AD-A099 326 FEDERAL AVIATION ADMINISTRATION TECHNICAL CENTER ATL--ETC F/G 17/7
DISCRETE ADDRESS BEACON SYSTEM (DABS) COMPUTER PERFORMANCE/TEST--ETC(U)
APR 81 D FISHER, J PINO, D FOX
UNCLASSIFIED FAA-CT-81-7 FAA-RD-81-12 NL

END
DATE
FILMED
6 81
DTIC

Report No. FAA-RD-81-12
FAA-CT-81-7

(12)

# DISCRETE ADDRESS BEACON SYSTEM (DABS) COMPUTER PERFORMANCE/TEST AND EVALUATION

Donald Fisher
Joseph Pino
Daniel Fox

FEDERAL AVIATION ADMINISTRATION TECHNICAL CENTER
Atlantic City Airport, New Jersey 08405

INTERIM REPORT

APRIL 1981

Prepared for

U. S. DEPARTMENT OF TRANSPORTATION
FEDERAL AVIATION ADMINISTRATION
Systems Research & Development Service
Washington, D. C. 20590

# NOTICE

**Technical Report Documentation Page**

| | | |
|---|---|---|
| 1. Report No. ⑱ ⑲<br>FAA-RD-81-12 | 2. Government Accession No.<br>AD-A099 326 | 3. Recipient's Catalog No. |
| 4. Title and Subtitle ⑥<br>DISCRETE ADDRESS BEACON SYSTEM (DABS) COMPUTER PERFORMANCE/TEST AND EVALUATION, | | ~~5. Report Date~~ ⑪<br>April 1981 |
| | | ~~6. Performing Organization Code~~ ⑫ 58 |
| 7. Author's) ⑩<br>Donald Fisher, Joseph Pino, ~~and~~ Daniel Fox | ⑭ | 8. Performing Organization Report No.<br>FAA-CT-81-7 |
| ~~9. Performing Organization Name and Address~~<br>Federal Aviation Administration<br>Technical Center<br>Atlantic City Airport, New Jersey 08405 | | 10. Work Unit No. (TRAIS) |
| | | 11. Contract or Grant No.<br>034-241-510 |
| 12. Sponsoring Agency Name and Address<br>U.S. Department of Transportation<br>Federal Aviation Administration<br>Systems Research and Development Service<br>Washington, D.C. 20590 | | 13. ~~Type of~~ Report and Period Covered ⑨<br>Interim rept.<br>Dec 1979 — Sept 1980 |
| | | ~~14. Sponsoring Agency Code~~ |
| 15. Supplementary Notes | | |

16. Abstract

This document describes tests conducted on the Discrete Address Beacon System (DABS) engineering model sensor with the release 6.4 software package to measure the performance characteristics of the computer subsystem distributive architecture. Tests were conducted for various aircraft load conditions in three specific areas: system data bus contention, global memory address space utilization, and processor utilization. Both the methods of conducting these tests and the results obtained are described.

It was concluded that system data bus contention is not a problem with the distributive architecture used. Release 6.4 of the DABS software uses less than 20,480 words of the available 24,576 word global memory address space in 24 of the 29 active processors. This leads to the conclusion that no problem should be experienced in expanding the size of the processor local memories from 8,192 words to 12,288 words. Additionally, an expansion of the local memories to 16,384 words appears feasible with minor software changes. The expansion of local memory will enable each processor to perform more functions. This will reduce the total number of processors required and lead to less complexity and a smaller overall volume for DABS.

| | | | |
|---|---|---|---|
| 17. Key Words<br>Discrete Address Beacon System (DABS)<br>Computer Performance/DABS | | 18. Distribution Statement<br>Document is available to the U.S. public through the National Technical Information Service, Springfield, Virginia 22161 | |
| 19. Security Classif. (of this report)<br>Unclassified | 20. Security Classif. (of this page)<br>Unclassified | 21. No. of Pages<br>52 | 22. Price |

Form DOT F 1700.7 (8-72)

411863

# METRIC CONVERSION FACTORS

## Approximate Conversions to Metric Measures

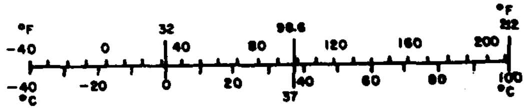
| Symbol | When You Know | Multiply by | To Find | Symbol |
|---|---|---|---|---|
| | | **LENGTH** | | |
| in | inches | *2.5 | centimeters | cm |
| ft | feet | 30 | centimeters | cm |
| yd | yards | 0.9 | meters | m |
| mi | miles | 1.6 | kilometers | km |
| | | **AREA** | | |
| $in^2$ | square inches | 6.5 | square centimeters | $cm^2$ |
| $ft^2$ | square feet | 0.09 | square meters | $m^2$ |
| $yd^2$ | square yards | 0.8 | square meters | $m^2$ |
| $mi^2$ | square miles | 2.6 | square kilometers | $km^2$ |
| | acres | 0.4 | hectares | ha |
| | | **MASS (weight)** | | |
| oz | ounces | 28 | grams | g |
| lb | pounds | 0.45 | kilograms | kg |
| | short tons (2000 lb) | 0.9 | tonnes | t |
| | | **VOLUME** | | |
| tsp | teaspoons | 5 | milliliters | ml |
| Tbsp | tablespoons | 15 | milliliters | ml |
| fl oz | fluid ounces | 30 | milliliters | ml |
| c | cups | 0.24 | liters | l |
| pt | pints | 0.47 | liters | l |
| qt | quarts | 0.95 | liters | l |
| gal | gallons | 3.8 | liters | l |
| $ft^3$ | cubic feet | 0.03 | cubic meters | $m^3$ |
| $yd^3$ | cubic yards | 0.76 | cubic meters | $m^3$ |
| | | **TEMPERATURE (exact)** | | |
| °F | Fahrenheit temperature | 5/9 (after subtracting 32) | Celsius temperature | °C |

---

*1 in ≡ 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures, Price $2.25, SD Catalog No. C13.10:286.

## Approximate Conversions from Metric Measures

| Symbol | When You Know | Multiply by | To Find | Symbol |
|---|---|---|---|---|
| | | **LENGTH** | | |
| mm | millimeters | 0.04 | inches | in |
| cm | centimeters | 0.4 | inches | in |
| m | meters | 3.3 | feet | ft |
| m | meters | 1.1 | yards | yd |
| km | kilometers | 0.6 | miles | mi |
| | | **AREA** | | |
| $cm^2$ | square centimeters | 0.16 | square inches | $in^2$ |
| $m^2$ | square meters | 1.2 | square yards | $yd^2$ |
| $km^2$ | square kilometers | 0.4 | square miles | $mi^2$ |
| ha | hectares (10,000 $m^2$) | 2.5 | acres | |
| | | **MASS (weight)** | | |
| g | grams | 0.035 | ounces | oz |
| kg | kilograms | 2.2 | pounds | lb |
| t | tonnes (1000 kg) | 1.1 | short tons | |
| | | **VOLUME** | | |
| ml | milliliters | 0.03 | fluid ounces | fl oz |
| l | liters | 2.1 | pints | pt |
| l | liters | 1.06 | quarts | qt |
| l | liters | 0.26 | gallons | gal |
| $m^3$ | cubic meters | 35 | cubic feet | $ft^3$ |
| $m^3$ | cubic meters | 1.3 | cubic yards | $yd^3$ |
| | | **TEMPERATURE (exact)** | | |
| °C | Celsius temperature | 9/5 (then add 32) | Fahrenheit temperature | °F |



# TABLE OF CONTENTS

APPENDICES

## LIST OF ILLUSTRATIONS

# LIST OF TABLES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| ACP | azimuth change pulse |
| ALU | arithmetic and logic unit |
| ARIES | Aircraft Reply and Interference Environment Simulator |
| ARP | azimuth reference pulse |
| ATARS | Automatic Traffic Advisory and Resolution Service |
| ATC | air traffic control |
| ATCRBS | Air Traffic Control Radar Beacon System |
| CIDIN | Common ICAO Data Interchange Network |
| CM | channel management |
| DABS | Discrete Address Beacon System |
| DEC | Digital Equipment Corporation |
| DL | data link |
| FAA | Federal Aviation Administration |
| HSMAT | high speed memory access test |
| I&P | interrogator and processor |
| ICAO | International Civil Aviation Organization |
| IF | intermediate frequency |
| IPC | intermittent positive control |
| PM | performance monitor |
| RF | radiofrequency |
| S | surveillance |
| ST | sensor track |
| TI | Texas Instruments Incorporated |

## EXECUTIVE SUMMARY

During the test and evaluation activity reported herein, the performance characteristics of the distributive architecture of the computer subsystem within the Discrete Address Beacon System (DABS) engineering model sensor were measured. All measurements were made using the computer performance measurement system designed and built by personnel of the Federal Aviation Administration (FAA) Technical Center. Measurements were made with release 6.4 of the DABS software using traffic scenarios generated by the Aircraft Reply and Interference Environment Simulator (ARIES).

Two basically identical ARIES scenarios containing 400 target aircraft were used. The first contained a mixture of 50 percent Air Traffic Control Radar Beacon System (ATCRBS) and 50 percent DABS transponder equipped aircraft. The second scenario contained all DABS transponder equipped aircraft. A third scenario containing all ATCRBS transponder equipped aircraft was also available for testing. However, it had been determined during previous testing that the release 6.4 DABS software was unable to process the peak sector ATCRBS target load generated by this scenario. This was due to software deficiencies, primarily within the channel management functions, which have been corrected in later releases of the DABS software. Due to these deficiencies the all ATCRBS scenario was not used during these tests.

As previously reported in the DABS Baseline Test and Evaluation Report, FAA-RD-80-36 (reference 1), asynchronous replies, normally referred to as fruit, were found to have minimal or no impact on overall system performance. Therefore, to ensure test repeatability, these tests were conducted without fruit.

Due to the distributive architecture of the DABS computer subsystem, a study of certain performance characteristics was deemed necessary as an input towards procurement of subsequent DABS sensors for field implementation. The DABS distributive architecture ties together 36 processors, each with its own 8,192 word internal or local memory, through a series of data buses to a shared central or global memory. All interprocessor communication is handled through data storage in this global memory. Each processor is programmed to process only one function or a small group of functions. During these tests 29 of the 36 processors contained active software modules. All measurements were confined to these 29 active processors.

This distributive architecture allows for ease of system modification and expansion. Additional functions can be added by increasing the number of processors. It also allows for function modification without changing code in any processor but the one being modified.

Tests were conducted in three specific performance areas: data bus contention, global memory address space utilization, and processor utilization.

Data bus contention measurements were made to determine the effect on system operation of the contention between the 29 active processors in accessing data stored in the global memories. Characteristic curves were developed relating the average delay experienced in accessing a data bus to the average load on the data bus. Measurements showed that average delays were relatively stable for bus loads of up to 30 percent busy on ensemble data buses and 88 percent busy on global data buses.

With the DABS software most ensemble data buses exhibited loads of between 10 and 15 percent busy when tested with the mixed and all DABS traffic

scenarios. The two global data buses exhibited loads of between 35 and 45 percent busy for global A and between 10 and 17 percent busy for global B. It was concluded that no appreciable contention effect exists.

Ensemble data bus loads of over 30 percent busy were found to exhibit increasing contention effects. A load of approximately 30 percent busy was measured for one ensemble data bus with the DABS software. Further investigation, however, showed that two of the four processors attached to this ensemble data bus contained the two busiest functions in terms of global memory accesses. A large reduction in this data bus load should be experienced by separating these two functions on separate ensemble data buses. This should be further investigated if contention on this data bus is suspected of degrading system operation. However, no system degradation was found during this test series.

A second series of tests investigated the utilization of the 32,768 word global memory address space. This address space is currently divided between an 8,192 word local memory and a 24,576 word window into the 640,672 word global memory. Measurements were made to assess the impact of increasing the size of the local memory to 12,288 words. This change would necessitate a corresponding decrease in the global memory window size to 20,480 words. The location of the global memory window is determined by the contents of software loadable bias registers. Any change in the size of the window can affect the frequency with which the bias registers must be reloaded. Therefore, a change in local memory size directly affects the amount of software overhead required for setting the bias registers.

Data collected with the mixed and all DABS scenarios indicated that most of the DABS software routines currently utilize less than 20,480 words of the available global memory window. Therefore, an expansion in the size of the local memory to 12,288 words should cause no system problems. In fact, 11 of the 29 active processors use less than 16,384 words of the available global memory window, and an expansion of local memory size to 16,384 words appears feasible. The expansion of local memory will enable each processor to perform more functions. This will reduce the total number of processors required and lead to less complexity and a smaller overall volume for DABS.

In a third series of tests, processor utilization was measured for the active processors in the system. This was accomplished by dividing the tasks of each processor into appropriate subtasks and measuring the time spent in each subtask. These data were reduced to produce graphs showing average processor utilization as a function of azimuth. In addition, a graph of target loading as a function of azimuth was generated in order that processor utilization in terms of target loading could be ascertained.

Of the 47 software tasks for which measurements were made, only 5 showed signs of possible processor overload. Three of these were ATCRBS related functions. The other two were DABS data link related functions which approached processor saturation under loads of approximately 400 DABS targets within 90° of azimuth.

Several of the processors, mainly those containing performance monitor functions, were very lightly utilized. The tasks of these processors are likely candidates for relocation if it is found desirable to reduce the number of processors in the computer subsystem or to accommodate additional functions.

It was concluded that:

1. No data bus contention problem exists with the DABS software.

2. Processor local memory can be expanded to 12,288 words with no impact on system operation. Furthermore, an increase to 16,384 words appears feasible.

3. The only processor utilization problems found were due to peak sector loads. In the case of ATCRBS, software modifications have already been identified to rectify the problem. In the case of DABS, no system degradation was found, but potential problems exist with loads of approximately 400 DABS equipped aircraft within a 90° azimuth wedge for two data link tasks.

It is recommended that the processor utilization tests be repeated with later versions of the DABS software to measure the effectiveness of the software changes which have been made. Additionally, tests should be conducted on the software tasks not covered in this test series.

# INTRODUCTION

## OBJECTIVE.

The objective of this test and evaluation activity was to evaluate the performance of the computer subsystem within the Discrete Address Beacon System (DABS) engineering model sensors. The specific performance areas investigated were system data bus contention/utilization, global memory address space utilization, and DABS processor utilization. The results reported in this document are based on tests conducted at the Federal Aviation Administration (FAA) Technical Center.

## BACKGROUND.

The DABS engineering model sensors built by Texas Instruments Incorporated (TI) contain a unique computer subsystem architecture. This architecture consists of 36 processors tied together through a large central or global memory by a system of data buses. This architecture, referred to as a distributive architecture, was selected because of its potential to greatly improve total system reliability and simplify system software development.

During the period of initial hardware fabrication and software development, a software simulation package was developed by TI to study the effects of contention among the processors for access to the system data buses and global memory. This simulation package was also used to study interaction between software modules. The simulation study showed that, due to low predicted utilization for the system data buses, contention effects would be very low and could safely be ignored.

During final system development it became necessary to add additional processors to the system configuration. An engineering analysis indicated that the additional processors would only slightly increase system data bus utilization and that contention effects would remain negligible. However, the software simulation package was not modified to verify these conclusions.

During factory acceptance testing an attempt was made to measure data bus utilization. These measurements showed bus utilizations which were much higher than predicted. Although no degradation in sensor operation was evident, a more detailed study of the DABS sensor computer subsystem performance was determined necessary as an input toward procurement of field implementable sensors.

Therefore, following delivery of the DABS engineering model sensors to the FAA Technical Center, the test and evaluation activity reported on herein was instituted.

# DISCUSSION

## TEST APPROACH.

The computer performance testing was conducted on the DABS sensor located in Elwood, New Jersey. Testing was conducted under the following system constraints:

1. The DABS sensor software was "frozen" with TI software release 6.4. A load tape for this release was built and was used throughout the computer performance test effort.

2. All computer performance testing was conducted using a terminal radar scan rate of 4.75 seconds and a maximum sensor range of 60 nautical miles.

3. Simulated traffic scenario inputs generated by the Aircraft Reply and Interference Environment Simulator (ARIES) were used exclusively in determining the DABS computer performance characteristics.

The general test approach used in all computer performance test activities was as follows:

1. Computer performance measurement system probes were connected to selected points within the DABS sensors.

2. The computer performance measurement system was set up to monitor selected DABS sensor performance data associated with the selected probe points.

3. The DABS software was executed within the DABS sensor.

4. Computer performance measurement system data collection was initiated.

5. After 1 minute ARIES was started with an appropriate traffic scenario to provide a simulated traffic load to the sensor.

6. Approximately 15 minutes of data were collected with the computer performance measurement system.

7. All computer performance measurement system data tapes were saved for later reduction and analysis.

This sequence was repeated for each test run during which a subset of the desired performance data were collected for one of the available traffic scenarios. This procedure was repeated until all desired performance data were collected for each of the traffic scenarios.

## SYSTEMS DESCRIPTION.

DISCRETE ADDRESS BEACON SYSTEM. The DABS is a cooperative surveillance and communication system for air traffic control (ATC). Each aircraft is assigned a discrete address or unique code which permits data link communication to or from a particular aircraft. The data link operates integrally with DABS surveillance interrogations and replies.

The DABS sensor has two modes of operation: Air Traffic Control Radar Beacon System (ATCRBS) and DABS. The sensor uses the available processing time first for ATCRBS functions and then for DABS functions. When in the ATCRBS mode, DABS transmits an ATCRBS/DABS All-Call interrogation which is similar to today's ATCRBS interrogation with an additional (P4) pulse. An ATCRBS transponder is unaffected by the P4 pulse and responds with a normal ATCRBS reply. A DABS transponder recognizes the interrogation as a DABS All-Call interrogation and responds with a DABS All-Call reply containing its discrete address.

After determining the position and velocity of a DABS-equipped aircraft from successive DABS All-Call replies, the sensor places the target on its roll-call list. On a subsequent discrete interrogation, the DABS transponder can be locked-out from replying to All-Call interrogations, thereby eliminating unwanted replies.

A functional block diagram of the sensor is illustrated in figure 1. It consists of three major subsystems: (1) the interrogator and processor (I&P) subsystem, (2) the computer subsystem, and (3) the communication subsystem.

The I&P subsystem consists of an antenna, transmitter, multichannel receiver, modulator control unit, DABS and ATCRBS reply processors, and a sensor clock which is synchronized with WWVB.

The DABS software resides in the computer subsystem and performs radio-frequency (RF) channel management, DABS and ATCRBS surveillance processing, data link processing, and network management tasks.

