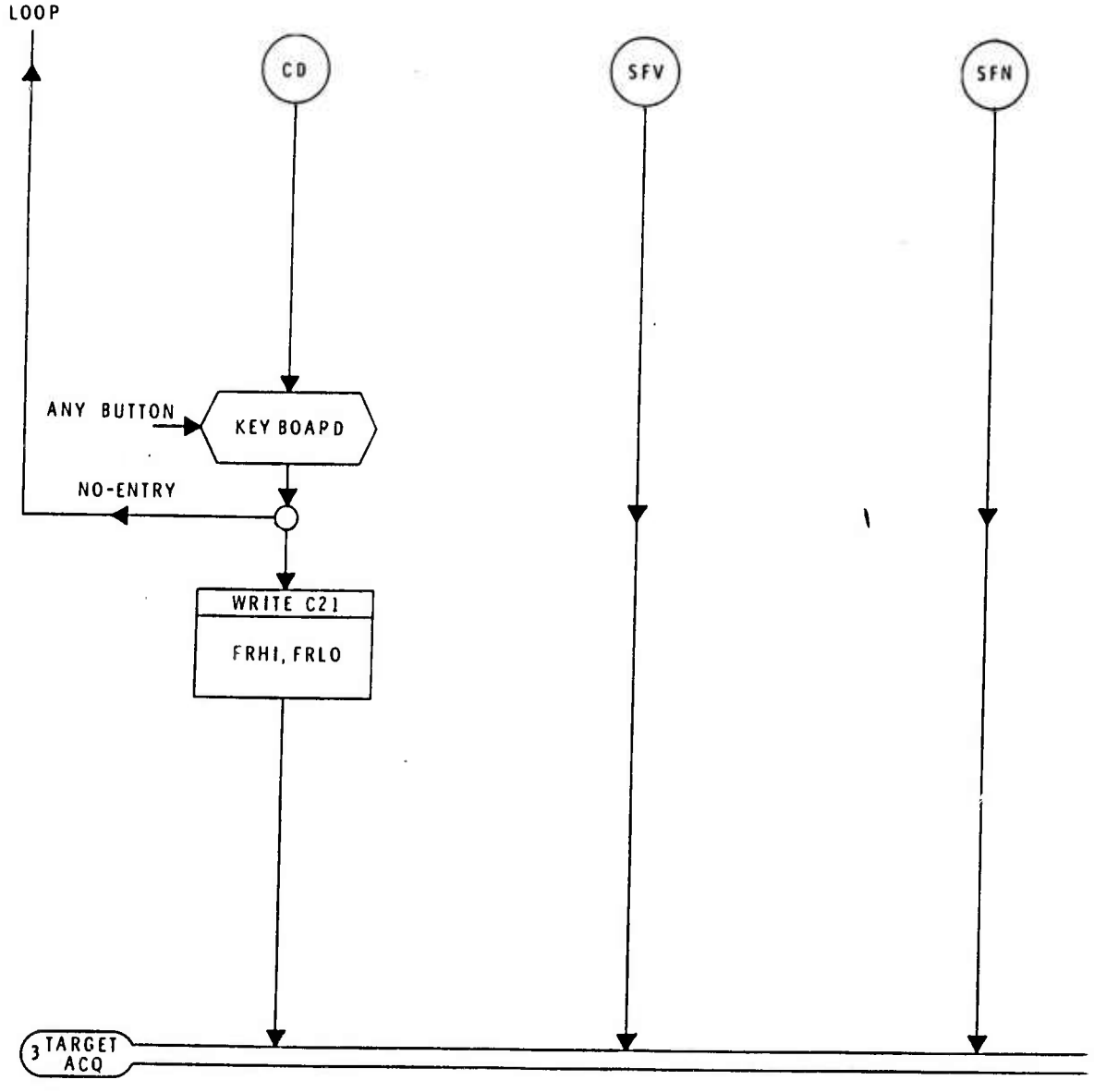
B

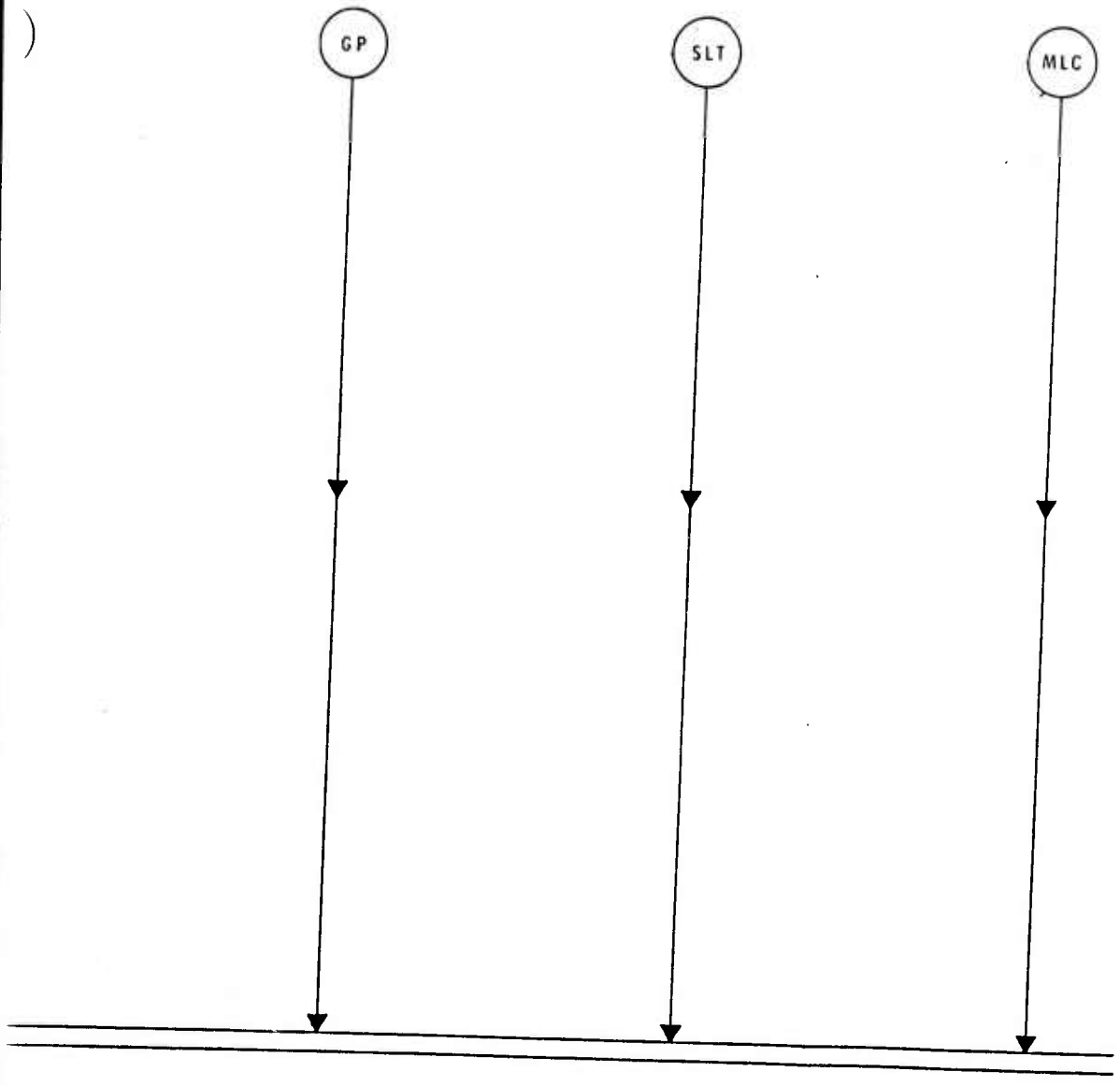

CD
SFV
SFN
2 SCENE LOCK
INIT IIS
BOOT
T18
PNVS
LOOP
BANK SWITCH
FRAME RATE REDU
LOOP (NO-ENTRY)
NO-SFN
GARBER FILTER
