ESD-TR-75-78

MTR-3015

AN OUTLINE OF THE APPLICATIONS SOFTWARE FOR THE RIDS EXPERIMENTAL TDM BUS

SEPTEMBER 1975

Prepared for

DEPUTY FOR DEVELOPMENT PLANS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
Hanscom Air Force Base, Bedford, Massachusetts

Project No. 6370
Prepared by
THE MITRE CORPORATION
Bedford, Massachusetts
Contract No. F19628-75-C-0001

Approved for public release; distribution unlimited.

REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.

EMERY F. BOOSE
Project Leader
MITRE D-92

KEITH HANDSAKER
Project Engineer

FOR THE COMMANDER

CARL A. SEGERSTROM
Acting Director, Technology
Deputy for Development Plans

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: ESD-TR-75-78
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): AN OUTLINE OF THE APPLICATIONS SOFTWARE FOR THE RIDS EXPERIMENTAL TDM BUS
5. TYPE OF REPORT & PERIOD COVERED
6. PERFORMING ORG. REPORT NUMBER: MTR-3015
7. AUTHOR(s): P.R. Cloud
8. CONTRACT OR GRANT NUMBER(s): F19628-75-C-0001
9. PERFORMING ORGANIZATION NAME AND ADDRESS: The MITRE Corporation, Box 208, Bedford, MA 01730
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Project No. 6370
11. CONTROLLING OFFICE NAME AND ADDRESS: Deputy for Development Plans, Electronic Systems Division, AFSC, Hanscom Air Force Base, Bedford, MA 01731
12. REPORT DATE: SEPTEMBER 1975
13. NUMBER OF PAGES: 33
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
15. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report): Approved for public release; distribution unlimited.
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
18. SUPPLEMENTARY NOTES
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

MIL-STD-1553

SOFTWARE

TDM BUS

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

MITRE, in support of the Electronic Systems Division, is presently working on the Radio Information Distribution System (RIDS). Under this project (6370), a rudimentary experimental TDM bus has been built that is compatible with the constraints of MIL-STD-1553 (USAF). This report outlines the applications software generated in support of a demonstration of some of the bus capabilities. These included the servicing of an AN/ARC-50 UHF radio and an array of cockpit indicators, initiation

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) of visual and audible warnings based on the values of the parameters being monitored and displayed, and the generation of fault diagnostic displays when an operator makes an illegal switch setting.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I

INTRODUCTION

MITRE, in support of the Electronic Systems Division, is presently working on the Radio Information Distribution System (RIDS). This project (6370) is concerned with the use of multiplex busing techniques in aircraft to simplify inter-subsystem communication and to exploit the synergistic potential of the suite of radio information equipments.

One part of the task has been to assemble a rudimentary data distribution network which conforms to the constraints of the military standard, Aircraft Internal Time Division Multiplex Data Bus, MIL-STD-1553 (USAF). A skeletal version of a "standard" multiplex bus, see Figure 1, consisting of a Controller--a PDP-9--, a Bus Control Interface Unit, and two remote terminals interconnected by a transmission line--a shielded twisted pair--has been in operation for some months.

The availability of a working system has provided the opportunity to investigate further the flexibility of an avionics multiplex bus to provide a range of useful functions on-board an aircraft. To demonstrate some of these capabilities, a number of subsystems, e.g. an AN/ARC-50 radio, cockpit indicators, etc., have been integrated into the experimental system to serve as "bus users". The present report is an outline of the software developed to implement, and support, these activities.

The earlier sections briefly describe the organization of the basic message handling software. This was developed earlier in the RIDS program and has been documented previously (1); it is summarized herein to provide background for the reader who does not have ready access to that report.

The later sections describe the bus capabilities being demonstrated, and the applications software developed for that purpose.

The last section contains a brief summary and some software oriented observations.

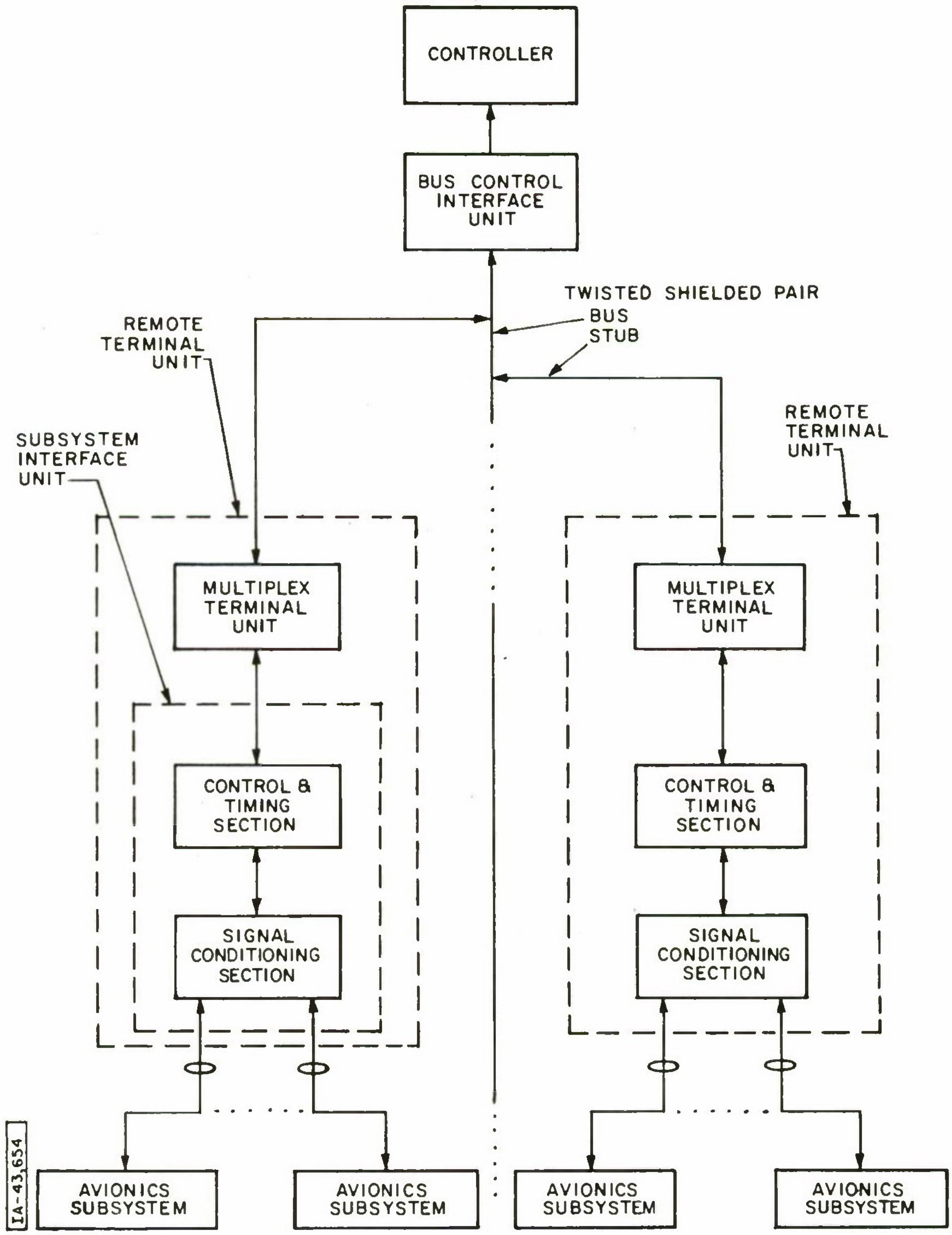


Figure 1 A "STANDARD" MULTIPLEX BUS
MIL-STD.-1553 (USAF)

SECTION II

SOFTWARE FOR THE LABORATORY DEMONSTRATION OF THE EXPERIMENTAL BUS

When designing a software package that is intended to perform a number of different functions, it is both desirable and expedient to partition the programs so that their interdependence is minimal. For this reason, the message handling software developed for the control of the multiplex bus was structured to permit the transfer of information between the subsystems serviced by the bus essentially independently of the nature of those subsystems and the specific content of the data being transferred. The interface between the message control programs and the application dependent processing was confined to a single entry/exit routine, which permitted processing of the data word content to any degree of complexity--as required by the application--without necessitating modification of the basic control programs. As a consequence of this, the additional programming required to service the AN/ARC-50 UHF radio and the miscellaneous cockpit displays, e.g., instrument gauges, warning displays, etc., was primarily in the area of applications processing.

The following sections outline the programs that were generated to interpret, and display, the content of the data words being transferred between the subsystems, via the bus. In addition, to permit the reader to appreciate the relationship between the application and control software, a brief description is given of the basic message control cycle.