The communication subsystem performs all functions related to processing of data transmitted to and received from

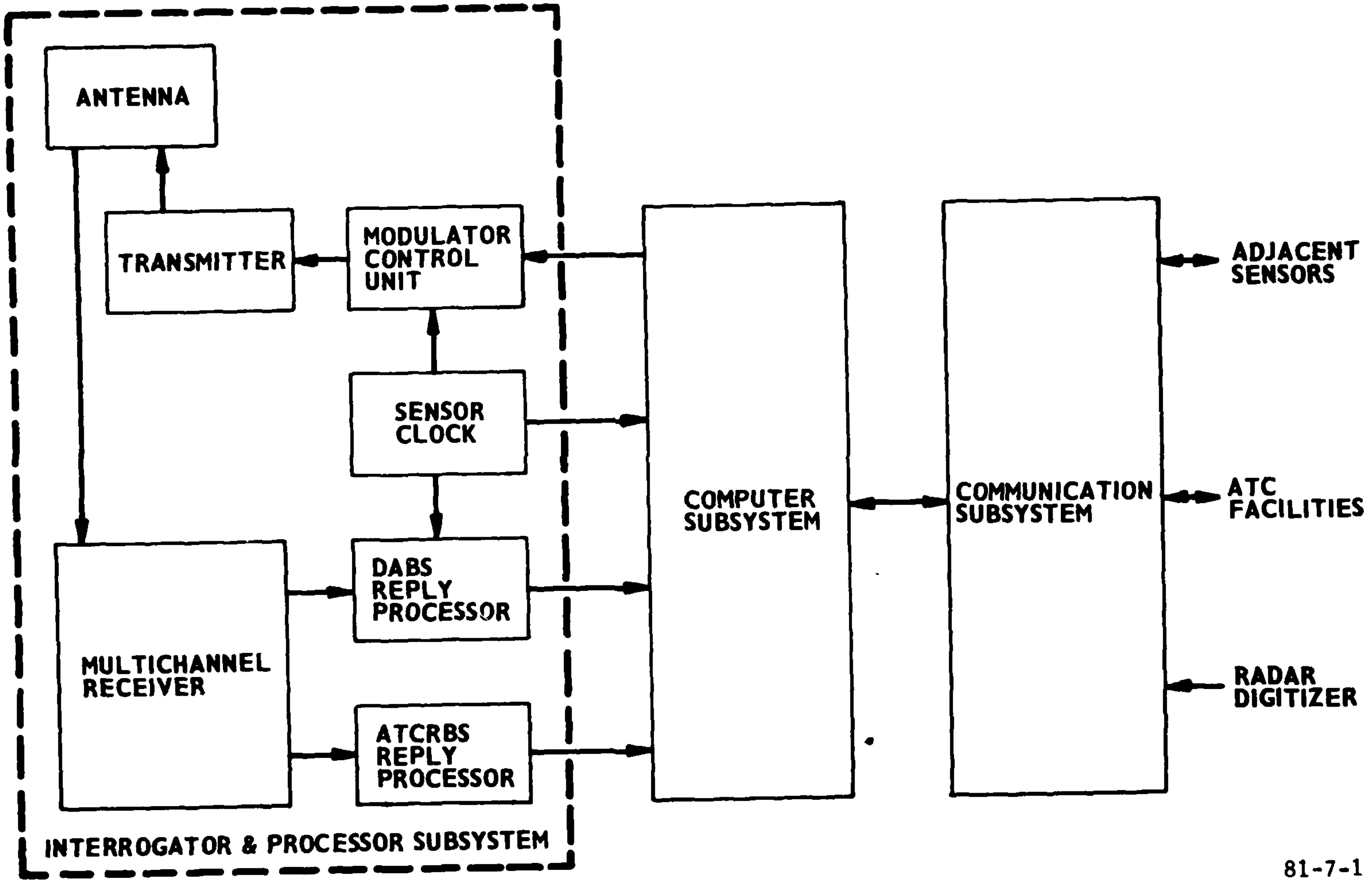


FIGURE 1. DABS SENSOR: FUNCTIONAL BLOCK DIAGRAM

adjacent DABS sensors, ATC facilities, and colocated radar digitizers.

Computer Subsystem Architecture. The DABS engineering model sensors incorporate a distributive computer subsystem architecture which features multiple use of common hardware modules such as processors, memory, couplers, and data buses. The application of redundancy at the module level supports the high reliability requirement of DABS. This architecture consists of 36 individual processors tied together through a central or global memory by a system of data buses. All communication among processors is through the global memory, such that each processor with its subtasks becomes an independent subsystem. This independence of tasks, along with the relative ease of adding processors to the system, greatly enhances the ability to modify or add tasks with little or no impact on the remainder of the tasks. Figure 2 illustrates a functional block diagram of the DABS computer subsystem.

Each processor consists of two arithmetic and logic units (ALU), voting logic for the ALU's, and 8,192 words of local error correcting code memory. The code of a processor is executed simultaneously by both its ALU's. Whenever memory access requests are generated, the outputs of both ALU's are checked for agreement. The voting logic will allow requests to be processed only when the following criteria are all satisfied:

1. Both requests generate the same memory address.

2. Memory operations being compared are of the same type (read or write).

3. During a memory write cycle the data from both ALU's are the same. If the requests do not match, the processor involved is immediately halted in order to prevent erroneous data from being passed to memory.

The processors used in the engineering model sensor make use of a 15-bit memory address. Although addresses are normally specified as 16 bits, only the most significant 15 bits are available on the memory address bus. The least significant bit specifies which 8-bit byte of the 16-bit word is used for byte type operations and is only used internally to the processor.

Of the 32,768 words which can be addressed by a processor, 8,192 words are contained in its local memory. The remaining 24,576 words are contained in global memory. As global memory far exceeds the 24,576 word addressing space available to the processor, the global memory address is expanded from the 15-bit processor generated address to a 20-bit address whenever the address generated exceeds the bounds of the local memory. The 20-bit address is formed by adding the contents of a software-loadable bias register to the processor generated address. By changing the contents of the bias register, the processor may establish a "window" into any area of global memory. Two bias registers are available and the global memory window can be divided into two separate "windows" located in different areas of global memory. Changing the location of the desired global memory window requires software to change the bias register settings.

The DABS sensor under test contained 36 processors. These processors were divided into groups referred to as ensembles. Each ensemble contains four processors. Of the 36 processors, 1, used for data extraction, was in an ensemble by itself. Three others, used for communication processing, shared an ensemble. The remaining 32 processors were divided between eight ensembles

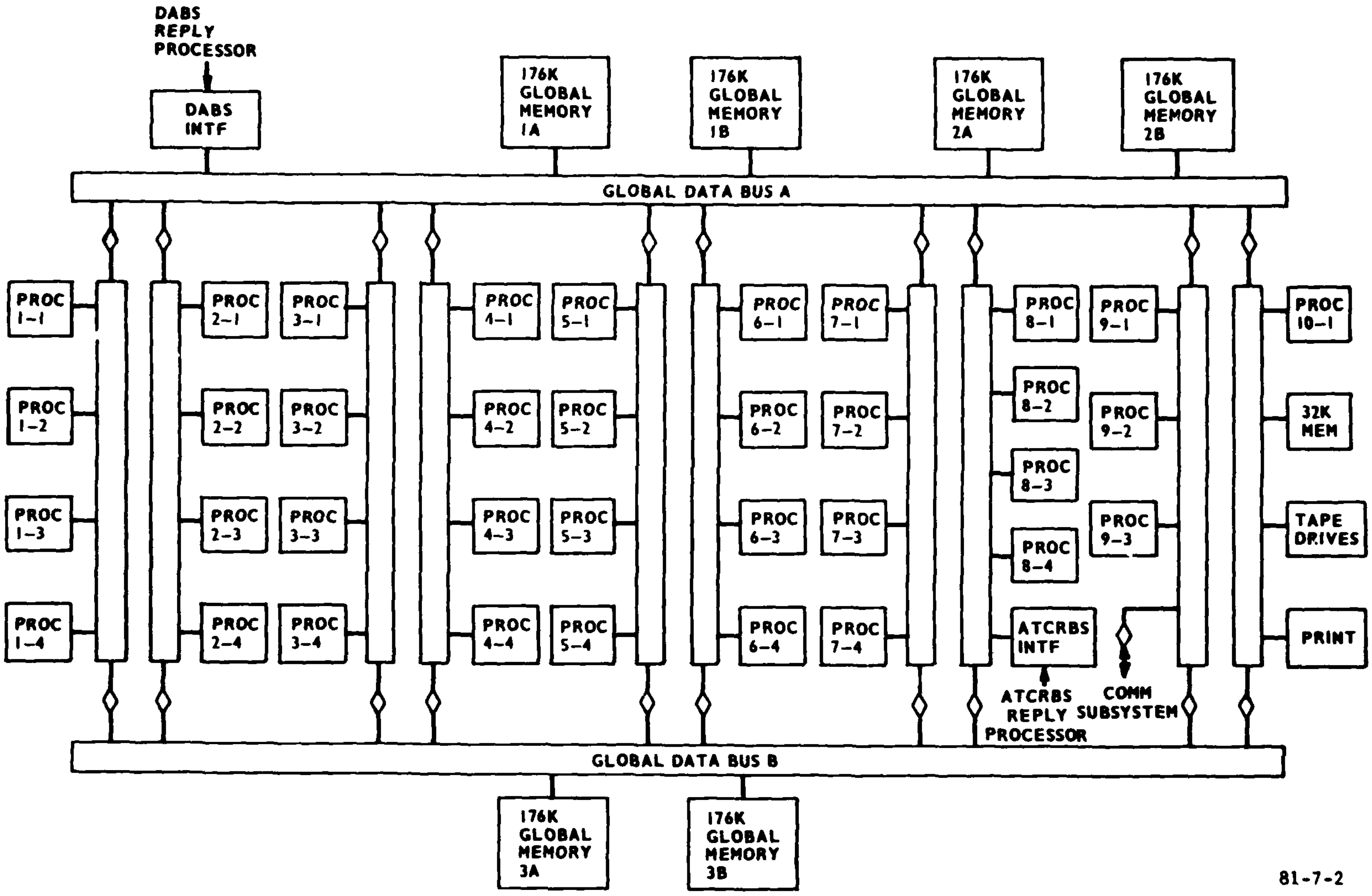


FIGURE 2. DABS COMPUTER SUBSYSTEM: ARCHITECTURAL BLOCK DIAGRAM

with 4 processors per ensemble. The processors of each ensemble were connected to an ensemble data bus through which they communicated with the remainder of the computer subsystem.

The 10 ensemble data buses were connected through coupler units to two global data buses to which the global memory units were attached. Three duplexed global memory units were provided in the sensor being tested. Two of these were connected to one of the global data buses, known as global A, and the third was connected to the other global data bus, global B.

Each of the global memory units consists of two 180,224 word error correcting memory units which operate in parallel with each other. During system operation the two memory units are initialized under software control, such that one executes both read and write operations while the other executes only write operations. If an error which is not correctable by the error correcting logic occurs, the software configures the failed memory unit off-line and continues operating using the other memory unit.

Control of the ensemble and global data buses resides in the priority circuitry on each data bus. This circuitry has an input and an output corresponding to each possible requester (processor or coupler) for access to the data bus.

When no processor or coupler has a memory access request pending, all outputs from the priority circuitry are set to enable memory accesses. When a processor or coupler requires use of a data bus to complete a global memory access, it generates an access request signal to the priority circuitry. This signal causes the priority circuitry to reset the enable signals for all other requesters of lower priority. After delaying for approximately 150 nanoseconds, the requester checks its enable signal. If the enable signal is set, this requester is the highest priority requester currently seeking an access to this data bus and may continue to take control of the data bus. If the enable signal is not set, another requester of higher priority is also seeking an access to this data bus and this requester must wait until the higher priority requester is serviced. When the higher priority requester is serviced, the enable signal for this requester will be reenabled, signalling this requester to continue.

When the requester is signaled by the priority circuitry that it has priority for access to the data bus, it checks to see if the data bus is currently busy. A single control line enabled by any requester on the data bus indicates the busy status of the control bus. If the data bus is not busy, the requester enables the busy line and takes control of the data bus. If the data bus is busy, the requester must wait until the bus busy control line is reset before taking control.

For the entire time that a requester is waiting for availability of the data bus, it holds its access request signal to the priority circuitry active. This signal is reset when the requester takes control of the data bus. Therefore, the duration of the access request signal is a direct measure of the delay experienced between generation of a request for bus access by a processor and the processor obtaining control of the bus.

To ensure that one requester is not locked out from receiving global memory accesses because of priority, the priority order of the requesters is not fixed. Priority is established as a rotating sequence. As each bus access is made, the priority order is rotated by one. The highest priority requester becomes lowest, and the second highest becomes highest. This ensures that all requesters will eventually become highest in priority and be guaranteed an access to the data bus.

Sensor Software. Release 6.4 of the DABS sensor software, adapted to the Elwood sensor with a 4.75 second (terminal) antenna scan rate and a maximum range of 60 nautical miles, was used throughout the computer performance testing. This software release requires 30 active processors. One of these, used for data extraction, will not be active in an operational sensor and, therefore, was not monitored during these tests. Two additional processors were used in the test configuration; the first for an extension to the data extraction software, and the second for a Common International Civil Aviation Organization (ICAO) Data Interchange Network (CIDIN) driver test package. The CIDIN driver test software generated the proper responses to communication messages transmitted to adjacent sensors and ATC facilities and, thus, allowed testing to progress without dependence on other facilities. No measurements were made on these two processors. The remaining four processors were configured as redundant spares, and remained unused during the test runs.

Each of the 36 processors had a unique load module which depended on the system tasks being performed in the processor. The 36 load modules were copied into global memory from the system load tape at system startup time and remained available in global memory throughout the test run. During system initialization the load modules were copied from global memory into the local memory of the associated processor. If a processor failed, recovery would consist of automatic reloading of the failed processor load module from global memory into the local memory of a spare processor and restarting of all processors in the system.

During sensor operation all executable codes are contained in local memory. Additionally, any temporary data storage required by a particular task are also contained in local memory. Only system data files and data required to be passed from one DABS processor to another are stored in global memory.

AIRCRAFT REPLY AND INTERFERENCE ENVIRONMENT SIMULATOR. The ARIES was designed by Lincoln Laboratory to simulate DABS and ATCRBS target replies, ATCRBS fruit replies, air-ground data link messages, and radar reports. The equipment consists of interrogation receiving circuitry, reply generation circuitry, and a computer with associated peripheral equipment and is housed in two standard racks. A complete description of ARIES is contained in Report No. FAA-RD-78-96 (reference 2).

The interrogation interface between the DABS sensor and the ARIES is at the RF level. The replies generated by ARIES are injected into the DABS sensor at the intermediate frequency (IF) level of the receiver.

Along with simulated traffic, ARIES can generate a simulated ATCRBS fruit environment. The average fruit rate can be controlled by setting parameters in a file on the ARIES system disk. In addition to the beacon data, ARIES can provide simulated digitized radar data. The radar targets correspond to the associated simulated beacon targets. The reported coordinates are those seen by a primary radar whose antenna rotates with the beacon antenna about the same axis. The radar reply probability (the probability that a beacon target will also have a radar return) for ARIES can be controlled by setting parameters in a file on the ARIES system disk.

Azimuth information in the form of azimuth change pulses (ACP) and azimuth reference pulses (ARP) can be generated by the ARIES. The rate at which these pulses are generated can be controlled by means of internal ARIES switches.

Traffic Scenarios. Two traffic scenarios were selected to test the DABS under anticipated ATC environments in the future. These scenarios were based on a predicted capacity traffic environment for the Los Angeles Basin area. Both scenarios had identical flight paths, but the mixture of aircraft transponder type (DABS or ATCRBS) varied between the scenarios.

The first scenario, referred to herein as the mixed scenario, contained a random mixture of approximately 50 percent DABS transponder equipped and 50 percent ATCRBS transponder equipped aircraft. The second scenario had all DABS transponder equipped aircraft. Figures 3A and 3B show the variation with time of the DABS and ATCRBS loads, respectively, for the mixed scenario. Figure 3C shows the variation with time of the total aircraft load, which was identical for all scenarios.

The traffic scenarios employed had all targets within approximately a 110° wedge (from 160° to 270° in azimuth). In addition, downlink COMM B messages were generated for 20 percent of the DABS aircraft.

COMPUTER PERFORMANCE MEASUREMENT SYSTEM. The computer performance measurement system was designed to collect computer performance data on the DABS computer subsystem. The system consists of a Digital Equipment Corporation (DEC) model PDP 11/55 computer, a data collection unit, and probes for connection to the DABS sensor. A description of this system is given in appendix A.

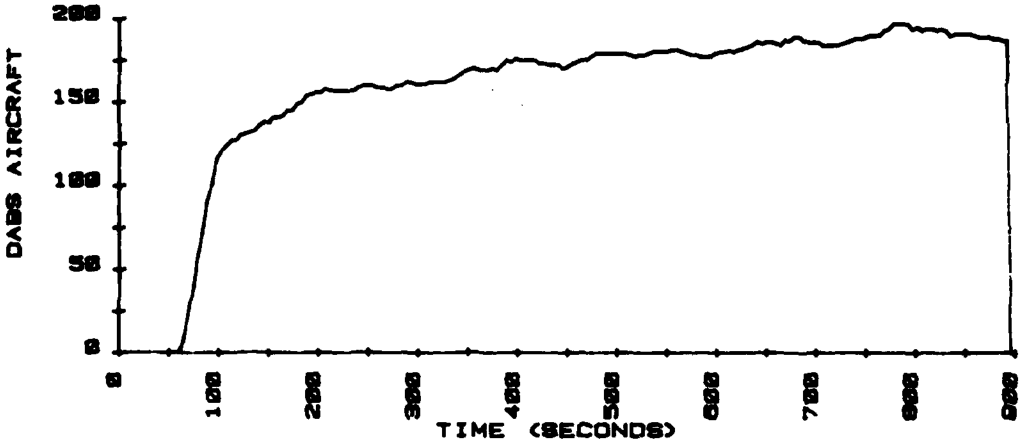
HIGH SPEED MEMORY ACCESS TEST SOFTWARE. During portions of the computer performance testing it was found desirable to impose a constant processing load on all of the processors. In addition, it was necessary to vary the level of the load which was imposed on the ensemble and global data buses from run to run. A diagnostic software package, the High Speed Memory Access Test (HSMAT), was used for this purpose.

This program simultaneously executes in all 36 processors. As is the case with the DABS operational software, the executable code is copied into the local memory of each processor during initialization and is executed from local memory. The test is designed to measure the number of global memory accesses granted to each processor in the system for various levels of attempted accesses per second.

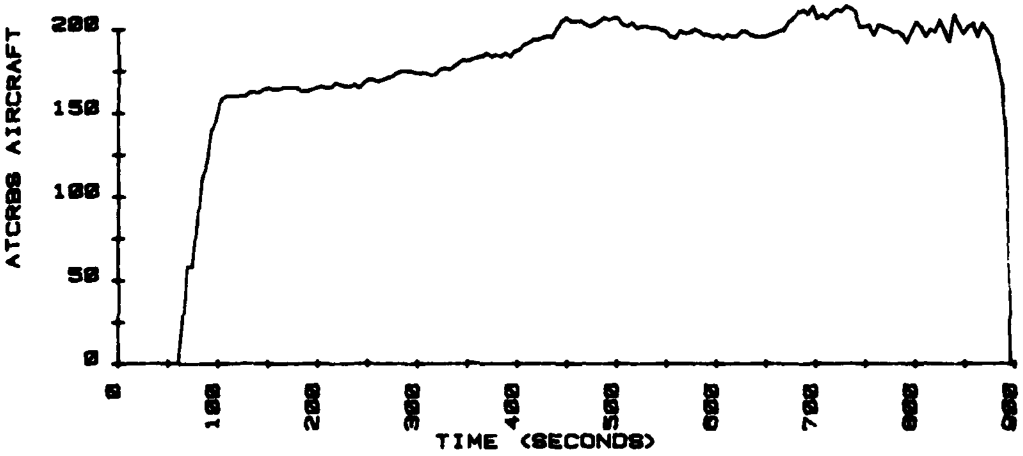
When initialized, the program requests entry of a "delay" parameter and clears a global memory table containing an entry for each processor in the system. The delay parameter is used to control operation of two program loops. When started, the program loads the delay parameter into a counter location in local memory and then loops, decrementing the counter by one until the counter contains zero. At this time the global memory table entry for this processor is incremented by one, the counter reloaded, and the decrementing loop reentered. When the program is terminated, the contents of the global memory table are printed showing the number of iterations executed by each individual processor.