S21
128 x 128 PNVS1032 TEST IMAGE
GARBER FILTERED 128 x 128 IMAGE
S21
128 x 128 IMAGE
S21
S21
DISPLAY IIS



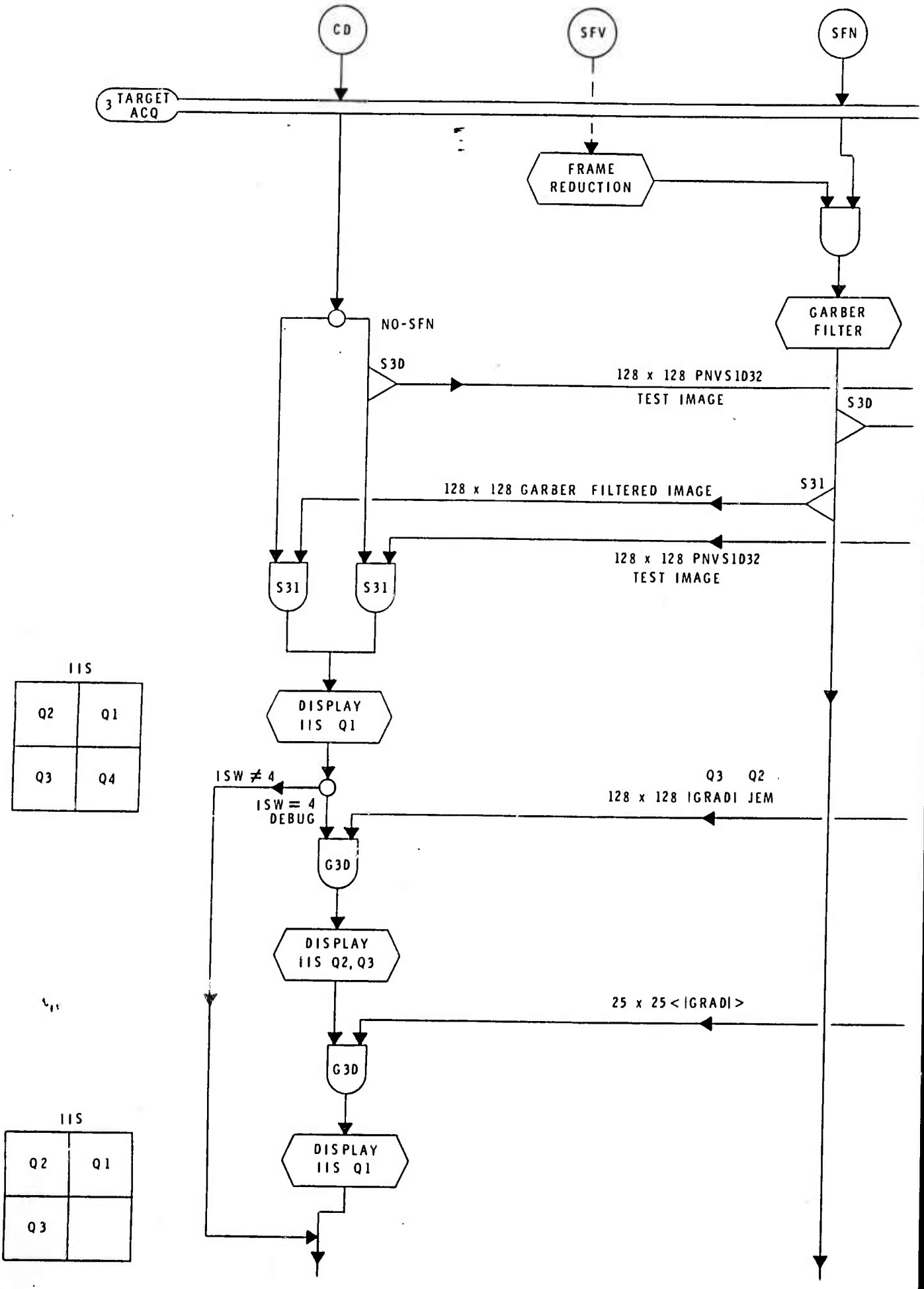
NATS SYSTEM DATA FLOW
REV 2.1 - JUNE '83