Outline of Message Control Software

The software package developed to control the experimental bus served two distinct purposes. One of these was the message handling function itself; the other consisted of a number of programs which permitted the majority of the message handling routines to be debugged without recourse to the hardware it was intended to control. To the level of complexity implemented in the experimental bus work, this test/simulation software has been extremely useful. It enabled the basic message cycle to be completely debugged prior to interfacing the control programs with the bus hardware, and subsequently allowed the bus controller to act as a sophisticated signal generator for the debugging of the bus subsystems in the later stages of system integration. The main functions implemented in the message handling software are listed in Tables 1a and 1b.

For convenience of conceptual design, together with the desirability of developing a modular computer program, the message

TABLE 1a

CONTROL SOFTWARE FUNCTIONS: OPERATIONAL

- Initiates and controls three modes of message transfer

  a) Controller to Remote Terminal
  b) Remote Terminal to Controller
  c) Remote Terminal to Remote Terminal

- Samples subsystems serviced by the remote terminals at one of six rates: 32, 16, 8, 4, 2, 1 per unit time period.

- Sequences messages to obtain a quasi-uniform loading of the multiplex bus throughout each unit time period.

- Checks validity of status word associated with each response component, and prints out address of subsystem location if an error is indicated.

- Distributes data words contained in incoming response components to core locations accessible to applications software.

TABLE 1b

CONTROL SOFTWARE FUNCTIONS: TEST AND DEVELOPMENT

In the course of developing the MUX bus control software, the following options within the basic message processing cycle were included primarily for the purpose of debugging and testing.

- Option 1: Include/Exclude all Command component transfers.
- Option 2: Include/Exclude Command component transfers to Bus Controller Interface Unit.
- Option 3: Include/Exclude Command component transfers to internal print buffer.
- Option 4: Option 3 can be Changed/Not Changed without recycling program.
- Option 5: Include/Exclude all Response component transfers.
- Option 6: Response component from BCIU/Response component from response simulation in internal buffer.
- Option 7: Minor cycle initiation under clock control/No clock control.
- Option 8: Include/Exclude "no response" processing.

control function--a subset of the overall bus control function--was subdivided into a number of discrete operations:

- Message discrimination

- Message synchronization

- Command/Response component transfer to/from the control unit to the bus

- Message validation

- Common response processing

These activities must be repeated for each Command/Response message sequence; the latter consisting of a command component generated by the bus controller, and its associated response component originating from an MTU(s). Thus, if the bus load consists of n Command/Response sequences per second, the basic set of message handling operations must also be executed n times per second.

The message handling cycle is shown diagrammatically in Figure 2. Two additional operations are included which do not constitute part of the "per message" activities listed above. One of these--sequence initiation--is required for entry into the repetitive message handling loop when the bus system is switched on. The other--applications, or special response responsing--consists of the routines dependent on the content of the data words, and comprises the bulk of the software generated for the present demonstration.

The figures given in parentheses--under the blocks in Figure 2--refer to the number of memory cycles and machine language instructions, respectively, used to program the routines on a PDP-9 computer (memory cycle time one microsecond). The reader is referred to Reference 1 for a discussion of the significance of these numbers.

The routine that is of more immediate interest is the Common Response processing, since it contains the interface between the message handling cycle and the applications processing used to interpret the content of the Data Words. The position of the Data Words within a Command/Response message sequence is established by the bus protocol defined in MIL-STD-1553 (USAF). In the particular case of a remote terminal to controller message sequence, a single Command word is transferred to a given MTU--determined by the appropriate address field. The MTU responds with a Status word and a number of data words--determined by the content of the word count field in the Command word. In the bus controller the incoming

IA-43,657

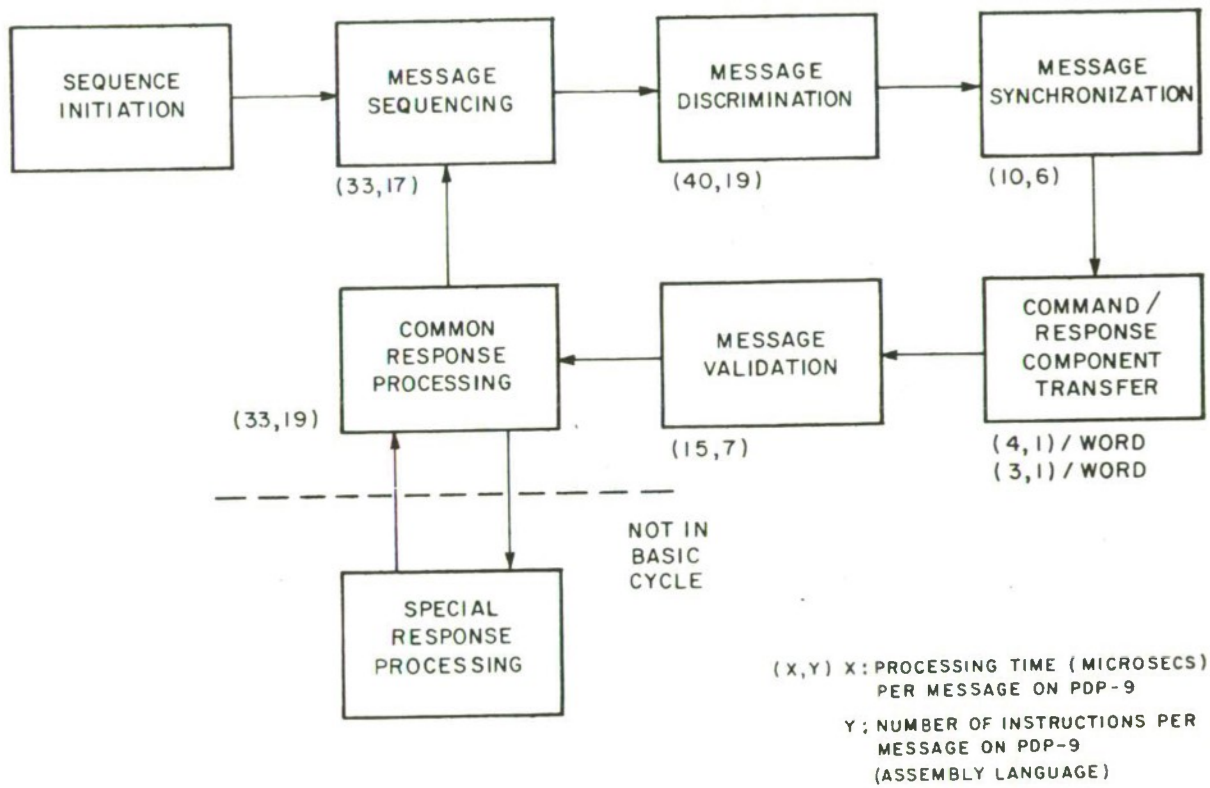


Figure 2 EXPERIMENTAL BUS CONTROL SOFTWARE : BASIC CYCLE

response component is placed in a data buffer, and each word is then processed according to a predetermined sequence.

The first word--the Status word--is checked for validity, and if determined to be in error, a message is printed on the TTY, associated with the controller, giving the terminal and subaddress from which the response component was received. The validation criteria are common to all Status words, and the latter are handled similarly for all messages.

The remaining words in the response component are Data words which, in general, require unique handling dependent on the subsystems at which they originated. The calling of, and return from, the appropriate application routine(s) required for each data word constitutes the processing interface between the conceptually distinct functions of message transfer and message usage. The flexibility of processing necessary to cope with the different types of Data word content, e.g., navigational data, flight control information, etc., is obtained by means of a system of pointers which is best described diagrammatically, see Figure 3. Each command/response message sequence, that has data words in the response component, has a processing word stored contiguously with the message's command word in core. The processing word points to a table of pointers--with an entry for each data word in the response component. Each of the pointers initiates the applications processing required by its associated data word.

As has been mentioned previously, the basic message control software, and in particular, the Common Response processing, was structured to permit message handling independent of the types of equipment being serviced by the bus. Thus, the task of generating additional software resulting from the selection of particular equipments, e.g., AN/ARC-50, gyro heading repeater, etc., for demonstration, resolved itself into one of developing applications software--as distinct from bus control programs. The following sections outline the multiplex bus capabilities that have been demonstrated, the message flow that was required to implement the demonstration, and the applications software that was developed to utilize the Data words transferred.