For each iteration through the program, each processor makes two accesses to global memory: one to read the current contents of the iteration counter, and the second to write the incremented value. The time required to complete one iteration and, therefore, the time between pairs of global memory accesses is directly proportional to the value of the delay parameter, or number of times the local memory counter is decremented per iteration.

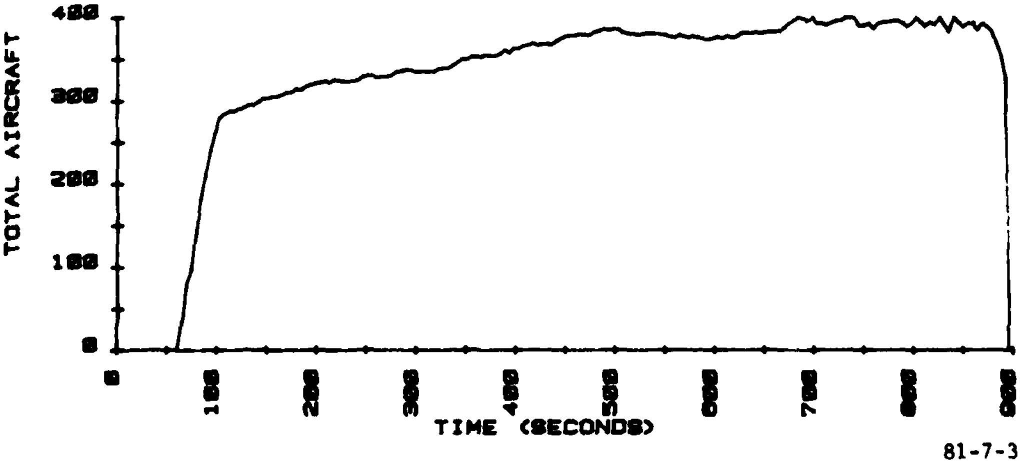
A. MIXED SCENARIO - DABS AIRCRAFT LOADING



B. MIXED SCENARIO - ATCRBS AIRCRAFT LOADING



C. ALL SCENARIOS - TOTAL AIRCRAFT LOADING



81-7-3

FIGURE 3. TRAFFIC SCENARIOS: AIRCRAFT LOAD VERSUS TIME

## TEST CONFIGURATION.

During the computer performance testing the Elwood DABS sensor was configured as shown in figure 4. The ARIES was used as a source of simulated targets and was configured for a terminal radar environment with a 4.75-second scan rate.

The fruit replies generated by ARIES are totally random. Therefore, repeated runs of the same scenario with fruit result in different inputs to the DABS sensor. The difference is due to the random fruit. During previous testing, as reported in the DABS Baseline Test and Evaluation Report, FAA-RD-80-36, it was found that for the fruit rates anticipated at operational sites (approximately 4,000 fruit replies per second) only one software module experienced a measurable effect. Furthermore, the effect on that module (ATCRBS reply-to-reply correlation) had been found to be minimal. Therefore, in order to keep the test runs as repeatable as possible, fruit replies were not generated during these tests.

The ARIES radar reply probability was set to 50 percent. This was the largest value which could be processed by ARIES with the target scenarios used without overloading the available radar interface to the DABS sensor.

The DABS sensor was loaded with the release 6.4 DABS software, including an extra data extraction processor and the CIDIN driver software package. A list of the software load modules used and the DABS processors in which each was executed is included in table 1. Site adaptation was included to specify a 4.75-second scan rate and a 60-nautical mile maximum range.

Signal probe units were installed in the DABS computer subsystem to monitor signals on the ensemble data buses and global data buses. One probe was installed on each of nine ensemble data buses. The six inputs of each probe were connected as shown in table 2. Two probes were installed on each global data bus. The 12 inputs of each probe pair were connected as shown in table 3.

Memory address probe units were connected to the breakpoint panel interface connectors on the DABS front panel as required for specific tests. The azimuth probe cables from the computer performance measurement system were connected to the ACP and ARP outputs of the ARIES.

## DATA COLLECTION.

The computer performance tests were divided into four series of test runs: (1) data validation, (2) data bus contention/utilization, (3) global memory address space utilization, and (4) processor utilization.

DATA VALIDATION. The data validation test runs were designed to verify that the data collection scheme, using the computer performance measurement system, provided valid data on the operation of the DABS computer subsystem. The tests were designed to show if differences in sampling rate or sampling technique (time or azimuth) would affect the data distributions obtained during the test runs.

For these tests the HSMAT program was loaded and executed simultaneously in all 36 processors. The HSMAT program caused a constant level of activity to be imposed on all processors, ensemble and global data buses, and global memory.

The computer performance measurement system was connected to measure the activity between the four processors of ensemble No. 1 and global memory. Specific measurements included:

1. Duration of the access request signal for each of the four processors. This measurement gave the duration of the delay experienced by each processor in accessing the ensemble data bus.

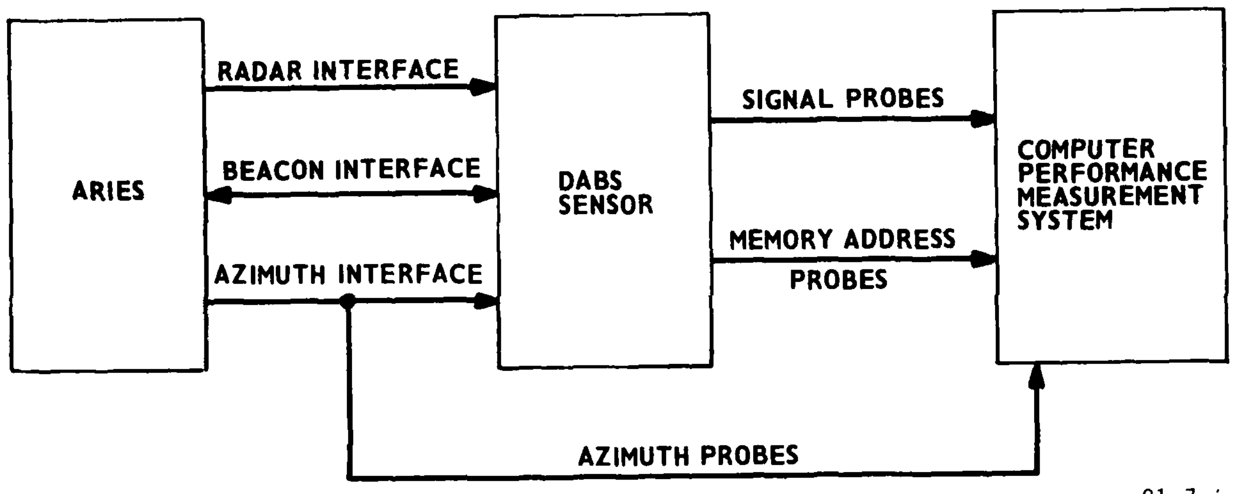


FIGURE 4. TEST CONFIGURATION

# TABLE 1. SENSOR LOAD MODULE LISTING

| Ensemble | Processor | Load Module | Sub Tasks |
|---|---|---|---|
| 1 | 1 | SS001X | CM: Beacon Formatting<br>CM: Target List Update |
| 1 | 2 | SS014X | ST: DABS Track Update |
| 1 | 3 | SS005X | ST: ATCRBS Track Initiation |
| 1 | 4 | SS01BX | SS: Spare Processor |
| 2 | 1 | SS002X | CM: Transaction Preperation<br>CM: Transaction Update |
| 2 | 2 | SS016X | ST: ATCRBS Track Update |
| 2 | 3 | SS017X | ST: DABS Track Initiation<br>IPC: Incoming Conflict Table Processing |
| 2 | 4 | SS01AX | SS: Spare Processor |
| 3 | 1 | SS012X | CM: Beacon Formatting |
| 3 | 2 | SS031X | ST: ATCRBS Target To Track Correlation |
| 3 | 3 | SS006X | DL: Output Post Processing |
| 3 | 4 | SS015X | DL: Output Processing |
| 4 | 1 | SS008X | ST: ATCRBS Target to Track Association |
| 4 | 2 | SS003X | ST: DPMS Roll-Call Processing<br>ST: DPMS All-Call Processing |
| 4 | 3 | SS004X | ST: Radar/Beacon Correlation<br>PM: Process Incoming PM Messages |
| 4 | 4 | SS00AX | IPC: Coarse Screen<br>PM: Hardware Configuration Tasks |
| 5 | 1 | SS00CX | IPC: Master Resolution |
| 5 | 2 | SS00DX | IPC: Sector Processing |
| 5 | 3 | SS00FX | IPC: Controller Alert<br>IPC: PWI/IPC Detect No. 2<br>PM: Process IPC Status Message |
| 5 | 4 | SS011X | DL: Input Processing |
| 6 | 1 | SS013X | NM: DABS Network Processing<br>NM: ATCRBS Network Processing |
| 6 | 2 | SS009X | MR: Message Routing<br>MR: Zero Address Message Processing<br>PM: Issue Sensor Status Report |
| 6 | 3 | SA01FX | NM: Process Incoming Network Message |
| 6 | 4 | SS00EX | NM: Send Northmark Messages<br>NM: Send Track Alert Message<br>NM: ATCRBS Radar Range Mask<br>PM: Adjacent Sensor Status Check<br>IPC: PWI/IPC Detect No. 1 |
| 7 | 1 | SS019X | SS: Primary Standby Spare |
| 7 | 2 | SS00BX | ST: Surveillance Data Formatting and Dissemination<br>PM: DABS Roll-Call CPME Check<br>PM: ATCRBS CPME Check |
| 7 | 3 | SA01EX | PM: Service I&P Channels<br>PM: Process Sensor Software Inputs<br>PM: Process Sensor Hardware Inputs<br>PM: Declare Sensor Status<br>PM: Analyze CPME Test Results |
| 7 | 4 | SS010X | IPC: EPOCH Synchronization<br>IPC: State Vector Deletion<br>IPC: Data Link Message Construction<br>IPC: Outgoing Conflict Table Processing |

|  |  |  |  |
|---|---|---|---|
| 4 | 3 | SS004X | ST: Radar/Beacon Correlation<br>PM: Process Incoming PM Messages |
| 4 | 4 | SS00AX | IPC: Coarse Screen<br>PM: Hardware Configuration Tasks |
| 5 | 1 | SS00CX | IPC: Master Resolution |
| 5 | 2 | SS00DX | IPC: Sector Processing |
| 5 | 3 | SS00FX | IPC: Controller Alert<br>IPC: PWI/IPC Detect No. 2<br>PM: Process IPC Status Message |
| 5 | 4 | SS011X | DL: Input Processing |
| 6 | 1 | SS013X | NM: DABS Network Processing<br>NM: ATCRBS Network Processing |
| 6 | 2 | SS009X | MR: Message Routing<br>MR: Zero Address Message Processing<br>PM: Issue Sensor Status Report |
| 6 | 3 | SA01FX | NM: Process Incoming Network Message |
| 6 | 4 | SS00EX | NM: Send Northmark Messages<br>NM: Send Track Alert Message<br>NM: ATCRBS Radar Range Mask<br>PM: Adjacent Sensor Status Check<br>IPC: PWI/IPC Detect No. 1 |
| 7 | 1 | SS019X | SS: Primary Standby Spare |
| 7 | 2 | SS00BX | ST: Surveillance Data Formatting and Dissemination<br>PM: DABS Roll-Call CPME Check<br>PM: ATCRBS CPME Check |
| 7 | 3 | SA01EX | PM: Service I&P Channels<br>PM: Process Sensor Software Inputs<br>PM: Process Sensor Hardware Inputs<br>PM: Declare Sensor Status<br>PM: Analyze CPME Test Results |
| 7 | 4 | SS010X | IPC: EPOCH Synchronization<br>IPC: State Vector Deletion<br>IPC: Data Link Message Construction<br>IPC: Outgoing Conflict Table Processing |
| 8 | 1 | SA01DX | ST: ATCRBS Reply to Reply Correlation |
| 8 | 2 | SS007X | IPC: Add New Aircraft<br>IPC: EPOCH Update<br>PM: CPME Scheduled Tasks |
| 8 | 3 | SS029X* | DEX: Data Collection |
| 8 | 4 | SA020X | SS: Spare Processor |
| 9 | 1 | SC021X | COM: Surveillance Receive<br>COM: Surveillance Transmit |
| 9 | 2 | SC022X | COM: CIDIN Input<br>COM: CIDIN Output |
| 9 | 3 | SXCDNX* | SS: CIDIN Driver |
| 10 | 1 | SX024X* | DEX: Data Extraction |

NOTE:

CM - Channel Management
DL - Data Link
MR - Message Routing
NM - Network Management
PM - Performance Management
SS - Spare
ST - Sensor Tracking
COM - Communications
DEX - Data Extraction
IPC - Intermittent Positive Control
* - Although an active processor during testing, no measurements were made on these processors.

2

## TABLE 2. ENSEMBLE DATA BUS PROBE POINTS

| Probe Input | Signal Name | Signal Description |
|---|---|---|
| 1 | TLAREQ0 | Processor 1 Access Request |
| 2 | TLAREQ1 | Processor 2 Access Request |
| 3 | TLAREQ2 | Processor 3 Access Request |
| 4 | TLAREQ3 | Processor 4 Access Request |
| 5 | TLAV | Ensemble Data Bus Busy |
| 6 |  | Not Used |

## TABLE 3. GLOBAL DATA BUS PROBE POINTS

| Probe Input | Signal Name | Signal Description |
|---|---|---|
| 1-1 | TLAREQ0 | Ensemble 1 Access Request |
| 1-2 | TLAREQ1 | Ensemble 2 Access Request |
| 1-3 | TLAREQ2 | Ensemble 3 Access Request |
| 1-4 | TLAREQ3 | Ensemble 4 Access Request |
| 1-5 | TLAREQ4 | Ensemble 5 Access Request |
| 1-6 | TLAREQ5 | Ensemble 6 Access Request |
| 2-1 | TLAREQ6 | Ensemble 7 Access Request |
| 2-2 | TLAREQ7 | Ensemble 8 Access Request |
| 2-3 | TLAREQ8 | Ensemble 9 Access Request |
| 2-4 | TLAV | Global Data Bus Busy |
| 2-5 |  | Not Used |
| 2-6 |  | Not Used |

2. Duration of the ensemble data bus busy signal.

3. Duration of the access request signal for the coupler between the ensemble No. 1 data bus and global data bus A.

4. Duration of the global data bus A busy signal.

Although operation of the HSMAT program was totally independent of azimuth, ARIES ACP and ARP data were fed to the computer performance measurement system. This provided a fixed dummy azimuth interval for testing data in the azimuth mode. It also provided a fixed scan time interval to aid in data reduction.

As detailed in table 4, data collection sample rates were varied from 300 to 50 sample intervals per second in time mode. In azimuth mode, a sample rate of 32 ACP's per sample interval (approximately 107 samples per second) was used. Each test was run twice to assure repeatability. For each test run data were collected with the computer performance measurement system for approximately 5 minutes while the HSMAT program was executing in the processors.

SYSTEM DATA BUS CONTENTION. The system data bus contention tests were designed to measure delays experienced on global memory accesses due to contention between processors. The intent of these tests was to measure the variation in these delays as a function of system load. Additionally, the tests were designed to measure utilization of the ensemble and global data buses as a function of system load.

For these tests, two separate test series were run. In the first, the HSMAT program was used to generate fixed loads on the system. The delay parameter was used to establish the load for each test run. This parameter was varied from 5 to 40 as shown in table 5 (tests 1 through 20). The computer performance measurement system was connected to measure the activity between the four DABS processors of ensemble No. 1 and global memory. The same measurements were made as those detailed for the data validation tests. For each test run data were collected with the computer performance measurement system for approximately 5 minutes, while the HSMAT program was executing with the delay parameter set to the selected value. A separate test run was made for each value of the delay parameter.

TABLE 4. DATA VALIDITY TEST MATRIX

| Test | Sensor Conditions | Measurement Conditions |
|---|---|---|
| 1 | HSMAT - Delay = 10 | Ensemble No. 1, 300 Samples/sec |
| 2 | HSMAT - Delay = 10 | Ensemble No. 1, 250 Samples/sec |
| 3 | HSMAT - Delay = 10 | Ensemble No. 1, 200 Samples/sec |
| 4 | HSMAT - Delay = 10 | Ensemble No. 1, 150 Samples/sec |
| 5 | HSMAT - Delay = 10 | Ensemble No. 1, 100 Samples/sec |
| 6 | HSMAT - Delay = 10 | Ensemble No. 1, 50 Samples/sec |
| 7 | HSMAT - Delay = 10 | Ensemble No. 1, 32 ACP's/Sample Interval |

## TABLE 5. SYSTEM DATA BUS CONTENTION TEST MATRIX

| Test | Sensor Conditions | Measurement Conditions |
|---|---|---|
| 1 | HSMAT - Delay = 5 | Ensemble No. 1, 100 Samples/sec |
| 2 | HSMAT - Delay = 6 | Ensemble No. 1, 100 Samples/sec |
| 3 | HSMAT - Delay = 7 | Ensemble No. 1, 100 Samples/sec |
| 4 | HSMAT - Delay = 8 | Ensemble No. 1, 100 Samples/sec |
| 5 | HSMAT - Delay = 9 | Ensemble No. 1, 100 Samples/sec |
| 6 | HSMAT - Delay = 10 | Ensemble No. 1, 100 Samples/sec |
| 7 | HSMAT - Delay = 11 | Ensemble No. 1, 100 Samples/sec |
| 8 | HSMAT - Delay = 12 | Ensemble No. 1, 100 Samples/sec |
| 9 | HSMAT - Delay = 13 | Ensemble No. 1, 100 Samples/sec |
| 10 | HSMAT - Delay = 14 | Ensemble No. 1, 100 Samples/sec |
| 11 | HSMAT - Delay = 15 | Ensemble No. 1, 100 Samples/sec |
| 12 | HSMAT - Delay = 16 | Ensemble No. 1, 100 Samples/sec |
| 13 | HSMAT - Delay = 17 | Ensemble No. 1, 100 Samples/sec |
| 14 | HSMAT - Delay = 18 | Ensemble No. 1, 100 Samples/sec |
| 15 | HSMAT - Delay = 19 | Ensemble No. 1, 100 Samples/sec |
| 16 | HSMAT - Delay = 20 | Ensemble No. 1, 100 Samples/sec |
| 17 | HSMAT - Delay = 25 | Ensemble No. 1, 100 Samples/sec |
| 18 | HSMAT - Delay = 30 | Ensemble No. 1, 100 Samples/sec |
| 19 | HSMAT - Delay = 35 | Ensemble No. 1, 100 Samples/sec |
| 20 | HSMAT - Delay = 40 | Ensemble No. 1, 100 Samples/sec |
| 21 | DABS Software | Ensembles Nos. 1&2, 100 Samples/sec |
| 22 | DABS Software | Ensembles Nos. 3&4, 100 Samples/sec |
| 23 | DABS Software | Ensembles Nos. 5&6, 100 Samples/sec |
| 24 | DABS Software | Ensembles Nos. 7&8, 100 Samples/sec |
| 25 | DABS Software | Ensembles Nos. 9&1, 100 Samples/sec |
| 26 | DABS Software | Ensembles Nos. 1&2, 32 ACP's/Sample Interval |
| 27 | DABS Software | Ensembles Nos. 3&4, 32 ACP's/Sample Interval |
| 28 | DABS Software | Ensembles Nos. 5&6, 32 ACP's/Sample Interval |
| 29 | DABS Software | Ensembles Nos. 7&8, 32 ACP's/Sample Interval |
| 30 | DABS Software | Ensembles Nos. 9&1, 32 ACP's/Sample Interval |

In the second test series, measurements were made with the DABS software executing in the sensor. The computer performance measurement system was connected to measure the activity between the eight processors of the two ensemble combinations defined in table 5 (tests 21 through 30) and global memory. Specific measurements made were as follows:

1. Duration of the access request signals for each of the eight processors.

2. Duration of the ensemble data bus busy signals for the two ensembles.

3. Duration of the four access request signals from the couplers of the two ensemble data buses to the two global data buses.