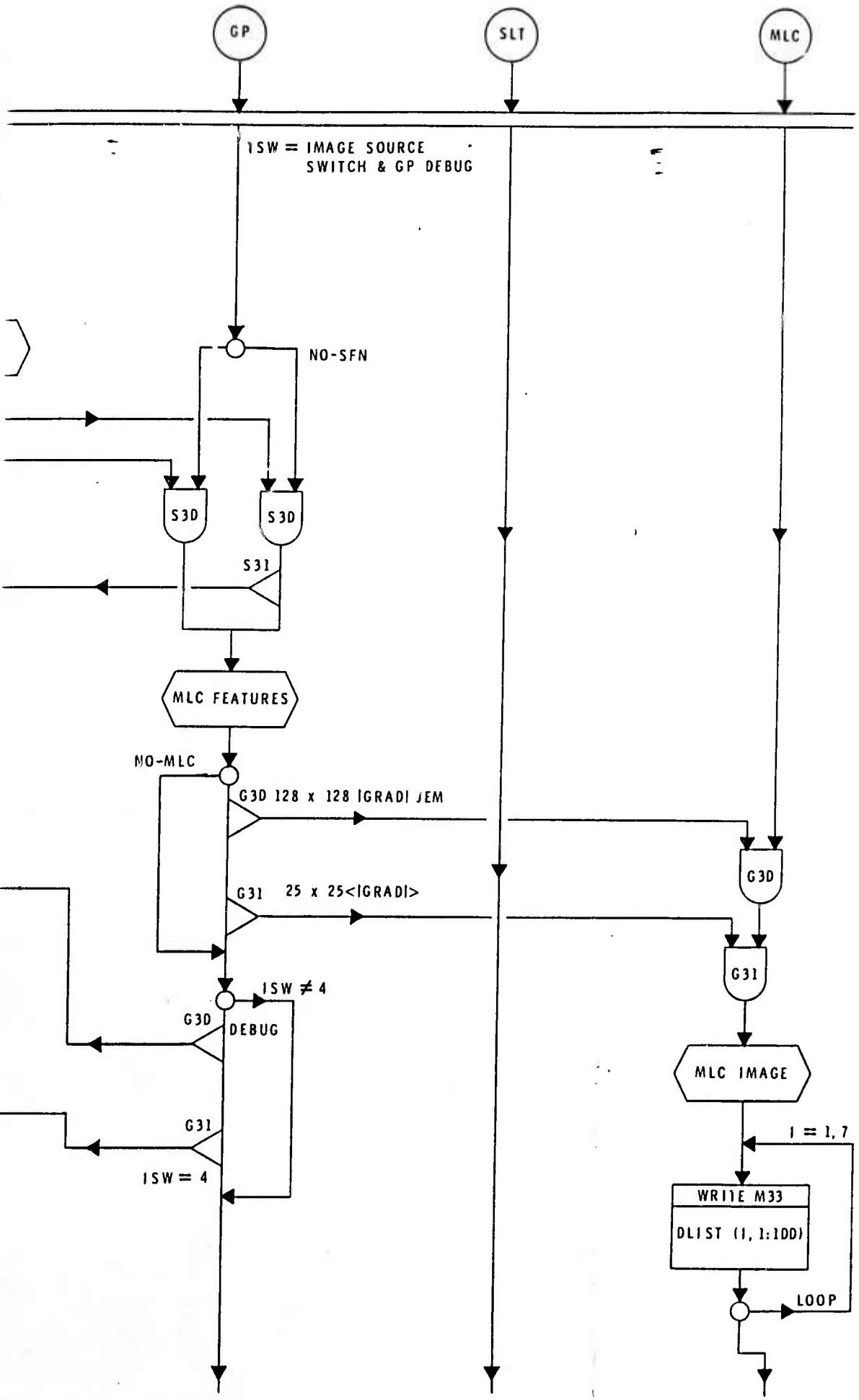
2 CONT.
SCENE LOCK





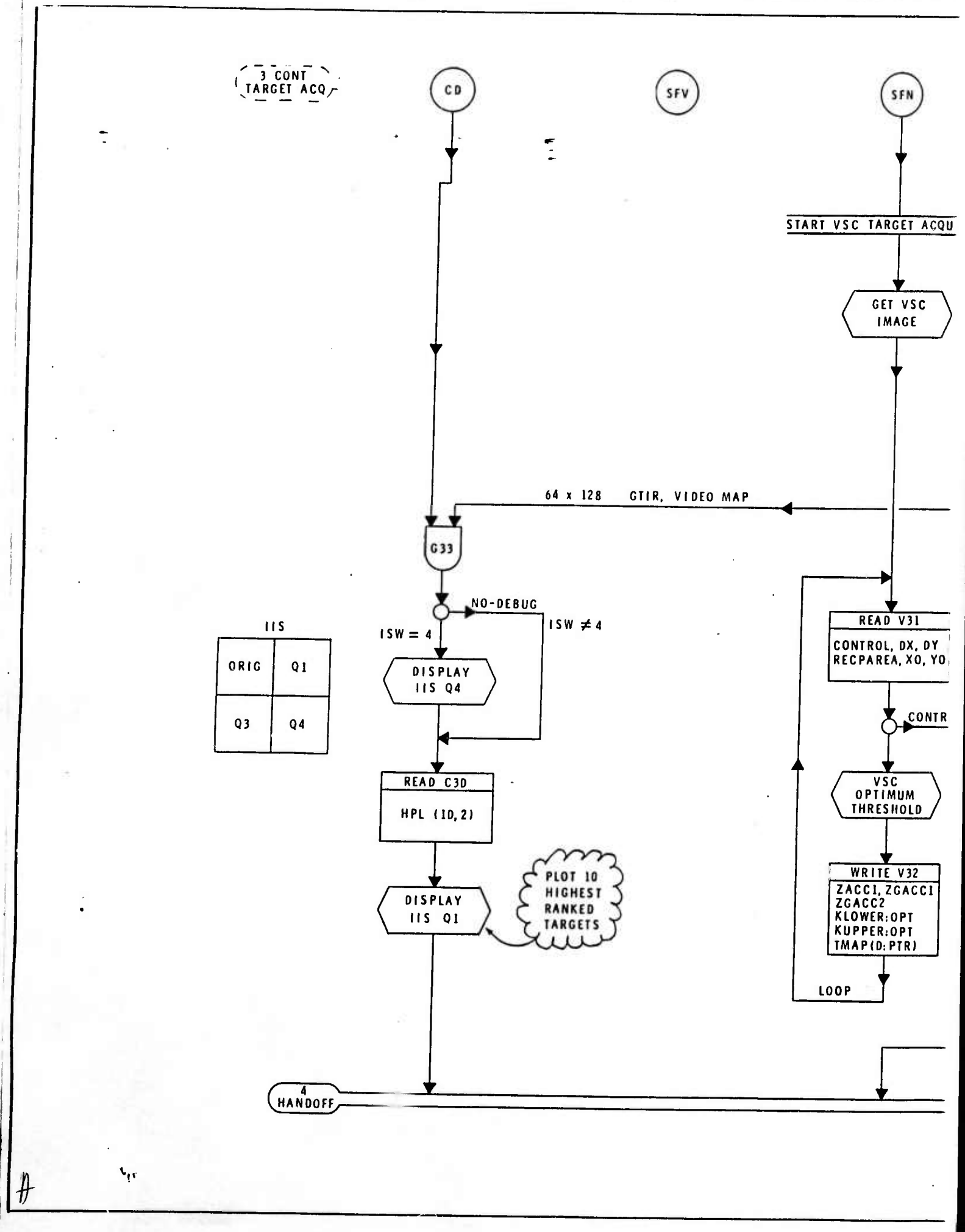
NATS SYSTEM DATA FLOW
REV 2.1 - JUNE '83