Multiplex Bus Capabilities Being Demonstrated

The demonstration--by experiment--of the capabilities of the multiplex bus falls into two fairly distinct phases. The first of these shows how elementary operations, such as data transfer, function control, etc., can be implemented on the bus. The method of displaying that these activities are indeed taking place, is by panel lights that monitor the contents of the two port memory associated with the subsystem interface unit. The second part of

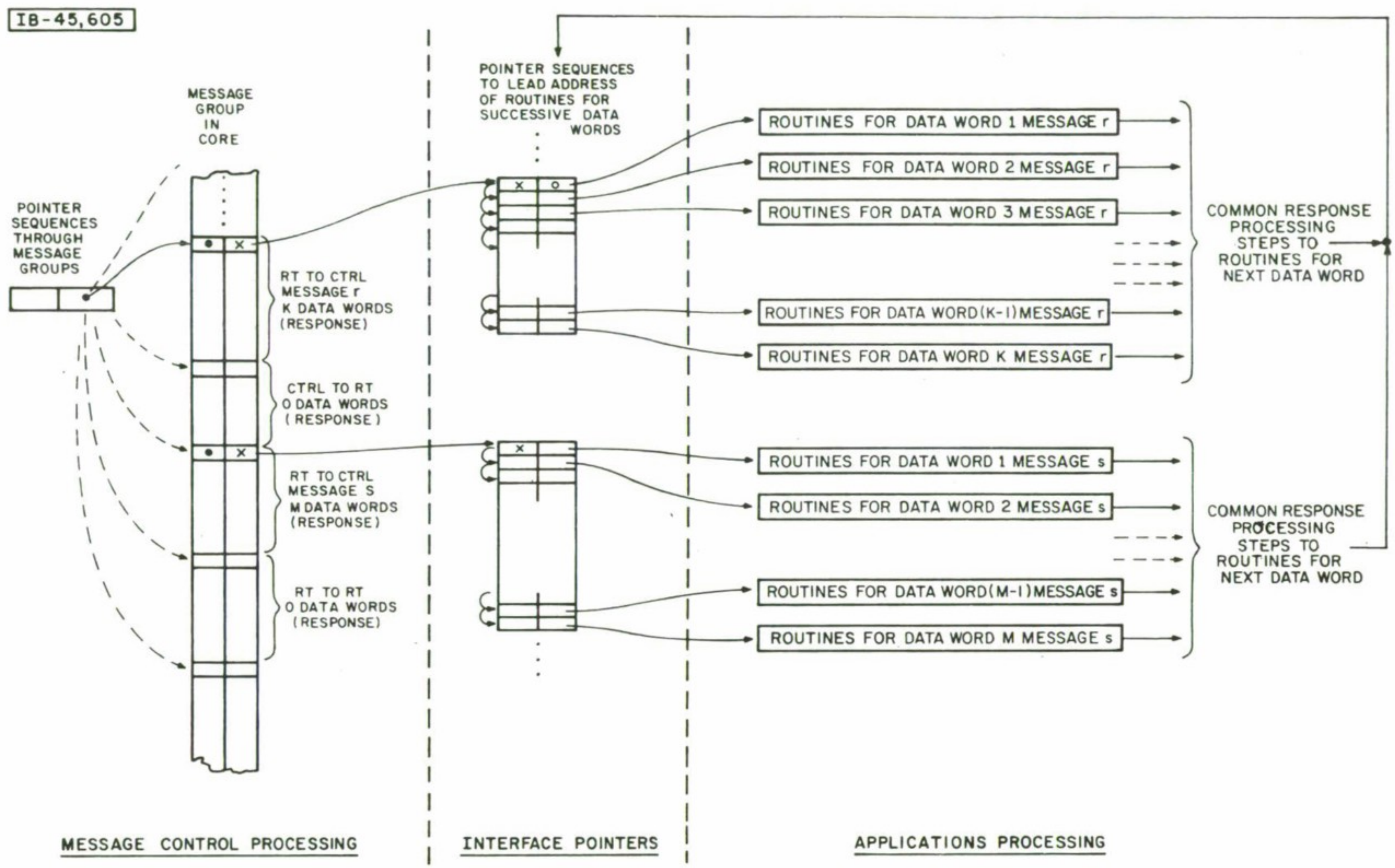


Figure 3 INTERFACE BETWEEN MESSAGE CONTROL AND APPLICATIONS PROCESSING

the demonstration is more operationally oriented. It includes the remote control of an AN/ARC-50 UHF radio, remote display of various aircraft parameters on standard cockpit indicators, threshold monitoring with diagnostic/warning display of critical conditions, etc.

A brief description of the capabilities being demonstrated is given below.

Data Transfer

The purpose is to show the transfer of a Data word between SSIU locations, via the bus controller.

The controller interrogates, at a predetermined rate, a given location in an SSIU's memory--defined by an MTU address, subaddress and word count--and transfers the content to some other location, similarly defined. The transfer is via the controller, and uses a terminal/controller message sequence followed by a controller/terminal transfer. The "applications" processing of the Data word is purely that of translation; from the response buffer in which the incoming Data word is stored, to a Data word location in an outgoing message.

Function Control

The purpose is to exhibit the remote initiation of a function with a single discrete.

The bus controller interrogates--at a predetermined rate--a given location in an SSIU's memory. The Data word--transferred from that location to the controller-- is tested for the condition of a "switch" bit. When clear, (0), a Data word containing all zeros, is transferred to another SSIU location. When the "switch" bit is set, (1), a Data word containing 107070 and 070707 on alternate transmissions, is placed in that location. Thus, the effect is to remotely activate a function, as represented by the flashing lights.

The message handling processing in the bus controller consists of first initiating a terminal/controller and then a controller/terminal message sequence. The application processing involves interrogation of the incoming Data word to determine the condition of the switch bit; conditional branching to alternate routines to set the appropriate content of a core location to perform the function; followed by transferring the contents to a predetermined Data word in a message addressed to the given location in the SSIU's memory.

Signal Sampling Rates

The purpose is to demonstrate the different rates at which data can be sampled from a subsystem, and transferred between subsystems.

The bus controller interrogates a data bit at a given SSIU location. When the bit is set, a data word containing fifteen zeros and a one is transferred to some other SSIU location. Each time the data bit is interrogated, i.e. at the sampling rate, the one bit in the outgoing word is shifted one place to the left. After unit time has elapsed, the number of positions through which the bit has been stepped corresponds to the sampling rate.

The above procedure is repeated for four source/sink pairs operating at different sampling rates--1, 2, 4, and 8 samples per unit time. The net effect is a set of stepping panel lights, each incrementing at a rate at which the particular source is being sampled and transferred to the sink location.

Operation of the AN/ARC-50 UHF Radio

The purpose of the AN/ARC-50 demonstration is to remotely control the radio, e.g., frequency selection, operating mode, transmitter power, etc., and to display its parameters by panel indicator and CRT.

The data transfer, by means of terminal/controller and controller/terminal message sequences, is identical in form, although different in content, to that used in the demonstrations outlined above; however, the applications processing is considerably more extensive. This arises from the need to extract eight parameters, describing the condition of the radio, from the incoming Data words. In general any given parameter can take one of several states, e.g., the frequency setting can range between 200.00 MHz and 399.95 MHz in 0.05 MHz increments. Subsequent to their extraction, the parameters can be displayed--an operator option--in tabular format on a CRT. The rate at which the radio's controls are sampled is compatible with the response time of the radio control unit--T/R unit combination--and the transfer of the control signals via the multiplex bus causes negligible degradation in response time of the control mechanism.

Remote Display of Aircraft Parameters on Cockpit Indicators

The purpose of the demonstration is to show how flight/aircraft parameters, e.g., gyro compass heading, fuel remaining, etc., can be sensed at various locations and displayed on standard cockpit indicators, and in summary format on a CRT, using the multiplex bus as the transfer medium.

In terms of the data processing involved, the message handling routines are those used for all the other demonstrations outlined above. The applications processing was similar in form, but different in detail to that used above.

Monitoring and Diagnostic Functions

The purpose of the demonstration is to show that monitoring and diagnostic functions can be implemented using the multiplex bus.

When the data transfer between subsystems is via the bus controller, it is an obvious extension from the control and display of subsystem parameters, to their monitoring--and fault diagnosis--based on their condition. Using the "fuel remaining" and hydraulic pressure parameters extracted above, threshold tests on their levels are made, and flashing DANGER symbols are superimposed on the summary display when the values fall below predetermined levels. In addition, an audible warning is activated by the same stimulus.

The function of fault diagnosis is also demonstrated using the operator options for selection of the CRT display format. If, inadvertently, the switch settings are set for two displays simultaneously, a diagnostic display is automatically called up that informs the operator that an illegal switch setting has been made.

Another implementation of the fault monitoring activity is based on the content of the Status word(s) that form part of each Command/Response message sequence. Each Status word is checked by the bus controller; if an error condition is indicated, the address of the terminal--and subaddress--are printed on the TTY.

Outline of "Response Handling" Software for the Laboratory Demonstration

Detailed discussion of specific programs is rather tedious, particularly for those readers not involved in their development. Thus, it is proposed to confine attention to the operational type demonstration software, and then to keep the description at a functional level, rather than to the details of the coding.