4. Duration of the two global data bus busy signals.

System data bus contention data were collected for sample rates of 100 samples per second in time mode and 32 ACP's per sample in azimuth mode. All tests were conducted with the mixed and all DABS ARIES traffic scenarios.

GLOBAL MEMORY UTILIZATION. The global memory utilization test runs were designed to determine what portion of the 24,576 word address space reserved for global memory accesses was actually utilized by each processor. These data were used to study the potential impact of expanding the current 8,192 word processor local memory to 12,288 words or greater.

Three tests were conducted with each ARIES scenario. For each test the memory address monitors were connected to a different set of 10 processors. All active processors were measured. The computer performance measurement system was set up to collect address samples for the processors at a rate of 300 samples per second.

PROCESSOR UTILIZATION. The processor utilization test runs were designed to measure the percent of time during which each processor was actively processing data. As the software executes continuously, looping through each of the subtasks checking global memory until it finds data to be processed, this measurement can not be made directly.

Each subtask was studied and broken down into sequences of code which reflected the presence of real work. In all, 47 such subtask sequences were identified. They are listed in table 6. These task sequences were further divided into four test groups for data collection. The entry and exit locations of the code sequences were identified. The computer performance measurement system was configured to measure the time spent executing code between the entry and exit locations. Data were collected with the computer performance measurement system while the DABS software was executed.

During preliminary testing it was discovered that measurements taken with the DABS software were not reliable due to a program error within the intermittent positive control (IPC) software logic of release 6.4. This error would randomly cause IPC function processors to stop executing. Any other function which shared a processor with an IPC function was affected as it would also be halted. As the IPC functions have been replaced by the Automated Traffic Advisory and Resolution Service (ATARS) functions, no measurements were taken on the IPC functions during processor utilization testing. In order to complete the desired measurement activity without the necessity for constant tedious data validation, the IPC functions were flagged as nonoperational. This allowed functions sharing a processor with an IPC function to execute normally during all test runs.

## TABLE 6. PROCESSOR UTILIZATION TEST MATARIX

| Test | Ensemble | Processor | Software Task |
|---|---|---|---|
| 1 | 1 | 1 | CM: Target List Update — Wait on Transaction Update |
| | 1 | 1 | CM: Target List Update — Wait on Transaction Preparation |
| | 1 | 1 | CM: Target List Update Active |
| | 1 | 1 | CM: Beacon Scheduler — 4 MS Wait |
| | 1 | 1 | CM: Beacon Scheduler — Main Scheduler Active |
| | 1 | 1 | CM: Beacon Scheduler — Allocator Active |
| | 1 | 1 | CM: Beacon Scheduler — All-Call Active |
| | 3 | 1 | CM: Beacon Formatter |
| | 2 | 1 | CM: Transaction Update |
| | 2 | 1 | CM: Transaction Update Process One Roll-Call Reply |
| | 2 | 1 | CM: Target Preparation Released Target List |
| | 2 | 1 | CM: Target Preparation |
| 2 | 1 | 3 | ST: ATCRBS Track Initiation File Purge |
| | 1 | 3 | ST: ATCRBS Track Initiation File Move |
| | 1 | 3 | ST: ATCRBS Track Initiation |
| | 8 | 1 | ST: ATCRBS Reply to Reply Correlation |
| | 2 | 2 | ST: ATCRBS Track Update — Bin Processing |
| | 2 | 2 | ST: ATCRBS Track Update |
| | 4 | 1 | ST: ATCRBS Target to Track Association |
| | 2 | 3 | ST: DABS Track Initiation |
| | 1 | 2 | ST: DABS Track Update — Bin Processing |
| | 4 | 2 | ST: DPMS Roll-Call Processing |
| | 4 | 2 | ST: DPMS All-Call Processing |
| 3 | 4 | 3 | PM: Process Incoming Messages |
| | 4 | 3 | S: Radar Beacon Correlation — Radar Processing |
| | 4 | 3 | S: Radar Beacon Correlation — ATCRBS Processing |
| | 4 | 3 | S: Radar Beacon Correlation — DABS Processing |
| | 4 | 3 | S: Radar Beacon Correlation — Radar Only Processing |
| | 3 | 2 | S: Target to Track Correlation |
| | 3 | 3 | DL: Output Post Processing |
| | [illegible] | [illegible] | DL: Output Processing |

| | | | |
|---|---|---|---|
| 2 | 1 | 3 | ST: ATCRBS Track Initiation File Purge |
| | 1 | 3 | ST: ATCRBS Track Initiation File Move |
| | 1 | 3 | ST: ATCRBS Track Initiation |
| | 8 | 1 | ST: ATCRBS Reply to Reply Correlation |
| | 2 | 2 | ST: ATCRBS Track Update — Bin Processing |
| | 2 | 2 | ST: ATCRBS Track Update |
| | 4 | 1 | ST: ATCRBS Target to Track Association |
| | 2 | 3 | ST: DABS Track Initiation |
| | 1 | 2 | ST: DABS Track Update — Bin Processing |
| | 4 | 2 | ST: DPMS Roll-Call Processing |
| | 4 | 2 | ST: DPMS All-Call Processing |
| 3 | 4 | 3 | PM: Process Incoming Messages |
| | 4 | 3 | S: Radar Beacon Correlation — Radar Processing |
| | 4 | 3 | S: Radar Beacon Correlation — ATCRBS Processing |
| | 4 | 3 | S: Radar Beacon Correlation — DABS Processing |
| | 4 | 3 | S: Radar Beacon Correlation — Radar Only Processing |
| | 3 | 2 | S: Target to Track Correlation |
| | 3 | 3 | DL: Output Post Processing |
| | 3 | 4 | DL: Output Processing |
| | 5 | 4 | DL: Input Processing |
| | 4 | 4 | PM: Hardware Configuration |
| | 6 | 1 | NM: DABS Network Processing |
| | 6 | 1 | NM: ATCRBS Network Processing |
| 4 | 6 | 2 | MR: Message Routing |
| | 6 | 2 | MR: Zero Address Message Processing |
| | 6 | 2 | PM: Sensor Status Report |
| | 7 | 2 | ST: Surv Data Formatting and Dissemination |
| | 7 | 2 | PM: DABS Roll-Call CPME Check |
| | 7 | 2 | PM: ATCRBS CPME Check |
| | A | 2 | PM: PM: CPME Scheduled Tasks |
| | 6 | 4 | PM: Adjacent Sensor Status |
| | 6 | 4 | PM: Misc. Northmark Tasks |
| | 7 | 3 | PM: Service I&P Channels |
| | 7 | 3 | PM: Sensor Hardware Inputs |

2

## TEST RESULTS AND ANALYSIS

### DATA VALIDITY.

The data collected during the data validity test series were reduced for analysis in the following ways:

1. Frequency of signal occurrence summaries and total signal activity summaries were obtained for each signal measured for each test run. Summaries were averaged over the simulated 4.75-second scan interval provided by the ARIES ARP's.

2. Time histograms were obtained for each signal measured for each test run. The histograms showed the relative frequency with which the signal duration measured fell within specific time intervals.

3. For selected test runs, a statistical comparison, as described in appendix B, was made on the data collected. This comparison tested the statistical difference between the data collected for a particular signal on one test run and the data for the same signal obtained on another test run.

In all cases, analysis revealed no differences in the data distributions obtained during the various test runs for measurements taken on the same signal.

### SYSTEM DATA BUS CONTENTION.

The data for each HSMAT test were summarized based on the simulated 4.75-second scan rate. Histograms of signal durations were prepared for each signal monitored. The reduced data were plotted to provide the following data bus loading graphs:

1. Average delay between generation of access request by the processor and the granting of the access to the ensemble data bus as a function of total ensemble data bus accesses per second (figure 5A).

2. Average ensemble data bus busy as a function of total ensemble data bus accesses per second (figure 5B).

3. Average delay between generation of access request by the processor and the granting of the access to the ensemble data bus as a function of ensemble data bus busy (figure 5C).

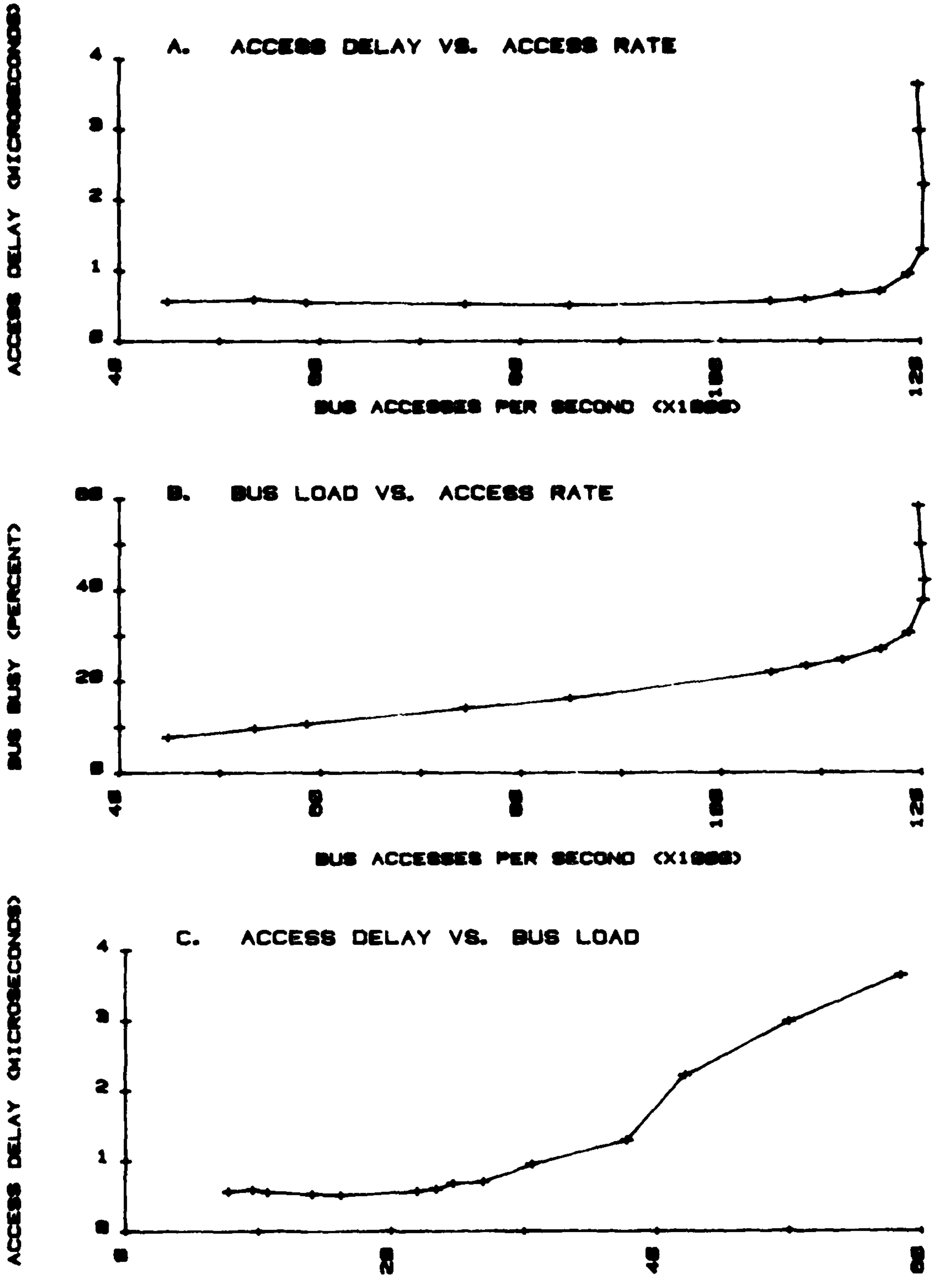
4. Average delay between generation of access request by the coupler and the granting of the access to global data bus A as a function of global data bus A accesses per second (figure 6A).

5. Average global data bus A busy as a function of global data bus A accesses per second (figure 6B).

6. Average delay between generation of access request by the coupler and the granting of the access to global data bus A as a function of global data bus A busy (figure 6C).

From figure 5A it can be seen that for ensemble data bus loads of under 115,000 accesses per second, the average delay experienced was relatively constant. This load level corresponded to an ensemble data bus load of 30 percent busy. Similarly, from figure 6A it can be seen that the global data bus average delay increased slightly up to loads of 1,250,000 accesses per second. This load level corresponded to a global data bus load of 88 percent busy.

When an attempt was made to increase ensemble and global data bus loads above these levels, both the average delay experienced per access and the busy time of the data bus increased rapidly. The global data bus peaked out at a constant load of 93 percent busy with an ever increasing delay in accessing the global A data bus. Attempts to increase the



81-7-5

FIGURE 5. ENSEMBLE DATA BUS LOADING WITH HSMAT SOFTWARE

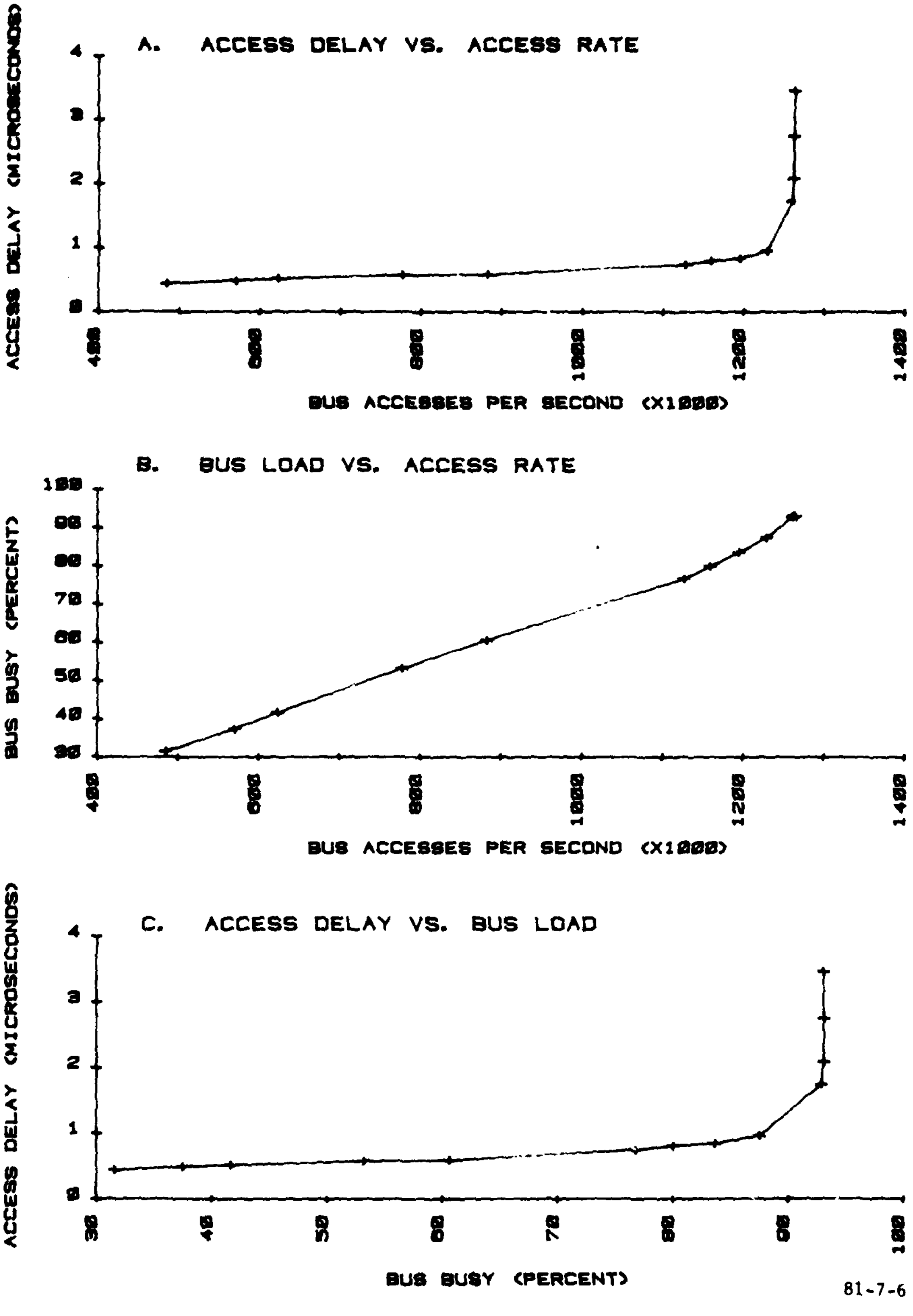


FIGURE 6. GLOBAL A DATA BUS LOADING WITH HSMAT SOFTWARE

bus ensemble data bus load above the level attained at 120,000 accesses per second (corresponding to approximately 40 percent bus busy) did increase the bus busy time. However, the number of accesses per second granted actually decreased. At load levels of this magnitude, decreasing the time interval between accesses to global memory did not increase the number of accesses per second granted, but rather, the rapidly increasing access delay experienced at these load levels caused a net decrease in the number of accesses per second granted.

The data for each data bus contention test with the DABS software were summarized over 10-scan intervals. This resulted in estimates of bus contention for five load conditions: no load, 300 mixed aircraft, 300 DABS aircraft, 380 mixed aircraft, and 380 DABS aircraft. The estimates for the global data buses are shown plotted in figure 7A along with the curves derived from the data measured with the HSMAT software. The data from the DABS software can be seen to fall upon the curve for the HSMAT software. However, the average delay for the DABS software corresponds with the lightest data bus loads imposed with the HSMAT software. This is well below the point at which data bus contention became a problem.