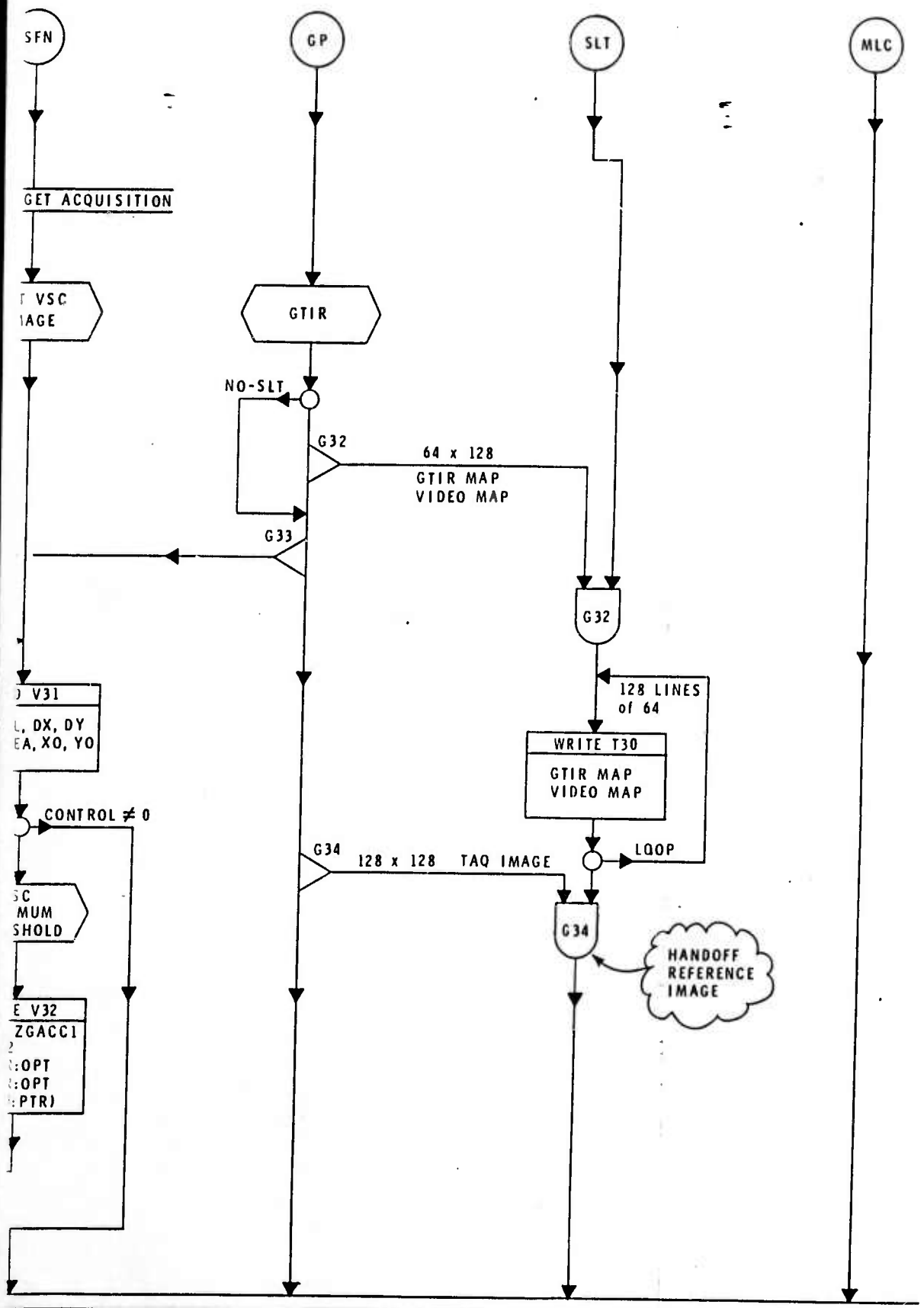

CD
SFV
SFN
3 TARGET ACQ
FRAME REDUCTION
GARBER FILTER
NO-SFN
S3D
128 x 128 PNVS1D32 TEST IMAGE
S3D
128 x 128 GARBER FILTERED IMAGE
S31
128 x 128 PNVS1D32 TEST IMAGE
S31
S31
IIS
Q2
Q1
Q3
Q4
DISPLAY IIS Q1
ISW ≠ 4
ISW = 4 DEBUG
Q3 Q2
128 x 128 IGRADI JEM
G3D
DISPLAY IIS Q2, Q3
25 x 25 < IGRADI >
G3D
IIS
Q2
Q1
Q3
DISPLAY IIS Q1



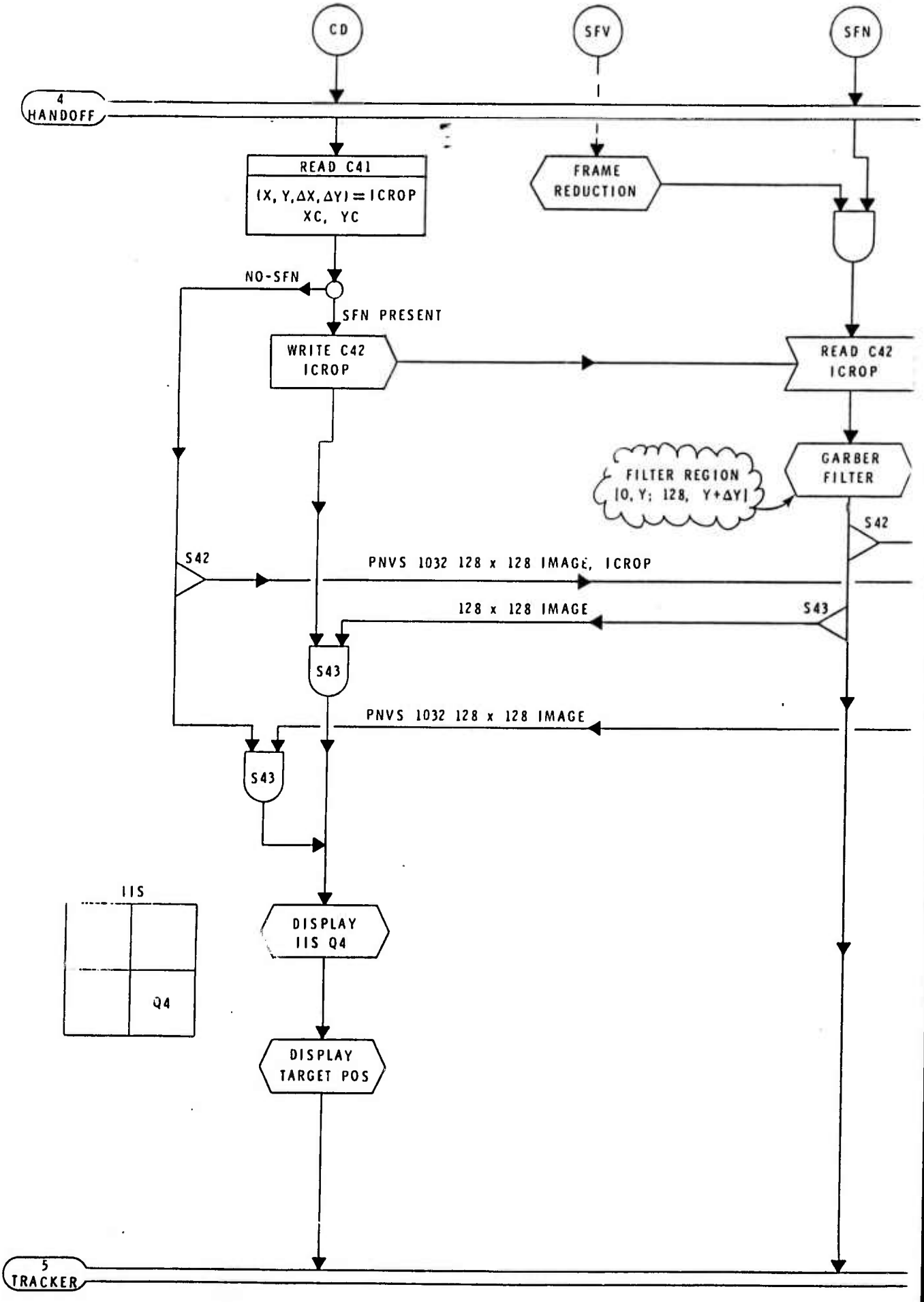
NATS SYSTEM DATA FLOW
REV 2.1 - JUNE '83