The processing used on all incoming messages is of two distinct types. One of these is common to all response components of the Command/Response protocol, and is concerned with routing the incoming Data words to the appropriate applications routines. The second part of the response handling is the application processing, which may vary from word to word, dependent on the content, e.g., navigational data, flight control data, etc.

If the requirement is for the Data word to be deposited in some specific location in core, then a simple procedure for this purpose would be activated. If, on the other hand, the Data word is packed with several signals destined for a number of different users, and necessitating different applications processing on each, then the initiation procedures are more complex. In all cases, whether single or multiple signals/users are involved, the processing flow is controlled by a sequence of pointers, shown diagrammatically in Figure 3. The pointer which forms the base of this "tree" is the sole point of contact between the bus control processing--associated with the transfer of data on the bus-- and that part of the applications processing being done in the computer handling the bus control function. In terms of the message handling software, therefore, the flexibility necessary to permit the bus system to service a wide range of subsystems resides in, and is confined to, the series of pointers which initiate the various applications routines required by the Data word content.

As a slight digression, it should be noted that even when the information content of the individual Data words is different, e.g., one carries range and another bearing data, there are frequently similar operations to perform on each; for example, scaling, BCD conversion, etc. Thus, it is possible, with judicious planning, to incorporate a considerable degree of commonality into the applications programs, but at a structural level below that of the bus control software.

It has already been mentioned that the invariant portion---the bus control programs--of the software handling the response components had already been developed, see Reference 1, prior to the request for the present demonstration. In consequence, the additional work required for the demonstration was confined to the development of the applications programs and their interfacing with the message control software. The latter was purely procedural consisting only of setting up tables of pointers, see Figure 3. The applications processing was specific to the type of subsystem being serviced, i.e. the UHF radio and the cockpit indicators, and is the subject of the descriptive material given below.

Outline of Applications Software for AN/ARC-50 Demonstration

The use of a multiplex bus system for the transfer of information between subsystems, or separated units of a subsystem, is conceptually straightforward. However, in practice, the constraints imposed by the desirable goals of standardization and future flexibility, may result in a message flow between units that appears on the surface somewhat redundant. This will be pointed out in the following discussions.

Information Flow. The information flow between the three units comprising the AN/ARC-50--when dedicated wiring is used--is shown in Figure 4a. In terms of a centralized bus control network, this information exchange translates into a word flow shown in Figure 4b, which in turn, in the context of the MIL-STD-1553 (USAF) bus protocol, evolves into the message interchange shown in Figure 4c. It will be recalled that the standard specifies the use of a Command/Response discipline; consequently, each information transfer requires a message sequence consisting of a command component and a response component, irrespective of whether the Data words are moved from terminal to controller or vice versa, see Figure 5. Each of these transfers incurs an overhead of two words--one Command word and one Status word--if the Data word transfer is to or from the controller; an overhead of four words--two Command and two Status--result from the use of the terminal to terminal mode of operation. For the AN/ARC-50 demonstration terminal/controller and controller/terminal transfers were used, since the information being moved was required at the controller for purposes of interpretation and display.

It should be stressed that the implementation of any given information transfer under the constraints of MIL-STD-1553 is not, in general, unique, and there is flexibility to adapt the number of message sequences used, etc., to conform with other conditions. In this context it can be seen that there is a marked difference in the number of data words required to operate the AN/ARC-50 as indicated in Figure 4a, and the number of Data words transferred on the bus, as shown in Figure 4c. There are several factors contributing to this disparity. The most significant of these is the use of a single remote terminal with only four subaddresses, in the initial implementation of the demonstration. This necessitated the use of a common subaddress for Data words intended for different units. For example, subaddress 4 is used for words to the T/R controller (H5), T/R box (H2, H3), and the frequency indicator(H1, H4, H6). Moreover, separate messages were used to each unit to permit easy changeover to the addition of a second MTU--with four subaddresses--at a later stage in the implementation. While the flow of redundant Data words incurred in the present case is somewhat artificial, resulting primarily from the unique conditions of the demonstration, analogous situations could arise in an operational system.

Applications Processing Sequence for AN/ARC-50 Data Words. The interface between the common response processing and the applications processing for the particular case of the AN/ARC-50 demonstration is shown in Figure 6a and b. All six words--the non-redundant subset of the AN/ARC-50 related Data words received from the remote terminal--are required to be transferred to other units of the radio subsystem. In addition, four of the words must be processed by the bus controller--an applications function--to

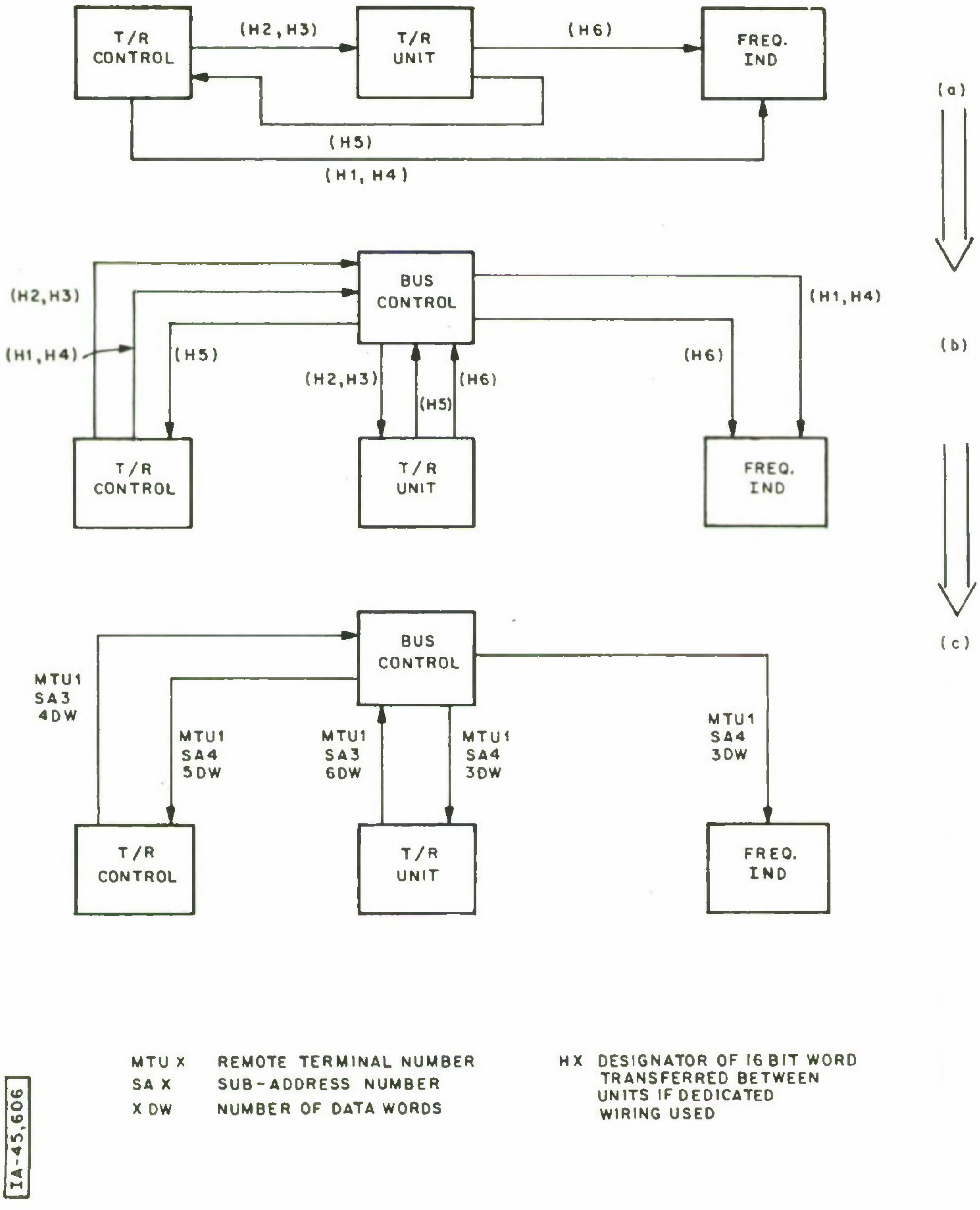
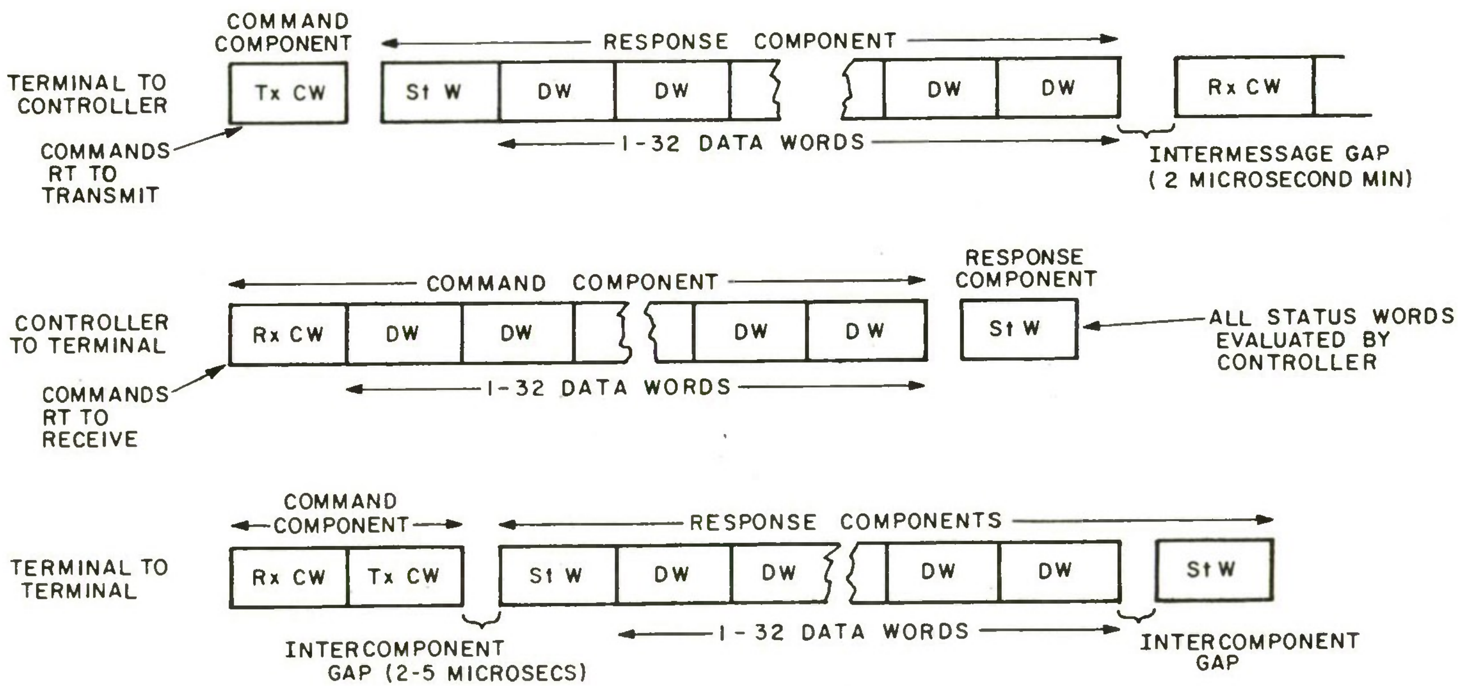


