IBM

Saturn Instrument Unit

SLCC PROGRAMMER'S REFERENCE MANUAL



Contract No. NAS 8-14000
MSFC-DRL-008A
Line Item No. 295
IBM No. 71W-00199

Prepared for the
GEORGE C. MARSHALL
SPACE FLIGHT CENTER
Huntsville, Alabama

SLCC PROGRAMMER'S REFERENCE MANUAL

CHANGE 4
1 MAY 1973

CLASSIFICATION AND CONTENT APPROVAL

DATA MANAGER APPROVAL

PROGRAM OFFICE APPROVAL

IBM Federal Systems Division, Electronics Systems Center/Huntsville, Alabama

CHANGE NOTICE

SLCC Programmer's
Reference Manual

1 May 1973

The attached pages constitute Change 4 to the subject document; they are to be added to, or are to replace pages in, this document.

Changed, revised, added, and deleted pages, as noted by the appropriate label and effectivity date at the lower edge of the page, are defined as follows.

- Change Page - Changes in content are identified by change bars or shading.
- Revised Page - Changes in technical content are so extensive or numerous that change bars or shading cannot be used effectively.
- Added Pages - The page is new to the document (contents may or may not be new).
- Deleted Page - Contents of a page or a sequence of pages are deleted from the document.

This change incorporates the following items.

UCN
16050

The following pages should be removed and replaced with the appropriate pages as indicated in the following table.

Pages to Be Removed	Pages to Be Inserted
Frontal Matter	
Title Page	Title Page
Section 2	
2-5, 2-6, 2-19, 2-20, 2-27/2-28	2-5, 2-6, 2-19, 2-20, 2-27/2-28
Appendix	
A-9, A-10	A-9, A-10

CHANGE NOTICE

1 OCTOBER 1972

SLCC PROGRAMMER'S
REFERENCE MANUAL

The attached pages constitute Change 3 to the subject document; they are to be added to, or are to replace pages in, this document.

Changed revised, added, and deleted pages, as noted by the appropriate label and effectivity date at the lower edge of the page, are defined as follows.

- Change Page - Changes in technical content are identified by change bars or shading.
- Revised Page - Changes in technical content are so extensive or numerous that change bars or shading cannot be used effectively.
- Added Page - The page is new to the document (contents may or may not be new).
- Deleted Page - Contents of a page or a sequence of pages are deleted from the document.

This change incorporates the following items.

UCN:	PTR:
14678	D-198

The following pages should be removed and replaced with the appropriate pages as indicated in the following table.

UCN	Pages to Be Removed	Pages to Be Inserted
	Frontal Matter	
	Title Page	Title Page
	Section IV	
14678	4-61, 4-62, 4-70A, 4-70B	4-61, 4-62, 4-70A, 4-70B
	4-71, 4-72, 4-81/4-82	4-71, 4-72, 4-81, 4-82
	Appendix	
D-198	A-9, A-10	A-9, A-10

SLCC PROGRAMMER'S REFERENCE MANUAL

(15 June 1972)

CHANGE NOTICE

The attached pages constitute Change 2 to the document. They are to be added to, or are to replace pages in, this document. Changes to the text are noted by vertical change bars in the margin of the text. Revised pages are so identified by a label on the bottom of the page.

This change incorporates the following PCR.

PCR

13756

The following pages should be removed and replaced with the appropriate pages as indicated in the following listing.

UCN	Pages to be Removed	Pages to be Inserted
13756	Frontal Matter Title Page Section IV 4-47 thru 4-50, 4-53, 4-54, 4-57, 4-58, 4-61 thru 4-70	Title Page 4-47 thru 4-50, 4-53, 4-54, 4-57, 4-58, 4-61 thru 4-70, 4-70A, 4-70B

SLCC PROGRAMMER'S
REFERENCE MANUAL

(15 March 1972)
CHANGE NOTICE

The attached pages constitute Change 1 to the document. They are to be added to, or are to replace pages in, this document. Changes to the text are noted by vertical change bars in the margin of the text. Revised pages are so identified by a label on the bottom of the page.

This change incorporates the following PCRs and Technical Directive.

PCR	TD
13748	N-IU-9-72

The following pages should be removed and replaced with the appropriate pages as indicated in the following listing.

	Pages to be Removed	Pages to be Inserted
	Title Page	Title Page
	Section IV	
13748	4-89, 4-90	4-89, 4-90
	Appendix A	
TD-N-IU-9-72	A-15, A-16	A-15, A-16

TABLE OF CONTENTS

Section	Paragraph	Title	Page
I		INTRODUCTION	1-1
	1.0	SCOPE	1-1
	1.1	CONTENTS	1-3
II		PROGRAM DEVELOPMENT	2-1
	2.0	SUMMARY	2-1
	2.1	PROGRAM STRUCTURE	2-3
	2.1.1	Names	2-5
	2.1.2	Size and Residency	2-6
	2.1.3	Test Program Communication Cells	2-7
	2.1.4	MDO Discrete Disallow Table	2-10
	2.1.5	Display Console Key Code	2-11
	2.1.6	General Storage	2-12
	2.1.7	Linkage For Programs and Routines	2-12
	2.1.8	Interrupt Processing	2-13
	2.1.9	Test Subroutine	2-14
	2.1.10	Forced Test End Subroutines	2-14
	2.1.11	Function Executor Routine Communication Cells	2-15
	2.2	TEST PROGRAM CONTROL PROCESSING	2-17
	2.2.1	Options	2-19
	2.2.2	Returns to Test Program Control	2-27
	2.2.3	Environment Check	2-29
	2.2.4	Test Program Requests for Function Executor Routines	2-29
	2.3	PROGRAM LOADING AND EDITING	2-33
	2.3.1	Test Program System Master Tape	2-35
	2.3.2	Card Input	2-35
	2.3.3	On-Line Program Editing	2-41
	2.3.4	Function Executor Routine Assembly and Loading	2-42
III		OPERATING SYSTEM SERVICES	3-1
	3.0	SUMMARY	3-1
	3.1	INTER-COMPUTER CORE TRANSFER	3-3
	3.1.1	Control Word Format	3-5
	3.1.2	Returns from the Remote Core-to-Core Subroutine	3-6
	3.2	SYSTEM LOGGING	3-7
	3.2.1	Logging By Test Program Control	3-7
	3.2.2	Logging By Function Executor Control (LCCC)	3-8
	3.2.3	Logging By Display Service Programs	3-8
	3.2.4	Logging Vehicle Data	3-9
	3.2.5	Logging by System Message Routine	3-9

TABLE OF CONTENTS (cont)

Section	Paragraph	Title	Page
	3.3	UTILITY SERVICES	3-11
	3.3.1	Data Conversion	3-11
	3.3.2	Accessing and Converting Time	3-13
	3.3.3	Printing Data	3-16
	3.3.4	Saving Register Data	3-17
IV	SYSTEM INPUT/OUTPUT		4-1
	4.0	SUMMARY	4-1
	4.1	LAUNCH VEHICLE INPUT/OUTPUT	4-3
	4.1.1	Discrete Input/Output	4-3
	4.1.2	LVDC Input/Output Interface	4-15
	4.1.3	Hardwire Analog Voltage Input	4-23
	4.1.4	Digital Data Acquisition System (DDAS)	4-25
	4.2	DISPLAY SYSTEM INPUT/OUTPUT	4-33
	4.2.1	Operational Display System	4-33
	4.2.2	Types of Displayed Data	4-33
	4.2.3	Digital Data and Slide Data	4-35
	4.2.4	Digital Data Symbols	4-35
	4.2.5	Display Screen Configuration	4-35
	4.2.6	Referencing of Displayed Data	4-36
	4.2.7	Display Descriptions	4-41
	4.2.8	DD Symbolic and Binary Tape Input	4-48
	4.2.9	Display Requests	4-71
	4.3	PERIPHERAL DEVICES INPUT/OUTPUT	4-83
	4.3.1	Magnetic Tape Input/Output	4-85
	4.3.2	Drum Input/Output	4-87
	4.3.3	Card Reader Input-LCC	4-88
	4.3.4	Line Printer Output	4-90
APPENDIX A		CALLING SEQUENCES	A-1
APPENDIX B		OPERATING SYSTEM ROUTINE LINKAGE	B-1
APPENDIX C		COMMON STORAGE AREAS	C-1
APPENDIX D		ABBREVIATIONS AND ACRONYMS	D-1
APPENDIX E		REFERENCES	E-1
APPENDIX F		DDTG Card Data Calculations	F-1

LIST OF ILLUSTRATIONS

Figure	Title	Page
2.1.1	Test Program Size and Residency	2-7
2.1.2	Test Program Communication Cells	2-9
2.1.3	MDO Discrete Disallow Table	2-10
2.1.4	Console Key Coding	2-11
2.1.5	Load Macro Control Word	2-14
2.1.6	Function Executor Routine Communication Cells	2-15
2.2.1	Table Structure for Test Program Options	2-22
2.2.2	Parameter Option Control Word and Core Structure	2-23
2.2.3	Parameter Validity Check	2-24
2.2.4	Parameter Storage Area for the LCCC and the MLC	2-25
2.2.5	Power Check Communication Cell	2-29
2.2.6	Execute Control Words	2-31
2.2.7	Terminate Control Words	2-31
2.3.1	Test Program Master Tape Generation	2-33
2.3.2	Test Program System Master Tape	2-36
2.3.3	INPUT Card Format	2-37
2.3.4	STAG Card Format	2-38
2.3.5	Patch Check Communication Cell	2-41
3.1.1	Control Word Format	3-5
3.2.1	Control Words for Logging Data	3-8
3.3.1	BCD Format of Greenwich Mean Time	3-14
3.3.2	Greenwich Mean Time Conversion Using Accumulators	3-14
3.3.3	Countdown Time Format	3-16
3.3.4	Register Storage Locations	3-17
4.1.1	Launch Vehicle Input/Output Data Flow	4-4
4.1.2	Issue Single Discrete Control Word	4-6
4.1.3	Issue Multiple MDO Control Words	4-6
4.1.4	Single Discrete Status Check Control Word	4-8
4.1.5	Multiple Discrete Status Check Control Word	4-9
4.1.6	MDO-MDI Timing Relationship	4-11
4.1.7	Issue MDO and Check MDI Response Control Word	4-11
4.1.8	Switch Selector Control Word	4-12
4.1.9	LVDC Input Data Communication Cell and Buffer Structure for Test Programs and Function Executor Routines	4-16
4.1.10	LVDC Data Word Conversion	4-17
4.1.11	LVDC Error Table Format (LCCC and MLC)	4-18
4.1.12	LVDC History Table Format (LCCC and MLC)	4-18
4.1.13	Output Storage for LVDC Data	4-19
4.1.14	LVDC Command and Control Words (SYNC Example)	4-19
4.1.15	Digital Data Acquisition System/RCA-110A Interface Data Word Format	4-26
4.1.16	DDAS Control Word and Storage	4-28

LIST OF ILLUSTRATIONS (cont)

Figure	Title	Page
4.1.17	Calibration In Progress Communication Cell	4-29
4.1.18	DDAS Read Request Word Format	4-30
4.1.19	Terminate or Delete Control Word	4-32
4.2.1	Display System Input/Output Data Flow	4-34
4.2.2	Vector Segment Calculation and Refresh Memory Word Formats	4-38
4.2.3	Display Console Grid Format for Display of Digital Data Mixed with Slide Data or Slide Data Alone	4-39
4.2.4	Display Description Categories and Data Types	4-43
4.2.5	Refresh Memory Space Allocations for Display Description Data	4-47
4.2.6	Display Description Binary Tape Format (Sheets AA through AE)	4-49
4.2.7	Display Description Tape Generation/Update System Flow	4-54
4.2.8	Display Request Control Word Format	4-71
4.2.9	Binary Control and Data Word Format	4-74
4.2.10	Packed BCD Control and Data Word Format	4-75
4.2.11	Preformatted Control and Data Word Format	4-77
4.2.12	Refresh Memory Format for Character Display	4-77
4.2.13	Refresh Memory Format for Vector Display	4-79
4.2.14	Display Memory (Drum or Core) Control and Data Word Format	4-80
4.2.15	Display Description Control Word Format	4-81
4.2.16	Operator Select or TV Transmit Control Word Format	4-81
4.3.1	Saturn Peripheral Device Assignments	4-84
4.3.2	Word Formats for Tape Operations	4-85
4.3.3	Drum Read and Write Control Words	4-88
4.3.4	Read Card, Write Printer Control Words	4-89
4.3.5	Page Position Control Word Format	4-92

LIST OF TABLES

Table	Title	Page
2.1-1	First Character IVAR Assignments	2-5
2.1-2	Second Character IVAR Assignments	2-6
2.1-3	Console Key Codes	2-11
4.1-1	Discrete Functions	4-5
4.1-2	Discrete Status Check Subroutines	4-8
4.1-3	LVDC Routine Regular Status Codes	4-20
4.1-4	LVDC Routine Special Status Codes	4-21
4.2-1	Minimum Requirements for Slide Data Used to Group Digital Data	4-40
4.2-2	Display Description Numbers by DD Category	4-42
4.2-3	Data Control Word Codes	4-74
4.3-1	Page Position Codes	4-91

SECTION I

INTRODUCTION

1.0 SCOPE

The Saturn Operating System (Operating System) contains many programs, routines, and subroutines which together provide the software control functions necessary for automatic checkout of the Saturn Launch Vehicle, (both Saturn V and Saturn I-B). The complexity of the use of this Operating System makes a detailed explanation essential.

The primary purpose of this document is to provide the test programmer with vital information needed for efficient use of the Operating System. Specific information needs of the test programmer include:

- What the test programmer is required to do in order to use the Operating System.
- What the Operating System provides for the test programmer.
- What the test programmer may choose to do with Operating System capabilities.
- How the programmer communicates with system hardware.

This manual is specifically prepared for experienced test programmers familiar with RCA 110A programming techniques. It is not intended for use as a hardware description, or as a user's manual for test engineers and test console operators.

1.1 CONTENTS

Generally, this manual informs the test programmer of data word formats, required control information, storage assignments, symbolic names for accessing data, program linkage and intercommunication techniques, and system MACRO instructions with their calling sequences and functions. This manual also contains functional explanations of various aspects of the Operating System specifically oriented toward the test programmer.

The major areas discussed in this manual are Test Program Development, Operating System Services, and System Input/Output. Each of these sections is divided into units of subparagraphs which contain a detailed discussion of the functional concepts.

Tabulated data appear in the appendices, which cover Calling Sequences, Operating System Routine Linkage, Common Storage Areas, and a Glossary of Acronyms and References.

A brief summary is included at the beginning of each section providing a quick reference and overview of the text. A functional point of view is given in reference to the detailed material covered in the section. Information is presented by order of importance wherever possible.

The following terms appearing in the text have been restricted to definitions given here. The word "program" refers to test programs, while the word "routine" refers to Function Executor routines (unless Operating System routines are discussed). The word "subroutine" refers to any program or routine called in and used by an operating core resident program or routine.

SECTION II

PROGRAM DEVELOPMENT

2.0 SUMMARY

This section presents information useful to the programmer from the planning stage to the final running stage of a test program. Test programs and Function Executor routines are often discussed together since both interface with the Operating System and perform some related functions; however, they are discussed separately whenever necessary.

Three units of development are presented:

- Program Structure
- Test Program Control Processing
- Program Loading and Editing

Program structure encompasses characteristics such as name, size, and standards for interfacing with the Operating System. These standards include communication cells, tables, counters, interrupts, linkage, and subroutines available or incorporated into programs and routines.

Test Program Control processing provides an interface between the test engineer and the test program. Starting with options, this section describes terms relating to program execution including returns, environment checks, and execution of Function Executor routines.

The Program Loading and Editing section begins with a description of a test program master tape. Various card formats and program editing capabilities are discussed; and also included is a paragraph concerning assembly and loading conventions for Function Executor routines.

2.1 PROGRAM STRUCTURE

Test programs and Function Executor routines have certain common characteristics termed design requirements which are basic to the development of test programs operating in the LCCC and the MLC. These characteristics are then described individually, as follows:

- Names
- Size and Residency
- Test Program Communication Cells
- MDO Discrete Disallow Table
- Display Console Key Code
- General Storage
- Program Linkage With Operating System
- Interrupt Processing
- Test Subroutines
- Forced Test End Subroutines
- Function Executor Routine Communication Cells

2.1.1 Names

Standards have been established for defining names of test programs, function executor routines, test subroutines, and externally assembled tables. All names must be unique, four characters in length, and the first IVAR used in the program, subroutine, or table. These names are assigned by utilizing the following format:

X Y a b

X must be alphabetic. It is selected to indicate the vehicle subsystem being tested, the vehicle stage being tested, or the function the program is to perform. Table 2.1-1 lists first characters and a brief explanation of their respective indications:

Table 2.1-1 First Character IVAR Assignments

Character	Indication
A	Operating System
B	S-IC Stage Propellants
C	Control System
D	Diagnostics
E	Operating System
F	Flight Computer
G	Guidance Platform
H	Spacecraft
I	Saturn IB Integrated Test
J	Operating System
K	IU Stage
L	Operating System
M	Measurements
N	Network and Mechanical Systems
O	S-IB Stage
P	S-II Stage Propellants
Q	S-II Stage
R	S-IVB Stage Propellants
S	Operating System
T	Telemetry
U	Utility Program
V	Saturn V Integrated Test
W	S-IC Stage
X	S-IVB Stage
Y	Verification Programs For SLCC Programs
Z	Postprocessing

Y must be alphabetic. It is selected to either indicate the source language or better define the program function. Table 2.1-2 lists second characters and gives a brief explanation of their respective indications.

Table 2.1-2 Second Character IVAR Assignments

Character	Indication
A	Test Program Coded in ATOLL Source Language
C	SDS/Sanders Generated Programs
D	930 DEE-6 Program
E	Saturn V Function Executor Routine
F	Table
H	092 DEE-6 Program
M	Machine Language Subroutine for ATOLL Programs
N	NASA-MSFC/KSC Generated Programs
P	Stand-Alone Utility Program
R	RCA Generated Programs
S	Stand-Alone Program
T	Test Program Coded in SLAP or SLAP Compatible Source Language
U	On-Line Utility Program
V	Programs for Saturn V only
X	Program Capable of Running at All Saturn Complexes
Z	Posprocessing for Specific Types of Test Programs

The third and fourth characters ab, can be either alphabetic or numeric, but must be selected to ensure uniqueness. The alphabetic characters "I" and "O" will not be used as either the third or the fourth characters.

2.1.2 Size and Residency

2.1.2.1 Test Programs

The size of a test program operating in the LCCC is limited to the area of core storage reserved for it (banks 6 and 7). The test program as defined herein consists of communication cells, the main routine, subroutines, tables, constants, control words, and messages. If more than two banks are required, some of these tables, constants, and messages may be placed on drum and called in when needed. Two 4000 word drum areas and one 1000 word area are reserved for test program use. The IVAR's assigned to these areas are SSSS, SSST, and SSSU.

Additional tape resident subroutines may be called into core when needed. The size of these test subroutines is limited to a maximum of 4093 words. When they are called for use, they may overlay any of the main test program resident in banks 6 or 7 except for the 20 communication cells. However, subroutine overlays must not overlay a bank boundary.