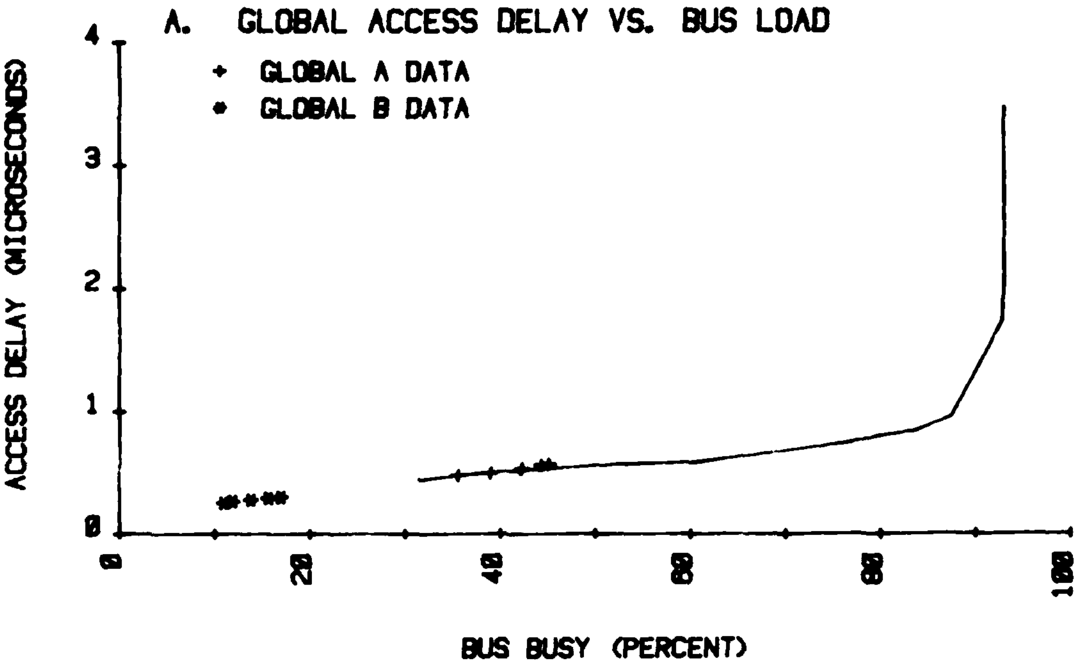
Figure 7B shows the average delay per access versus data bus busy data for nine ensemble data buses as measured with the DABS software for the same five load conditions. The curve derived from the data measured with the HSMAT software is included for reference. In this case, the correlation between the two sets of data is not as good. In fact, the data from the DABS software are consistently lower than the corresponding data from the HSMAT software. It is believed that the shift in the data is due to the fact that all global memory accesses in the HSMAT program came in pairs; while in the DABS software a more random distribution of times between individual accesses is present. For this reason, the delay versus load curves of figure 5 must be considered worst case. The actual delay experienced for a particular load should fall below the value given in the curve.

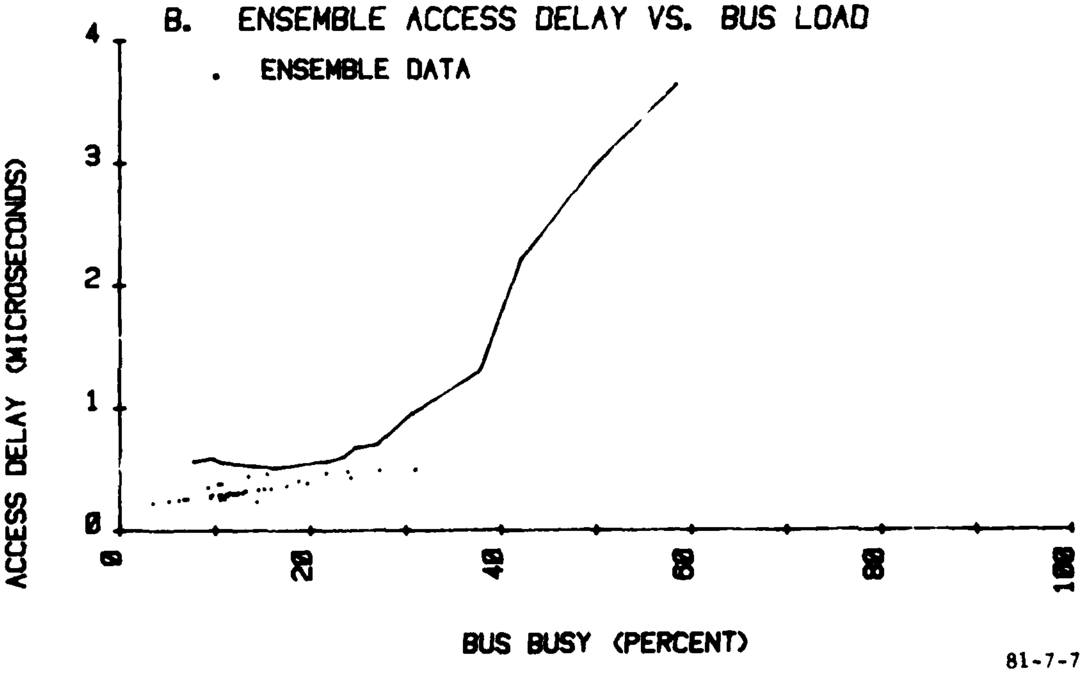
Further examination of the data revealed that only two of the ensemble data buses experienced loads of greater than 20 percent. Most ensemble data bus loads were in the 10 to 15 percent range with the DABS software. These values are well below the 30 percent level at which access delays started to increase with the HSMAT program.

A review of data bus access loading data in terms of number of accesses made per ensemble (see table 7) showed that ensemble 3 made the greatest number of accesses to global memory. Likewise, ensemble 4 had a high proportion of accesses, especially during those tests when this ensemble had large bus busy times.

A further breakdown of the access data for ensemble 3 revealed that processor 1 (the beacon formatting task of channel management) generated over 55 percent of the accesses for this ensemble in all test runs. This amounted to approximately 15 percent of all global memory accesses and was the greatest number of accesses for any single processor. During the mixed scenario tests, processor 3 of this ensemble (the ATCRBS target to track correlation task of sensor tracking) also generated a high number of accesses (18 percent of the ensemble and 7 percent of the total).

Ensemble 4, processor 4 (IPC course screening), likewise, generated a high number of accesses (six to eight percent of the total). Ensemble 4, processor 3 (radar beacon correlation task of sensor tracking) had a high number of accesses (six percent of the total) during the mixed scenario runs.





81-7-7

FIGURE 7. DATA BUS LOADING WITH DABS SOFTWARE

TABLE 7. DABS SOFTWARE — BUS ACCESSES

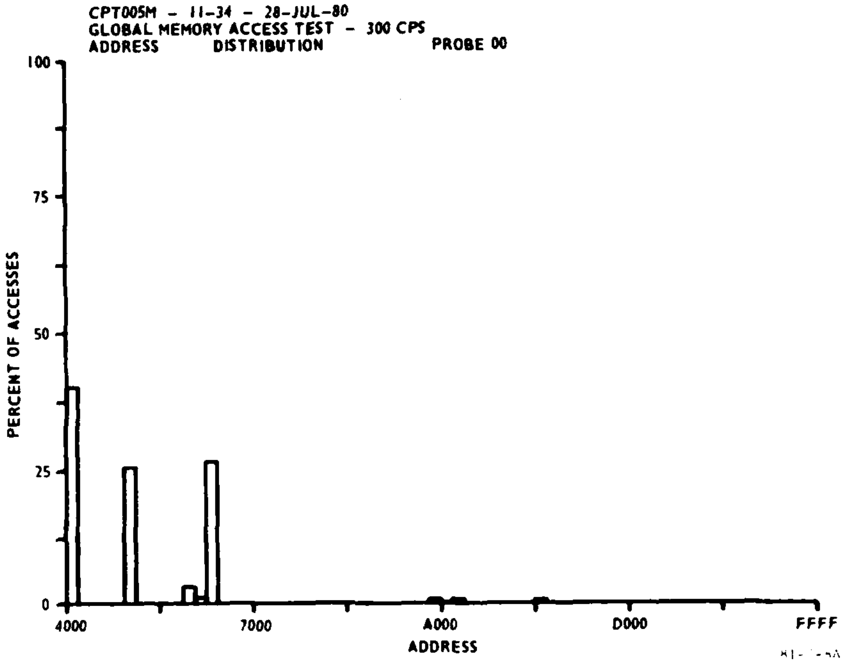
|  |  | Percent Of Global Memory Accesses |  |
|---|---|---|---|
| Ensemble | No Load | 190 ATCRBS 190 DABS | 0 ATCRBS 380 DABS |
| 1 | 11 | 11 | 13 |
| 2 | 11 | 7 | 10 |
| 3 | 19 | 24 | 20 |
| 4 | 14 | 18 | 14 |
| 5 | 12 | 11 | 14 |
| 6 | 9 | 7 | 8 |
| 7 | 11 | 10 | 9 |
| 8 | 8 | 8 | 7 |
| 9 | 5 | 4 | 5 |

The only other single processors to contribute greater than five percent of the total global memory accesses during the test runs were processor 4 of ensemble 7 and processor 1 of ensemble 1. Processor 4 of ensemble 7 (IPC deferred processing) generated six to seven percent of the total accesses during all tests. Processor 1 of ensemble 1 (The beacon formatting/ target list update task of channel management) generated seven percent of the accesses for the all DABS scenario tests.

GLOBAL MEMORY UTILIZATION.

The data collected during the global memory utilization tests were reduced to provide histograms of the addresses accessed by each processor. The data reduction software used provided the capability to specify the address range for which the histograms were produced. This range was set to encompass the processor generated addresses which cause accesses to be made to global memory. This created memory maps for addresses (in hexadecimal) of 4,000 through FFFF. Figure 8A illustrates an example of these histograms. In addition, graphs were produced showing only the lower four percent of these distributions (see figure 8B), thus, allowing the global memory space to be analyzed for address ranges containing no accesses. Bars that are open at the top indicate that the data are off the y-axis scale of the graph.

Analysis of these histograms revealed that for 11 of the processors, the total range of addresses accessed was less than 16,384 words (8,000 hexadecimal


A. 100% SCALE

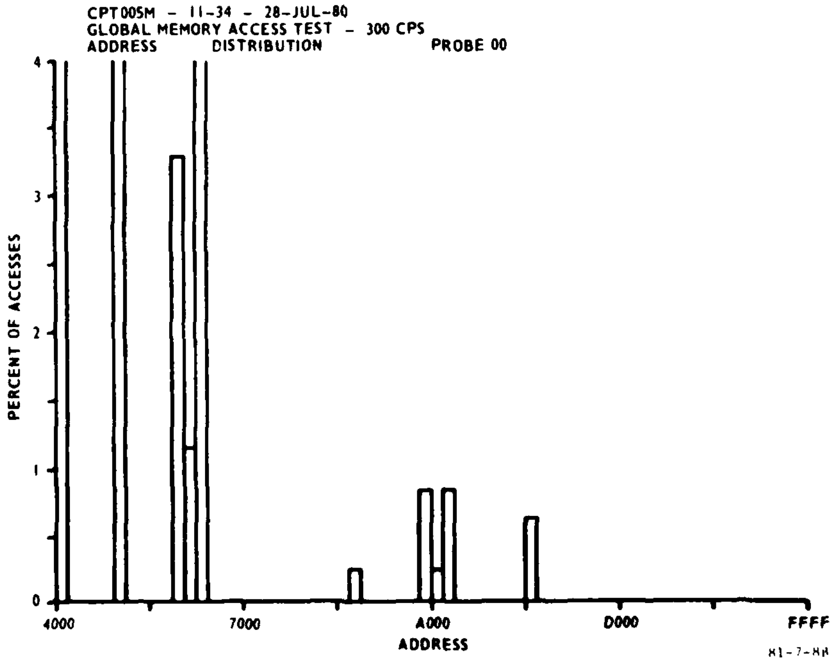

B. 4% SCALE

FIGURE 8. SAMPLE GLOBAL MEMORY ACCESS MAPS

bytes). This indicates that new values for the bias register settings can be found to enable these accesses to be made with no additional processor overhead and a local memory expanded from 8,192 words to 16,384 words.

An additional 13 processors utilized address ranges of less than 20,480 words (C000 hexadecimal bytes). These processors can execute without additional processor overhead with appropriate bias register setting changes and local memories of 12,288 words.

Only five processors would potentially require an increase in processor overhead to change the bias register settings more frequently with a local memory of 12,288 words. The global memory maps for these five processors are included in figures 9 to 13. In these processors, large expanses of the available address space are not used. A review of the software listings or recompilation of the data base may allow these processors to work with larger local memories and no increase in processor overhead to reload bias register settings.

The memory maps of the 13 processors having address ranges of greater than 16,384 words, but less than 20,480 words, also exhibit the property of large expanses of memory space not being utilized. A more detailed analysis of these may result in an ability to increase local memory size to 16,384 words with little increase in bias register overhead.

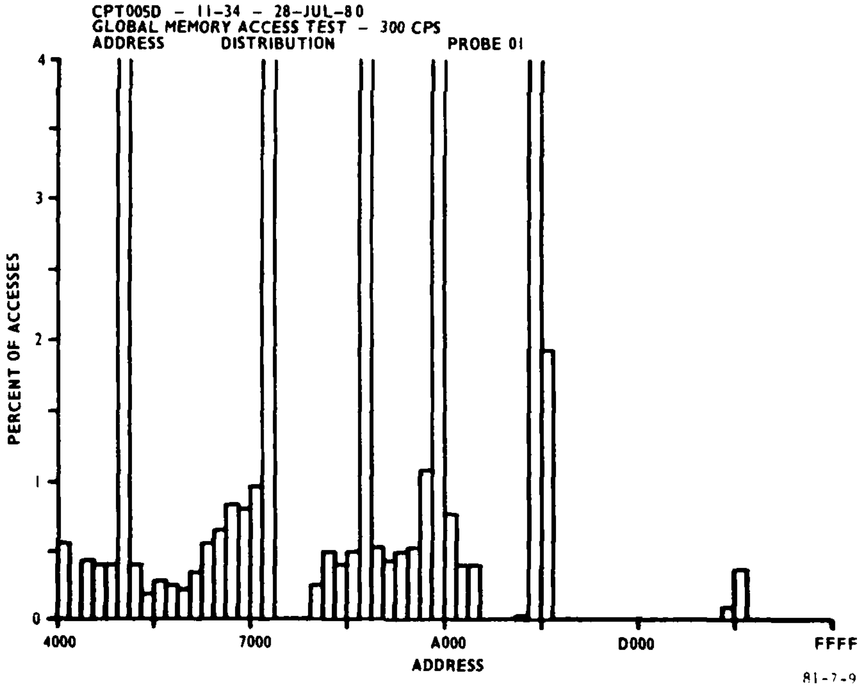


FIGURE 9. GLOBAL MEMORY ACCESS MAP, ENSEMBLE 1 PROCESSOR 2 (SS014X)

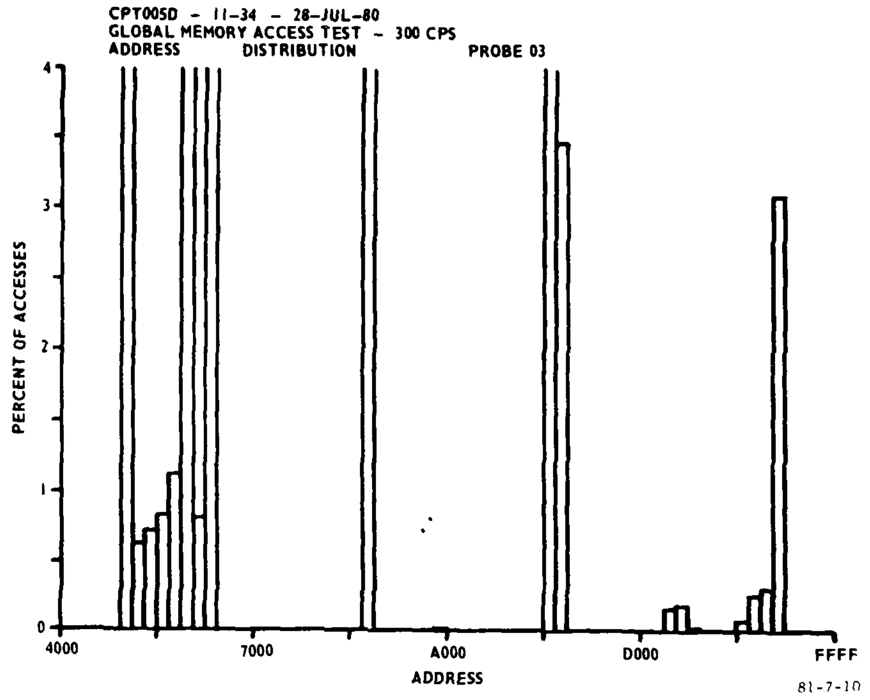


FIGURE 10. GLOBAL MEMORY ACCESS MAP, ENSEMBLE 2 PROCESSOR 1 (SS002X)

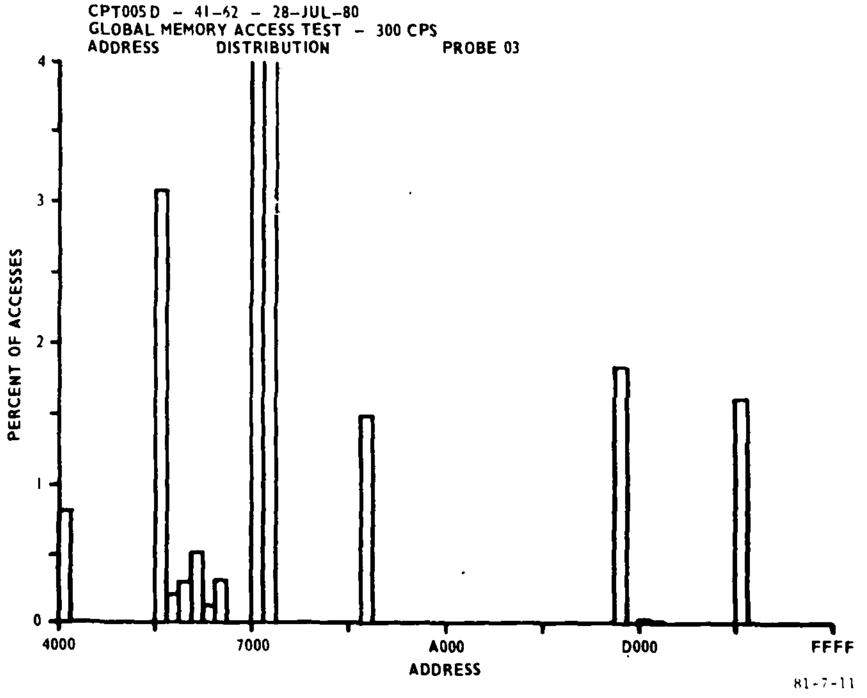


FIGURE 11. GLOBAL MEMORY ACCESS MAP, ENSEMBLE 4 PROCESSOR 4 (SS00AX)