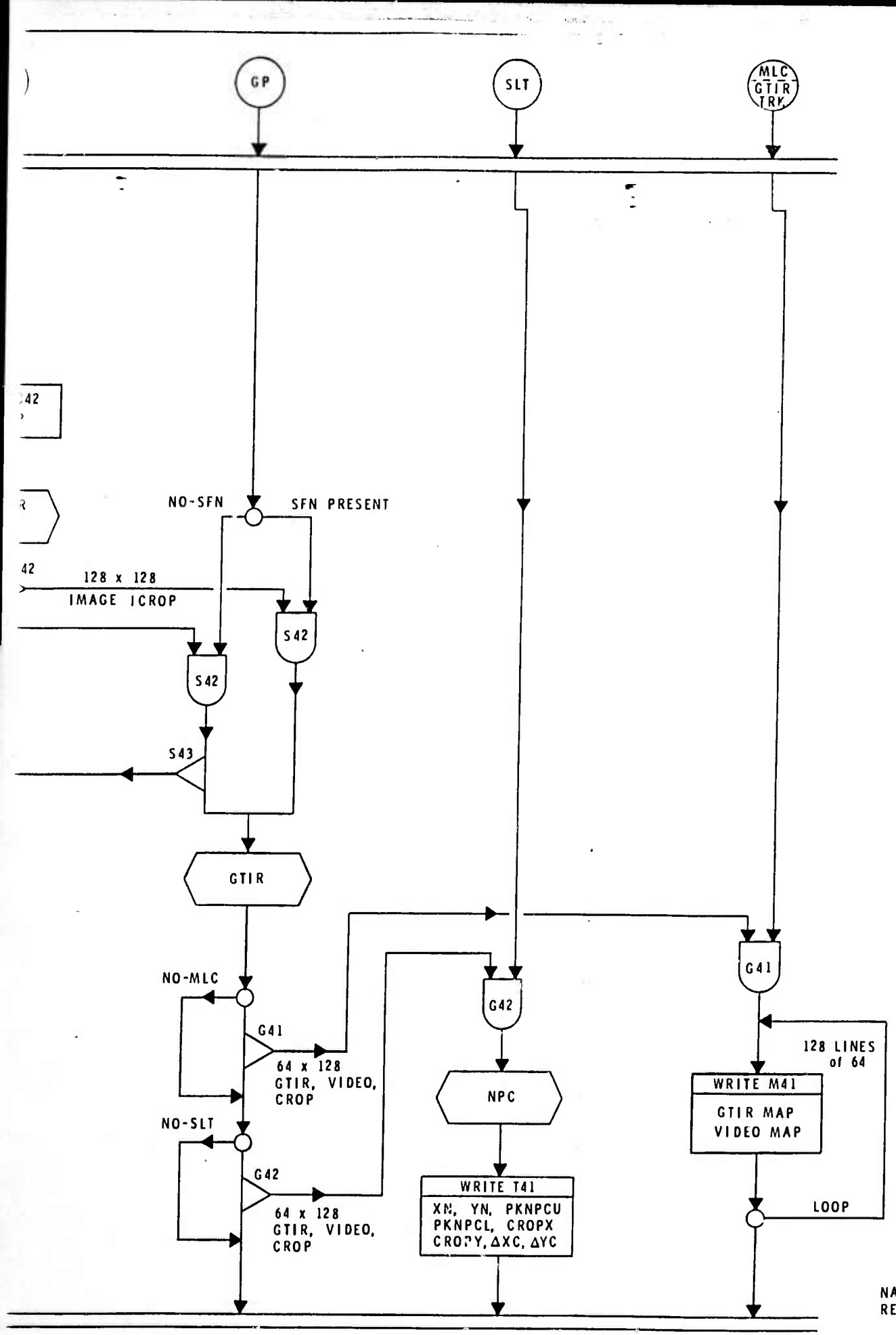
B


3 CONT
TARGET ACQ
CD
SFV
SFN
START VSC TARGET ACQU
GET VSC
IMAGE
64 x 128 GTIR, VIDEO MAP
G33
NO-DEBUG
ISW = 4
ISW ≠ 4
IIS
ORIG
Q1
Q3
Q4
DISPLAY
IIS Q4
READ C3D
HPL (10, 2)
DISPLAY
IIS Q1
PLOT 10
HIGHEST
RANKED
TARGETS
READ V31
CONTROL, DX, DY
RECPAREA, XO, YO
CONTR
VSC
OPTIMUM
THRESHOLD
WRITE V32
ZACC1, ZGACC1
ZGACC2
KLOWER:OPT
KUPPER:OPT
TMAP(0:PTR)
LOOP
4
HANDOFF