NOTE

Since test subroutines may overlay positions of a test program, the programmer should use caution, and code accordingly when using the overlay technique.

Figure 2.1.1 indicates the size and residency of the test programs and test subroutines.

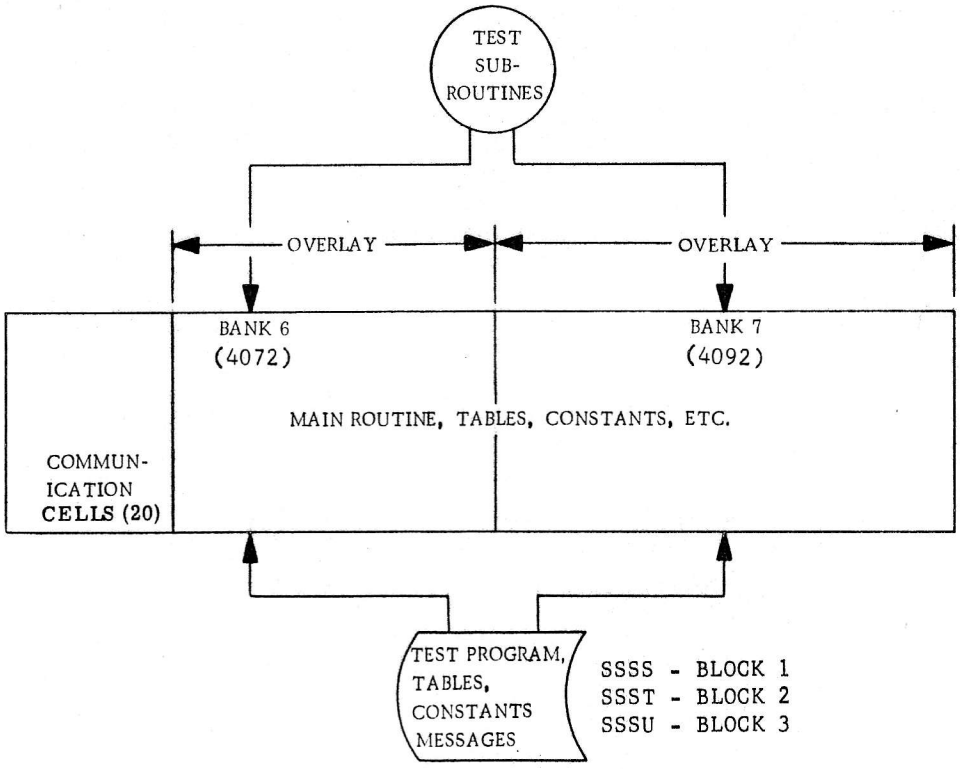


Figure 2.1.1 Test Program Size and Residency

2.1.2.2 Function Executor Routines

Critical function executor routines reside in core. Non-critical function executors are drum-resident, and are read into the last half of bank 3 in the MLC. Both core resident and drum resident function executors are limited to 3775_8 locations to allow for two locations to contain checksum data at the end of the function executor.

2.1.3 Test Program Communication Cells

Communication between the test programs in the LCCC and the Operating System is accomplished through communication cells, which are the first 20 words of a test program. Function Executor

routines in the MLC reserve the first ten words (see paragraph 2. 1. 11). The words in the communication cells must contain information as explained in the following list, with the exception that WORDS 3, 4, and 5 must be zero if the communication function described is not applicable to the test program. Figure 2. 1. 2 summarizes the contents of the communication cells. Octal, decimal, and relative addressing is feasible. Decimal designations are used here:

- WORD 1 - A jump instruction (JRA) to the first test program instruction to be executed. Used to initiate the execution of a test program.
- WORD 2 - Bits 14-0 must contain the address of the test program's Forced Test End routine. This routine is a short routine which the test program executes before being terminated. Bits 23-15 are not used.
- WORD 3 - Bits 14-0 contain the address, which may be an XVAR, of the test program's Programmable Interval Timer (PIT) No. 1 Interrupt routine. It is a short routine to be executed as a PIT No. 1 interrupt occurs. The test program must indicate that it is expecting an interrupt by setting bit 23 of this word to a one. Bits 22-15 are not used. Control is then given to the test program's PIT No. 1 Interrupt routine using a TSP instruction. PIT No. 1 Interrupt routine then returns control to Operating System at the address stored previously by the TSP instruction.
- WORD 4 - Bits 14-0 must contain the address (which may be an XVAR) of the test program's DDAS log mode (table full or half-full) Interrupt routine. This routine services DDAS log tables on the interrupt level. The test program must indicate it is expecting an interrupt by setting bit 23 to a one. Bits 22-15 are not used. Control is given to the DDAS Log Mode Interrupt routine for test programs by a TSP instruction. This routine then returns control to the Operating System via the address just stored by TSP instruction.
- WORD 5 - Bits 14-0 must contain the address of the area reserved for data input from the LVDC. Bit 23 is set when data has been entered into this area. This bit must be reset by the test program. Bit 22 of the word is set to a one each time an error message from the LVDC is received. This bit must also be reset by the test program. This cell is used to store LVDC data at the address specified. Bits 21-15 are not used.
- WORD 6 - This is a BCD word stating the hardware power necessary to execute this program. Each character indicates the status of a piece of equipment. If the character is a 0 or 1, power should be off or on, respectively. If the character is a ½ (blank), power may be on or off and does not affect the check.

  Each character position is associated with a particular unit. They are:

  Bits 23-18 - LVDC
  17-12 - ST-124M Platform
  11-6 - Control Computer
  5-0 - Control Computer Interface Unit

  This word indicates to the Operating System the environment checks that should be made prior to execution of the test program.

- WORDS 7-10 - A NOGO capability is provided where the test program can have the Operating System display options to terminate, restart, continue, or retry the test program. These words contain the address needed to transfer control when an option is chosen by the test engineer (refer to paragraph 2. 2. 1. 2 for a more detailed description of this procedure.)
- WORD 11 - This word contains all card keys allowed to execute this program. The master card key can execute any program and need not be defined in this word. A maximum of four card keys are allowed. If less than four card keys are defined, they must be right-justified.

BIT NUMBER				WORD NUMBER		
23	15 14		0	OCTAL	DECIMAL	RELATIVE
JRA INSTRUCTION TO START OF TEST PROGRAM				0	1	IVAR
			ADDRESS OF FORCED TEST END ROUTINE	1	2	IVAR+1
*			ADDRESS OF PIT NO. 1 INTERRUPT ROUTINE	2	3	IVAR+2
*			ADDRESS OF DDAS LOG MODE INTERRUPT ROUTINE	3	4	IVAR+3
*	* *		ADDRESS OF LVDC DATA INPUT BUFFER	4	5	IVAR+4
BCD WORD FOR POWER CHECKS				5	6	IVAR+5
			ADDRESS OF ABORT ROUTINE	6	7	IVAR+6
			ADDRESS OF RESTART ROUTINE	7	8	IVAR+7
			ADDRESS OF CONTINUE ROUTINE	10	9	IVAR+10
			ADDRESS OF RETRY ROUTINE	11	10	IVAR+11
CARDS KEYS ALLOWED TO EXECUTE THIS PROGRAM				12	11	IVAR+12
				13	12	IVAR+13
			3-WORD (12-CHARACTER) DESCRIPTIVE NAME OF TEST PROGRAM	14	13	IVAR+14
				15	14	IVAR+15
			ADDRESS OF INITIAL OPTIONS AND VALIDATION MASK	16	15	IVAR+16
***			AUTOMATIC EXECUTION INDICATOR	17	16	IVAR+17
				20	17	IVAR+20
			RESERVED FOR FUTURE USE	21	18	IVAR+21
				22	19	IVAR+22
				23	20	IVAR+23

* IN PROCESS INDICATOR - BIT 23
** ERROR MESSAGE INDICATOR - BIT 22
*** PSC INDICATORS - BITS 23 AND 22

Figure 2.1.2 Test Program Communication Cells

- WORDS 12-14 - These words contain a 12-character descriptive name of the test program.
- WORD 15 - Bits 14-0 contain the address of initial options to be displayed immediately before execution of the test program. If this word is zero, it is assumed no options are defined, and a message to execute the test program is then displayed by the Operating System. If the word is not zero, the initial options are defined and displayed.
- WORD 16 - This word is assembled with bit 23 in the ON state if the test program can be run under ATOLL PSC. If the test program uses either the test program drum scratch area (SSSS, SSST, or SSSU) or bank 7, bit 22 is assembled in the ON state; this indicates that the ATOLL Executive must be reloaded after execution of this test. If bit 21 is set to the on state when the program is assembled, the test program will automatically begin execution without its initial options being displayed. Test Program Control places PX (in BCD) into the test program parameter input buffer area.
- WORDS 17-20 - Reserved for future use.

NOTE

If the communication function for each cell does not apply to the test program, the cell should be set to zero.

2.1.4 MDO Discrete Disallow Table

Each test program operating in the MLC must define an MDO Discrete Disallow Table. This table is assembled separately from the program with which it is associated and is transferred to the MLC during execution. The table defines MDO's that are not to be issued by the program itself.

NOTE

Function Executor routines do not have MDO Discrete Disallow Tables.

The MLC has a maximum of 2016 MDO signals. This MDO Discrete Disallow Table consists of 84 data words containing a total of 2016 bits. Each bit represents a specific MDO. If the bit in a specific position is set to 1, this indicates that the associated MDO is not allowed to be issued by the test program in execution. If the bit position contains a 0, this indicates that the associated MDO may be issued. Figure 2.1.3 shows the structure of the MDO Discrete Disallow Table. Notice that in each word the low-order bit positions correspond to the smaller discrete numbers for the group.

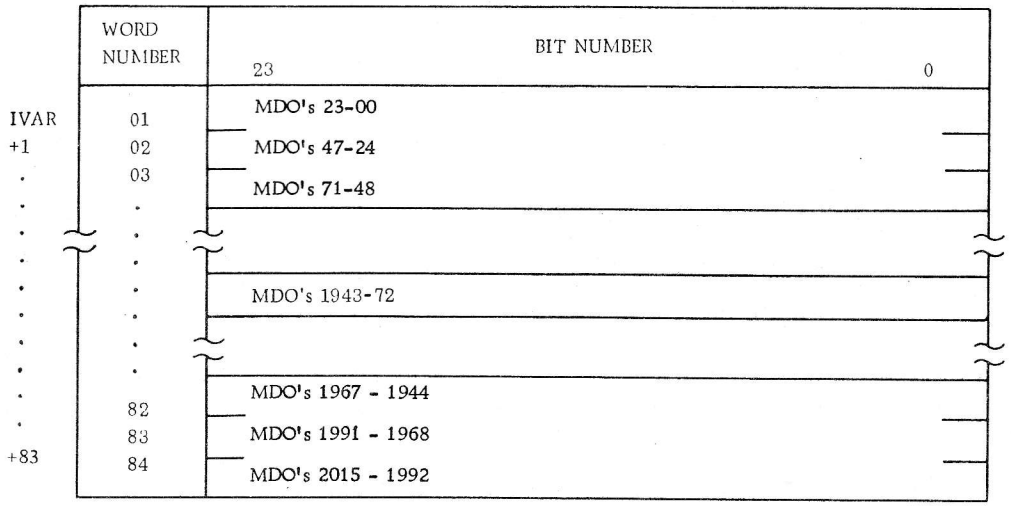


Figure 2.1.3 MDO Discrete Disallow Table

2.1.5 Display Console Key Code

The purpose of the display console key code is to allow an operator to communicate with specific test programs. The programmer places these codes in the eleventh (11) word of the test program communication cells. A maximum of four codes may be specified (see Figure 2.1.4). Since there are two pages, A and B, associated with each console, it is suggested that the programmer code this cell so that only page A be allowed to communicate with the program. The low-order bit position of each code is used to indicate page A of the console.

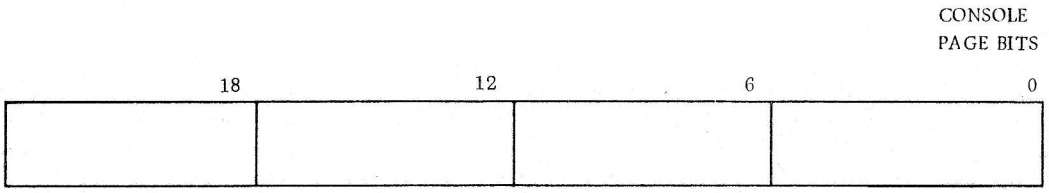


Figure 2.1.4 Console Key Coding

The binary value of this bit must be 0 to indicate page A when specifying the console codes given in Table 2.1-3.

Table 2.1-3 Console Key Codes

Console	Key Code	Console	Key Code	Console	Key Code
1	2	6	14	11	26
2	4	7	16	12	30
3	6	8	20	13	32
4	10	9	22	14	34
5	12	10	24	15	36

When a request is made to load and execute a test program, the key code of the test console and the test program name are transferred to Test Program Control. Test Program Control then compares the code of the console with the codes specified by the test program. If these codes compare, the test program is loaded into core and executed. If none of the specified codes compare, the request is automatically cancelled.

Any card key may be used to execute any Function Executor routine; however, when a routine is being executed by one card key, only that card key or the master card keys may alter or terminate the execution of the routine. A cell in bank zero (CME04) contains the card key code associated with the routine in execution. This code is in bits 18 through 23; the remaining bits will be zero.

2.1.6 General Storage

The Operating System provides test programs and Function Executor routines storage areas in bank zero of both computers for various essential functions. These functions are:

- Test Program Intercommunication
- Test Program linkage with Operating System
- Storage of Constants
- Storage of Control Data
- Storage of Parameters

Test program intercommunication is necessary because only one test program operates in the LCCC at one time, and because the current test program may have data for the next test program. The current test program would normally create a storage area in bank zero and place the data there. When the next test program is read in, it may access this data. There are 100 storage locations reserved in bank zero for this purpose, which are assigned to test programs by the Operating System group.

Portions of bank zero are also saved for test program linkage with the Operating System. The IVAR's of the Operating System routines are given in Appendix B.

Storage area is also reserved in bank zero for constants and parameter input areas. These areas may be used by all test programs. The IVAR's are given in Appendix C.

2.1.6.1 Special Counter

A special binary counter is provided by the Operating System for the time-out routines to use. This counter is set to zero at system initialization and is incremented every second. This time is in bank zero communication cell CCS50.

NOTE

For a more accurate program-system timing relationship, relative time should be accessed.

2.1.7 Linkage for Programs and Routines

One program (or routine) accesses another by a fixed address in bank zero which is assigned at assembly time. This address is provided by the Assembly Pool equate cards containing constants or absolute address locations. An absolute address (specific entry points), such as starting address of the program or routine, is referred to as an IVAR. This method of linkage is used by programs and routines.

NOTE

Any IVAR is called an XVAR when accessed by another program; XVAR indicates the program being used as a subroutine.

Locations have been reserved in bank zero for linkage with the Operating System. Programs and routines use a TSPO instruction indirect to link with various Operating System routines. The operand of the TSPO instruction is the Operating System routine IVAR. Transfer is then made to the

address stored in the bank zero IVAR location. This address is the first instruction of the desired Operating System routine. A list of Operating System routine IVAR's is given in Appendix B.

2.1.7.1 Test Program Linkage

When a test program is requested, it is loaded and executed by the Operating System. The test program may access various Operating System programs resident in the LCCC by using the calling sequences defined in Appendix A. The Operating System routines in the LCCC that may be used by test programs are given in Appendix B.

2.1.7.2 Function Executor Routine Linkage

When a function executor routine in the MLC is requested, the Operating System accesses the first communication cell of that routine and determines where its first instruction is located. Control is then transferred to this routine and it is executed. When the routine is operating, it may access those Operating System programs resident in the MLC by using the appropriate calling sequences appearing in Appendix A. When the function executor completes execution, it must return control to the Operating System with the transfer instruction TSPO SMFR, 7.

If a function executor routine is terminated, the termination is always recognized on a line 5 interrupt, and the Operating System accesses the address of the forced test end subroutine from the second communication cell of the routine. The Interrupt Processor transfers control to the forced test end subroutine after processing the line 5 interrupt. Housekeeping is performed, and the subroutine returns control to the Operating System.

The Operating System routines that a function executor routine may access are given in Appendix B.

2.1.8 Interrupt Processing

If the Operating System interrupts a test program or function executor routine, the system stores the contents of all six index registers and the left and right accumulators. After the interrupt has been serviced, these registers are restored and control is returned to the point in the program or routine of the interrupt. The test program or Function Executor routine is unaware that the interrupt has occurred.

NOTE

A test program must not use the following interrupt instructions:

- Activate Interrupts (ACI)
- Inhibit Interrupt (INI)
- Return After Interrupt (RAI)

2.1.8.1 Programmable Interval Timer (PIT) No. 1

Pit No. 1 is used by the test programmer for timing test programs in the LCCC. If a test programmer uses this timer, a PIT No. 1 subroutine must be provided within the test program. The address of the first instruction word of this subroutine must be placed in the third word of the test program communication cell.

When PIT No. 1 is used, the test program must set bit 23 of the subroutine address word in the communication cell. When the interrupt occurs, the Interrupt Processor checks bit 23. If it is a

one, control is then transferred to the subroutine. If a PIT No. 1 interrupt occurs, and bit 23 was a zero, an error condition occurs and a message is printed and displayed.

NOTE

Since all other system interrupts are inhibited during the PIT No. 1 Interrupt subroutine, its execution time must be of short duration.

Test programs must not use PIT Nos. 2 and 3.

2.1.9 Test Subroutines

A test program subroutine is a separate portion of the test program that resides on tape and is called into core by the test program when it is needed. They only overlay portions of the LCCC test programs in banks six and seven, except for the communication cells. The size of a subroutine may be up to a maximum of 4092 words. These subroutines do not contain Operating System communication cells, and therefore must be called and controlled by the associated test program.

When a test program is ready to use a subroutine, it transfers control to the Operating System with the macro instruction LOAD and its calling sequence, TSPO SUBR, 7 (Table A-6, Appendix A). A TSPO SUBR, 7 is used to give control to the system. The IVAR of the desired test subroutine is specified in the control word of the calling sequence. Figure 2.1.5 shows the format of the control word.

When the test subroutine has been loaded, either an error or a normal return is made. If the test subroutine cannot be loaded into core, the Operating System returns control to the error return. If the test subroutine has been loaded, control is transferred to the normal return.

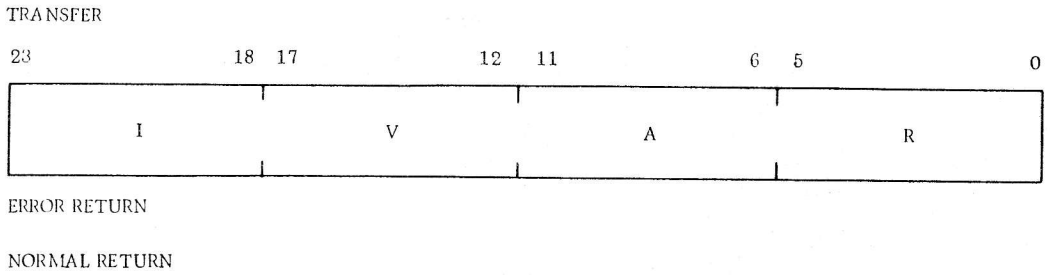


Figure 2.1.5 Load Macro Control Word

2.1.10 Forced Test End Subroutine

All test programs and function executor routines are required to contain a forced test end subroutine. This subroutine performs necessary housekeeping tasks when a test program or routine is terminated. This termination may be due to a condition found by the test program Operating System, or by a display console keyboard request. When a test program is to be terminated, the Operating System uses word 2 of the communication cells to determine the address of the forced test end routine. Control is then given to this subroutine. After the housekeeping tasks are performed, control is returned to Test Program Control by the program or routine.