Figure 4 EVOLUTION OF INFORMATION FLOW TO MESSAGE FLOW FOR AN/ARC-50 UHF RADIO

IA-43,655



(1) COMMAND COMPONENT: SEQUENCE OF CONTIGUOUS WORDS TRANFERRED FROM CONTROLLER TO REMOTE TERMINAL.

(2) RESPONSE COMPONENT: SEQUENCE OF CONTIGUOUS WORDS TRANSFERRED FROM RT TO CTRL OR RT TO RT.

Figure 5 MESSAGE FORMATS ON MULTIPLEX BUS

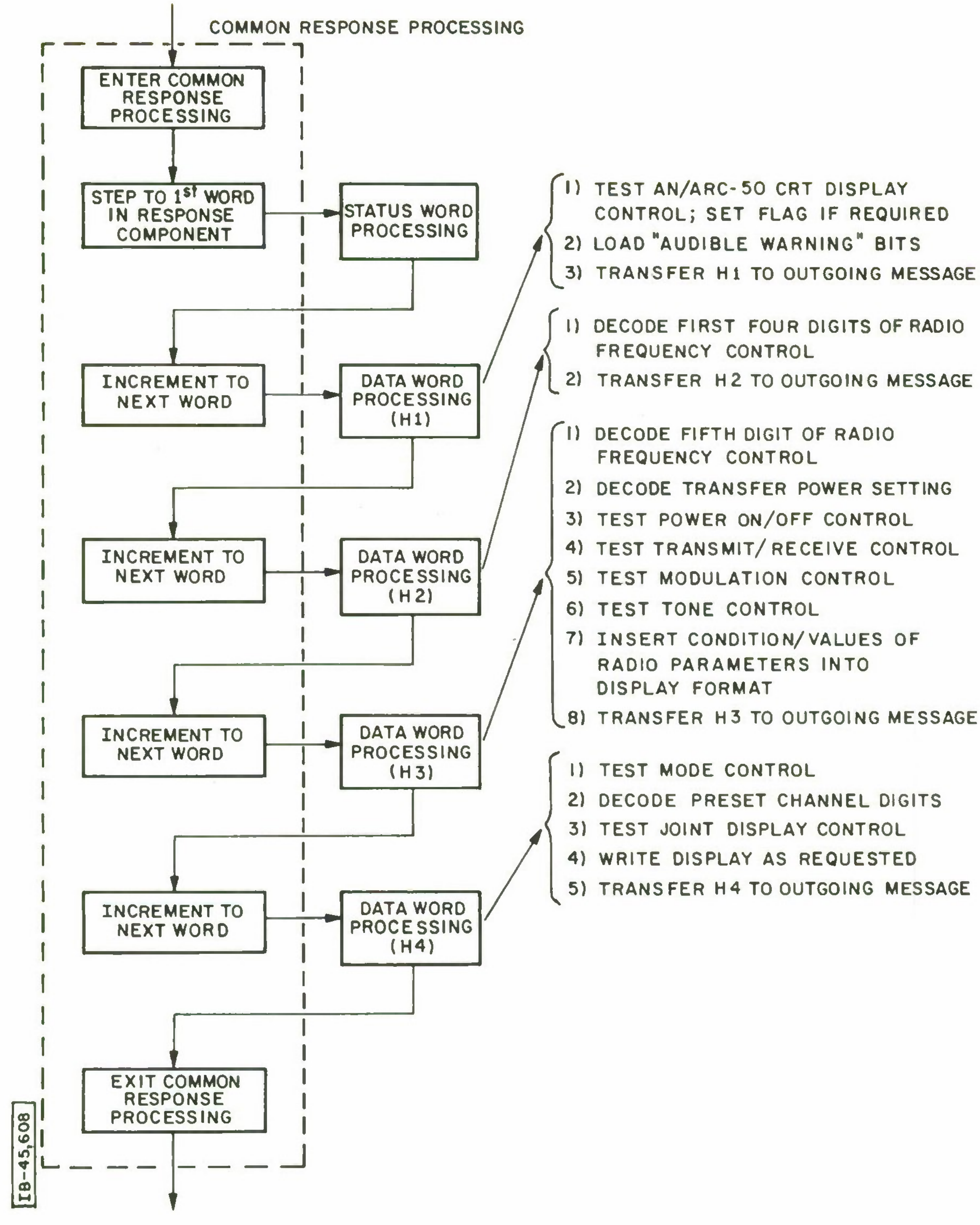


Figure 6a APPLICATIONS PROCESSING SEQUENCES FOR AN/ARC-50 DATA WORDS (MESSAGE M10102)