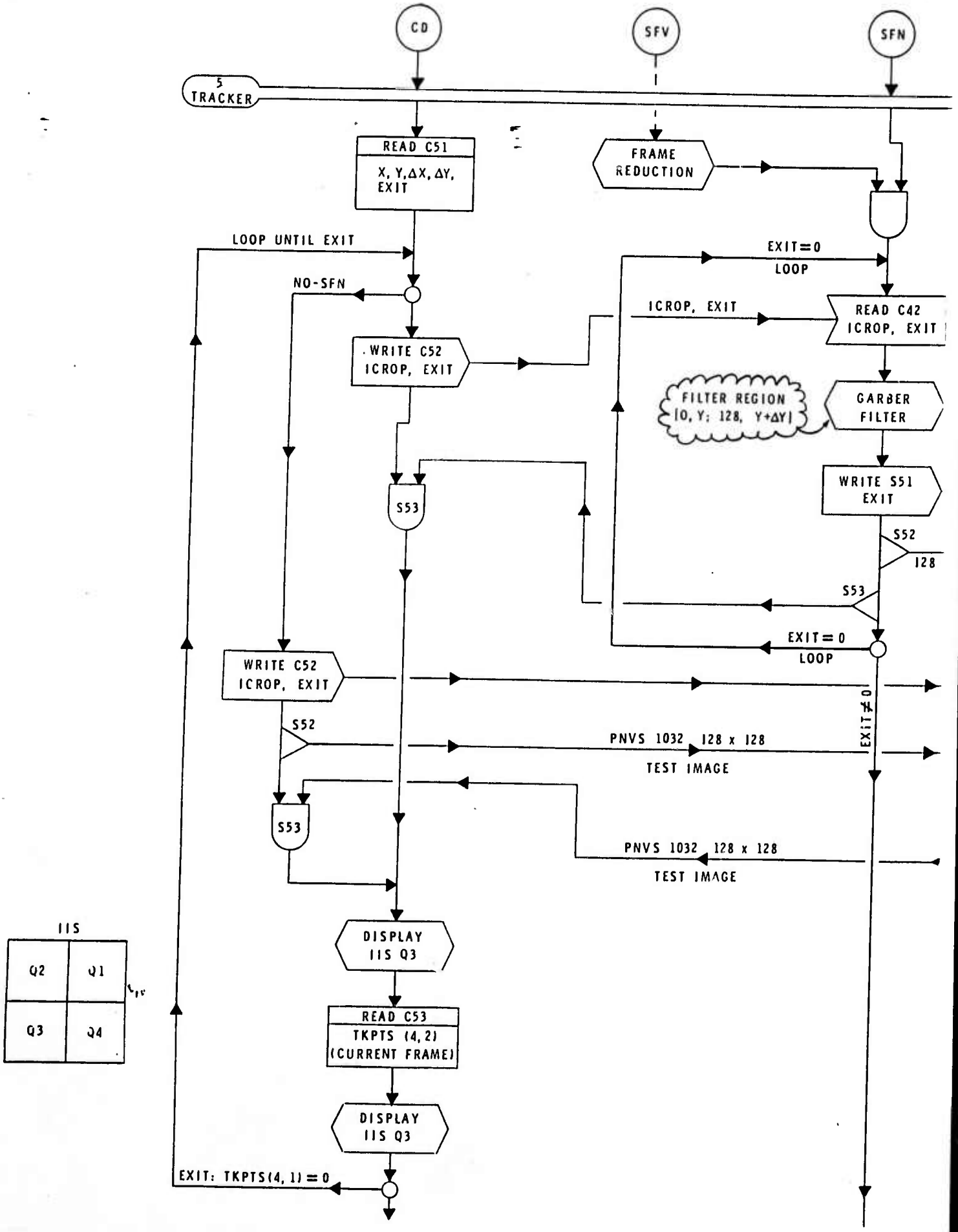


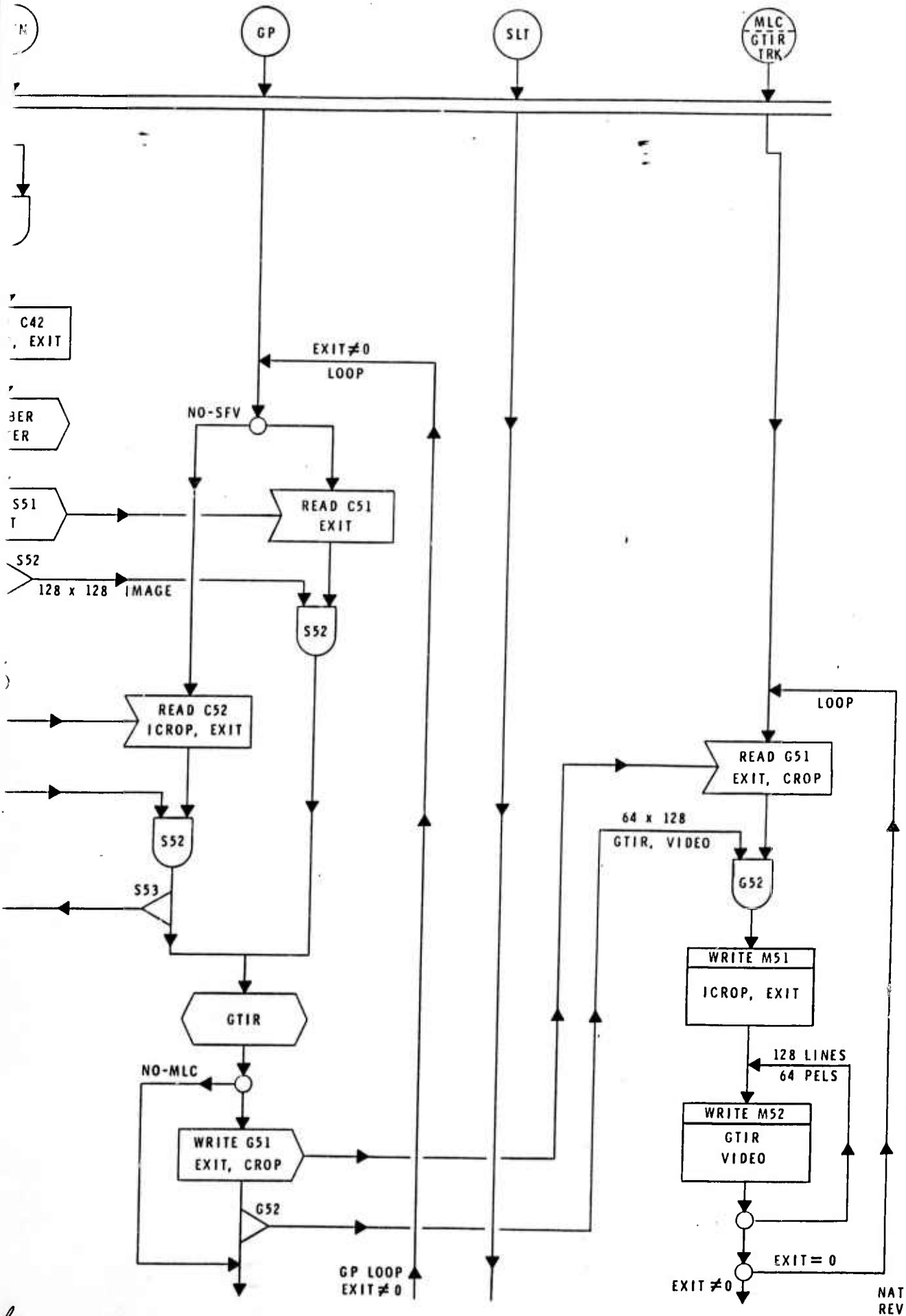
NATS SYSTEM DATA FLOW
REV 2.1 - JUNE '83


CD
SFV
SFN
4
HANDOFF
READ C41
(X, Y, ΔX, ΔY) = ICROP
XC, YC
FRAME
REDUCTION
NO-SFN
SFN PRESENT
WRITE C42
ICROP
READ C42
ICROP
FILTER REGION
(0, Y; 128, Y+ΔY)
GARBER
FILTER
S42
S42
PNVS 1032 128 x 128 IMAGE, ICROP
128 x 128 IMAGE
S43
S43
S43
PNVS 1032 128 x 128 IMAGE
IIS
Q4
DISPLAY
IIS Q4
DISPLAY
TARGET POS
5
TRACKER