2.1.11 Function Executor Routine Communication Cells

Function executor routines must provide communication cells similar to those of the test programs. The test programmer reserves the first ten locations in his routine as communication cells to interface with the Operating System resident in the MLC. The structure and contents of these cells are shown in the following figure.

BIT NUMBER 23 15	BIT NUMBER 14 0 0	WORD NUMBER OCTAL	WORD NUMBER DEC.	WORD NUMBER REL.
JRA	ADDRESS OF FUNCTION EXECUTOR ROUTINE	0	1	$IVAR
	ADDRESS OF FORCED TEST END SUBROUTINE	1	2	+1
	FUNCTION EXECUTOR NAME IN BCD	2	3	+2
	REINITIALIZATION INDICATOR	3	4	+3
	ADDRESS OF LVDC DATA INPUT BUFFER	4	5	+4
	RESERVED FOR FUTURE USE	5 6 7 10 11	6 7 8 9 10	+5 +6 +7 +8 +9

Figure 2.1.6 Function Executor Routine Communication Cells

For a function executor routine to display data via the Operating System's Remote Display and Print Service routine of the System Message program, a card key code of zero must be indicated in the control words. The message will then be displayed on the Display Console that requested the routine.

2.2 TEST PROGRAM CONTROL PROCESSING

The general processing accomplished by Test Program Control in loading and controlling the execution of a test program requires the test programmer's attention in several areas of development. These areas include:

- Option Capabilities
- Returns to Test Program Control
- Environment Checks
- General Storage Area

The following paragraphs describe four classes of options for the test programmer, three types of returns to Test Program Control after or during program execution, hardware power checks that may be required for program execution, and utilization of the contents of general storage locations.

2.2.1 Options

Test Program Control provides the capability to initiate the display of certain classes of options. Each class contains a number of different options. An option allows the test engineer to enter data via the keyboard in response to a displayed message specified by the Operating System or the test program. The messages types may take three forms. They are:

- Test program operation status
- Request for parameters
- Request for performance of specific program steps

The message describing the test program operation status would be Terminate, Restart, Continue, or Retry. The message describing a request for parameters may take the form of a descriptive message asking for a number of different constants, such as voltage limits, etc. The message describing a performance of a specific test program step would be either automatic or manual mode.

If a decision is required by the display console operator during a test function, the appropriate class of options can be displayed by the test program. These options are classified as:

- Initial Options
- Standard Options
- Test Programmer Options
- Parameter Options

A test programmer may request a display of these options either prior to or during the execution of his program. Test Program Control accepts the request and displays the class of options specified.

When the options are displayed, the operator selects the desired option by typing in the appropriate keyboard sequence. When the option is keyed in, the test program continues processing.

Test Program Control provides a MACRO instruction for each class of options. These instructions and their calling sequences are given in Table A-6, Appendix A.

2.2.1.1 Initial Options

Initial options are defined as those options that are required prior to the execution of a test program. These initial options may be structured by the test programmer in the same format as parameter options (refer to paragraph 2.2.1.4). If initial options are required, their address is specified in the fifteenth word of the test program communication cells (see Figure 2.1.2).

After a test program has been loaded into core, Test Program Control always checks this communication cell prior to execution of the program. If the cell is non zero, Test Program Control then displays the specified initial options.

When the options have been displayed, Test Program Control waits for the operator to enter the required options. When they are entered, Test Program Control validates them if required, and executes the test program.

If these initial options are not required, the test programmer should place a zero in this communication cell word. Test Program Control then displays the standard initial option: 'PROGRAM XXXX LOADED, TO EXECUTE ENTER $PX'.

2.2.1.2 Standard Options

Test Program Control provides the test programmer with the capability to display a class of standard options, standard options being those which are predefined by Test Program Control. If these options are displayed, the display console operator is expected to select one and enter his choice. The test programmer may choose to display these options for error conditions or any situation for which they are appropriate.

The choice of options in this class are:

- ENTER A TO TERMINATE
- ENTER B TO RESTART
- ENTER C TO CONTINUE
- ENTER D TO RETRY

Test Program Control provides the capability to display certain combinations of the above options. The possible combinations are A and B, AB and C, AB and D, or ABC and D. Notice that the A and B options are always displayed and the choice is for C and/or D. The means by which these combinations are selected is via the calling parameters XX of a MACRO instruction. This MACRO instruction and the calling sequence of instructions is given in Table A-6 of Appendix A. The instruction is identified by the NOGO MACRO name. The calling parameters XX are coded as either a C/or a D. If neither C nor D is desired, the XX parameters are left blank (bb).

If a test programmer chooses to display the standard options, he must provide processing subroutines for each option. The addresses of the first instruction of each subroutine are placed in the seventh through the tenth word of the communication cells. The seventh word contains the address of the first instruction word for the option A subroutine. The eighth word contains the address of the first instruction word for the option B subroutine, and so on (see **Figure 2.1.2**).

The coding for these cells could be a pseudo operation code of PZ5 and a symbolic name of the option subroutine.

When these options are displayed, the test engineer makes a selection by entering $PY. Y is the chosen option A, B, C, or D. Test Program Control recognizes the option and accesses the subroutine address contained in the communication cell. Control is then transferred to the appropriate subroutine.

2.2.1.3 Test Programmer Options

Test Program Control provides the capability for test programmers to specify various options to be displayed. These options are not predefined by Test Program Control; they are defined by the test programmer. These options have a format which consists of a header message followed by a maximum of four options.

The test programmer should specify these options as follows:

```
 H---------------H
*ENTER A TO XX----X
 ENTER B TO XX----X
 ENTER C TO XX----X
 ENTER D TO XX----X
```

HH----H is the header message and XX----X is the specified option

*The text for each option entry is limited to a maximum of 28 characters (7 words)

The test programmer may choose to request this class of options when a special condition occurs during the test function requiring a decision by the display console operator. The header message may contain information about the condition that occurred.

The header message and options are specified by a MACRO instruction given in Table A-6 of Appendix A. The name of this MACRO is OPTION. Since this class of options is completely specified by the test programmer, communication with Test Program Control is provided in the calling parameters. The calling parameters specify the address of the header message, the option identifiers, and information about the option.

The address is symbolized by the calling parameter HEADR. The option identifiers are A, B, C, and D. To display any number of these, the test programmer codes a one for the options to be displayed. If the option is not to be displayed, a zero is coded.

Test Program Control requires certain information about these options, as follows: the number of words in the option message, the address of the option message, and the address of the subroutine which performs the processing for the particular option. This information is supplied by the test programmer in a table. The address of the table is given in the MACRO instruction. The structure of this table is shown in Figure 2.2.1. The first option identifier (A) is associated with the first two words of the table. The second option identifier (B) is associated with the second two words of the table, and so on.

If a test programmer specifies this class of options to be displayed, Test Program Control looks at the address of the header message, and then accesses and displays the header message. Test Program Control then checks the option identifier to see which are to be displayed. If the identifier is coded non-zero, Test Program Control looks at the appropriate address, and next accesses and displays the option message.

When the options have been displayed, the test console operator selects an option by entering $PY, where Y is the option selected. Test Program Control accesses the corresponding address for the entered option and transfers control to the appropriate subroutine.

2.2.1.4 Parameter Options

Test Program Control provides the test program with the capability to display options requesting parameter inputs. This class of options may be used to request discrete numbers or some other type of parameters needed to continue processing. This request can be made during the test function.

Test Program Control provides a MACRO instruction to initiate a display of this class of options. This MACRO instruction is given in Table A-6, Appendix A. The MACRO name is DSPLY and the calling parameter is ADRS. This calling parameter specifies the storage address for the parameter option data and is contained in the calling sequence control word. Figure 2.2.2 shows the format of this control word and the general structure of the core locations which contain the parameter option data.

When the request for a display of parameter options is made, Test Program Control accesses the message, initiates the display, and waits for the parameter to be entered. The console operator enters the parameters by typing in $PZ-------Z where Z-------Z specifies the parameters. A maximum of 80 parameters is allowed. These parameters may be validated if the test programmer so desires. If the parameters are to be validated, control words for this validity check follow immediately after the parameter option message.

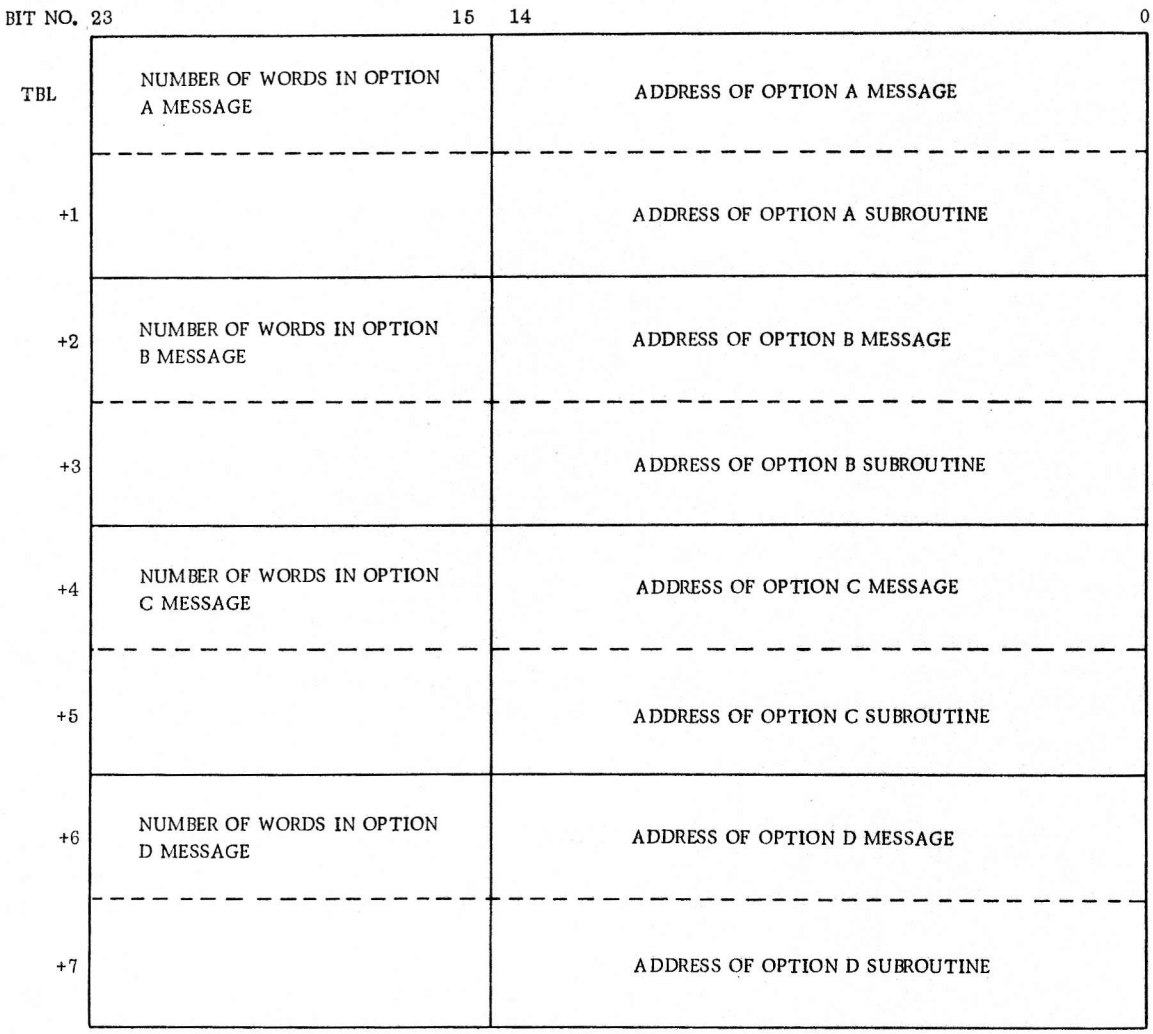


Figure 2.2.1 Table Structure for Test Program Options

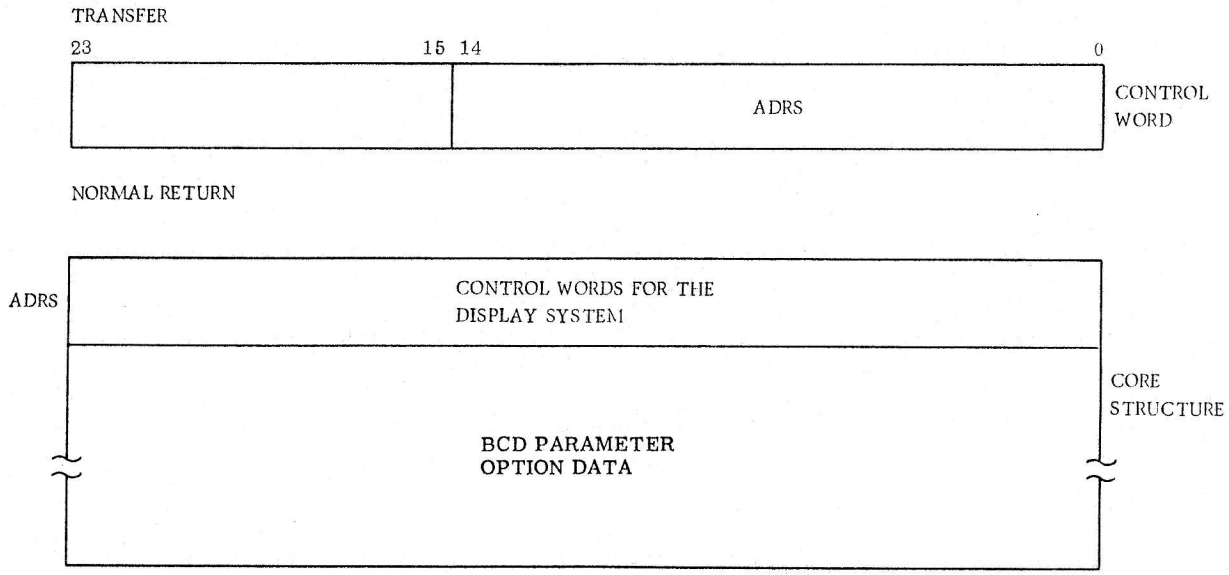


Figure 2.2.2 Parameter Option Control Word and Core Structure

2.2.1.4.1 Parameter Validity Check: There are two types of validity check control words. One type of control word contains the maximum number of input parameter characters expected, and immediately follows the message to be displayed. The second type of control word is the BCD validity code mask word which contains codes used to determine the valid parameter input characters. There may be as many mask words as required, depending on the number of parameter inputs. Figure 2.2.3 shows an example of the core structure and coding for validity check control words.

BCD validity code mask words are structured by the test programmer. The series of BCD validity code mask words may contain various codes as follows:

0-Anything valid
1-Alphabetic
2-Binary
3-Octal
4-Decimal
5-Alphanumeric
6-Plus or minus sign
7-The letter X only
8-Decimal point
9-Blank or no entry

A series of these codes are placed into a code word separated by a space. Each series of codes between spaces would allow the corresponding parameter character to be transmitted to the parameter area. The first series of codes corresponds to the first parameter character and so on. A space between the series of validity codes indicates the end of the codes for a parameter character. The structure of core and example of coding is given in Figure 2.2.3.

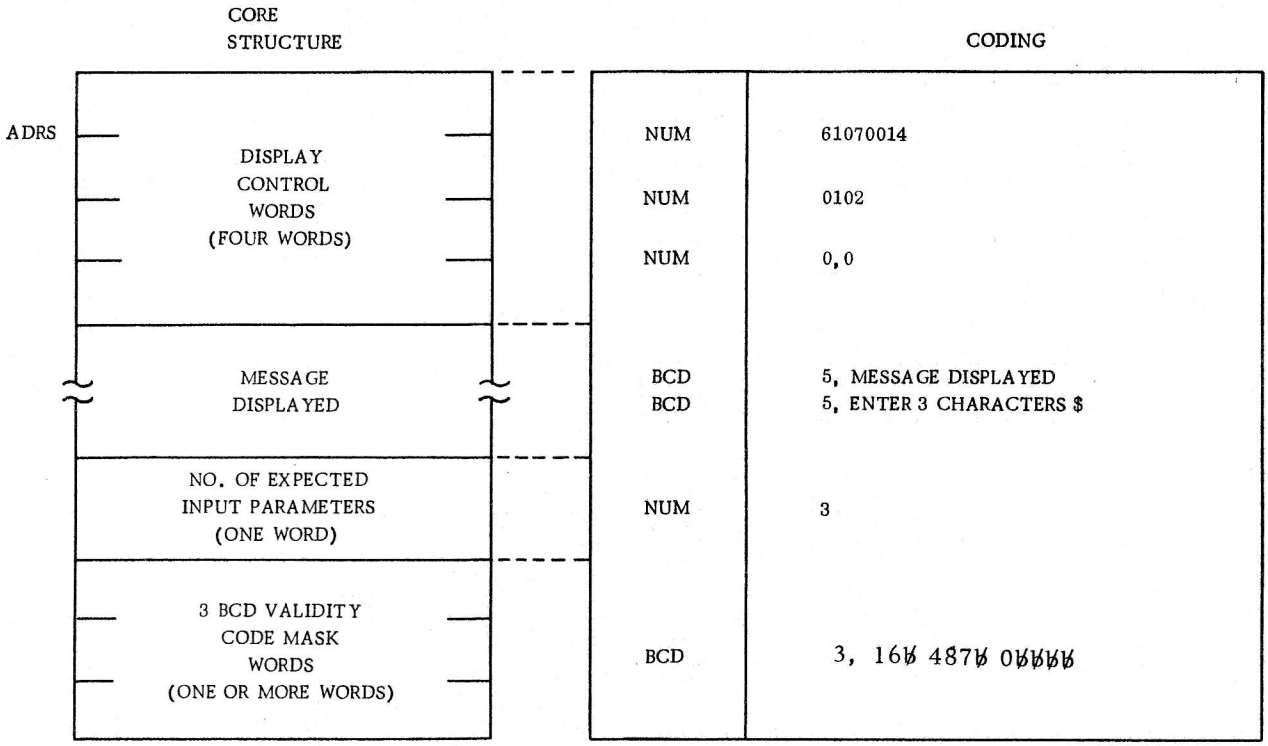


Figure 2.2.3 Parameter Validity Check

The example given indicates in the first code line that there are 14 words of data in the displayed message including the control words. The example specifies that a maximum of three characters are to be entered by the test engineer. If more than 3 characters are entered the input parameters are considered invalid. It is the responsibility of the test program to determine if the number of input parameters is valid. The last three words in the coded example are the mask words. The 16 indicates that the first character entered can be an alpha or a plus or minus sign. The 487 indicates that the second parameter character can be a decimal number, decimal point, or the letter X. The 0 indicates the third character can be anything. The blank (ß) indicates the end of the series of mask words. If the test programmer does not want the parameter validated, the word following the message to be displayed must be zero.