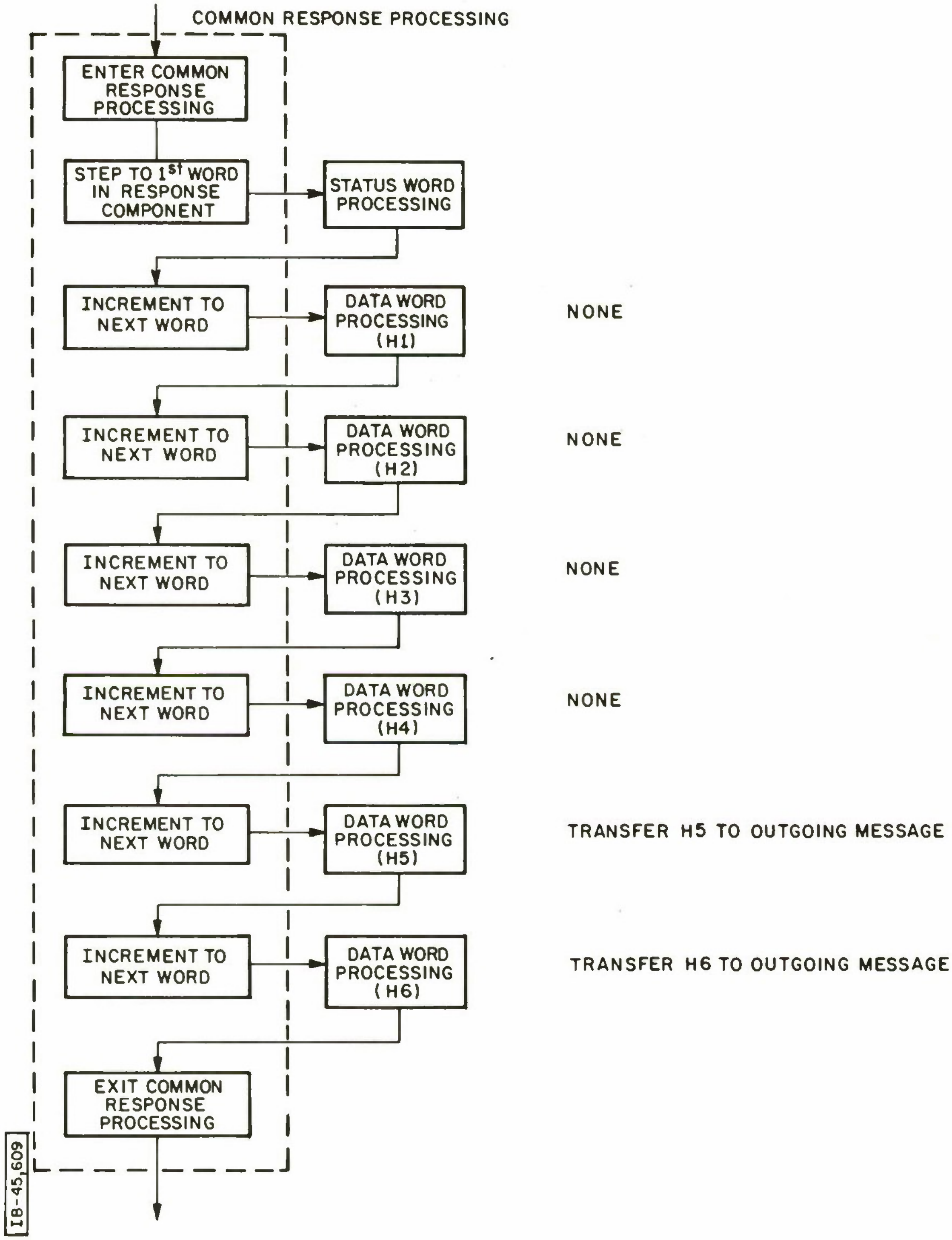


Figure 6b APPLICATIONS PROCESSING SEQUENCES FOR AN/ARC-50 DATA WORDS (MESSAGE M10102)

extract the information used in the AN/ARC-50 display and associated warning functions. The signals in these four words are formatted as shown in Figure 7. As the common response processing sequences through each of the incoming Data words in the response component that samples the radio subsystem's subaddress, it initiates the processing sequences appropriate to their information content. The activities performed are largely self explanatory, and further discussion of the individual routines is not warranted; however, a general observation is worth making. The subsystem selected for the demonstration, the AN/ARC-50, was not originally designed for use with a multiplex bus, and provides a good example of some of the problems involved in adapting "as is" equipment to bus usage. An example of this, in the area of applications processing, is the lack of uniformity in the representation of the information content in the Data words; the digit 3 as coded in the frequency word is different from a 3 in the preset channel, which in turn differs from a 3 used to denote a power setting. Such lack of standardization necessitates the use of a number of software routines with identical functions but dissimilar in detail, resulting in a considerably larger software package than might be anticipated.

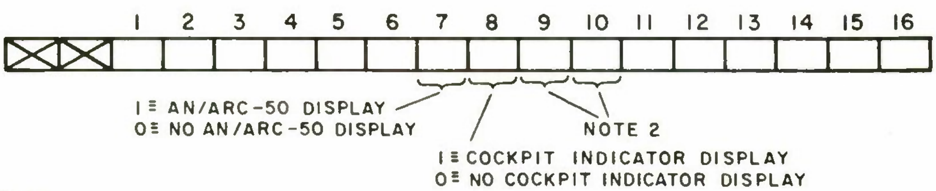
Outline of Applications Software for the Cockpit Indicator Demonstration

The approach to servicing the cockpit indicators by the multiplex bus is essentially identical to that used for the AN/ARC-50; however, the details differ somewhat.

Information Flow. The evolution of the MIL-STD-1553 version of the message flow between the cockpit indicators and their drivers from the information flow if dedicated wiring was used, is shown in Figures 8a to 8c. The same subaddresses, 3 and 4 of MTU1, as are used to service the AN/ARC-50 radio, are also used for the input and output data for the cockpit indicators. Since the message sequences for all the capabilities described in this report are loading the bus essentially simultaneously, and two words are required for the cockpit indicator data, eight words total will be used in subaddresses 3 and 4, i.e. six words for the radio, and two for the cockpit indicators. Two of the indicator drivers are sampled, and their data transferred to the indicators, four times per unit time, and one, the gyro repeater, twice per unit time; i.e. half and one quarter, respectively, of the sampling rate used for the radio control.

Applications Processing Sequence for the Cockpit Indicator Data Words. The interface between the common response processing and the applications processing for the cockpit indicators is shown in Figures 9a and b. The first six Data words of the response component are not associated with cockpit indicators, and in

WORD 1 (H1): T/R CONTROL TO FREQUENCY INDICATOR

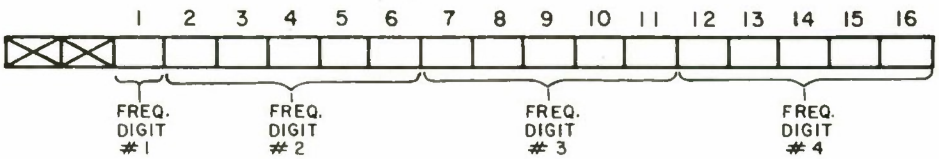


NOTES:

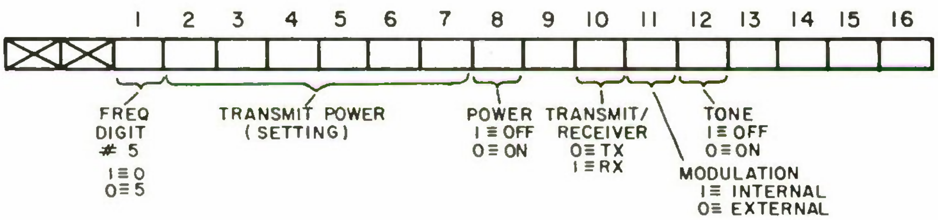
(1) WHEN BITS 7 AND 8 ARE BOTH SET A FAULT DIAGNOSTIC DISPLAY IS CALLED.

(2) BITS 9 AND 10 ARE USED BY BUS CONTROL TO ACTIVATE WARNING SIREN FOR FUEL AND HYDRAULIC PRESSURE PARAMETERS.