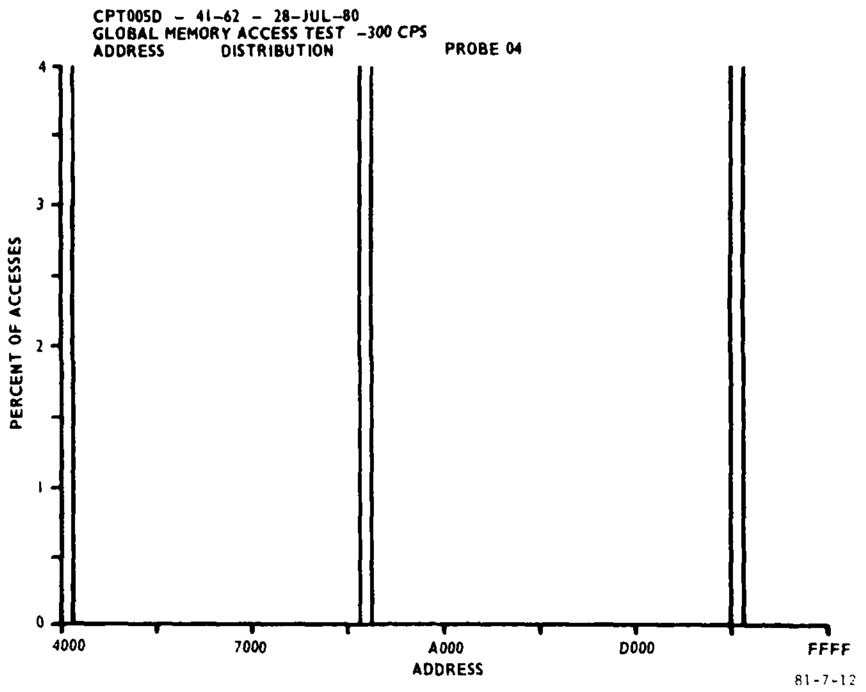


FIGURE 12. GLOBAL MEMORY ACCESS MAP, ENSEMBLE 5 PROCESSOR 1 (SS00DX)

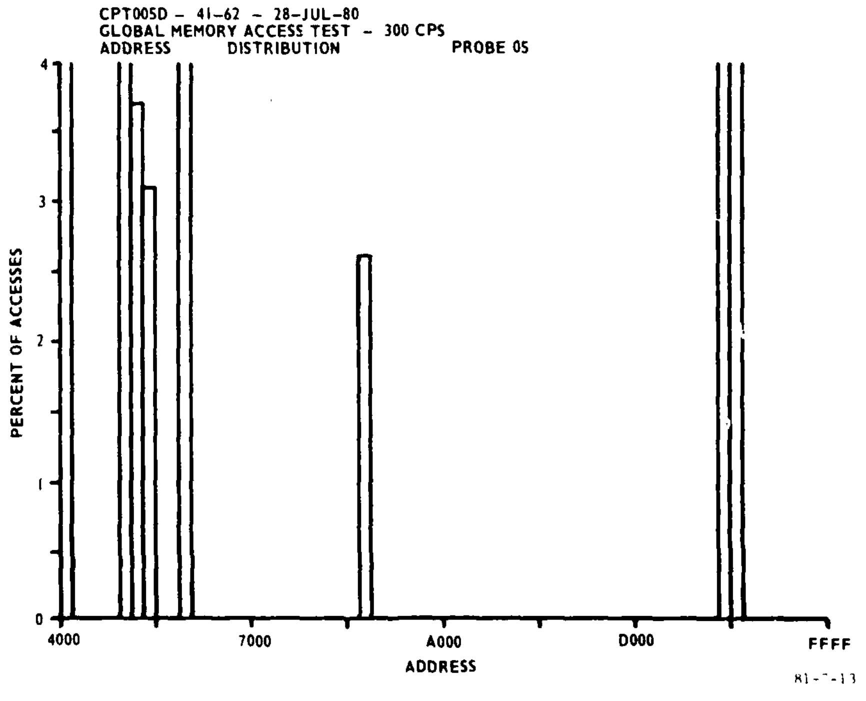


FIGURE 13. GLOBAL MEMORY ACCESS MAP, ENSEMBLE 5 PROCESSOR 2 (SS00DX)

## PROCESSOR UTILIZATION.

The data collected during the processor utilization tests were reduced to provide graphs showing average processor utilization as a function of azimuth for each of the 47 software tasks measured. Since the scenarios used build load gradually, graphs were produced for 10-scan intervals for which the aircraft loads were relatively constant. Loads chosen for analysis were: no load, 150 ATCRBS and 150 DABS, 190 ATCRBS and 190 DABS, no ATCRBS and 300 DABS, and no ATCRBS and 380 DABS. Figure 14 shows target loading as a function of azimuth in order that processor utilization in terms of target loading can be ascertained. Summaries of the data were obtained to provide average utilization data over the full scan. These data are tabulated in table 8.

As indicated in table 8, four of the software tasks resulted in no measurement data. An analysis showed that two of these tasks involved processing of adjacent sensor inputs. As these tests were run in a single sensor environment, no inputs were present and the data accurately reflected the fact that these two functions had no work to perform. The other two functions involved the processing of data link uplink messages. With no ATC scenario input and IPC inhibited, there was no work for these functions to perform. This was accurately reflected in the measurements.

An additional six software tasks resulted in identical measurements regardless of aircraft load. All six of these were performance monitor functions which are performed once per scan.

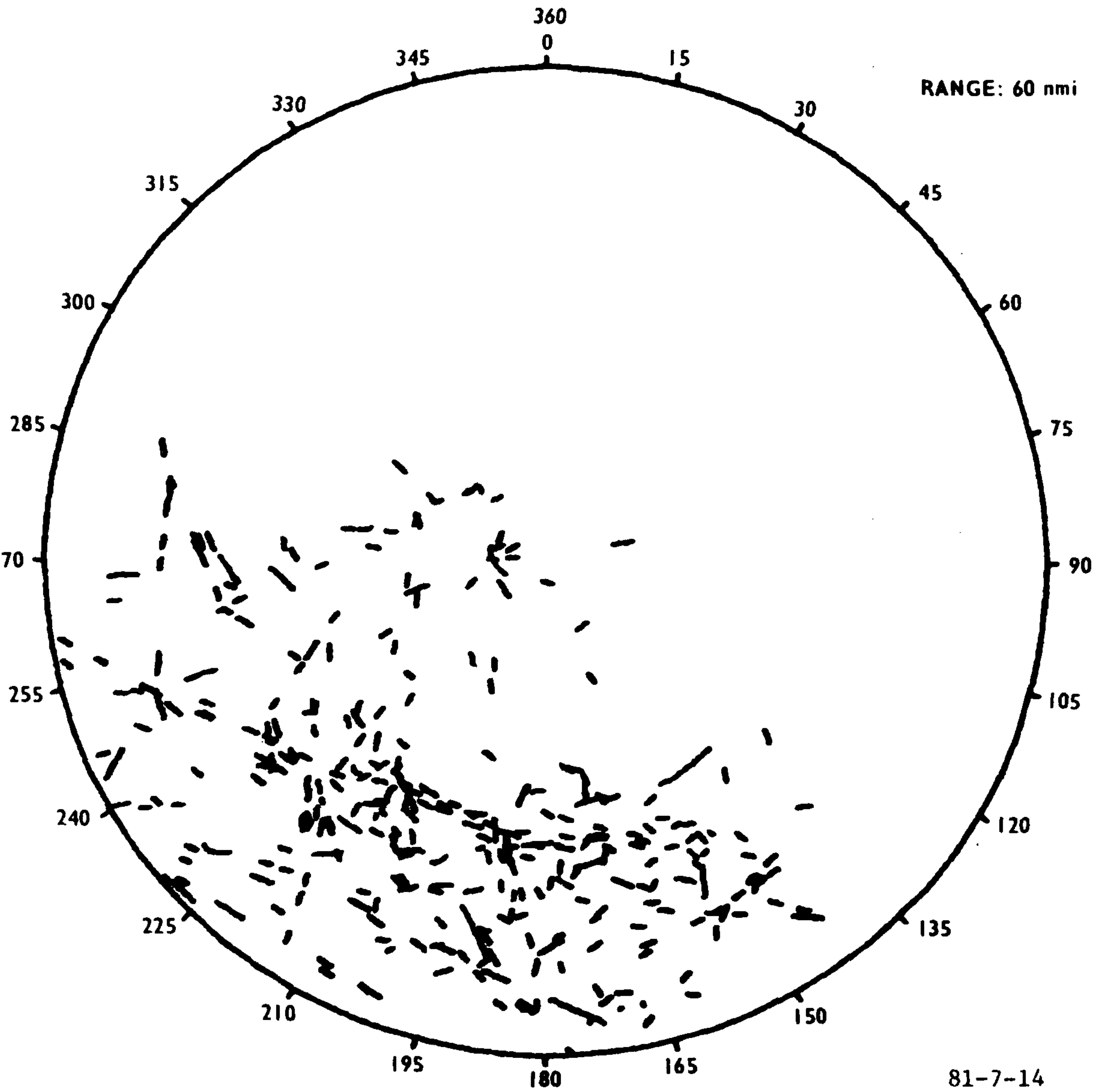
Five software tasks (identified in table 8) were found to have peak average utilizations approaching 100 percent during some part of the scan. The average utilization versus azimuth graphs for these functions are included in figures 15 through 19. Three of these, shown in figures 15, 16, and 17, are for ATCRBS tracking functions under aircraft loads of 190 ATCRBS and 190 DABS. Any further increase in the ATCRBS load above 190 targets in the 90° wedge would cause these three functions to saturate their respective processors.

The remaining two tasks (shown in figures 18 and 19) are data link downlink functions with aircraft loads of 380 DABS targets. These functions are also approaching saturation of their respective processors under the bunching conditions imposed by the scenario.

The data from table 8 were combined to provide average utilization data for each processor. These data are tabulated in table 9. Of the 22 processors which were measured, 4 also contain software tasks which were not measured (IPC tasks) and 8 others contain tasks which were discussed above. These are noted in table 9.

Of the remaining 10 processors, 2 others are thought to be particularly worthy of note. The first of these, ensemble 3 processor 3, containing load module SS012X, had a relatively constant utilization of approximately 25 percent. The graph of utilization versus azimuth for the only function contained in this processor (the beacon formatting task of channel management) for a load of 380 DABS targets is included as figure 20. This graph contains the highest fluctuations measured in the utilization for this processor.

Ensemble 7 processor 3, containing load module SS01EX, had the lowest utilization of any processor measured. Graphs for the two software tasks contained in this processor are included as figures 21 and 22. This processor is only utilized for brief periods of time immediately after azimuths of 0° and 180° in performing performance monitor tasks.



81-7-14

FIGURE 14. TARGET LOADING VERSUS AZIMUTH ALL SCENARIOS

# TABLE 8. TASK UTILIZATION DATA

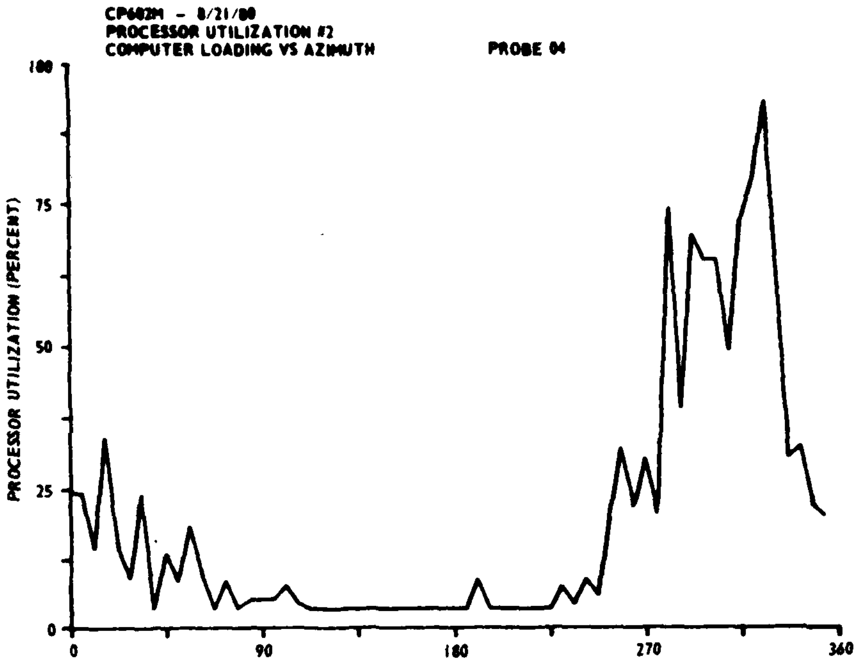
| | | Average Utilization Per Scan (Percent) | | | | |
|---|---|---|---|---|---|---|
| Processor | Software Task | No Load | 150 ATCRBS 150 DABS | 190 ATCRBS 190 DABS | 0 ATCRBS 300 DABS | 0 ATCRBS 380 DABS |
| 1-1 | CM: Target List Update<br>Wait for Transaction Update | 0.07 | 0.37 | 0.76 | 0.43 | 4.28 |
| | CM: Target List Update<br>Wait for Transaction Prep | 0.13 | 0.15 | 0.13 | 0.13 | 0.13 |
| | CM: Target List Update<br>Active | 0.65 | 0.85 | 0.96 | 0.86 | 1.48 |
| | CM: Beacon Scheduler<br>4 ms Wait | 82.43 | 67.66 | 63.77 | 65.35 | 52.87 |
| | CM: Beacon Scheduler<br>Main Scheduler Active | 0.00 | 0.88 | 1.31 | 0.99 | 2.80 |
| | CM: Beacon Scheduler<br>All-Call Active | 1.12 | 1.12 | 1.12 | 1.14 | 1.11 |
| 3-1 | CM: Beacon Formatter | 25.97 | 25.43 | 25.17 | 25.39 | 26.48 |
| 2-1 | CM: Transaction Update | 3.40 | 5.39 | 6.15 | 6.00 | 8.56 |
| | CM: Transaction Update<br>Process one Roll-Call Reply | 0.00 | 0.96 | 1.51 | 1.01 | 2.92 |
| | CM: Target Preparation<br>Released Target List | 0.00 | 0.54 | 0.90 | 0.59 | 1.92 |
| | CM: Target Preparation | 0.00 | 0.97 | 1.57 | 1.03 | 3.74 |
| 1-3 | ST: ATCRBS Track Initiation<br>File Purge | 18.04 | 23.76 | 24.46 | 22.75 | 18.09 |
| | ST: ATCRBS Track Initiation<br>File Move | 0.00 | 5.62 | 5.12 | 2.05 | 0.00 |
| | ST: ATCRBS Track Initiation | 0.00 | 6.94 | 3.91 | 4.83 | 0.00 |
| 8-1 | ST: ATCRBS Reply to Reply<br>Correlation | 5.55 | 12.06 | 13.79 | 5.61 | 5.53 |
| 2-2 | ST: ATCRBS Track Update<br>Bin Processing (See Note 3) | 3.57 | 14.94 | 19.52 | 3.64 | 0.14 |
| | ST: ATCRBS Track Update | 0.00 | 7.14 | 2.51 | 0.34 | 3.58 |
| 4-1 | ST: ATCRBS Target to Track<br>Association (See Note 3) | 4.95 | 31.71 | 38.01 | 5.05 | 4.96 |
| 2-3 | ST: DABS Track Initiation | 0.00 | 0.59 | 0.09 | 2.37 | 0.22 |
| 1-2 | ST: DABS Track Update<br>Bin Processing | 1.07 | 8.35 | 12.30 | 9.70 | 23.83 |
| | ST: DABS Track Update | 0.00 | 0.39 | 0.06 | 0.95 | 0.16 |
| 4-2 | ST: DPMS Roll-Call Processing | 0.00 | 7.07 | 12.04 | 9.12 | 23.91 |
| | ST: DPMS All-Call Processing | 6.92 | 11.94 | 7.41 | 21.83 | 8.23 |
| 4-3 | PM: Process Incoming Messages<br>(See Note 1) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | S: Radar Beacon Correlation<br>Radar Processing | 0.00 | 1.53 | 1.83 | 1.55 | 1.88 |
| | S: Radar Beacon Correlation<br>ATCRBS Processing | 0.00 | 9.38 | 7.56 | 0.29 | 0.00 |
| | S: Radar Beacon Correlation<br>DABS Processing | 0.00 | 8.10 | 8.56 | 14.77 | 20.68 |
| | S: Radar Beacon Correlation<br>Radar Only Processing | 22.44 | 46.45 | 54.37 | 21.30 | 19.63 |
| 3-2 | S: ATCRBS Target to Track<br>Correlation (See Note 3) | 2.14 | 22.53 | 34.40 | 1.40 | 0.00 |
| 3-3 | DL: Output Post Processing<br>(See Note 3) | 0.00 | 7.35 | 11.49 | 10.09 | 27.85 |
| 3-4 | DL: Output Processing<br>(See Note 3) | 0.00 | 9.27 | 15.59 | 12.51 | 31.88 |
| 5-4 | DL: Input Processing<br>(See Note 1) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 4-4 | PM: Hardware Configuration | 3.55 | 3.24 | 3.25 | 3.24 | 3.24 |
| 6-1 | NM: DABS Network Processing | 0.00 | 10.78 | 15.93 | 14.49 | 32.59 |
| | NM: ATCRBS Network Processing | 0.00 | 4.91 | 6.98 | 0.03 | 0.00 |
| 6-2 | MR: Message Routing<br>(See Note 1) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | | | | [illegible] | 2.33 | 4.55 |

|  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|
|  | ST: | **ATCRBS Track Initiation** | **0.00** | **6.**[illegible] | [illegible] | [illegible] | [illegible] |
| 8-1 | ST: | ATCRBS Reply to Reply Correlation | 5.55 | 12.06 | 13.79 | 5.61 | 5.53 |
| 2-2 | ST: | ATCRBS Track Update Bin Processing (See Note 3) | 3.57 | 14.94 | 19.52 | 3.64 | 0.14 |
|  | ST: | ATCRBS Track Update | 0.00 | 7.14 | 2.51 | 0.34 | 3.58 |
| 4-1 | ST: | ATCRBS Target to Track Association (See Note 3) | 4.95 | 31.71 | 38.01 | 5.05 | 4.96 |
| 2-3 | ST: | DABS Track Initiation | 0.00 | 0.59 | 0.09 | 2.37 | 0.22 |
| 1-2 | ST: | DABS Track Update Bin Processing | 1.07 | 8.35 | 12.30 | 9.70 | 23.83 |
|  | ST: | DABS Track Update | 0.00 | 0.39 | 0.06 | 0.95 | 0.16 |
| 4-2 | ST: | DPMS Roll-Call Processing | 0.00 | 7.07 | 12.04 | 9.12 | 23.91 |
|  | ST: | DPMS All-Call Processing | 6.92 | 11.94 | 7.41 | 21.83 | 8.23 |
| 4-3 | PM: | Process Incoming Messages (See Note 1) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
|  | S: | Radar Beacon Correlation Radar Processing | 0.00 | 1.53 | 1.83 | 1.55 | 1.88 |
|  | S: | Radar Beacon Correlation ATCRBS Processing | 0.00 | 9.38 | 7.56 | 0.29 | 0.00 |
|  | S: | Radar Beacon Correlation DABS Processing | 0.00 | 8.10 | 8.56 | 14.77 | 20.68 |
|  | S: | Radar Beacon Correlation Radar Only Processing | 22.44 | 46.45 | 54.37 | 21.30 | 19.63 |
| 3-2 | S: | ATCRBS Target to Track Correlation (See Note 3) | 2.14 | 22.53 | 34.40 | 1.40 | 0.00 |
| 3-3 | DL: | Output Post Processing (See Note 3) | 0.00 | 7.35 | 11.49 | 10.09 | 27.85 |
| 3-4 | DL: | Output Processing (See Note 3) | 0.00 | 9.27 | 15.59 | 12.51 | 31.88 |
| 5-4 | DL: | Input Processing (See Note 1) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 4-4 | PM: | Hardware Configuration | 3.55 | 3.24 | 3.25 | 3.24 | 3.24 |
| 6-1 | NM: | DABS Network Processing | 0.00 | 10.78 | 15.93 | 14.49 | 32.59 |
|  | NM: | ATCRBS Network Processing | 0.00 | 4.91 | 6.98 | 0.03 | 0.00 |
| 6-2 | MR: | Message Routing (See Note 1) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
|  | MR: | Zero Address Message Processing | 0.07 | 2.02 | 2.18 | 2.33 | 4.55 |
|  | PM: | Sensor Status Report (See Note 2) | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |
| 7-2 | ST: | Surv Data Formatting and Dissemination | 70.01 | 71.68 | 71.37 | 71.36 | 70.24 |
|  | PM: | DABS Roll-Call CPME Check | 0.00 | 2.30 | 3.66 | 3.02 | 7.44 |
|  | PM: | ATCRBS CPME Check | 0.00 | 2.55 | 3.58 | 0.01 | 0.00 |
| 8-2 | PM: | CPME Scheduled Tasks (See Note 2) | 0.07 | 0.07 | 0.07 | 0.07 | 0.07 |
| 6-4 | PM: | Adjacent Sensor Status (See Note 1) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
|  | PM: | Misc Northmark Tasks (See Note 2) | 0.04 | 0.04 | 0.04 | 0.04 | 0.04 |
| 7-3 | PM: | Service I&P Channels (See Note 2) | 0.24 | 0.29 | 0.31 | 0.29 | 0.29 |
|  | PM: | Sensor Hardware Inputs (See Note 2) | 0.84 | 0.84 | 0.84 | 0.84 | 0.84 |