2.2.1.4.2 Parameter Storage: Parameter storage is an area that has been reserved in bank zero of both computers for input data that has been entered via the console. Twenty-one words are reserved. The entire storage area structure is shown in Figure 2.2.4. The first word specifies the number of computer words that contain parameter characters. The next 20 words, although all may not be used, are reserved for the parameter characters. The letter P shown in Figure 2.2.4 is a control letter indicating that parameter characters follow, and should always be specified. Twenty words can accommodate a maximum of 80 BCD characters.

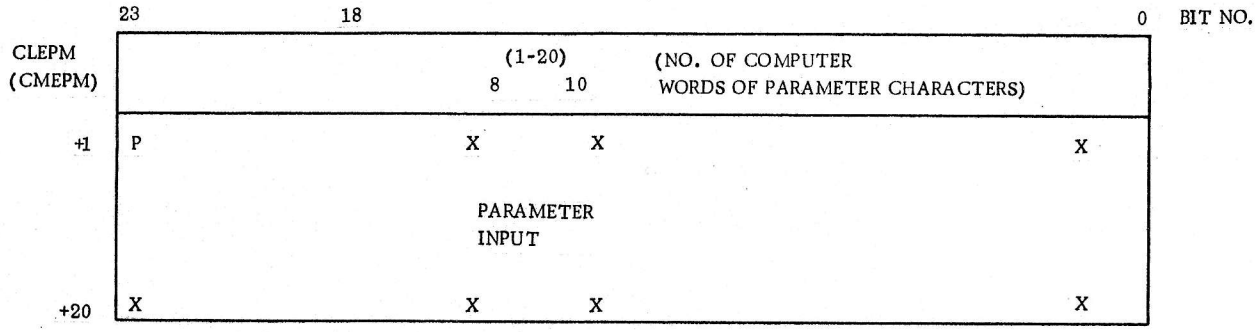


Figure 2.2.4 Parameter Storage Area for the LCCC and the MLC

A test programmer may access these parameters by using a bank zero load instruction. The operand of this instruction is the storage area IVAR plus the number for relative addressing. The storage area IVAR is different for each computer. The parameter storage area in the Launch Control Center Computer (LCCC) has the IVAR CLEPM; for the Mobile Launcher Computer (MLC), the IVAR is CMEPM.

P3XX 0000001

2.2.2 Returns to Test Program Control

Test Program Control provides returns to the Operating System for test programs operating in the LCCC. Test Program Control provides entry points when the following test program events occur:

- Self-termination and execution of another test program
- Abnormal termination
- Normal completion of the test function

Test Program Control provides a macro instruction for each type of return. These instructions and their calling sequences are given in Table A-6, Appendix A.

2.2.2.1 Terminate and Execute

A test program may terminate itself and execute another test program by passing control to Test Program Control entry point ST16 (see Table A-6, Appendix A.) The name of the new program to be executed is used in the calling sequence.

The self-terminating program may be either a test program or an ATOLL procedure, and may execute either another test program or an ATOLL procedure.

NOTE

If the self terminating program or ATOLL procedure has already been terminated, the request for terminate and execute will not be recognized.

2.2.2.2 Normal Completion

When a test program normally completes its test function, it usually returns control to Test Program Control. The entry point is identified as ST18. Test Program Control then displays a message that the test function is completed.

2.2.2.3 Abnormal Termination

An abnormal termination is defined as a cancellation or termination which has been caused externally either by Display Console inputs or program environment. Some possible causes for abnormal termination are:

- Program unable to be loaded
- Console termination via $TEND entry
- Console termination via $TERM entry
- Invalid card key codes requesting programs
- Switch Selector Function conflict
- Control Computer Interface Unit and the Control Computer not in the required state
- ST-124 Platform and LVDC power not in specified state
- Data Link down

If a test program is executing, and any of the above occur (except TEND), Test Program Control accesses the address of the test program's forced test end subroutine and transfers control to it. After housekeeping tasks are performed, the test program returns control to Test Program Control via ST19, using the transfer instruction TSPO ST19, 7. If a test program was not executing and any of the above occur, control is passed directly to Test Program Control via ST19. If a test program is terminated via a $TEND, the test program's forced test end routine is not entered and control is not returned to the Operating System but is retained by Test Program Control.

2.2.3 Environment Check

Test Program Control provides the capability to check the operating environment of a test program prior to execution. This environment check consists of a power on/off check of various hardware units. If a test programmer is not concerned with a specific hardware unit, he indicates this and Test Program Control will not make the check. The hardware units that are checked for power on/off are:

- Control Computer Interface Unit
- LVDC
- ST - 124M Platform
- Control Computer

When a power check is made of these units, Test Program Control makes use of the Data Link and DDAS. If these two units are not operable, a message to this effect is displayed.

2.2.3.1 Power Check Communication Cell Coding

For a hardware power check to be made, the test programmer codes the sixth word of the test program communication cell. If no checks are to be made, this entire cell should be zero. The following codes used for power checks are, in BCD:

0 - Power should be off
1 - Power should be on
ϕ - Power condition irrelevant

When a check is made for a specific hardware unit, the code is placed into the bit positions specified for that unit. Figure 2.2.5 shows the format for the power check communication cell.

If the results of the power check are not as specified, the test program is cancelled.

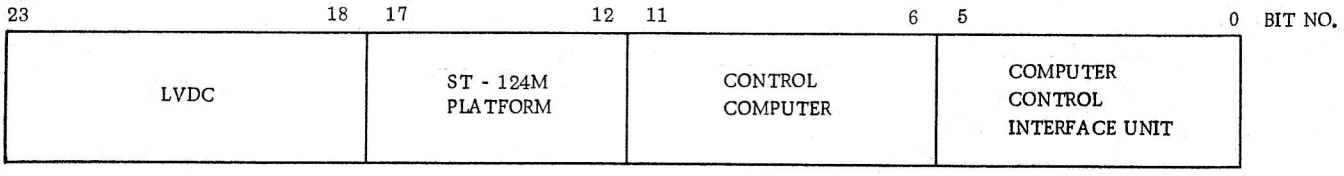


Figure 2.2.5 Power Check Communication Cell

2.2.4 Test Program Requests for Function Executor Routines

The Operating System provides the capability for test programs operating in the LCCC to request the execution or termination of a function executor routine resident in the MLC. The Function Executor Control program has a subroutine resident in the LCCC to process and transfer these requests to the MLC. Both requests use a TSPO SMFX, 7 to link with the function executor subroutine.

The macro name for each request is unique. The assembler inserts this unique name into the first control word of the calling sequence. The macro call instructions and the calling sequences are given in Table A-6, Appendix A.

2.2.4.1 Execute a Routine

When a test program requests a function executor routine to be executed, it must specify five parameters in the calling sequence control words as shown in Figure 2.2.6. The first control word is inserted by the assembler to indicate that the request is an execution request (FUNC). If this word is omitted or if there is an error within it, a Halt Interruptable instruction is inserted in the test program calling sequence and transfer is made to it. The five parameters are as follows:

- Routine IVAR
- Time Intervals (seconds)
- Number of Parameter Words
- Parameter Address
- Return Code

The routine IVAR is the 4 BCD character name of the routine that is to be executed. This is specified in the second control word.

The time interval (in seconds) is the time lapse between routine re-execution. This time must be specified in seconds and have three BCD digits. The limits of the time are $010_{10} <$SSS $< 999_{10}$. If the routine occurs once, then this parameter is BCD zeroes. The time parameter is specified in the third control word.

The number of parameter words indicates the number of RCA-110A core locations required to contain the parameters given to the routine for execution. This is specified in the fourth control word. The maximum number of parameter control words is 20.

The parameter address is the core location of the first word containing the parameters. This is specified in bits 14 through 0 of the fifth control word.

The return code indicates whether Function Executor Control is to return control to the test program after the routine has started execution, or immediately after transmitting the request across the Data Link to the MLC. This code is specified in bits 17 through 15 of the fifth control word. If this field is an octal 1, then control is to be returned after the request has been transmitted to the MLC. If this field is an octal 0, then control is to be returned after the routine has received control.

When the function executor request has been processed, or when the request has been sent to the MLC (depending on the return code), control is returned to the test program at the core location following the control words. The return information is placed in the left accumulator. The contents of the left accumulator and their significance are:

+ = Request cannot be executed
- = Routine has begun execution
0 = Request has been transmitted to the MLC

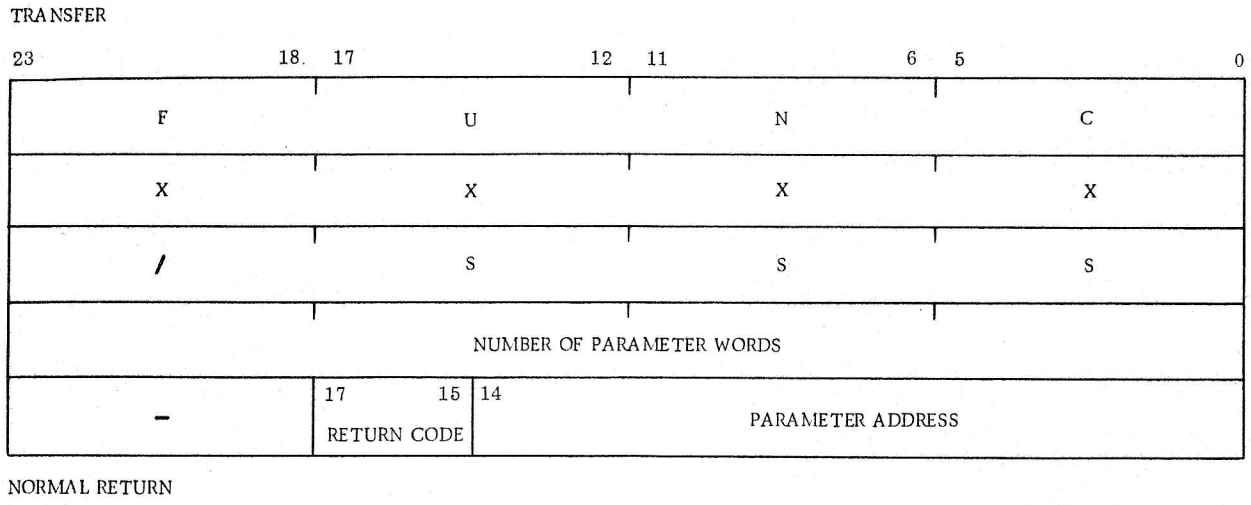


Figure 2.2.6 Execute Control Words

2.2.4.2 Terminate a Routine

A test program may terminate a function executor routine while it is currently in operation. The test program links to the function executor and specifies only one parameter, which is the IVAR of the routine to be terminated. The parameter is specified in the second control word of the calling sequence as shown in Figure 2.2.7. The first control word of the sequence contains the BCD characters FUNT. If this word is omitted or if there is an error within it, a Halt Interruptible instruction is inserted in the test program calling sequence and transfer is made to it.

When Function Executor Control stacks the termination request, it returns control to the core location following the IVAR control word. The left accumulator contains the status of the request. If the routine has been terminated, the sign bit (23) of the left accumulator is positive (0). If the routine has not been terminated, the sign bit (23) of the left accumulator is negative (1).

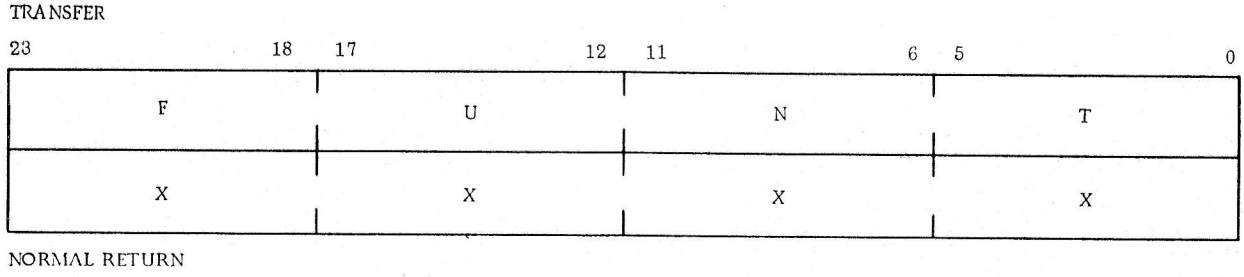


Figure 2.2.7 Terminate Control Words

2.3 PROGRAM LOADING AND EDITING

Once a test program is initially coded it is assembled by one of three assembly systems: SLAP 2, CHAPS or SOAR. SLAP 2 assembles programs on the RCA 110A computer; CHAPS assembles programs on the GE235 computer; and SOAR assembles programs on the IBM/360 computer.

The System Tape Arranging, Generating, and Editing program (STAG) is used to generate a system master test tape. Inputs to this program are the output binary cards, binary tapes, or a combination of both from the three assembly programs. Figure 2. 3. 1 shows the system flow for the generation of a system tape.

The Operating System also provides patching capabilities for test programs. Patching instructions are via keyboard and card input while the test programs are on-line.

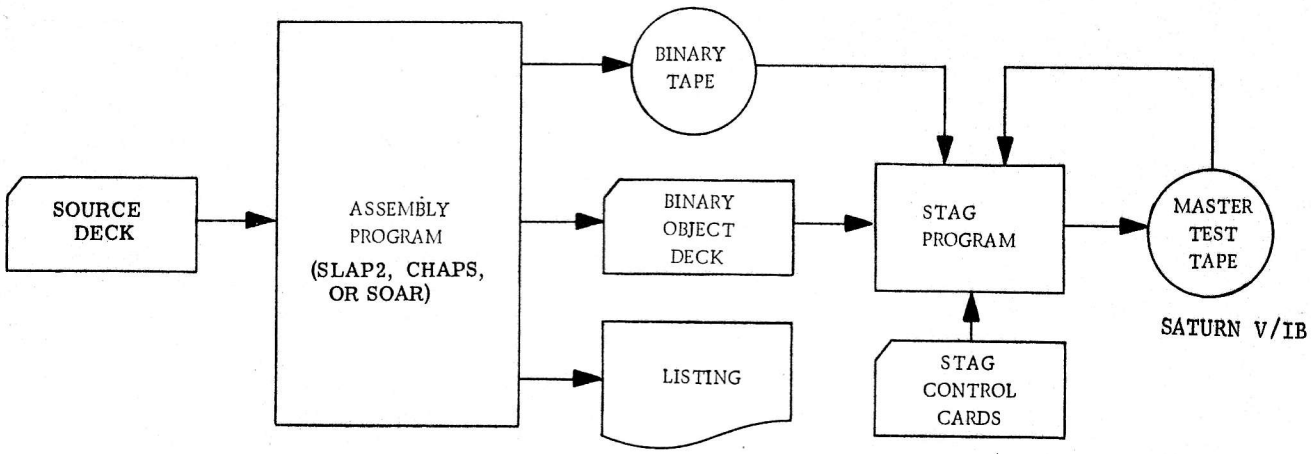


Figure 2. 3. 1 Test Program Master Tape Generation

2.3.1 Test Program System Master Tape

The Saturn Operating System provides the capability for test programmers to create and/or edit a Saturn Test Program System master tape. This tape normally contains all the vehicle test programs, and on-line diagnostic test programs which operate in the LCCC. When a request to load and execute a test program occurs, this system test tape is accessed by the Operating System and a search is made for the test program. This tape is to be resident on Magnetic Tape Station No. 2 which is attached to Input/Output Data Channel No. 1.

Each file of the test tape contains one complete test program. Within the file is a series of data blocks defining the test program. Each series of data blocks contains segments of the program which are grouped according to their location in memory (MLC, LCCC drum, LCCC core, LCCC tape). Each series of data blocks must be placed on the tape in the following order:

a. MLC (MDO Disallow Table)
b. LCCC Drum
c. LCCC Core
d. LCCC Tape

The system master test program tape format is shown in Figure 2.3.2. The first file contains a dictionary. This dictionary contains a file name and the file number for every test program existing on the tape. The file name is the test program IVAR. The file number indicates in which file the test program resides. This dictionary is read in and placed on drum. When a request for test program execution is made, the dictionary is searched to determine if the program request is valid.

Each test program file contains a group of blocks which contain all segments of the test program that is to reside in the same storage media. Thus, each file may contain as many as four groups of blocks. The first group contains the MDO Disallow Table. The second contains all segments which are to reside on the LCCC drum; the third contains all segments which are to reside in the LCCC core; and the fourth contains all the test subroutines which are to remain on tape.

Each block in the group usually contains a header. This header contains data as shown in Figure 2. 3. 2. Each block contains a specific segment of the program, which is usually a specific routine within the program. The last block of data in the file is a reference block which also contains a header. The data contained in the reference block is also shown in Figure 2. 3. 2.

2.3.2 Card Input

To assemble, generate, and edit a Saturn Master System Test Tape, certain control cards are necessary. There are three card types: INPUT, STAG, and END cards.

The INPUT card describes the type of input expected for the STAG program. This input is either binary cards, binary tape, or a combination of both. This input is the output from the SOAR, CHAPS or SLAP 2 assembler runs. Only one INPUT card is required for each STAG run. If there are two such cards, the second card takes precedence and the input associated with the first is deleted. In the case of a duplicate object program, the program that appears first is processed. The INPUT card also indicates whether an edit run is to be made (EDIT) or whether a block of program data should be printed (DUMP).