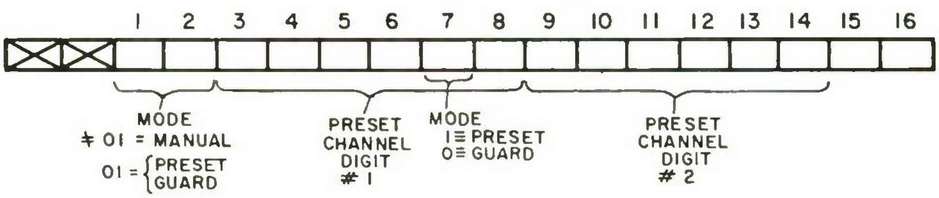
WORD 2 (H2): T/R CONTROL TO T/R UNIT



WORD 3 (H3): T/R CONTROL TO T/R UNIT



WORD 4 (H4): T/R CONTROL TO FREQUENCY INDICATOR



WORD 5 (H5): T/R UNIT TO T/R CONTROL

NO COMPUTER PERTINENT CONTENT

WORD 6 : T/R UNIT TO FREQUENCY INDICATOR

NO COMPUTER PERTINENT CONTENT

IB-45,607

Figure 7 SIGNAL FORMATS IN AN/ARC-50 DATA WORDS

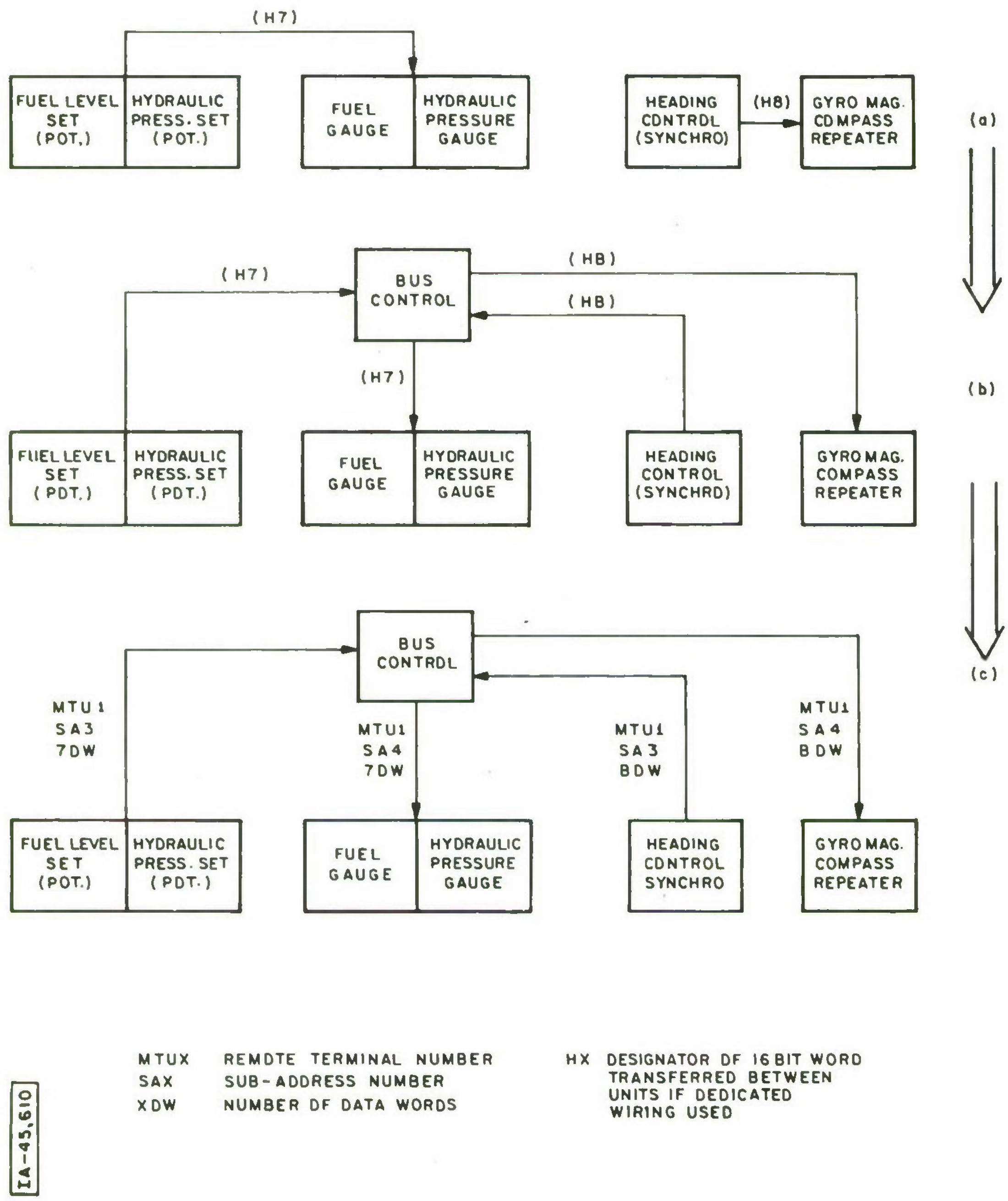


Figure 8 EVOLUTION OF INFORMATION FLOW TO MESSAGE FLOW FOR COCKPIT INDICATORS

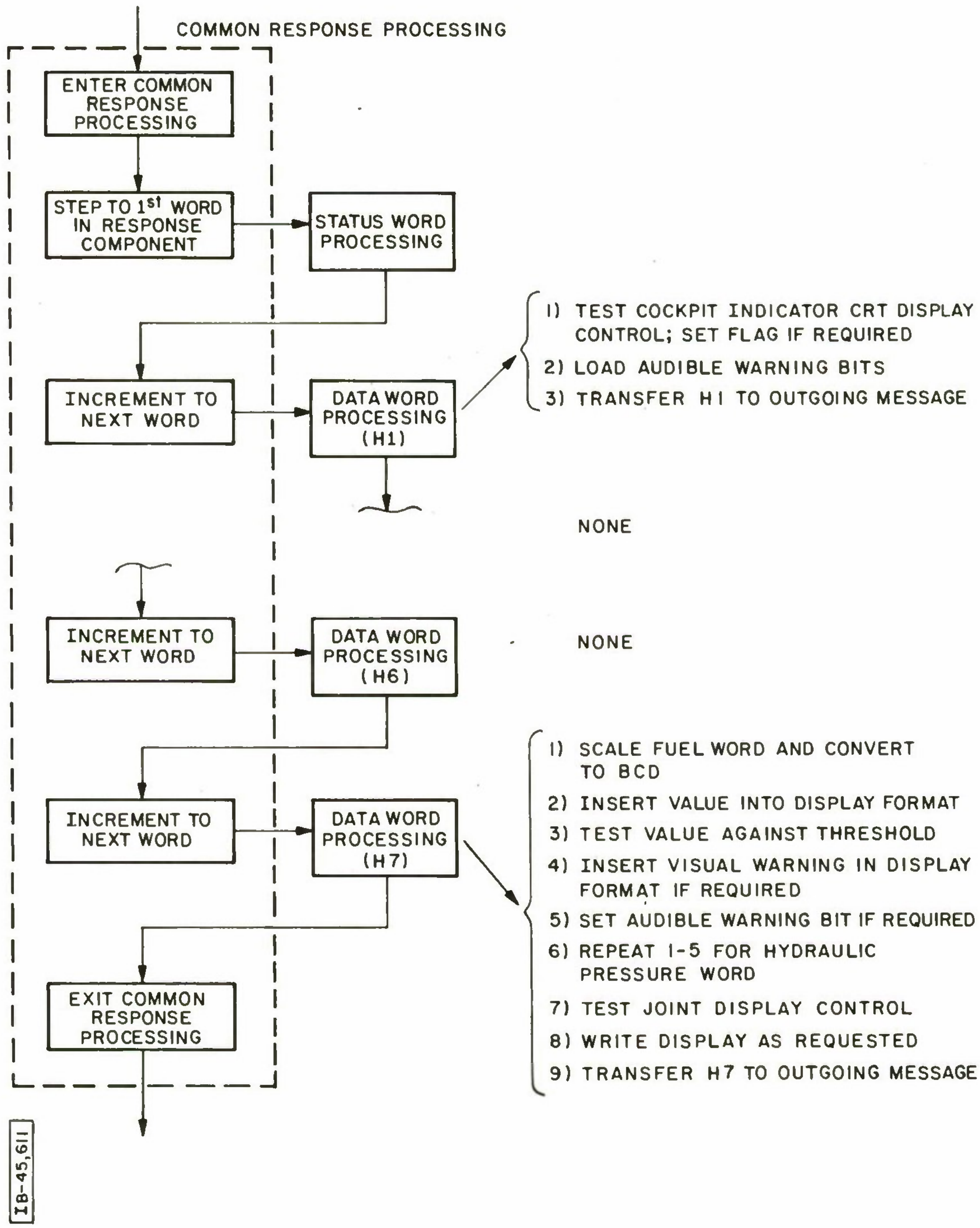


Figure 9a · APPLICATIONS PROCESSING SEQUENCES FOR COCKPIT INDICATOR DATA WORD (MESSAGE M 20102)