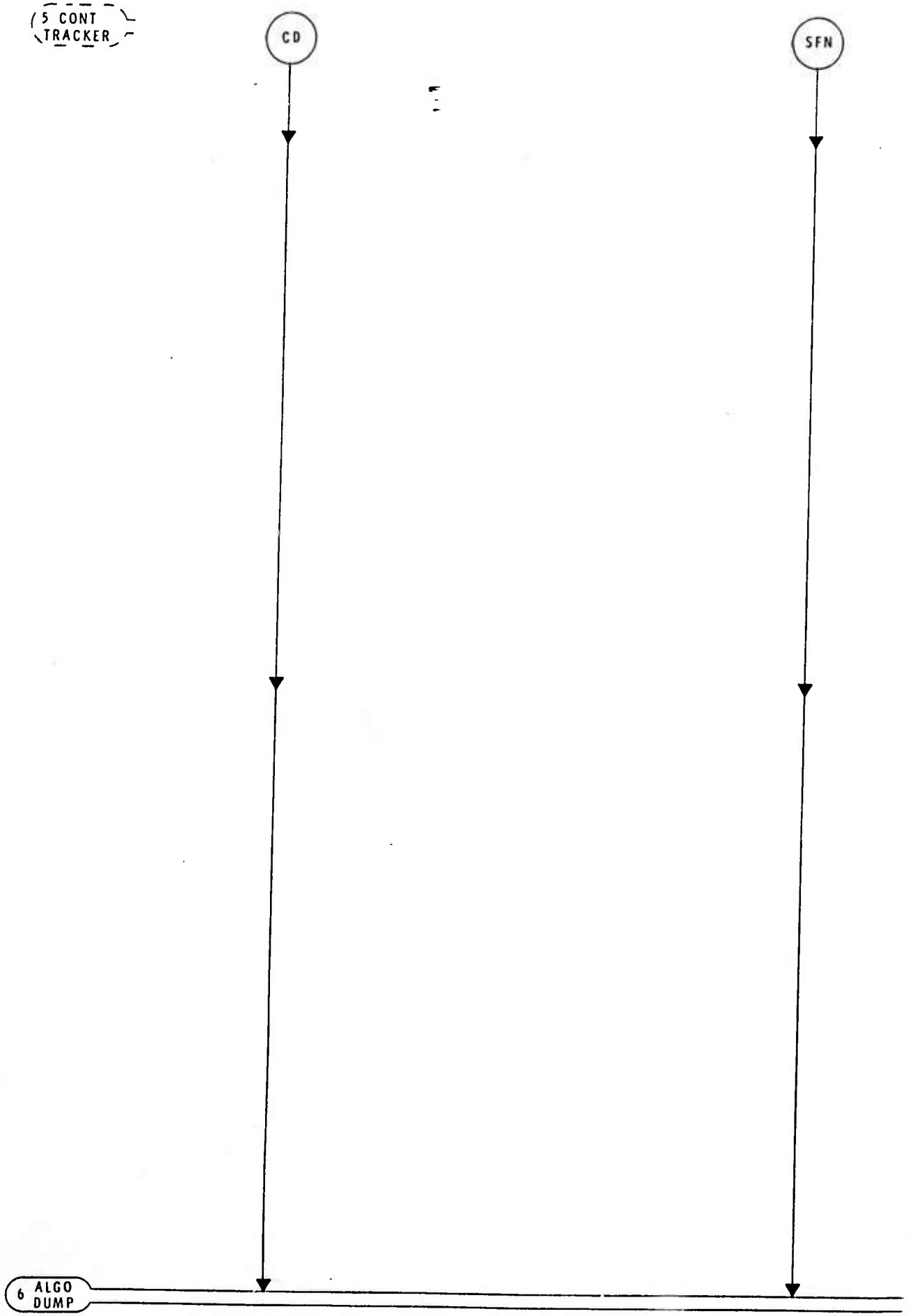


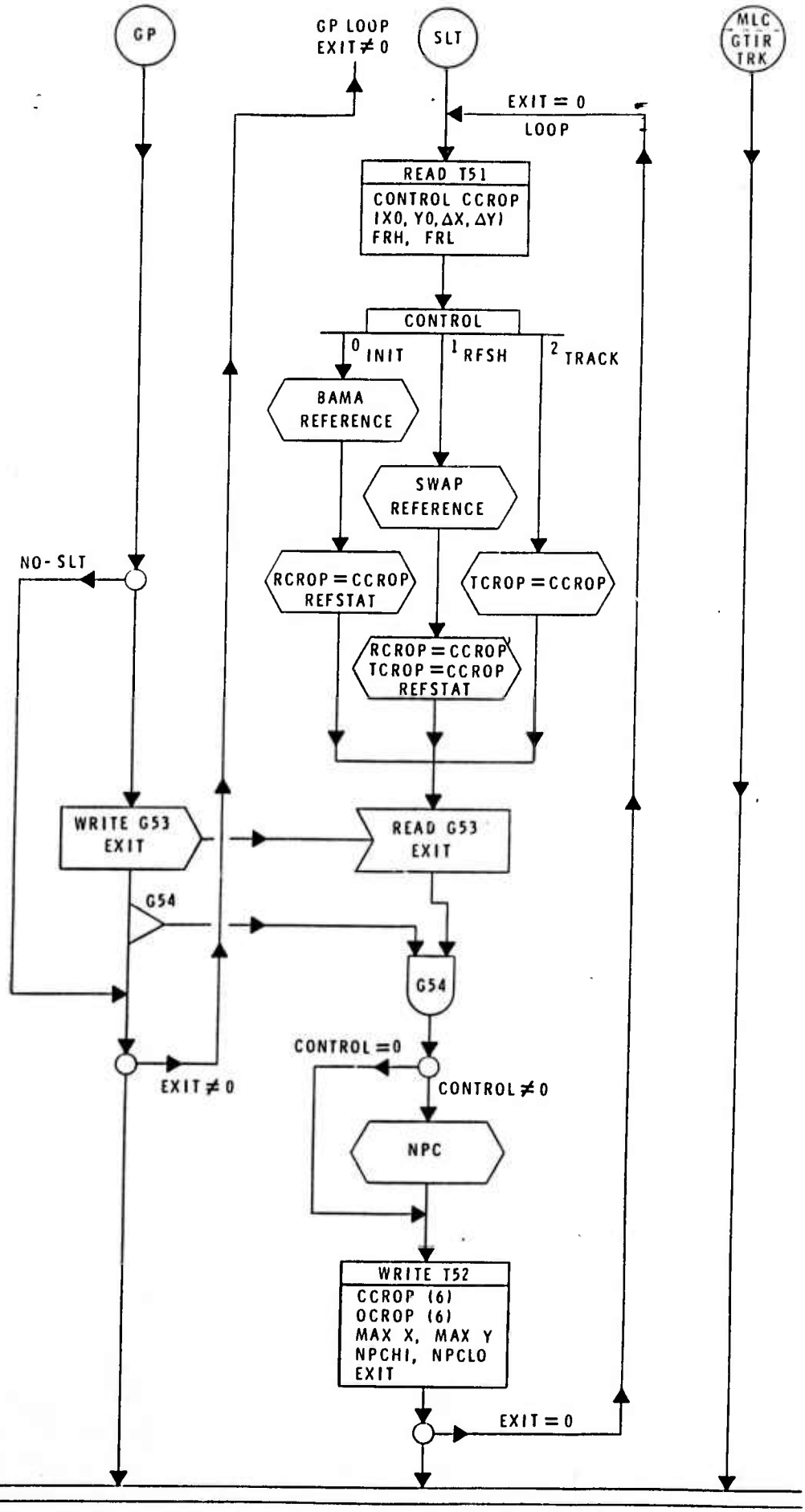
NATS SYSTEM DATA FLOW
REV 2.1 - JUNE '83