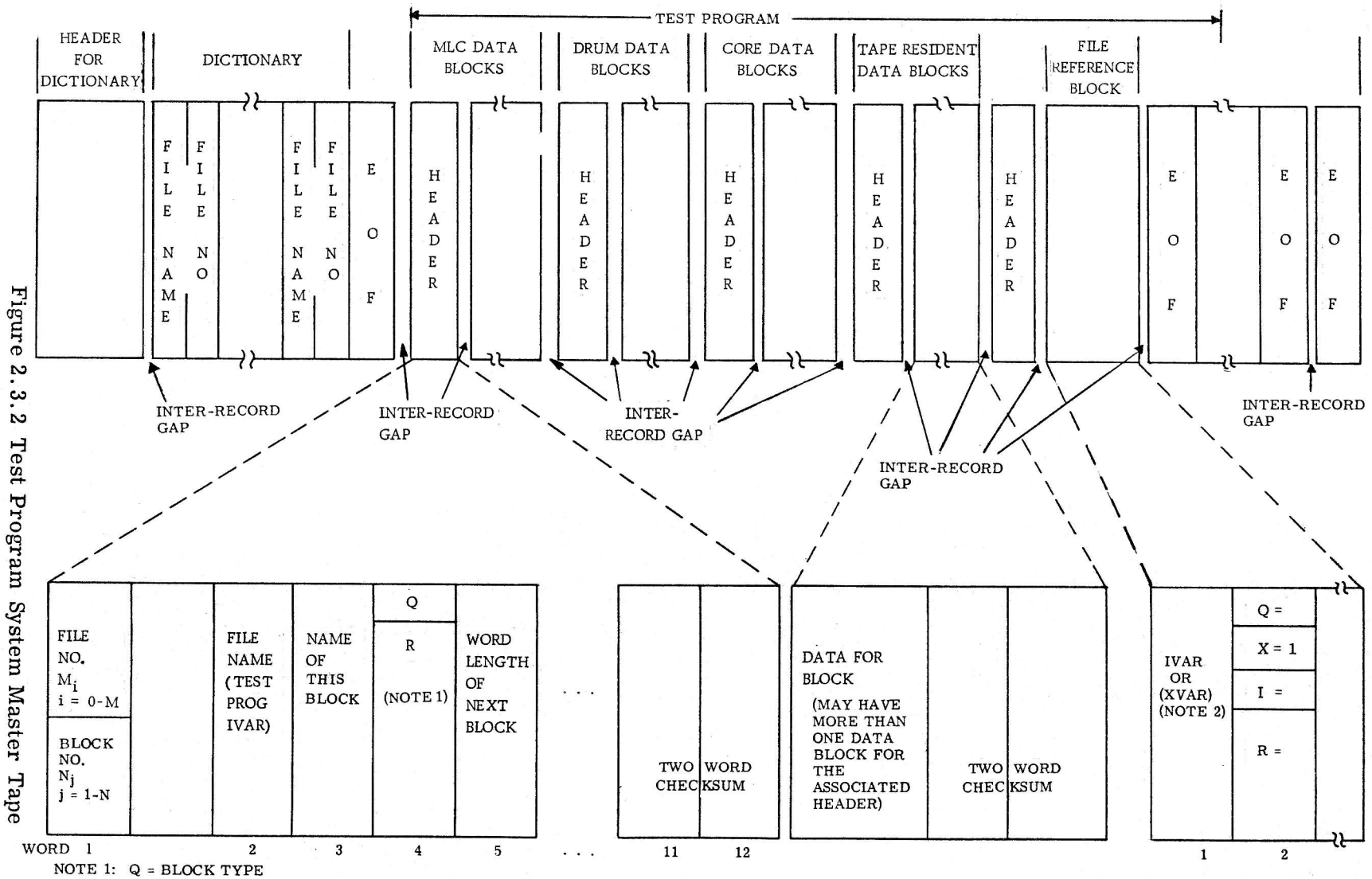


Figure 2.3.2 Test Program System Master Tape

NOTE 1: Q = BLOCK TYPE
BITS 18-23
R = ADDRESS
BITS 0-14

NOTE 2: Q = BLOCK TYPE (M, D, 0, Ø)
BITS 18-23
X = 1 INDICATES FIRST IVAR OF OVERLAY SEGMENT BIT 17
I = 0 FIRST WORD IN IVAR
I = 1 FIRST WORD IN IVAR
BIT 15
R = 15 BIT ACTUAL ADDRESS
BITS 14-0

The STAG card describes the file (test program) to be created or edited. One or more files may be generated or edited by repeating this card. The STAG card contains the file name (test program IVAR); a code indicating where the program data blocks are to be resident (core, drum, MLC, or tape); the reference address for the block of data; and the names of the various blocks of data that are to reside in one of the areas (core, drum, MLC, or tape).

The END card indicates the end of the STAG run. It causes a second End of File (EOF) to be written on the System Tape.

NOTE

If card input is to be used with the System/360 STAG, a //bMIXED card must precede the //SYSINbDDb* card.

2.3.2.1 INPUT Card Format

The format for the INPUT card is shown in Figure 2.3.3. The word INPUT is specified in card columns 8 through 12. Columns 16 through 80 specify the words CARD, TAPE, EDIT, or DUMP. If cards are used as input, the object deck immediately follows the INPUT card. The end of the object deck is indicated by a type of card punched in column 1. The rows punched are 0, 7, 9, 11 and 12.

If the input is tape, then the word TAPE is specified. If there are several tapes for input, the word TAPE is repeated and separated by commas. The number of times the word TAPE appears should agree with the number of input tapes.

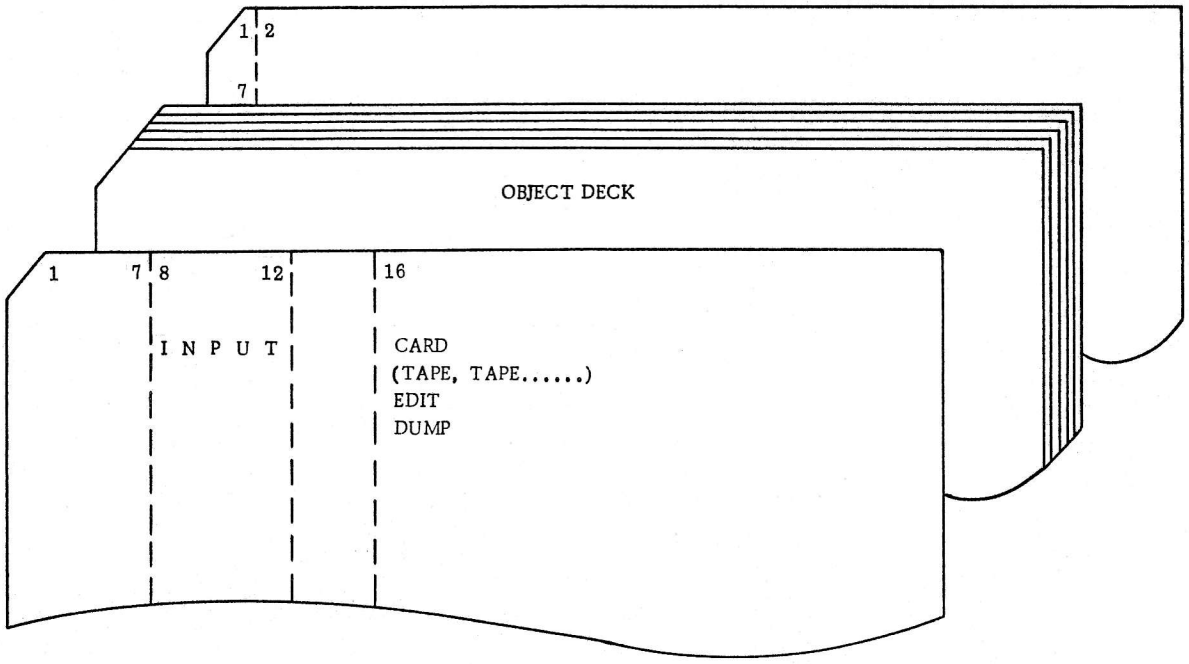


Figure 2.3.3 INPUT Card Format

If an edit run is to be made on an old master system test program tape, then the word EDIT is specified.

If a block of program data is to be printed, the word DUMP is specified.

NOTE

The DUMP capability is available only when the IBM System/360 computer is used. The EDIT and PATCH capability are only available on the RCA 1104 computer.

2.3.2.2 STAG Card Format

The STAG card describes a file to be created on the output tape. One or more files per output tape may be generated by repeating this card. The data on the STAG card may be continued to the next card if it is followed by a comma and then a blank (ϕ). The data on the next card must begin in column 16. The named entries must immediately follow each other and be separated by commas in most cases. Figure 2.3-4 shows the format for this STAG card.

The STAG Card format is described as follows:

- FILE NAME - This field specifies the character BCD test program IVAR to be used for this file.
- EDITID - This field is optional and is used only when the EDIT option is specified on the INPUT card. If EDITID is DROP, the file indicated by FILE NAME will not be copied from the old tape to the output tape. If EDITID is other than DROP, the file specified by FILE NAME will be inserted on the output tape following the file specified by EDITID. If the EDITID and FILE NAME are the same, the file specified by FILE NAME will replace the file on the old test tape. The cards specifying deletions, replacements, or insertions must be placed in the order of EDITID occurrence on the old master system test program tape.

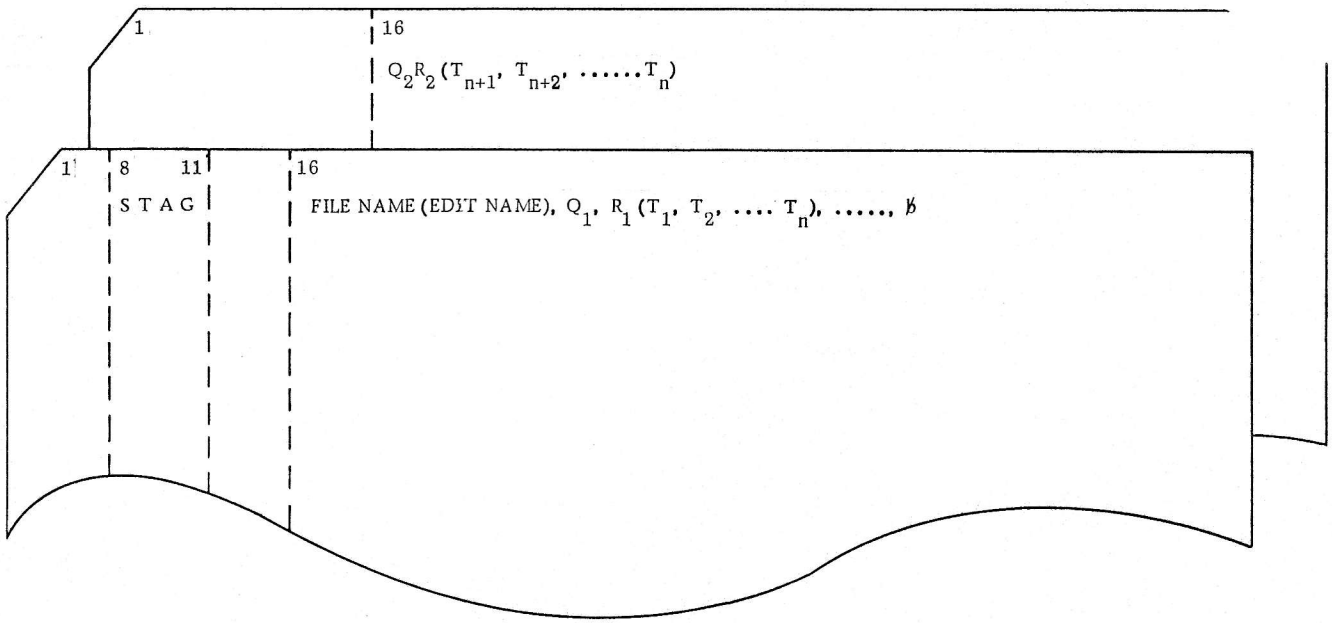


Figure 2.3.4 STAG Card Format

NOTE

DROP must not be used as a file name.

- Q is a code to indicate the storage media where the series of data program blocks are to reside. Q may be any one of the following:

  M - This code indicates that the data program blocks within each group are to be resident in the MLC.

  D - This code indicates that the series of data program blocks within each group are to be resident on the LCCC drum.

  0 (zero) - This code indicates that the series of data program blocks within each group are to be resident in core.

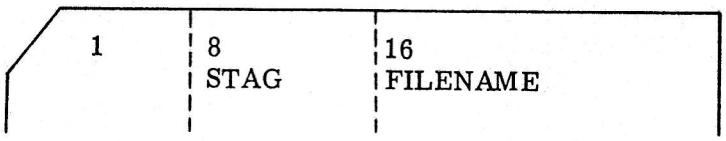
  O (oh) - This code indicates that the series of data program blocks within each group are to remain on tape and loaded into core when requested.

NOTE

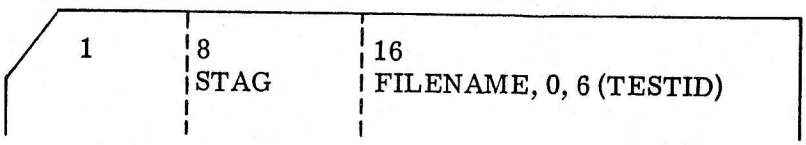
All the data program blocks associated with each code must be grouped together and each group must be specified in the following order:
(a) MLC (b) LCCC Drum (c) LCCC Core (d) LCCC Tape.

- R is the reference address for the series of data program blocks and can be a bank number (6, 7) or an IVAR. The first block in the remaining series is assigned this address and the remaining blocks follow according to their sequence on cards or tape.
- (T_1, T_2, . . . , T_n) is a set of prelinked program identifiers that make up one of a series of data program blocks. If it is not possible to link an XVAR, the XVAR address is inserted in the Instruction Address Word with a 7 in bits 17 through 15. Each successive reference to this XVAR will be linked to the address of the XVAR's first occurrence.

A test file can be generated by specifying only the TESTID for the FILE NAME as shown below:



This will be the format for a normal test program where all additional subprograms are implied by XVAR's. This will allow patching and possible program replacing at load time. Another way to specify and generate a test file is to format the STAG card as follows:



All the implied programs referenced by the TESTID are balanced between banks 6 and 7, and the TESTID is the same as the FILENAME. No implied program may have an ORG specification.

2.3.2.3 END Card Format

This card contains only the word END in columns 8 through 10.

2.3.3 On-Line Program Editing

The Operating System Utility Executive program provides test programmers with the capability to patch portions of their programs. This on-line editing can be done via display console keyboard commands while the program is resident in core or on drum. A test programmer may change, display, and print data from core or drum. The commands to perform these functions are given in Table A-9, Appendix A. The procedures that must be performed to enter these commands are:

a. Depress TYPE to activate the keyboard.
b. Type in the appropriate input as given in Table A-9.
c. Depress EDIT EXECUTE.

A test programmer may also enter changes to his program via card input. Patch card format is given in Figure 2.3.5. The first card is a control card containing the word PATCOR in columns 8 through 13. The next card is a data card containing a five-digit absolute octal address of the core location that is to be changed, and the eight digits of new date (in octal) that is to be placed into the specified address. The address is punched into columns 8 through 12, and the data is punched into columns 14 through 21. Column 13 contains a comma. If a number of patches is required, then a series of these data and address cards may be specified.

An END card and a blank card must be placed as the last cards in the patch card deck. Patches are read in at the time the test program is loading when SSW-4 is active (in the down position). Overlays called into core may also be patched by placing the desired patch cards in the reader and activating Sense Switch 4 while the overlay is being loaded.

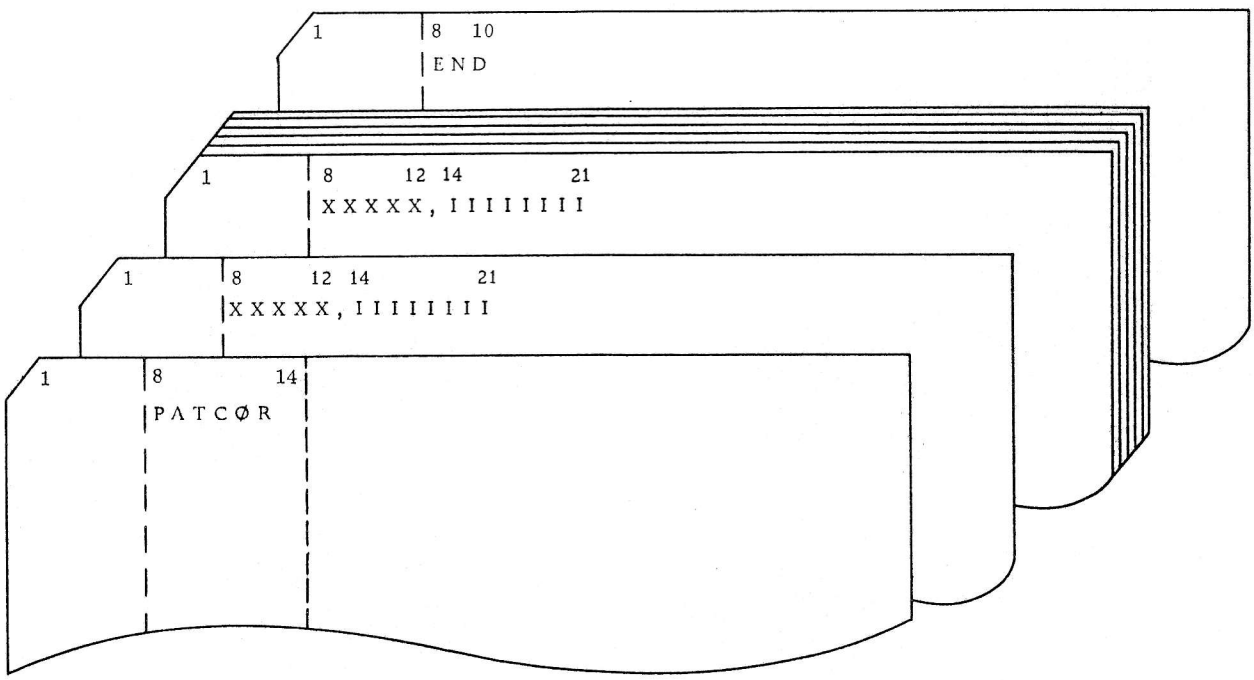


Figure 2.3.5 Patch Card Format

2.3.4 Function Executor Routine Assembly and Loading

A function executor routine may be resident in core or on drum. Both core-resident and drum-resident function executor routines can be assembled separately from the Operating System. They are placed on the Operating System Master Tape via the STAG program. The STAG control cards designate whether it should be placed in core or on drum. Residency depends on whether the routine has a critical monitoring activity. If the routine does critical monitoring, it will reside in core. It is loaded as an integral part of the system at load time. If a routine does not have critical monitoring activity, it resides on drum. When the routines are called from drum for execution, they will reside in bank 3 starting at location 34000.

Core-resident function executor routines are generated by STAG for loading into bank 3 of the MLC. The format of the STAG card used to generate these routines is as shown below:

Card Column 16

0, 3(FE50, GE01, GE16, CE00, ME01), etc.

Control cards for drum-resident function executors are in the format shown below:

Card Column 16

D, SFEB (XXXX),

where XXXX is the function executor routine IVAR.

SECTION III

OPERATING SYSTEM SERVICES

3.0 SUMMARY

The test programmer has at his disposal various Operating System services in the form of core-resident routines (in both the MLC and the LCCC) designed to save the test programmer time and effort. These routines move, write, and convert data whenever the programmer applies them. They are grouped and explained in the following three units:

- Inter-Computer Core Transfer Routine of Data Link Executive
- System Logging Program
- Utility Service Programs

The Inter-Computer Core Transfer is a routine that moves specified data between computers.

The System Logging routine provides a means of recording information available while the test program is running. The data is saved by this program for a post-processing analysis run.

The Utilities are a group of commonly used routines that provide many different services. There are four basic data conversion routines and six routines which access and convert time data. In addition there are two routines used for index register data storage.

3.1 INTER-COMPUTER CORE TRANSFER

The Remote Core-to-Core Transfer routine of the Data Link Executive program provides a communication link between the MLC and the LCCC. It gives the capability for a test program to read data from or write data into the MLC; and for a function executor routine to read data from or write data into the LCCC.

NOTE

The computer from which the read or write request is made is defined as the local computer; the other is the remote computer.

A maximum of 1026_{10} words may be transferred with one request. If a request is made, no other requests are accepted until the first one has been completely serviced.

The Operating System provides two macro instructions to read or write data. The names of the macros are RCOR and WCOR for a read and write data, respectively. These macro names and their calling sequences are given in Table A-7, Appendix A.

To read or write data, a test program or function executor routine must provide a TSPO instruction to transfer to SCTP, 7. The control words following this instruction contain the codes and parameters required by the Operating System from the calling program to perform the operation. Bits are set by the Operating System in return parameters to indicate the status of a request.