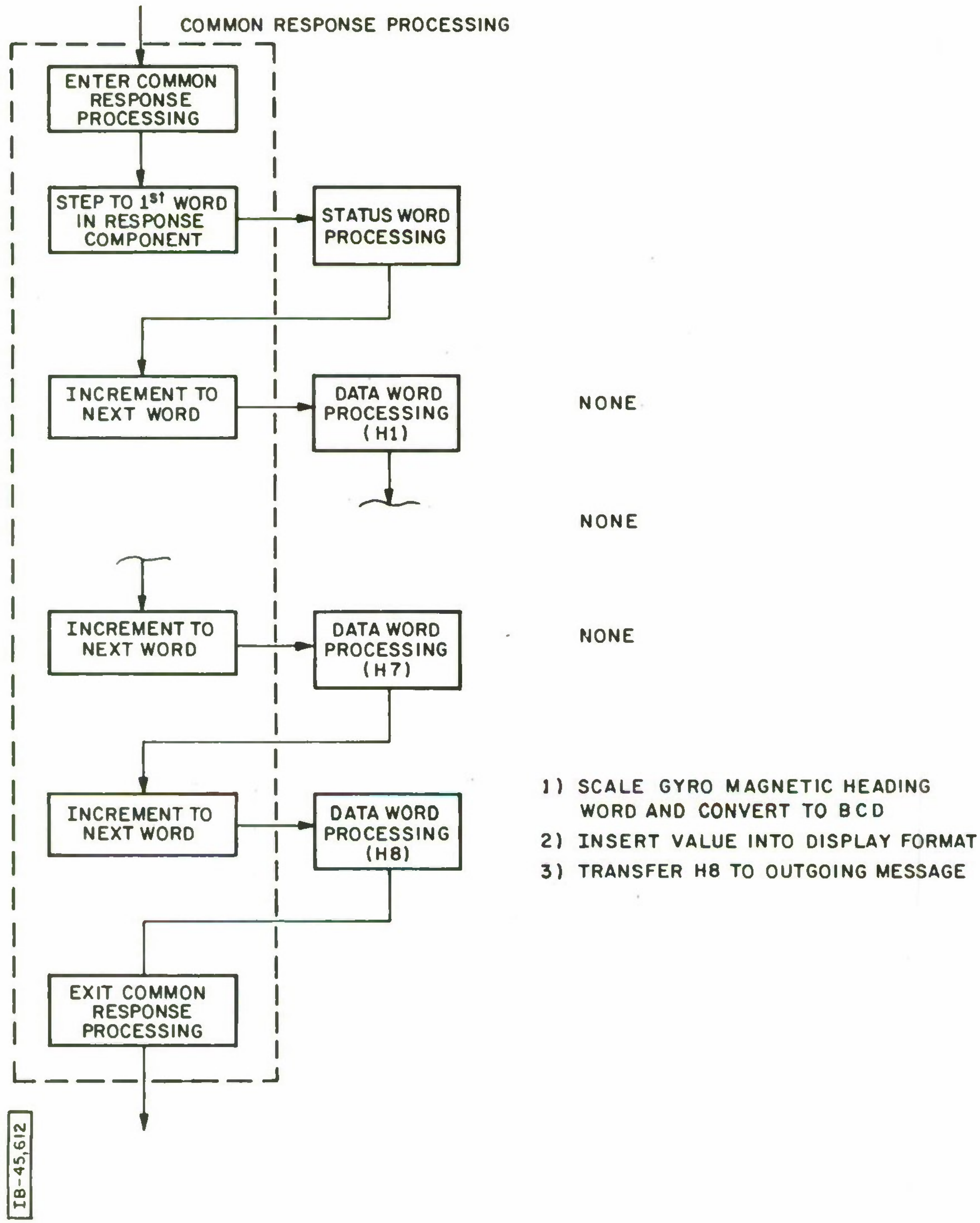


Figure 9b APPLICATIONS PROCESSING SEQUENCES FOR COCKPIT INDICATOR DATA WORD (MESSAGE M30301)

consequence are not subject to applications processing. The format and functional content of Data words 7 and 8 are shown in Figure 10. The information transfer between the indicators and their drivers is in the form of binary numbers; the A/D converters associated with the fuel and hydraulic indicators quantizing the driver (analogue potentiometer) settings into 8 bit words; the synchro converters produce 10 bit outputs. In all cases the processing consists of scaling the binary word--according to the range of the parameter being represented--, converting to 7 bit ASCII via BCD, and inserting the characters into a format for subsequent display on a Sanders 720 CRT.

Additional Demonstration Capabilities

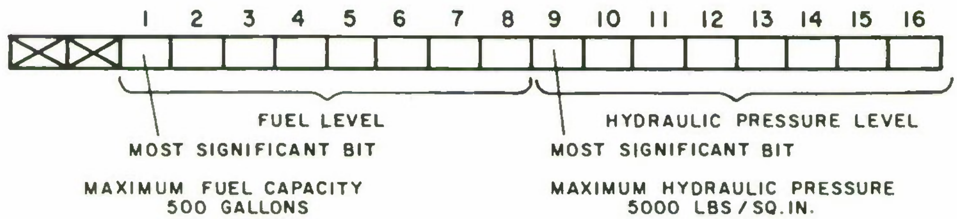
In addition to the applications software associated with the UHF radio and indicator demonstrations, some simple fault diagnostic capabilities were programmed.

One of these is derived from the validation of the Status word(s), which is a standard procedure on all response components. If the Status word is valid, processing of the Data words proceeds along the paths described previously. If, on the other hand, an error bit is set in the Status word, the processing branches to a routine which prints out--on a TTY--the MTU address and subaddress to which the message had been sent. Subsequent applications processing of the Data words in this response component is bypassed, and the message handler steps to the next message and proceeds routinely.

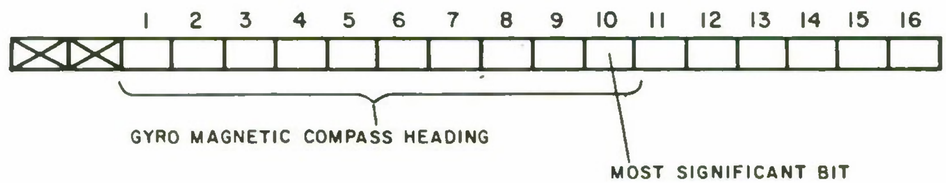
Another capability being demonstrated is the detection of operator errors and a display of the event. The operator is provided with two toggle switches which provide the following options: select AN/ARC-50 CRT display, cockpit indicator CRT display, or clear CRT display. A possible, but illegal, pair of switch settings calls both display formats in rapid alternating sequence. The applications processing contains a subroutine which interprets the bits, 7 and 8 in Data word H1, which are set by the switches. If the illegal setting is detected, a "fault diagnostic" display is automatically called that informs the operator that this condition exists.

The demonstration of the monitoring of flight parameters includes both the fuel and hydraulic pressure levels. The extraction of the latter from the incoming data words for display purposes, has already been described in the previous section. The applications processing associated with the monitoring function is a simple extension to that operation. The level of the parameter, as determined from the Data word, is compared with a preprogrammed threshold value. When the variable falls below the threshold level,

WORD 7 (H7): LEVEL CONTROLS TO FUEL GAUGE AND HYDRAULIC PRESSURE GAUGE



WORD 8 (H8): HEADING CONTROL TO MAGNETIC COMPASS REPEATER



IB-45,613

Figure 10 FORMAT AND FUNCTION CONTENT OF COCKPIT INDICATOR DATA WORDS a routine is initiated which superimposes, on the CRT display, an intermittent DANGER signal adjacent to the critical parameter. In addition, a bit is set in an outgoing Data word, which activates an audible alarm at the remote terminal.

SECTION III

SUMMARY AND CONCLUSIONS

The software that has been developed for a laboratory demonstration of the MITRE experimental avionics multiplex bus can be subdivided into two distinct types. One of these is associated with the common functions of message handling and bus control, and is relatively invariant with respect to the classes of equipment being serviced by the network. This processing has been discussed in some detail in a previous report (1), and is summarized briefly herein. The second type of software is application specific, and is largely dependent on the equipments serviced by the bus and the functional interactions that they have one with another. The applications programs developed to implement a small subset of the potential capabilities of an avionics multiplex information distribution network have been described. The latter include the remote control of an AN/ARC-50 UHF radio and a corroborative CRT display of its operational parameters, the remote display of various flight parameters on an array of cockpit indicators, the monitoring of critical flight variables, and the initiation of visual and audible warning signals when the values fall below some predetermined threshold levels, and finally the automatic presentation of a fault diagnostic display when an operator makes an illegal switch setting.

The distinction between message control processing and applications processing, made previously, is also of significance when considering the overall data processing load associated with bus usage. By judicious distribution of the message control functions between a special and general purpose (g.p.) processor, the residual message control processing requirements per message sequence, on the latter, can be small. The resultant contribution to the overall load on the g.p. processor is then a direct function of the duty cycle at which the avionics bus is operated. At 100% duty cycle, using short messages, the message handling activities could well monopolize the capacity of a medium size g.p. processor; at bus loads corresponding to the estimated data transfer requirements of representative aircraft missions, the message control load would be essentially negligible.

While the applications processing load is also duty cycle dependent, it is, in addition, extremely vulnerable to ill conceived system design. The potential for avionics bus usage appears virtually limitless; however, embedded in the worthwhile applications are many seemingly inconsequential functions that would be "nice to do". The ready accessibility of flight data, control data, etc., at the computer gives an illusion of being able to acquire additional capability with little penalty. The problem is, of course, that the capacity of the g.p. processor will rapidly be saturated by the execution of the software needed to implement these cosmetic functions, particularly if they are associated with parameters being sampled at a high data rate.

REFERENCES

1) Digital Avionics Group, "Final Report for Project 6370," ESD-TR-75-70, July 1975.