NOTE:

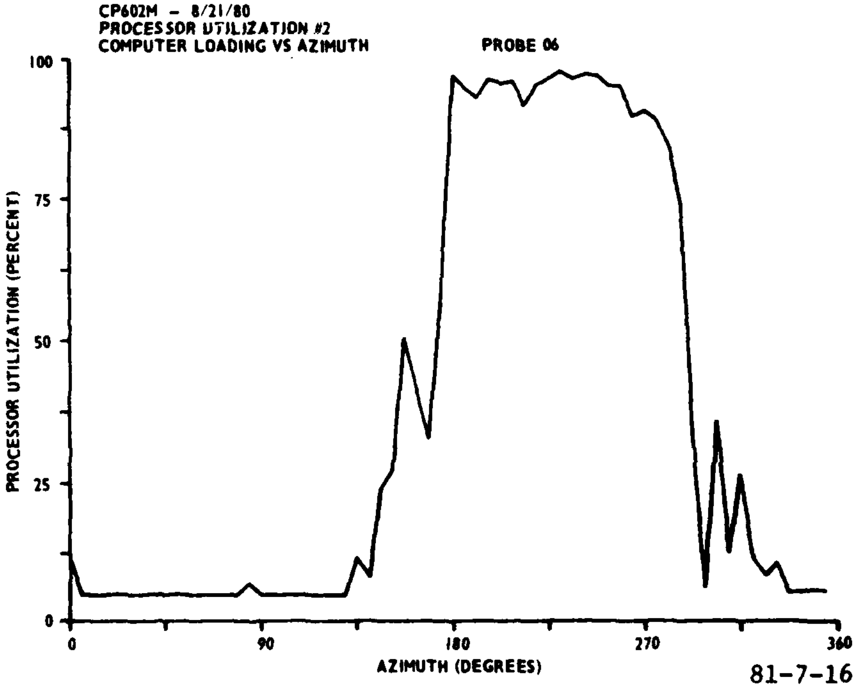
1. These tasks process inputs from the external world which were not present in the test configuration. Therefore, these tasks never cycled.
2. During all test runs, the utilization distributions for these tasks remained constant, regardless of aircraft load applied to the sensor.
3. Utilization for these tasks approached 100 percent during the scan interval.

2



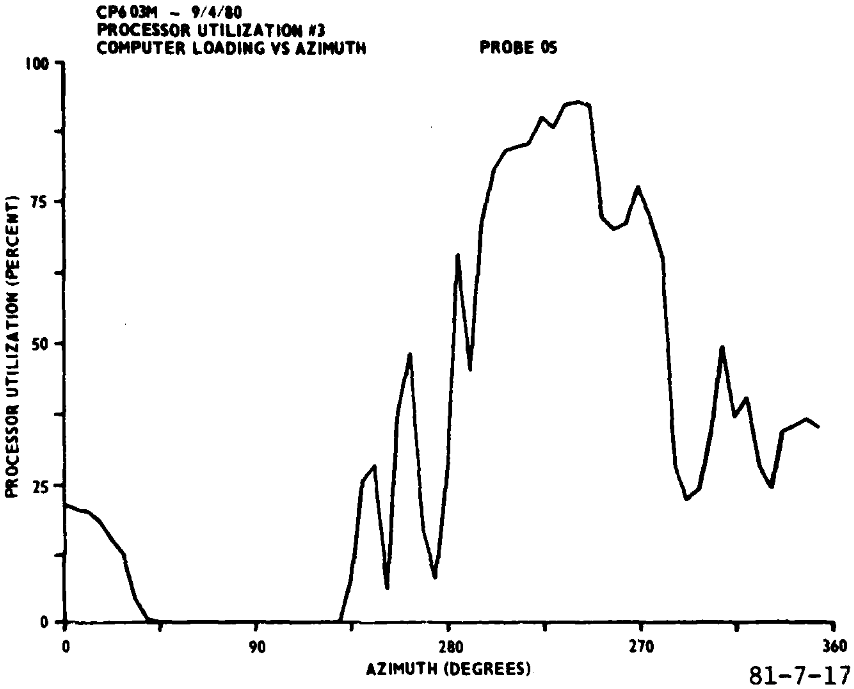
81-7-15

FIGURE 15. TASK UTILIZATION — ST: ATCRBS TRACK UPDATE (BIN PROCESSING)



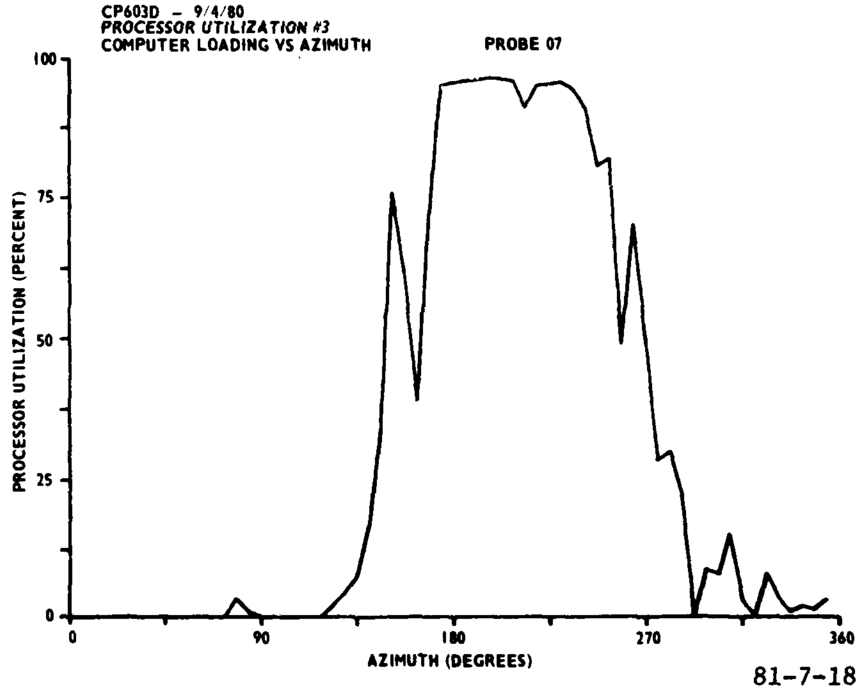
81-7-16

FIGURE 16. TASK UTILIZATION — ST: ATCRBS TARGET TO TRACK ASSOCIATION



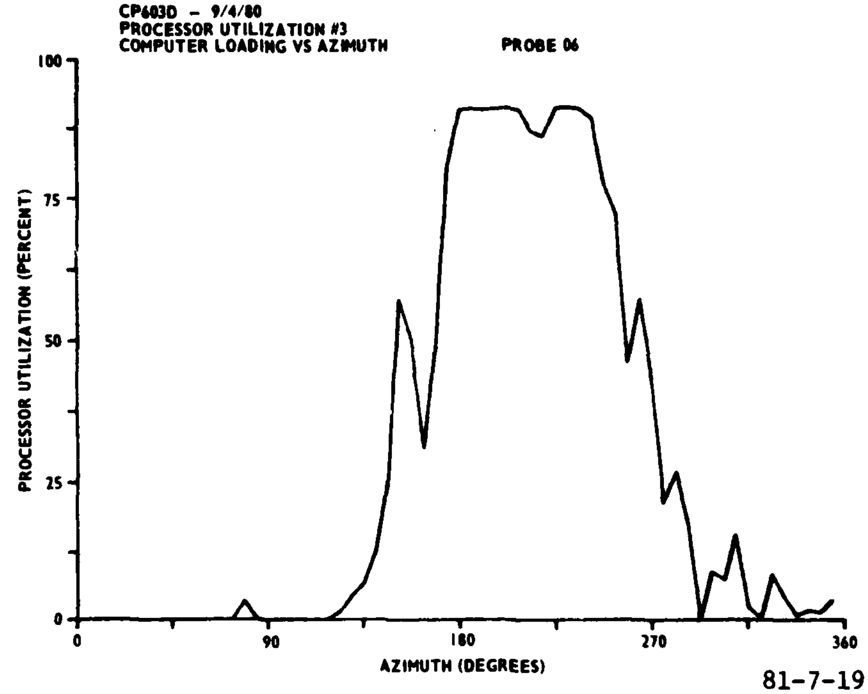
81-7-17

FIGURE 17. TASK UTILIZATION — S: ATCRBS TARGET TO TRACK CORRELATION



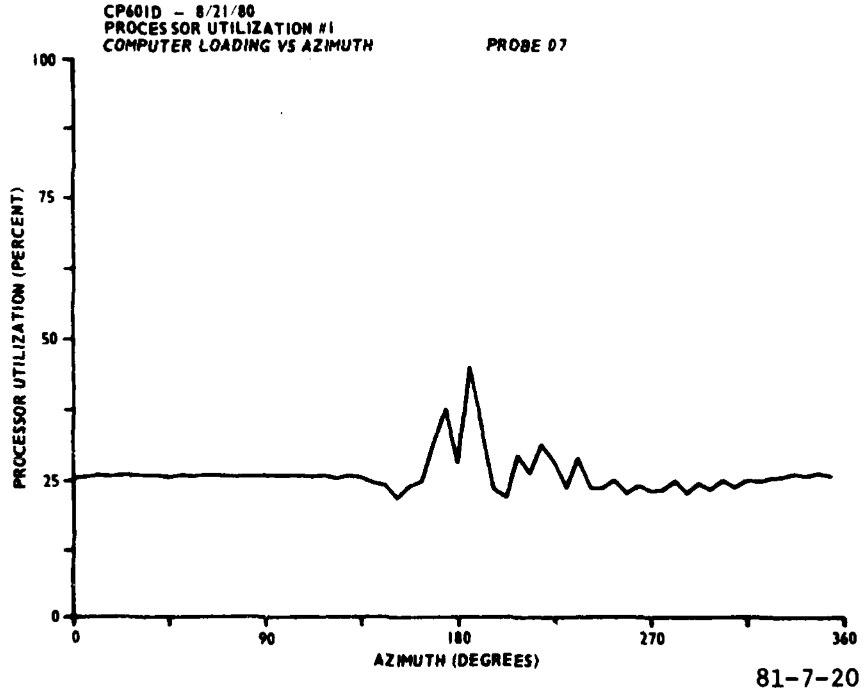
81-7-18

FIGURE 18. TASK UTILIZATION — DL: OUTPUT PROCESSING



81-7-19

FIGURE 19. TASK UTILIZATION — DL: OUTPUT POST PROCESSING



81-7-20

FIGURE 20. TASK UTILIZATION — CM: BEACON FORMATTER

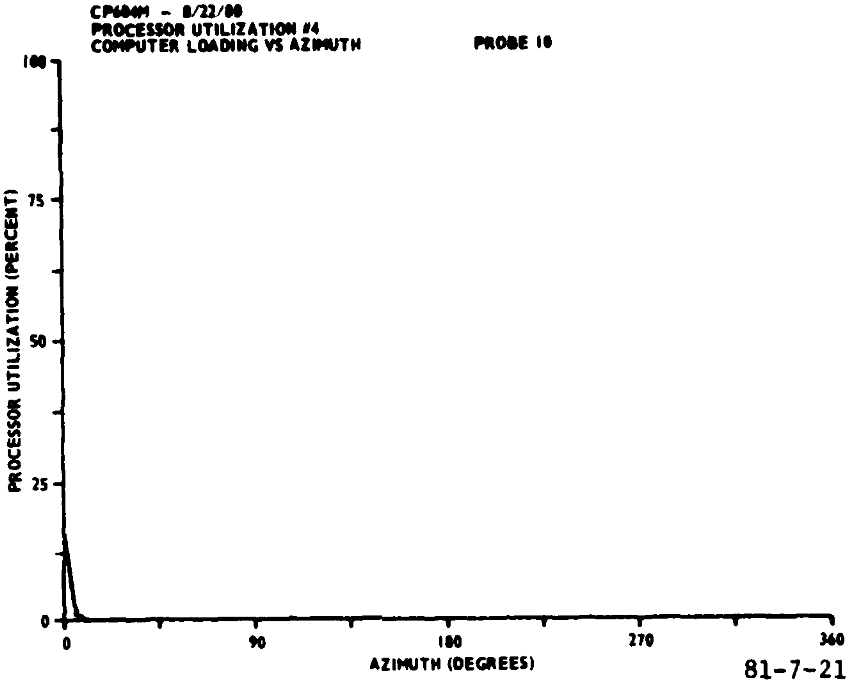
## TABLE 9. PROCESSOR UTILIZATION DATA

| | | | Average Utilization Per Scan (Percent) | | | |
|---|---|---|---|---|---|---|
| Load Module | Processor | No Load | 150 ATCRBS 150 DABS | 190 ATCRBS 190 DABS | 0 ATCRBS 300 DABS | 0 ATCRBS 380 DABS |
| SS001X | 1-1 | 84.45 | 71.08 | 68.10 | 68.95 | 62.72 |
| SS002X | 2-1 | 3.40 | 7.86 | 10.13 | 8.63 | 17.14 |
| SS003X | 4-2 | 6.92 | 19.01 | 19.45 | 30.95 | 32.14 |
| SS004X | 4-3 | 22.42 | 65.46 | 72.32 | 37.91 | 42.19 |
| SS005X | 1-3 | 18.04 | 26.32 | 33.49 | 29.63 | 18.09 |
| SS006X | 3-3 | 0.00 | 7.35 | 11.49 | 10.09 | 27.85 |
| SS007X | 8-2 | 0.07 | 0.07 | 0.07 | 0.07 | 0.07 |
| SS008X | 4-1 | 4.95 | 31.71 | 38.01 | 5.05 | 4.96 |
| SS009X | 6-2 | 0.12 | 2.07 | 2.23 | 2.38 | 4.60 |
| SS00AX | 4-4 | 3.55 | 3.24 | 3.25 | 3.24 | 3.24 |
| SS00BX | 7-2 | 70.01 | 76.53 | 78.61 | 74.39 | 77.68 |
| SS00EX | 6-4 | 0.04 | 0.04 | 0.04 | 0.04 | 0.04 |
| SS0011X | 5-4 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| SS0012X | 3-1 | 25.97 | 25.43 | 25.17 | 25.39 | 26.48 |
| SS0013X | 6-1 | 0.00 | 15.69 | 22.91 | 14.52 | 32.59 |
| SS0014X | 1-2 | 1.07 | 8.74 | 12.36 | 10.65 | 23.99 |
| SS0015X | 3-4 | 0.00 | 0.59 | 0.09 | 2.37 | 0.22 |
| SS0016X | 2-2 | 3.57 | 22.08 | 22.03 | 3.98 | 3.72 |
| SS0017X | 2-3 | 0.00 | 0.59 | 0.09 | 2.37 | 0.22 |
| SA01DX | 8-1 | 5.55 | 12.06 | 13.79 | 5.61 | 5.53 |
| SS01EX | 7-3 | 1.07 | 1.13 | 1.15 | 1.13 | 1.13 |
| SS031X | 3-2 | 2.14 | 22.53 | 34.40 | 1.40 | 0.00 |

NOTE:

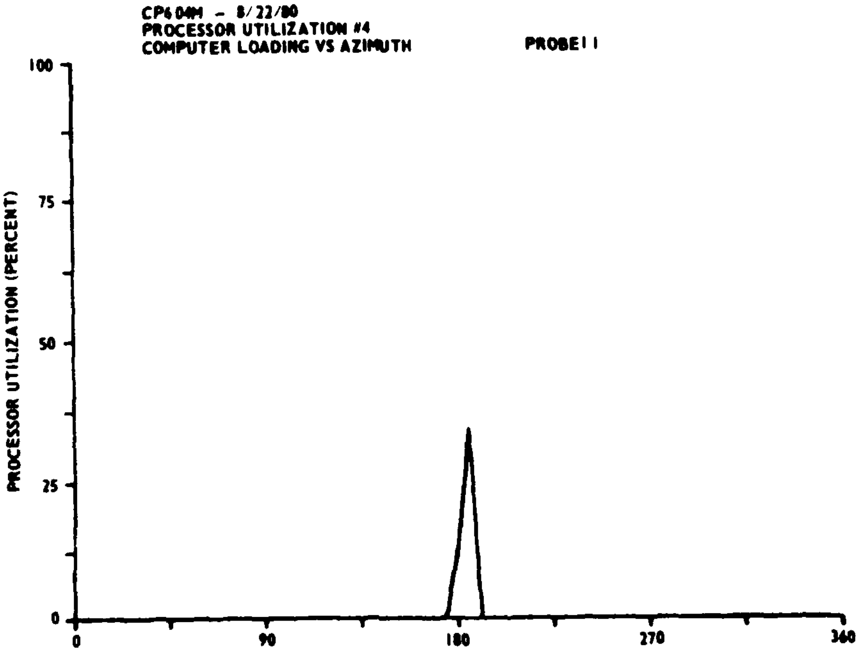
1. These processors contain other functions for which measurements were not made.