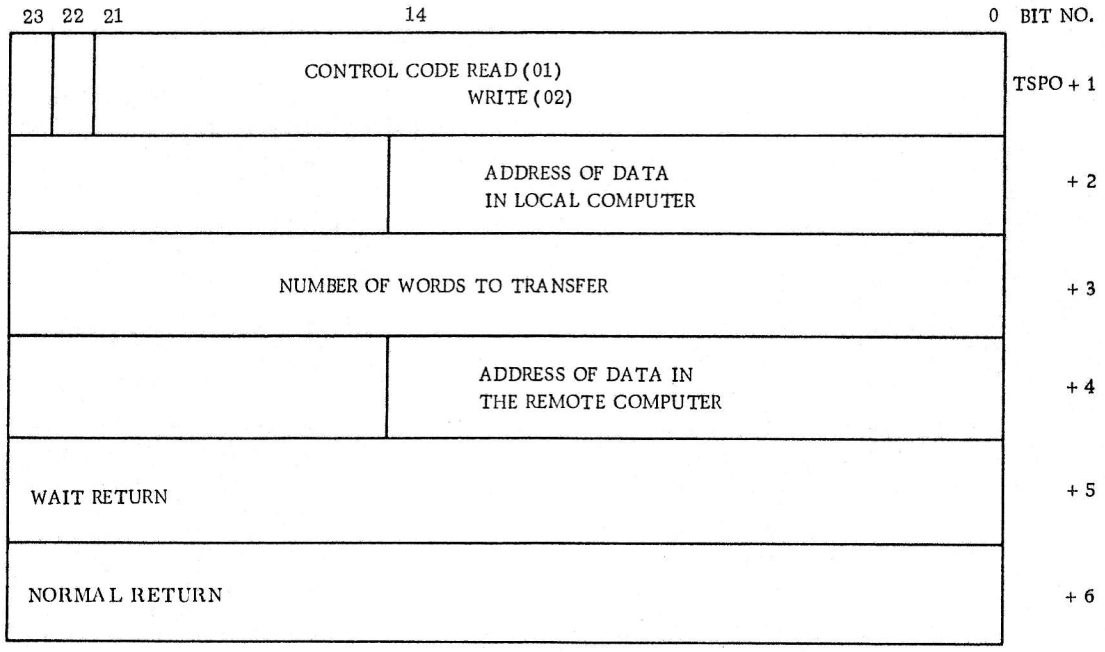
3.1.1 Control Word Format

The control word format contains parameters for data transferred from one computer to the other. The test programmer must supply the Operating System with these parameters in order to perform a data transfer. These parameters are supplied in macro instructions as operands (Refer to Table A-7, Appendix A.) When the macro is assembled, the parameters are placed in a format as shown in Figure 3.1.1.

The control word format contains the four parameters supplied by the test programmer necessary to perform the transfer. The parameters are as follows:

- Control code
- Starting core address in local computer
- Number of words to transfer
- Address of starting core location in remote computer.

The first control word following the TSPO instruction contains the control codes and two bit positions for the conventional operation status codes to be checked by the calling program for success or failure. The control code is used to determine the type of operation. The read operation is coded 01; the write operation is coded 02. Bits 21 through 0 are reserved for this code.



NOTE

BITS 22 AND 23 OF THE TSPO + 1 WORD INDICATE THE FOLLOWING AFTER THE TRANSFER OPERATION
IF THEY = 10, THE OPERATION SUCCEEDED
IF THEY = 01, THE OPERATION FAILED
IF THEY = 00, THE OPERATION HAS NOT YET COMPLETED

Figure 3.1.1 Control Word Format

After a status return, the status code is supplied by the Operating System. Bits 23 and 22 are used for this code, which indicates whether the data transfer has succeeded or failed.

The second control word contains the address of the first data word in the local computer. Thus for a read, this address is the location in the local computer where the data is to be placed; for a write this address is the location in the local computer from which data is accessed. Both are 15-bit addresses.

The third control word contains the number of words to be transferred. A maximum of 1026_{10} words may be transferred. The entire word is reserved for this information.

The fourth control word contains the address of the first data word in the remote computer. For a read, this address is the location where data is to be accessed; for a write this address is the location where data is to be placed. This is a 15-bit address.

3.1.2 Returns from the Remote Core-To-Core Subroutine

Returns are made from the Remote Core-to-Core subroutine to the requesting test program or function executor when the Remote Core-to-Core subroutine is busy or the request has been stacked.

If Remote Core-to-Core is busy, control is returned to the requesting program's WAIT transfer instruction, which immediately follows the four control words (refer to Figure 3.1.1).

When the request has been stacked, control is returned to the location following the WAIT return.

The Remote Core-to-Core subroutine sets either bit 23 or 22 of the Y+1 word of the calling macro to indicate success or failure in transferring data. The test programmer may analyze the status of the operation by checking the return codes (bits 23 and 22) as required. If bit 23 equals 1, the operation was successful; and if bit 22 equals 1, the operation failed.

3.2 SYSTEM LOGGING

The Operating System provides a data logging service for postprocessing and evaluation of vehicle checkout. Four types of logging functions are discussed in this section as follows:

- Test Program Control and Function Executor Control Data
- Vehicle Data
- Display Data
- IODC Hardware Malfunction Messages

Test Program Control and Function Executor Control data is defined as the control data which provides an interface between test programs and the Operating System. Vehicle data is defined as information data that is transferred between test programs and the vehicle. Display data is defined as data which is transferred between test programs and the Display System.

The System Logging program logs all data requested by other Operating System programs or by test programs and function executors (refer to the program description for the System Logging program in Volume I of this document for word formats of data logged on log tape). This program logs the following specific control data which is common to all requests:

- BCD ID name of requesting program
- BCD IVAR name of test program in progress
- Variable word containing logged data identifier
- Number of data words logged
- Computer indicator (LCCC or MLC)
- GMT from IODC 7 (2 bits)

If a test programmer chooses to log data, he may use the calling sequence given in Table A-7 of Appendix A. The program requesting logging service links to the System Logging program with a TSPO SLOG, 7. The test programmer must specify calling parameters in the control words of the calling sequence. These parameters are:

- BCD four-character IVAR of the test program requesting data to be logged
- Variable word which identifies the data to be logged
- Absolute address of the first data word to be logged
- Number of data words to be logged.

Figure 3.2.1 shows the format for these parameters.

When a request is made and cannot be stacked, the System Logging program returns control to the full return core location, which immediately follows the four control words. If the request has been stacked or the data has been logged, a return is made to the normal return core location with bit 23 set in the fourth control word to indicate the status (Figure 3.2.1).

When a test programmer has data to be logged, he should not duplicate data that the Operating System logs. The data that the system logs is explained in subsequent paragraphs.

3.2.1 Logging by Test Program Control

When Test Program Control interfaces with a test program, it logs control data associated with the request, execution, and processing of the test program. This data includes:

- IVAR of the test program
- Operator number and page

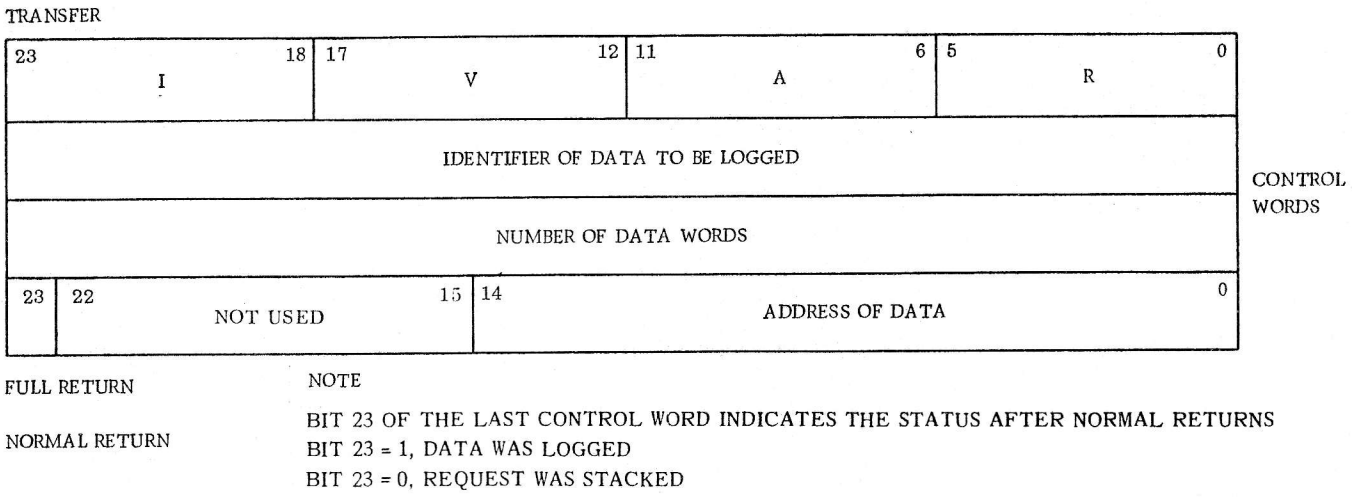


Figure 3.2.1 Control Words for Logging Data

- Descriptive name of the test program
- Starting GMT (27 bits) of the test program
- Ending GMT (27 bits) of the test program
- Starting countdown time of the test program
- Type of termination of the test program (see paragraph 2.2.2)
- Cause of external abnormal termination (see paragraph 2.2.2.3)

3.2.2 Logging By Function Executor Control (LCCC)

When a function executor routine is called, either by display input or by a test program, Function Executor Control in the LCCC logs the request. The logged data is as follows:

- The origin of request CONS for display input or TEST for a test program request.
- Console control words
- Request mode (FUNC or FUNT)
- IVAR of function executor routine requested
- Time interval requested
- Parameters or address of parameters

3.2.3 Logging By Display Service Programs

The display programs of the Operating System log all data transmitted to and from the Display Console. The data logged as input from the Display Console is:

- Function Executor routine commands and parameters
- Test program commands and parameters
- Utility commands and parameters
- Discrete commands and status requests

Data that is sent to the console for display is also logged. This information includes:

- Display control words
- Variable displayed data such as BCD, raw binary, and preformatted

3.2.4 Logging Vehicle Data

All data sent to and received from the vehicle is logged. The type of data logged includes LVDC and discrete data.

LVDC data logged includes:

- History Status table when full
- Error messages from the LVDC
- Number of data words that could not be transmitted to the LVDC due to a Data Link failure.

The Discrete Executive initiates logging at both computers. There are a number of different types of discrete tables each containing specific data. These tables are:

- LDO and MDO Profile table
- LCCC and MLC Discrete Status tables for IODC's 5 and 7
- General Discrete Log table (LDI, LDO, and MDI Changes)
- MDO Issue table

Specific discrete data that is logged is as follows:

- Discrete numbers
- Address of first invalid entry
- Initial GMT (27 bits) for LCCC and MLC
- Current GMT for DO's
- Relative time (18 bits) for DI's
- Discrete issued because of DI activity
- DO complement group number
- DI group number
- Status of discrete group

3.2.5 Logging by System Message Routine

All IODC device error messages are logged by the System Message routine. The type of data includes the following:

- Specific device which failed
- IODC commands upon which failure occurred
- Control words
- Errors sensed
- Number of attempts

General and remote messages which are printed are also logged by the System Message program.

3.3 UTILITY SERVICES

The Operating System provides routines that test programmers may use to perform various utility functions. These utility functions include accessing, converting time, printing data from memory or drum, and saving index registers and accumulators. These routines are available in both computers and may be used by test programs and Function Executor routines.

3.3.1 Data Conversion

The Operating System provides the capability for test programmers to convert various types of coded data. The types of conversion available are:

- Octal to Binary
- Binary to Octal
- Binary to BCD
- BCD to Binary

The first three types of conversion are accomplished by utility routines provided by the Operating System. The BCD to Binary conversion is performed by the normal conversion instructions - BTD, BTDI, DTB and DTBI. The conversion tables required for these instructions can be accessed in bank zero by the test programmer.

The calling sequences used to link with the utility routines are given in Table A-8, Appendix A. There are no control words required for these routines. The data to be converted is placed in the accumulators prior to transferring to the routines. The Operating System then places the converted numbers in the accumulators and returns control to the location following the Test Program or Function Executor routine transfer instruction. The test programmer may code this location as required.

3.3.1.1 Octal to Binary (SUOB)

This data conversion routine accesses an eight-digit octal number and converts it to an equivalent binary number. If there are any leading blanks, they are converted to zeroes. The test programmer places this octal number into both accumulators prior to transferring control to the routine. The four least significant digits and the four most significant digits must be placed into the left and right accumulators, respectively. Control is transferred with the instruction TSPO SUOB, 7. When the Operating System completes the conversion, the converted binary number is placed into the right accumulator so that the test program may access it.

3.3.1.2 Binary to Octal (SUBO)

This data conversion routine accesses a 24-bit binary number and converts it to an equivalent eight-digit octal number. The test programmer must place this binary number into the left accumulator prior to transferring control to this routine. Control is transferred with the instruction TSPO SUBO, 7. When the Operating System completes the conversion, the converted eight-digit octal number is placed into both accumulators. The four least significant digits and the four most significant digits of this number are placed into the left and right accumulators, respectively.

3.3.1.3 Binary to BCD (SUBD)

This data conversion routine accesses a double-precision binary number and converts it to equivalent BCD characters. This double-precision number has a decimal point to the left of Bit 23 in the right accumulator, and must not exceed ±999.999. If the number is positive, it is not necessary

to code the BCD plus sign. The test programmer must place this binary number into both accumulators prior to transferring control to the routine. Control is transferred with the instruction TSPO SUBD, 7. When the Operating System completes the conversion, the converted BCD number is placed into both accumulators. The left accumulator contains the decimal point and three fraction digits. If the number is negative, the right accumulator contains the sign and three integer digits. If the number is positive, the right accumulator contains four BCD digits.

NOTE

If the test programmer tries to convert a number larger than ±999.999, the Operating System returns the number 1999.999.

3.3.1.4 BCD to Binary

The Operating System provides integer and fraction conversion tables for BCD-to-binary conversions. The test programmer may use the normal conversion instructions to convert BCD characters to a binary representation. The address of these tables is equated to a symbolic table name. The IVAR's and the tables are defined as follows:

- STSW - Integer Table
- STSF - Fraction Table

3.3.2 Accessing and Converting Time

The Operating System provides the capability for test programmers to access three forms of time and convert them to a desired specific data format. The three forms of data are Greenwich Mean Time, Relative Time, and Countdown Time. These may be converted to two types of data format: BCD and binary.

Greenwich Mean Time is hardware-counted in Gray Code. A test programmer may access this time in binary or in BCD. The binary Greenwich Mean Time is 27 bits long and is in hours, minutes, seconds, and milliseconds. The BCD Greenwich Mean Time has the format HHMM/SS.XXX, where HH is the hour, MM are the minutes, and SS are the seconds. An example of this BCD format is 2359/59.999. It should be noted that the hours and minutes are in a military time format.

Relative time is a millisecond binary count. It is set to zero at initialization time. The test programmer may use this for time-out procedures. He may access the current relative time or he may have the Operating System check for a specific time lapse. The time interval is specified by the test programmer.

Countdown time is a binary count in seconds. The test programmer may access this time supplied by the Operating System in both binary and BCD formats.

To access these various forms of time and specific data format, the test programmer may use the calling sequences and coding examples as given in Table A-8, Appendix A. The Operating System has only one return to the test program or function executor routine. The return is to the core location which follows the last coded instruction in the calling sequence. The test programmer may code this location as he requires.

3.3.2.1 Read Greenwich Mean Time and Convert to Binary (SREB)

The test programmer transfers to this routine in either computer with the instruction TSPO SREB, 7. He need not specify any input parameters to the Operating System.

In the LCCC, this routine reads raw Greenwich Mean Time and converts it to an equivalent 27-bit binary number. In the MLC, this routine computes GMT from an LCCC-supplied reference and a PIT reference and converts it to an equivalent 27-bit binary number. A new GMT reference is supplied by the LCCC with every LDI-MDO transmission.

When the routine completes the conversion, it returns control to the core location which follows the test programmer's transfer instruction. The binary Greenwich Mean Time value will be in the left and right accumulator. The left accumulator contains the four most significant bits in positions 3 through 0. The right accumulator contains the 23 least significant bits in positions 22 through 0.

3.3.2.2 Read Greenwich Mean Time and Convert to BCD (SRED)

This routine reads raw Greenwich Mean Time and converts it to an equivalent BCD number. The test programmer transfers to this routine with the instruction TSPO SRED, 7. He need not specify any input parameters to this routine. When the routine completes the conversion, the converted Greenwich Mean Time number is placed into the three core locations which follow the test programmer's transfer instruction. This format is HHMM/SS.XXX, where HH is the hour, MM are the minutes, SS are the seconds, and XXX are the milliseconds. Figure 3.3.1 shows the core layout and bit configurations of this data. The routine returns control to the test program at the next core location which follows the return data. The test programmer may code this location as he requires.

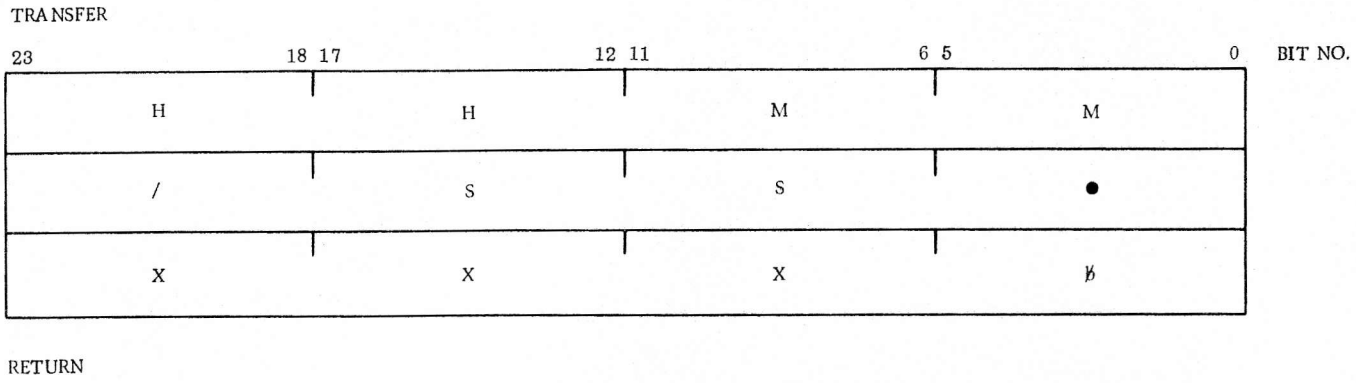


Figure 3.3.1 BCD Format of Greenwich Mean Time

3.3.2.3 Convert Greenwich Mean Time from Binary to BCD (SRCN)

This routine accesses Greenwich Mean Time in binary and converts this number to an equivalent BCD Greenwich Mean Time. The test programmer transfers to this routine with the instruction TSPO SRCN, 7. He must specify 27 binary bits of GMT in the accumulators. The four most significant bits (MSB) are placed into bit positions 3 through 0 of the left accumulator. The 23 least significant bits (LSB) are placed in bit positions 22 through 0 of the right accumulator (see figure 3.3.2). When this routine completes the conversion, the data is placed into the three locations which follow the test programmer's transfer instruction. This format is HHMM/SS.XXX, where HH is the hours, MM are the minutes, SS are the seconds, and XXX are the milliseconds. Figure 3.3.1 shows the core layout for this data. This routine also returns control to the core location which follows the return data. The test programmer may code this location as he requires.