CD
SFV
SFN
5
TRACKER
READ C51
X, Y, ΔX, ΔY, EXIT
FRAME REDUCTION
LOOP UNTIL EXIT
NO-SFN
EXIT=0
LOOP
ICROP, EXIT
READ C42
ICROP, EXIT
WRITE C52
ICROP, EXIT
FILTER REGION
(0, Y; 128, Y+ΔY)
GARBER FILTER
WRITE S51
EXIT
S53
S52
128
S53
EXIT=0
LOOP
WRITE C52
ICROP, EXIT
S52
EXIT≠0
PNVS 1032 128 x 128
TEST IMAGE
S53
PNVS 1032 128 x 128
TEST IMAGE
IIS
Q2
Q1
Q3
Q4
DISPLAY
IIS Q3
READ C53
TKPTS (4, 2)
(CURRENT FRAME)
DISPLAY
IIS Q3
EXIT: TKPTS(4, 1) = 0



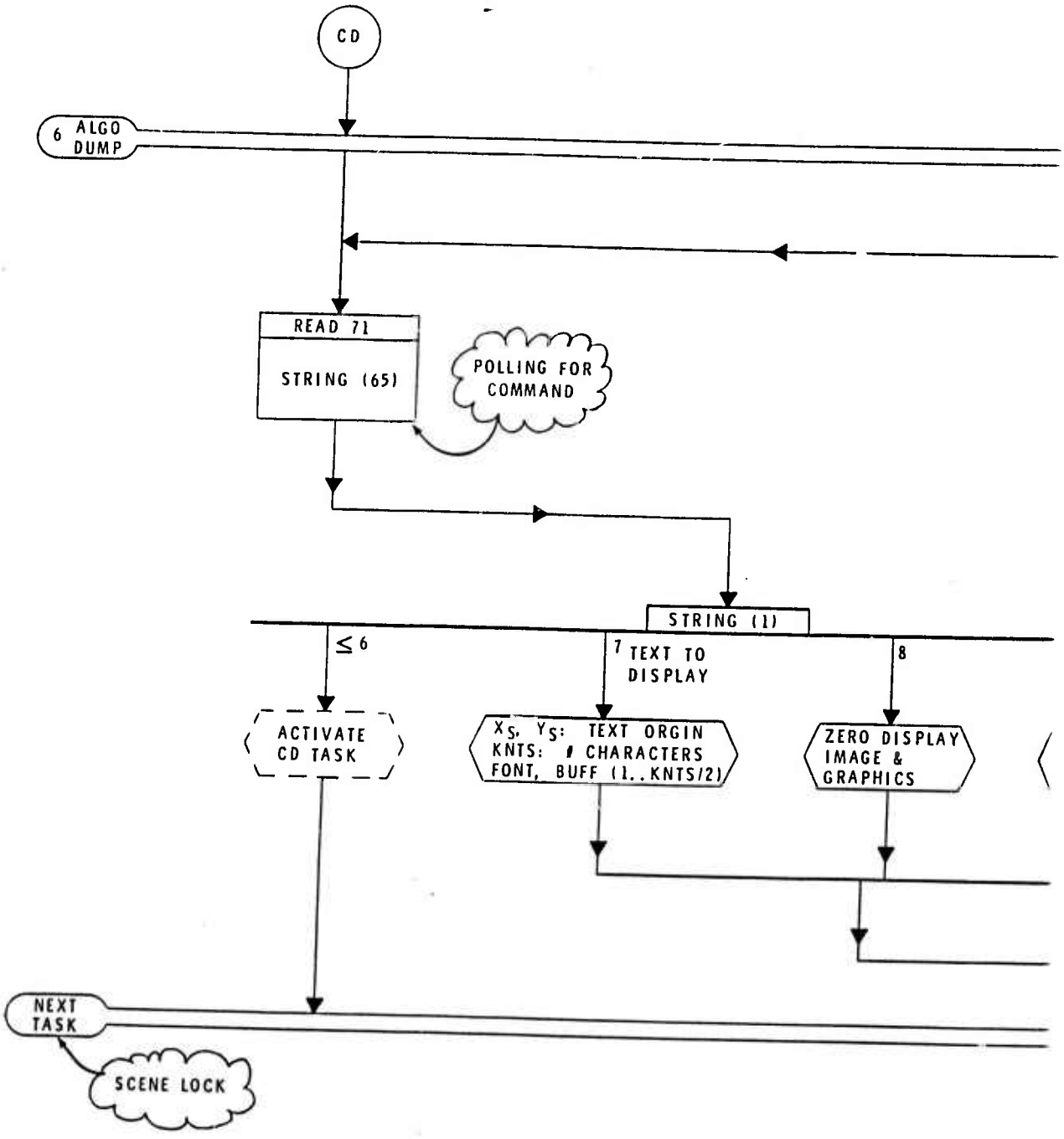
NATS SYSTEM DATA FLOW
REV 2.1 - JUNE '83

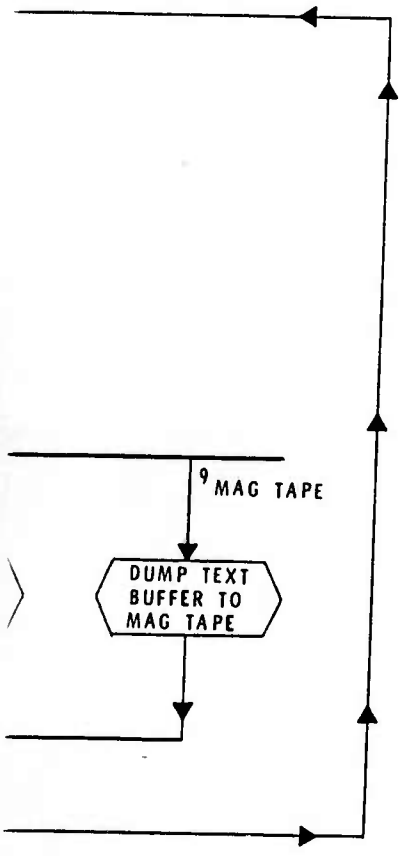

5 CONT
TRACKER
CD
SFN
6 ALGO
DUMP



NATS SYSTEM DATA FLOW
REV 2.1 - JUNE '83

B


CD
6 ALGO DUMP
READ 71
STRING (65)
POLLING FOR COMMAND
STRING (1)
≤ 6
7 TEXT TO DISPLAY
8
ACTIVATE CD TASK
XS, YS: TEXT ORGIN
KNTS: # CHARACTERS
FONT, BUFF (1..KNTS/2)
ZERO DISPLAY IMAGE & GRAPHICS
NEXT TASK
SCENE LOCK


9 MAG TAPE
DUMP TEXT BUFFER TO MAG TAPE