2. See test for comments relating to utilization data.



81-7-21

FIGURE 21. TASK UTILIZATION — PM: SERVICE I&P CHANNELS



87-1-22

FIGURE 22. TASK UTILIZATION — PM: SENSOR HARDWARE INPUTS

No measurements were taken for seven of the active processors. Four of these contained IPC functions. As these are being replaced by the new ATARS software, they were not measured. Two others contained the communication software. These were eliminated from this effort due to a lack of adequate program listings for these functions on the release 6.4 software system, and because a multisite configuration would be needed to adequately measure these functions. The remaining processor contained the network management software to process incoming network messages. These messages are only available in a multisite environment.

## SUMMARY OF RESULTS

The following performance characteristics of the DABS computer subsystem were determined during this test activity:

1. Average delay for a processor to access the ensemble data bus measured with HSMAT software was relatively constant for bus loads of less than 30 percent. Values measured ranged from 510 to 700 nanoseconds. When ensemble data bus loads of over 30 percent were applied with the HSMAT program, the average delay per access increased rapidly to over 3.5 microseconds for bus loads of 60 percent.

2. Ensemble data bus loads measured with the release 6.4 DABS software were mainly in the range of 10 to 15 percent. Average access delays measured were between 225 and 500 nanoseconds. These values are substantially less than those measured with the HSMAT program for comparable bus loadings. Only one ensemble data bus had a bus load exceeding 30 percent. This ensemble contained the two busiest processors in terms of accesses generated to global memory: the beacon formatting task of channel management and the ATCRBS target-to-track correlation function of sensor tracking.

3. Average delay for an ensemble data bus to access the global data bus through its coupler, as measured with the HSMAT software, increased slightly with load for global data bus loads of up to 88 percent busy. Values measured ranged from 450 to 850 nanoseconds. For global data bus loads in excess of 88 percent busy, the average delay per access increased rapidly.

4. Global A data bus loads measured with the release 6.4 DABS software were in the range of 35 to 45 percent busy; global B data bus loads were in the range of 10 to 17 percent busy. Average access delays to the global data buses were relatively the same with both the HSMAT software and the DABS software.

5. Eleven processors used less than 16,384 words of the available global memory address space. Thirteen others used less than 20,480 words of the available address space. Only five processors used over 20,480 words of the available global memory address space. All of these contained large gaps of unused addresses.

6. Processor utilization for the following three ATCRBS software tasks approached 100 percent with the 190 ATCRBS, 190 DABS target load:

| | |
|---|---|
| SS016X ST: | ATCRBS Track Update (Bin Processing) |
| SS008X ST: | ATCRBS Target to Track Association |
| SS031X S: | ATCRBS Target to Track Correlation |

7. Processor utilization for the following two software tasks approached 100 percent with a 380 DABS target load:

| | |
|---|---|
| SS015X DL: | Output Processing |
| SS006X DL: | Output Post Processing |

8. Two processors had particularly low utilizations. Load module SS01EX (performance monitor) was only utilized for brief periods of time at north and south. Load module SS012X (beacon formatting task of channel management) had an average utilization of approximately 25 percent, with no peaks greater than 50 percent.

## CONCLUSIONS

The following conclusions have been made with respect to the performance of the Discrete Address Beacon System (DABS) computer subsystem of the DABS engineering model sensors:

1. No data bus contention problem exists for ensemble data bus loads of less than 30 percent and global data bus loads of less than 88 percent.

2. Although no contention problem was found with ensemble 3, an improvement in system performance could possibly be experienced by separating the tasks of ensemble 1 processors 1 and 3 (beacon formatting task of channel management and Air Traffic Control Radar Beacon System (ATCRBS) target-to-track correlation task of sensor tracking).

3. The processor local memories can be expanded from their current 8,192 words to 12,288 words with little or no impact on system operation. For the most part, only a change in the values currently loaded into the bias registers is required. An expansion of the processor local memories to 16,384 words appears feasible. More extensive software changes would be required than for a 12,288 word local memory.

4. The distributive architecture employed in the computer subsystem of the DABS engineering model is a viable means of implementing the types of processing performed in DABS.

5. Processor utilization is not considered a problem with the release 6.4 DABS software except for the three ATCRBS software tasks. It is believed that overloading of these tasks is the primary cause of the inability to conduct tests with the all ATCRBS Aircraft Reply and Interference Environment Simulator (ARIES) scenario.

6. The heavy utilization of the two data link tasks may cause a potential problem. Problems would only be manifested with large numbers of data link COMM B messages within a narrow azimuth window.

7. Several of the processors are relatively lightly utilized.

## RECOMMENDATIONS

The following recommendations are made:

1. Further study should be made into the effects of expanding local memory size to 16,384 words. This study should include the effect of reorganizing the data base tables located in global memory.

2. The processor utilization tests of this activity should be repeated for later releases of the Discrete Address Beacon System (DABS) software. In particular, the following areas should be emphasized:

a. Performance with the all Air Traffic Control Radar Beacon System (ATCRBS) scenario.

b. Variations in data link loading.

3. A similar analysis should be performed for those functions which were not measured during these tests and new functions which have been added. Examples of these are: multisite, intersite communication, and radar tracking.

# REFERENCES

1. Holtz, M., et al., Discrete Address Beacon System (DABS) Baseline Test and Evaluation, Interim Report, FAA-RD-80-36, April 1980.

2. Spencer, D. A. and Goon, M., The Aircraft Reply and Interference Environment Simulator (ARIES), FAA-RD-78-96, Lincoln Laboratory, ATC-87, 22 March 1979.

# APPENDIX A

# COMPUTER PERFORMANCE MEASUREMENT SYSTEM

The computer performance measurement system was designed and fabricated by Federal Aviation Administration (FAA) Technical Center personnel to collect performance data on the Discrete Address Beacon System (DABS) computer subsystem. The system consists of a Digital Equipment Corporation (DEC) model PDP 11/45 computer, a data collection unit, and probes for connection to the DABS sensor. A block diagram of the computer performance measurement system is shown in figure A-1.

## PROBES.

Three types of probes are used as buffer amplifiers/line drivers between the DABS sensor and the data collection unit: (1) signal probes, (2) memory address probes, and (3) azimuth data probes.

SIGNAL PROBES. The signal probes are used to detect the logic level which exists at the specific signal point to which the probe is attached. Each signal probe unit contains the logic necessary to monitor the logic levels for six independent signal points. For ease of testing, a signal probe unit was installed on the motherboard of each ensemble data bus to be measured. In addition, two signal probe units were installed on the motherboard of the global data buses. The signal probe units were wired to connectors mounted on the DABS front panel. Any six signal probe units can be connected to the data collection unit simultaneously.

MEMORY ADDRESS PROBES. The memory address probes are used to detect memory address data from the processors of the DABS computer subsystem and remote these data to the data collection unit. The memory address probes were designed to interface to the breakpoint panel interface connector on the DABS front panel. Each memory address probe unit contains the logic necessary to monitor the 15-bit memory address bus and signals indicating that valid address data are available on the bus. Twelve memory address probe units are available for connection to any combination of the 36 processors in the DABS computer subsystem.

AZIMUTH PROBES. The azimuth probes are mounted in the Aircraft Reply and Interface Environment Simulator (ARIES) and are used to amplify the azimuth change pulse (ACP) and azimuth reference pulse (ARP) signals for distribution to the data collection unit.

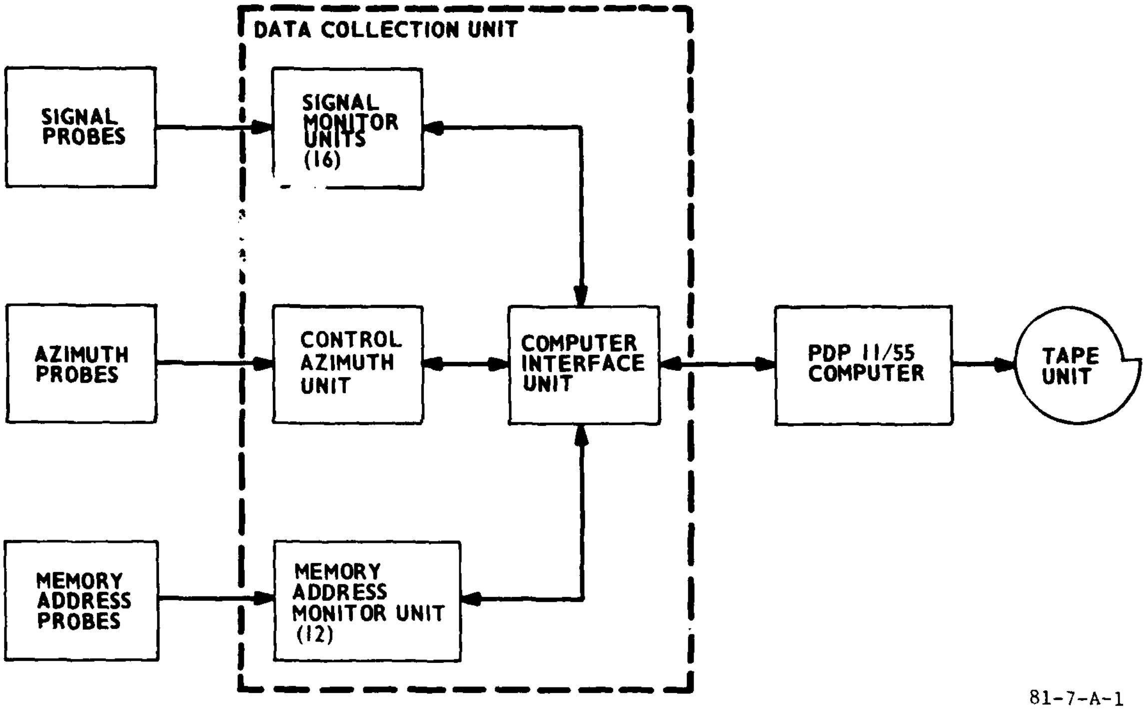
## DATA COLLECTION UNIT.

The data collection unit contains all of the logic necessary to monitor and process the data forwarded to it by the various probe units. The data collection unit contains 16 signal monitor units, 12 memory address monitor units, a control/azimuth unit, and a computer interface unit.

## SIGNAL MONITOR UNIT.

The signals to be monitored by each signal monitor unit are determined by a patchboard mounted in the data collection unit. The 36 signals from the six connected signal probe units provide the inputs to the patchboard. The outputs from the patchboard are wired to the inputs of the 16 available signal monitor units. Separate patchboards were wired for each combination of signals desired during testing.

Each signal monitor unit provides three separate measurements on the input logic signal: (1) a measure of the cumulative active time over the measurement interval, (2) a measure of the duration of the first active period during the measurement interval, and (3) a count of the number of inactive to active transitions which occur during the measurement interval. Time measurements are made with a resolution of 25 nanoseconds.



81-7-A-1

FIGURE A-1. COMPUTER PERFORMANCE MEASUREMENT SYSTEM FUNCTIONAL BLOCK DIAGRAM

## MEMORY ADDRESS MONITOR UNITS.

Each memory address monitor unit processes data on the activity of a processor. Four measurements are made of the activity within the processor being monitored: (1) the value of the first address detected on the memory address bus during the measurement interval; (2) a measure of the cumulative time spent executing code between two specified memory addresses during the measurement interval, (3) a measure of the time spent executing code between the two specified memory addresses for the first occurrence of the address pair during the measurement interval, and (4) a count of the number of times the specified address pair is executed during the measurement interval. Time measurements of the memory address monitor unit are made with a resolution of 1 microsecond.

## CONTROL/AZIMUTH UNIT.

The control/azimuth unit controls the overall operation of the data collection unit, and monitors the ACP and ARP signals to determine the exact azimuth of the ARIES/DABS sensor. Based on control information loaded at startup time, either time or azimuth may be used to control the data collection process and determine the length of the measurement interval.

When the control/azimuth unit detects the end of a measurement interval it causes the current contents of all signal monitor units and memory address monitor units to be latched into output registers. The monitor units are then reset and a new measurement interval started.

## COMPUTER INTERFACE UNIT.

The computer interface unit interfaces the data collection unit to the DEC model PDP 11/55 computer. During system initialization this interface is used to transfer control information from the computer to the data collection unit. This control information specifies which of the 28 monitor units are to collect data during this run, whether the measurement interval is to be specified in time or azimuth, and the length of the interval in either time or ACP's.

During data collection, the computer interface unit is notified by the control/azimuth unit each time data are latched into the output registers of the monitor units. These data are then sequentially transferred to the computer under control of the computer interface unit.

## PDP 11/55 COMPUTER.

The PDP 11/55 is used to control the operation of the computer performance measurement system, and to record the data received from the data collection unit onto magnetic tape. At the start of a data collection run, control information is read from a previously stored disk file and transferred to the data collection unit. This same information is recorded in the first record of the data collection tape. Following initialization, all data transferred from the data collection unit to the computer are recorded onto the data collection tape.

## DATA REDUCTION.

All data collected are reduced using software on a DEC model PDP 11/45 computer. Data reduction software currently available includes printout of system configuration data, data summarization, signal time and count frequency distributions, address frequency distributions, and processor loading distributions.

# APPENDIX B

## STATISTICAL VALIDATION OF HARDWARE DISTRIBUTIONS

### INTRODUCTION.

During the development of the data collection techniques for the computer performance tests, a major concern was the validity of the data distributions obtained with the computer performance measurement system. As the data to generate time and address distributions were obtained using sampling techniques, the question arose as to whether the distributions would be affected by sampling rates or sampling technique. If so, a sampling bias could be assumed to exist and the distributions would not necessarily reflect the true distributions of the system.

Therefore, a statistical test of the data collected by the computer performance measurement system was designed to test the statistical validity of the data. The test was designed around a scheme in which the identical program was repeatedly executed in the Discrete Address Beacon System (DABS) computer subsystem. The computer performance measurement system was used to collect data under various sampling conditions. The data were then compared, as described below, to test the statistical equivalence of the distributions obtained from each test run.

### STATISTICAL TESTS.

In order to validate the distributions obtained, classic goodness-of-fit tests were applied to the data. These tests produced a test statistic which was evaluated by comparison to the distribution of that statistic. Statistical inference was then drawn from the values derived. Three different goodness-of-fit tests were used: The Kolmogorov-Smirnov (K-S) test, the chi-square test, and Patnaik's power test.

**KOLMOGOROV-SMIRNOV TEST.** The K-S test is based on examining the largest deviation between two empirical cumulative distribution functions at given points along the cumulative distribution function. The K-S test deals with a theoretical distribution based on the largest deviation among the data points. The distribution is characterized by the number of data points. The cutoff points of the theoretical distribution used in evaluating two empirical distributions results from an approximation based on the number of points observed.

The data collected were divided into 64 class intervals, and the points selected for comparison designated as the end points of each class interval. The cutoff points were established at significance levels of 0.1, 0.05, and 0.01 where significance level is defined as the uncertainty of declaring significant differences.

**CHI-SQUARE TEST.** The chi-square test is based on examining the difference between two distributions at each class interval. The differences between the distributions are used to derive a chi-square statistic.

Normally, the chi-square test is used to test an experimentally obtained distribution against a standard or control distribution. For these tests, the control distribution was defined as the distribution which was considered more stable. For example, [illegible] comparing distributions obtained at various sampling rates, the data collected at the highest sampling rate were considered more stable and were used as the control distribution. When the distributions were considered equally stable (e.g., comparisons of data collected under identical *sampling conditions*) the designation of the control distribution was made arbitrarily. Once a distribution had been defined as the control distribution, all other distributions of interest were tested against this same control distribution.

The number of class intervals were not necessarily the same from test to test. A rule generally followed, and used in these tests, was to combine adjacent class intervals in both distributions when the frequency or expected value in the control distribution was less than three. This rule is applied due to the fact that one of the basic assumptions of the chi-square distribution (that the distribution of intervals follows a multinominal distribution) is violated for such cases. With low or zero frequencies, the multinominal representation is unsatisfactory. As a result, the number of class intervals and, hence, the theoretical chi-square distribution changed from test to test.

PATNAIK'S POWER TEST. Patnaik's power test determines the "power" of the chi-square test, where power means the ability to detect inherent differences when such differences actually exist. A tradeoff exists between significance level and power. Should a larger uncertainty in declaring a significant difference be allowed, the ability to correctly detect a significant difference increases. The reverse is also true.

Patnaik's test tells us the power at specified significance levels for a given chi-square test. This is accomplished by utilizing the chi-square value as one of several "noncentrality" parameters and calculating a "noncentral" chi-square distribution via some approximations to a central chi-square distribution.

## DATA ANALYSIS.

SAMPLE SIZE VALIDATION. Initially, data were analyzed to determine an appropriate sample size to be used in testing data validity. For these tests, a data tape was used which had been collected by the computer performance measurement system at a sample rate of 100 samples per second while the HSMAT software was executing in the DABS computer subsystem. Data from this tape were reduced for three different signal monitors and five different sample sizes: 0.5, 5, 20, 50, and 100 scans. During each scan approximately 477 observations (samples) had been recorded. This provided sample sizes ranging for 238 to 47,700.

The five distributions for each signal monitor were compared against each other for significant differences. Any differences detected could be attributed to sampling, reflecting the fact that the sample size was too small to form a representative distribution.

Comparison of the distributions obtained for 0.5 scans with those for 5 scans resulted in K-S values far below the cutoff point for a 0.1 significance level. The chi-square test indicated some variation, but the values were still below the 0.1 significance level cutoff point. Patnaik's power test, however, indicated that a significance level of 0.1 was not high enough to obtain a plausible figure for power.

Comparisons of the distributions for 5 scans with those for 20 scans gave significantly better results. Variation between these two distributions was negligible, and it was concluded that five scans provided a satisfactory sample size for testing the distributions obtained for the different sample rates.

SAMPLE RATE VALIDATION. For these tests data tapes were used which had been collected by the computer performance measurement system at various data rates while the HSMAT software was executing. The data rates tested are listed in table B-1. For each of the five tests, distributions for seven signal monitor's were compared.

Without exception, the 35 sampling mode comparisons showed minute variations. This implies that no significant difference exists between the distributions, and:

a. Fifty samples per second is a satisfactory sample rate.

b. Thirty-two ACP's per sample (approximately 107 samples per second) is asynchronous to events in the DABS and is also satisfactory as a sample rate.

Ergo, the sampling rates used to collect data with the computer performance measurement system did not affect the distributional forms generated by the DABS hardware data. All sample rates, from 50 to 300 samples per second and both sample modes (time and azimuth) are satisfactory for further analysis of signal characteristics.

Based on the findings of this test, it was recommended that all further testing with the HSMAT software be conducted at sample rates of 100 samples per second in time mode. It was further recommended that testing with the DABS software be conducted in azimuth mode at rates of either 32 ACP's per sample or 64 ACP's per sample.

TABLE B-1. SAMPLE RATES VALIDITY TEST

| Test | Control Distribution | Comparison Distribution |
|---|---|---|
| 1 | 300 Samples/Second | 300 Samples/Second |
| 2 | 300 Samples/Second | 100 Samples/Second |
| 3 | 300 Samples/Second | 50 Samples/Second |
| 4 | 32 ACP's/Sample | 32 ACP's/Sample |
| 5 | 32 ACP's/Sample | 100 Samples/Second |

FILMED

8