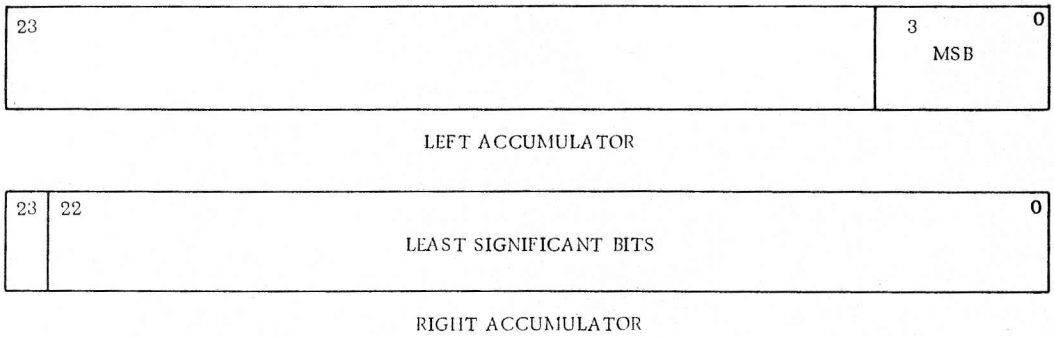


Figure 3.3.2 Greenwich Mean Time Conversion Using Accumulators

3.3.2.4 Read Relative Time (SRRT)

This routine provides the necessary processing for test program to access the current relative time. The test programmer causes transfer to this routine with a TSPO SRRT, 7. Input parameters need not be specified. The routine senses this time from the sense register and places it into the left accumulator. The time is a binary count in milliseconds. After accessing the time, the routine returns control to the core location following the test program transfer instruction, which the test programmer may code as required. The routine saves and reloads the contents of the index registers but does not save and reload the contents of the accumulators.

3.3.2.5 Read Relative Time and Check for Delay (SRDC)

This routine accesses relative time and checks for a time interval lapse. The test programmer transfers to this routine with the instruction TSPO SRDC, 7. He must supply two input parameters: one is the base reference relative time, which is placed into the right accumulator; the other is the amount of the time interval in which the delay check is to be made. The routine determines whether the interval has or has not elapsed. This millisecond interval (in binary) is placed in the core location which follows the test program transfer instruction. When the routine completes processing, it places the return information in the left accumulator. If the accumulator is negative, the time interval has elapsed. If the accumulator is positive, the time interval has not elapsed. The accumulator contains the amount of time left before the interval expires.

3.3.2.6 Read Countdown Time (SCDC) - LCCC Only

This routine accesses the raw countdown time from the Input/Output Sense Register and converts it to a BCD and binary format. This time is T ±(days, hours, minutes, and seconds). This routine may be accessed by the instruction TSPO SCDC, 7. The test program is not required to specify any input parameters when the routine is accessed. When the Operating System completes the routine, the converted time is placed into the five core locations which follow the transfer instruction. Coding to generate these locations is given in Table A-8, Appendix A. The format of the converted information is shown in Figure 3.3.3. The first three locations contain the countdown time in BCD format. An example of this countdown time is T + 02 2359.59. The next two core locations contain the binary equivalent of this time: the 23 least significant bits are placed in the first location; the four most significant bits are contained in the second location and are right justified. It should be noted that the time may be negative. If it is, the binary return information is in two's complement form. When the Operating System completes this routine, it returns control to the sixth core location following the transfer. The test programmer codes this return line as required. To obtain countdown clock time at the MLC, the user must request a DDAS read using MODE = 7 in the calling sequence (refer to Table A-7, Appendix A for a description of the calling sequence).

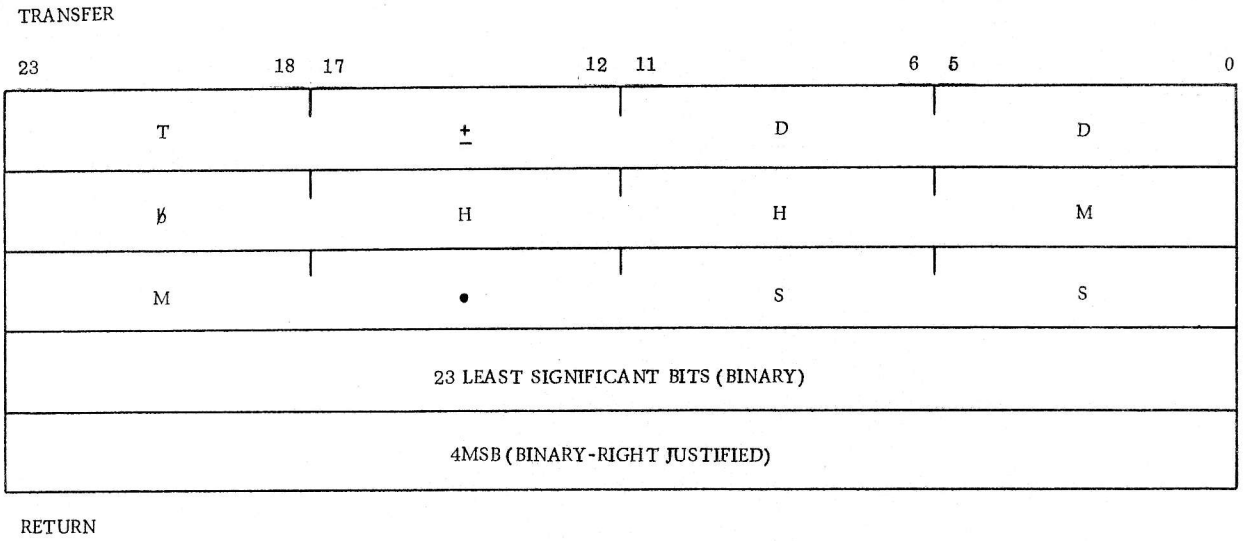


Figure 3.3.3 Countdown Time Format

A test programmer may want to know whether countdown time is holding or counting. The Operating System provides a communication cell which indicates this. This cell may be accessed by referring to its symbolic name SCDI. If the cell is equal to zero, countdown time is decrementing. If the cell is not equal to zero, then countdown time is holding.

3.3.3 Printing Data

The Operating System provides the test programmer with the capability of printing data from LCCC core. The data is printed in octal form at the computer where the request originated; all HSM parity errors are flagged by the letters PE which are printed to the left of the location contents. Two routines provide the processing for the storage devices.

3.3.3.1 Print Core Data Routine (SUDC) - LCCC Only

This routine prints data contained in HSM of the LCCC after printing the following header information: data start and end address, current date, relative GMT, tape identification, and the index register contents.

The starting and ending 15-bit addresses of the designated data must be specified in the left and right accumulators respectively. The test programmer transfers control to this Print Core routine with the instruction TSPO SUDC, 7. The Operating System returns control to the next location which follows the TSPO instruction. The test programmer codes this location as required.

3.3.4 Saving Register Data

The Operating System has two routines which provide the capability to save and restore data in the computer registers and accumulators. The test programmer will usually provide the Operating System with an identical address for storing and accessing register data. This address is specified in the second word of the calling sequence and may be 15 bits long. The test programmer transfers control to these routines by coding calling sequences as given in Table A-8, Appendix A.

When the Operating System completes the execution of these routines, control is returned to the third location of the calling sequence following the address control word.

3.3.4.1 Store Register Data (SSXR)

This utility routine will store the contents of the right and left accumulators and the six index registers. Data will be stored into eight locations; the starting address is specified by the test program. The test program transfers control to this routine with the instruction TSPO SSXR, 7. The Operating System will store the contents of specific registers into specific locations for the eight storage words. Figure 3.3.4 shows the eight-word layout for the register data. The first two locations will contain the right and left accumulator, respectively. The next six locations will contain the six index registers.

ADRS	RIGHT ACCUMULATOR
+1	LEFT ACCUMULATOR
+2	INDEX REGISTER 1
+3	INDEX REGISTER 2
+4	INDEX REGISTER 3
+5	INDEX REGISTER 4
+6	INDEX REGISTER 5
+7	INDEX REGISTER 6

Figure 3.3.4 Register Storage Locations

3.3.4.2 Reload Register Data (SRXR)

This routine will restore the data of the right and left accumulator and six index registers previously stored by a store register data request. The test programmer must transfer control to this routine with the instruction TSPO SRXR, 7. If the test programmer requires data reloaded which is different from data that was previously stored, he must first place the data into the locations intended for specific registers as shown in Figure 3.3.4.

SECTION IV

SYSTEM INPUT/OUTPUT

4.0 SUMMARY

This section encompasses all data movement to and from the Launch Vehicle, the Display Consoles, and the various peripheral devices via the two computers (the LCCC and the MLC). The major segment of data movement is initialized by test programs.

System input/output is divided into the following three units:

- Launch Vehicle
- Display System
- Peripheral Devices

The Launch Vehicle data flow is accomplished using the discrete Input/Output Data Channels (IODC's), the input/output sense lines, address lines and registers, and the Digital Data Acquisition System (DDAS). The IODC's, sense lines, and address lines provide both input and output while DDAS provides only input to the Operating System and the test programs.

The Display System hardware is the means by which the test engineer communicates with the test program. Three software programs provide and control input/output data for the Display System.

The types of output information displayed for the test engineer include options, tutorial messages, and fixed and/or variable data. He in turn provides instructions to the test program by entering input data through the display console keyboard. The alternatives are provided by the test programmer.

4.1 LAUNCH VEHICLE INPUT/OUTPUT

The Saturn Operating System provides a number of programs which have the capability to communicate with various vehicle systems and subsystems. These programs include:

- Discrete Service
- LVDC
- Analog Input
- DDAS Interface

These programs are used to transfer both data and control signals between the Saturn Launch Vehicle and the test programs operating in the LCCC. These Operating System programs are resident in both computers, with the exception of the DDAS Executive program, which resides only in the LCCC.

The test programs and function executor routines access the Operating System programs by using the appropriate calling sequences. These sequences are given in the tables of Appendix A.

Figure 4.1.1 shows the general flow of data in relation to the test programs, function executor routines, Operating System programs, and the vehicle.

4.1.1 Discrete Input/Output

The Operating System provides test programmers with the necessary processing to check status of or issue discretes in the LCCC and the MLC.

The Operating System Discrete Service programs consist of subroutines (applicable ones shown in Table 4.1-1) which may be requested manually or automatically.

These subroutines accomplish the following:

- Issue single or multiple discretes
- Check status of single or multiple discretes
- Issue switch selector channel functions
- Issue an MDO and check the MDI response within a specified MDO time interval

The Operating System provides macro names and calling sequences, which enable the test programs and function executor routines to call the discrete routines, supply parameters, and link with the Discrete Service programs. Table 4.1-1 shows the macro name, IVAR linkage, and discrete routine function. The table also indicates whether the test programs in the LCCC or the function executor routines in the MLC can access the routine.

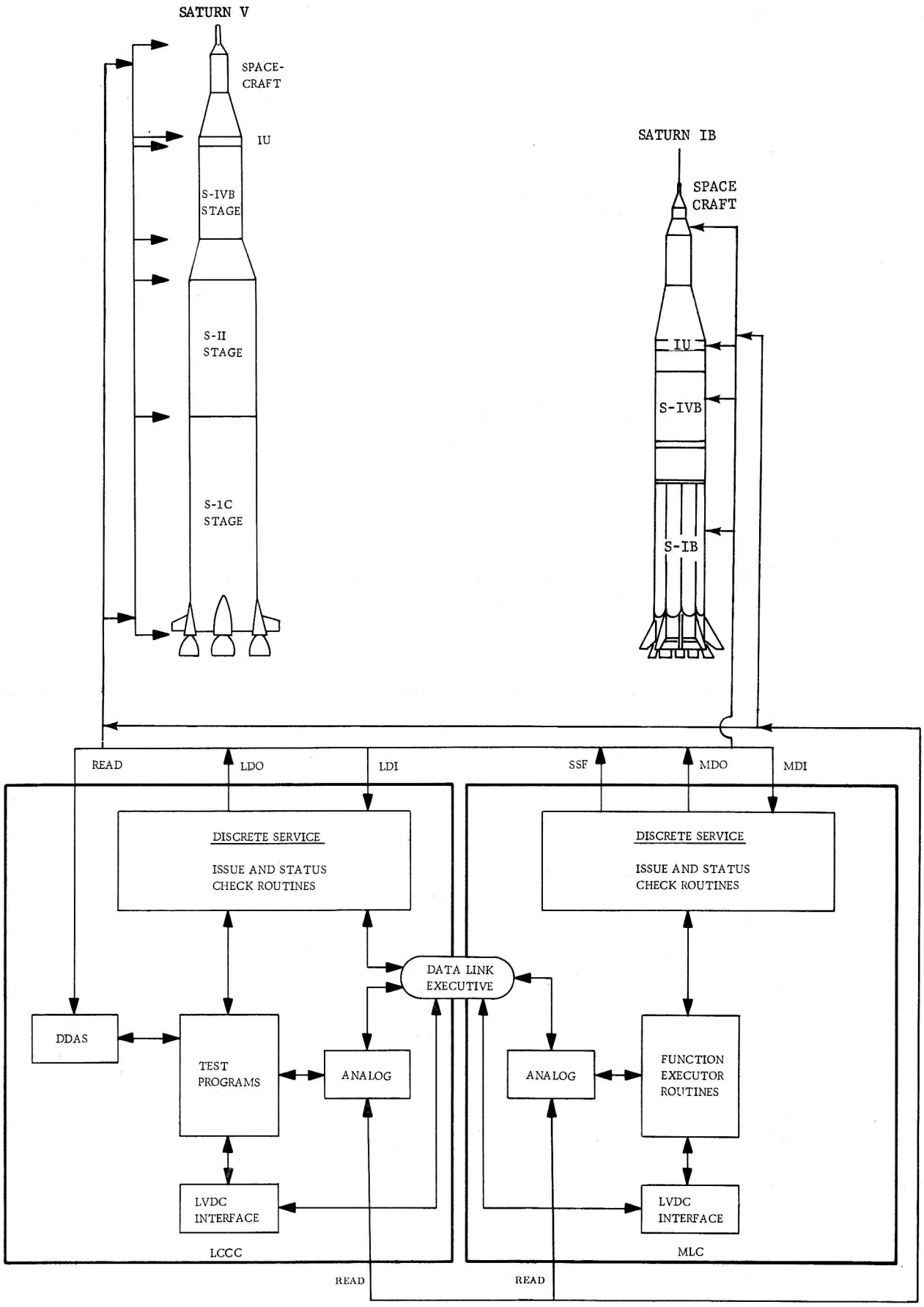


Figure 4.1.1 Launch Vehicle Input/Output Data Flow

Table 4.1-1 Discrete Functions

MACRO Function	MACRO Name	IVAR Linkage	Computer
Issue Single MDO	SMDO	SACA	LCCC/MLC
Issue Multiple MDO	MMDO	SADA	LCCC/MLC
Issue Single LDO	SLDO	SAPA	LCCC
Issue Pulsed MDO & Check MDI	PDOCK	SAJA	LCCC/MLC
Issue Switch Selector Function	SWSEL	SAKA	LCCC/MLC
Check Status of Single MDI	SMDI	SAEA	LCCC/MLC
Check Status of Multiple MDI	MMDI	SAFA	LCCC/MLC
Check Status of Single MDO	CMDO	SANA	LCCC/MLC
Check Status of Single LDI	SLDI	SAGA	LCCC
Check Status of Multiple LDI	MLDI	SAHA	LCCC
Check Status of Single LDO	CLDO	SAMA	LCCC

4.1.1.1 Issue Single or Multiple Discretes

The operating System provides the test programmer the capability to issue a single discrete or issue a group of discretes. There are three issue discrete subroutines; the calling sequences for these subroutines are given in Table A-2, Appendix A, of Volume I. The subroutines are:

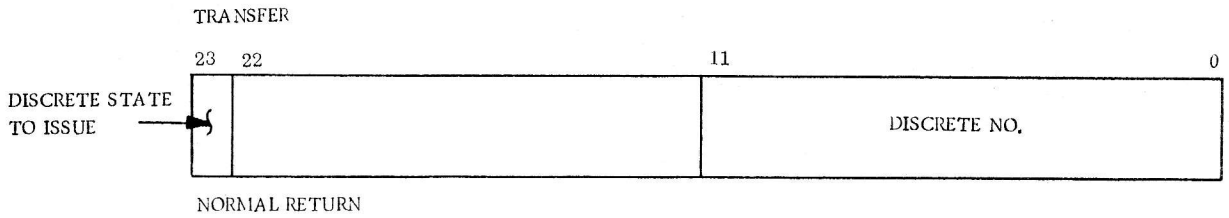
- Issue Single MDO
- Issue Single LDO
- Issue Multiple MDO's

The Issue Single MDO subroutine may be accessed by either a test program operating in the LCCC or a Function Executor routine operating in the MLC. The test program links to this subroutine with a TSPO SACA, 7 and specifies two calling parameters in the control word which follows the transfer instruction.

The two calling parameters are:

- MDO number
- MDO state to issue

These are specified in bits 11 through 0 and bit 23, respectively, as shown in Figure 4.1.2.



BIT 23 = 1 : ISSUE LDO (MDO) ON

BIT 23 = 0 : ISSUE LDO (MDO) OFF

Figure 4.1.2 Issue Single Discrete Control Word

The Issue Multiple MDO subroutine may be accessed both by test programs operating in the LCCC, and by Function Executor routines operating in the MLC. This subroutine will issue a group of discretes from the MLC. The test program or routine may link to this subroutine with a TSPO SADA, 7. The programmer must specify four calling parameters. They are:

- High and Low Discrete Numbers
- State Mask Word
- Issue Mask Word

These parameters are specified in the control words as shown in Figure 4.1.3.

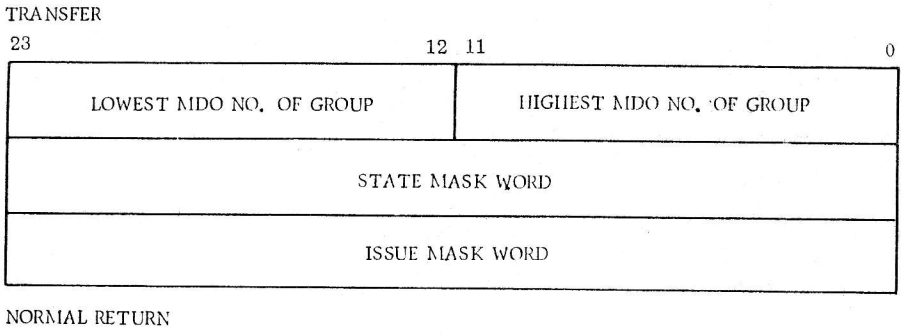


Figure 4.1.3 Issue Multiple MDO Control Words

When multiple discretes are specified, a maximum of 24 may be issued. The low and high numbers specified must be inclusive of the group of 24 discretes and are specified in bits 23 through 12 and 11 through 0, respectively, in the first control word.

The state mask word for the discrete group contains a 24-bit configuration indicating which of the two states to issue. If a bit is a 1 or 0, the corresponding discrete is issued on or off, respectively. The high- and low-order bit of this word corresponds to the high and low number, respectively, of the discrete group. The mask word is specified in the second control word and must be octal and right justified.

The issue mask word of the discrete group contains a bit configuration indicating which discretes are to be issued within the group. If a bit is a 1 or 0, the corresponding discrete is to be issued or not issued in the state specified by the state mask word. The high- and low-order bit of the issue mask word corresponds to the high and low discrete of the group. The issue mask word is specified in the third control word of the calling sequence, and is octal and right justified.

When a discrete operation is complete (either subroutine), control is returned to the core location following the control word(s) of the calling sequence. This core location is specified as the NORMAL return.

The left and right accumulators contain the return information pertaining to the operation. If the operation is successful the sign of the left accumulator is positive. If not, the sign of the left accumulator is negative and contains codes in bits 9 through 0 which indicates the type of failure. The codes and possible failures are:

01 - Discrete Out Inhibited

02 - Discrete Out Disallowed

03 - Data Link Failed

04 - System in Idle Mode

05 - Discrete Out Parity Error

06 - MTOAD Failure

10 - Calling Parameter Input Error

If the left accumulator contains codes 01 or 02, the right accumulator contains the mask word for the inhibited or disallowed discretes. If a bit of the right accumulator is a 1, the corresponding discrete is either inhibited or disallowed. This mask word is octal and right justified; bit configuration corresponds to the input mask words.

4.1.1.2 Check Single or Multiple Discrete Status

The Discrete Service programs provide the capability of checking the state (on or off) of a single discrete or a group of 24 or less discretes. There are a total of six subroutines provided for the various discrete functions; four of these subroutines check the status of the following single discretes:

- Discretes In to the MLC (MDI)
- Discretes In to the LCCC (LDI)
- Discretes Out from the MLC (MDO)
- Discretes Out from the LCCC (LDO)

Two subroutines check the status of a group of 24 or less multiple discretes; they check the status of multiple MDI's and LDI's.

Table 4.1-2 shows pertinent data for each subroutine.

Table 4.1-2 Discrete Status Check Subroutines

Discrete Function	Discrete Type	Computer Checked	MACRO Name	Link Operand	Accessed By	
					Test Program	Function Executor Routine
Single MDI	IN	MLC	SMDI	SAEA	X	X
Single MDO	OUT	MLC	CMDO	SANA	X	X
Multiple MDI	IN	MLC	MMDI	SAFA	X	X
Single LDI	IN	LCCC	SLDI	SAGA	X	
Single LDO	OUT	LCCC	CLDO	SAMA	X	
Multiple LDI	IN	LCCC	MLDI	SAHA	X	

A description of the subroutine calling sequences is given in Table A-2, Appendix A, of Volume I.

The two single MDI and LDI Subroutine calling sequences require two calling parameters, which are specified in the control words as shown in Figure 4.1.4. These parameters are the discrete number and the expected status code.

The discrete number is the number of the single discrete whose status is to be checked. Since the discretes checked are both discretes in to the computer, the numbers range from 1 through 3024. This calling parameter is specified in bits 11 through 0.

The expected status code indicates the discrete state which is to occur when checked. This code is specified in bit 23. If the state is to be on, then bit 23 is a one; if the state is to be off, then this bit is a zero.

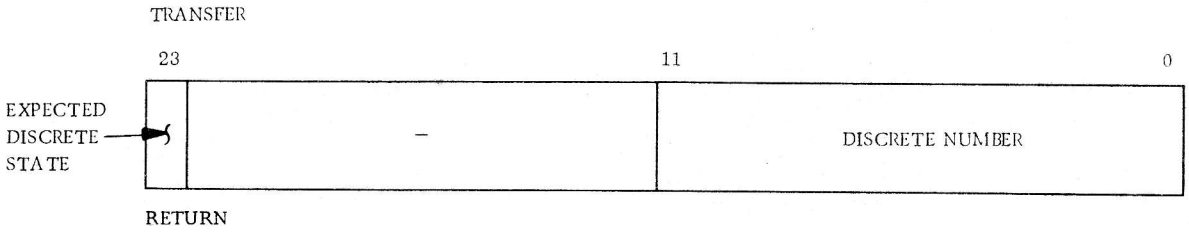


Figure 4.1.4 Single Discrete Status Check Control Word

The single MDO and LDO subroutine calling sequences require only one calling parameter, which is the discrete out number. Since the discretes checked are discretes out from the computer, the number ranges from 01 through 2015. This parameter is specified in bits 11 through 0 of the control words shown in Figure 4.1.4.

The Multiple MDI and LDI subroutines each require three parameters specified in the two control words as shown in Figure 4.1.5. The parameters are the low and high discrete numbers, and the expected status mask word. The low and high discrete numbers specify a number of discretes to check at one time.

NOTE

The numbers specified must not fall outside of a predefined group of 24 discretes.

These groups are as shown in Figure 2.1.1 of paragraph 2.1.1. The low and high discrete numbers are specified in bits 23 through 12 and bits 11 through 0, respectively.

The expected status mask word is a binary-bit configuration indicating the state of the specified discrete group, and must be right justified. The high-order bit indicates the expected status of the mask word high-order discrete number. The low-order bit of this mask word corresponds to the low-order discrete number. This mask word is specified in the second control word of the calling sequences.

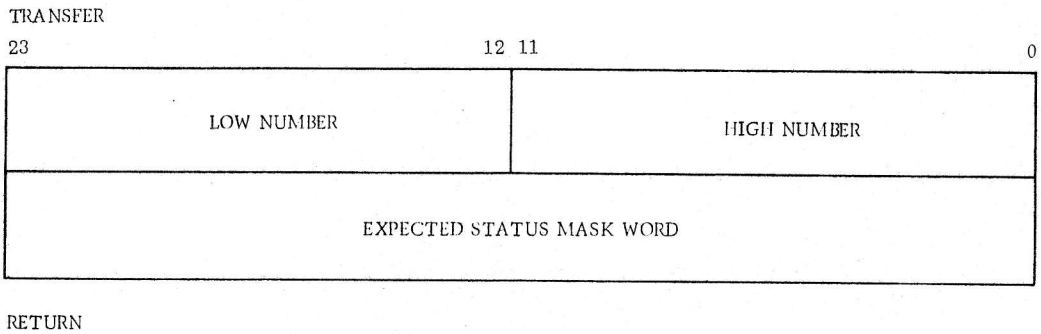


Figure 4.1.5 Multiple Discrete Status Check Control Word

When the status has been checked for either operation, control is returned to the location following the control word(s). This is a normal return. Information pertaining to the status of the operation is returned in the left and right accumulators.

The return data for the single and multiple MDI and LDI operations indicates whether the status of the discretes was or was not as expected. If the status was as expected, bit 23 of the left accumulator is zero. If the status of the discretes was not as expected, bit 23 of the left accumulator is a one.

If the left accumulator has bit 23 set to one for the multiple discrete check, the right accumulator tells which discretes had an unexpected status. If a discrete is of unexpected status, the corresponding bit is a one. The high-order bit corresponds to the high discrete number.

Return data for the single discrete out checks indicates whether the discrete was on or off. If the discrete out was on, bit 23 of the left accumulator is zero and the remaining bits are non-zero. If the discrete out was off, bit 23 is a one and the remaining bits are non-zero.

It should be noted that three of the discrete status check subroutines can be accessed by test programs operating in the LCCC to check discretes in the MLC. Requests from the LCCC to check discretes of the MLC are sent over the Data Link. If the Data Link fails before the status check is made, this fact is indicated by setting the left accumulator to zero.

4.1.1.3 Issue MDO and Check MDI

An Operating System Discrete program provides test programs resident in the LCCC with the capability to issue an MDO and check whether an MDI response has occurred within a specified time interval. When the MDO is issued, a timer is set at the same instant. The subroutine then checks whether the MDI has returned before the time interval has elapsed. If the MDI did not respond within the specified time interval, then applicable information is placed into the accumulators. If the MDI did respond within this specified interval, the time difference between the MDO issuance and the MDI response is returned to the test program. Figure 4.1.6 shows the timing relationship.

The calling sequence used for this operation is given in Table A-2, Appendix A, of Volume I. The test programmer links to this subroutine with a TSPO SAJA, 7. He must specify the calling parameters in the two control words of the calling sequence. These parameters are:

- MDO number and state issued
- MDI number and returned state
- Maximum time in milliseconds in which MDI response is expected
- State to set MDO after time expiration

The format for these parameters is shown in Figure 4.1.7. The first control word contains the MDO state to be issued and the MDO and MDI numbers. The bit positions for these parameters are also shown in the figure. The second control word contains the state to set the MDO after the time has expired, the MDI state to expect, and the maximum time in milliseconds. (see figure 4.1.7).

When the Operating System completes the operation, return information is placed into the left and right accumulators. Control is then transferred to the core location immediately following the the second control word. The left and right accumulators contain the status information and error codes. The contents of the left accumulator are indicated as follows:

- + Time difference between MDO issuance and MDI response (see figure 4.1.6)
- 0 MDO issued, but no MDI response
- - Inhibited MDO, disallowed MDO, or improper MDI response

If the left accumulator is negative, bits 8 through 0 contain specific error codes. These codes are:

001 - Inhibited MDO

002 - Disallowed MDO

003 - Data Link Failure

004 - System in Idle Mode

005 - MDO Parity Error

006 - MTOAD Failure

010 - Calling Parameter Error

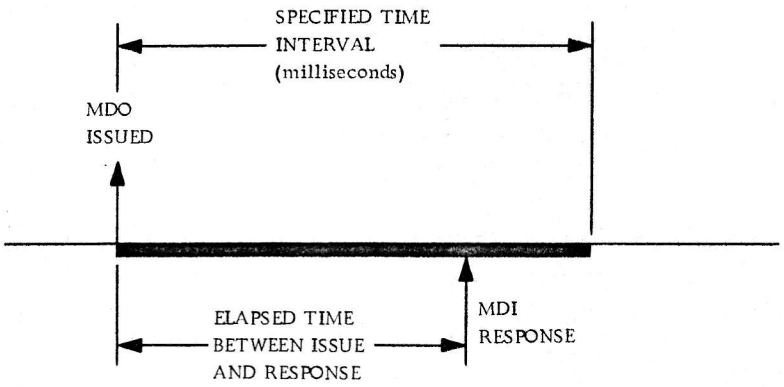


Figure 4.1.6 MDO-MDI Timing Relationship

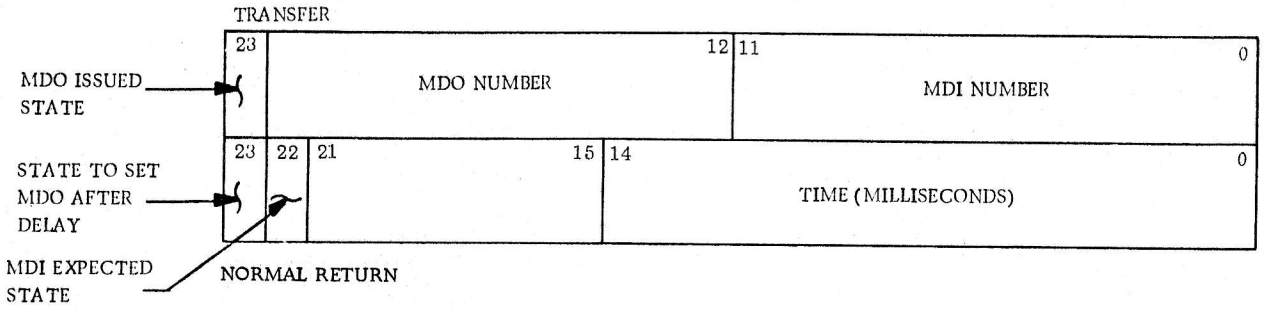


Figure 4.1.7 Issue MDO and Check MDI Response Control Word

Additional coded error information may be placed into the right accumulator if the left accumulator is negative. The additional error information is right justified. These codes are:

01 - MDO inhibited on first pass

02 - MDO disallowed on first pass

03 - MDO inhibited on second pass

04 - MDI did not respond within specified time interval

4.1.1.4 Issue a Switch Selector Function

The Operating System provides the capability to issue commands to a specific vehicle stage via a Switch Selector channel. A test program resident in the LCCC or a Function Executor routine in the MLC may issue a command.

The calling sequence used to issue a Switch Selector command is given in Table A-2, Appendix A, of Volume I. The test program or routine links to this subroutine with a TSPO SAKA, 7. The test programmer specifies four calling parameters in the control words of the calling sequences as shown in Figure 4.1.8. The parameters are as follows:

- Channel Number
- Selected Stage Code
- Read Command Code
- Complement Channel Number

The channel number indicates which of the 128 possible commands is to be issued. This is specified in bits 7 through 0.

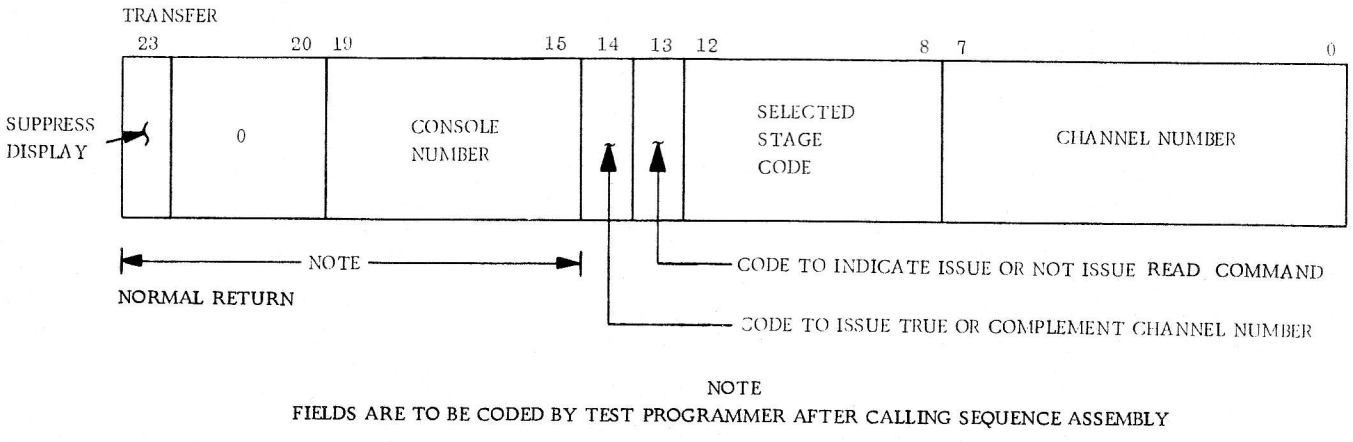


Figure 4.1.8 Switch Selector Control Word

The selected stage code indicates the stage to receive the command. The stage codes are specified in bits 12 through 8. The stage codes are as follows:

- 01 = S-IC
- 02 = S-II
- 04 = S-IVB
- 10 = S-IB
- 20 = IU

The read command code indicates that the command is or is not to be issued when verification of the signal command code is received. This code is specified in bit 13. If the command is to be issued, bit 13 is a one; if not, the bit is a zero.

The complement channel number code indicates whether the complement or true channel number is to be issued. This code is specified in bit 14. If the complement is to be issued, bit 14 is a one; if not, this bit is a zero.

The control word contains two additional fields in the high-order bit positions. These fields are coded after the use of the MACRO instruction, since the MACRO only specifies the four above-mentioned fields. These two fields are the display console number and suppress message display.

The display console number field is specified in bits 19 through 15. If the test programmer wants the Operating System to display the error message on the console requesting the program, the console number field is set to zero. Since any console may request execution of a Function Executor routine, the test programmer coding a routine may specify the Display Console on which a message is to be displayed.

The suppress message display field is specified in bit 23. If the test program is to have its own message displayed, this must be set to one.

NOTE

Bits 22, 21, and 20 of the control word must be zero.

When the discrete program has issued the command, it returns control to the core location which follows the control word. The left and right accumulator contain the operation status information. The left accumulator indicates the following:

+ = Command successfully issued
- = Hardware failure
0 = Environment failure

For a successful issue of a Switch Selector function, the left accumulator is positive. No message is displayed. The right accumulator is zero.

A hardware failure indicates that a command has not been verified in the vehicle, the left accumulator is negative. The right accumulator contains error code 14.

Environment failures result in Switch Selector functions not being issued. The left accumulator is zero. An environment failure code is placed into bits 8 through 0 of the right accumulator. These codes and the failures are as follows:

01 = Calling parameter error

02 = Channel and stage inhibited by ESE

03 = Stage power off

04 = Stage zero signal not present
05 = Stage is hardware-inhibited
06 = Manual panel using Switch Selector
07 = Interface not clear
10 = Inhibited MDO during issue
11 = Inhibited MDO by ESE during Read
12 = IU power off
13 = LVDC using Switch Selector
14 = Data Link failure
15 = System in IDLE mode
16 = Read verify signal not received

NOTE

For all error codes, the Switch Selector command is not issued. If the Switch Selector issue fails, the complement channel number code is not automatically issued, but must be specified for issue by the test programmer in the calling parameter.

4.1.2 LVDC Input/Output Interface

The Operating System's Launch Vehicle Digital Computer (LVDC) Interface program enables test programs and function executor routines to communicate with the LVDC in the following ways.

- Read data from or write data to the LVDC.
- Notify the LVDC that data is available.
- Command the LVDC to perform a specific routine (status command).

The LVDC Interface program is resident in both the LCCC and the MLC. The LCCC program accepts the requests to perform these functions and sends the request to the MLC program, which performs actual LVDC data input/output processing. This data and signal flow is shown in Figure 4.1.1 of this volume.

4.1.2.1 Program Linkage

Program linkage between the LVDC and the test program or function executor routine establishes communication with the LVDC. The LVDC Interface program provides calling sequences which may be used to request execution of the LVDC data and control functions. These calling sequences are given in Appendix A of Volume I.

The first instruction of the calling sequence is a transfer instruction to link the test program or function executor routine with the LVDC Interface program. This instruction is a TSPO SFLT, 7. The next instruction of the calling sequence contains the control word, and the control data for each type of request. The control word format is shown in subsequent paragraphs.

The next two instructions are reserved for error and normal returns from the LVDC Interface program. If this program detects that the Data Link failed or that the operation was not executed, return is made to the error location. The left accumulator will contain error information in bit 23. If the Data Link failed, bit 23 is a zero. If the operation was not executed, bit 23 is a one. If the operation was successfully completed, the LVDC Interface program transfers control to the normal return core location which is coded as required by the programmer. The following is a coding example for performing an LVDC read:

Description	Coding
Enter	TSPO SFLT, 7
Control Word	BCD, READ
Error Return	TRA ERROR
Normal Return	As Required

4.1.2.2 Test Program and Function Executor Routine LVDC Communication Cells

Word 5 of test program and word 4 of function executor routine communications cells contains the address of the LVDC data input buffer. It is reserved for those programs which communicate with the LVDC. Bits 14 through 0 of the cell contain the address of the test program's or function executor routine's LVDC data input buffer. Bits 23 and 22 of this word are used for control information. When data has been received from the LVDC, the address of the input buffer is

accessed, data is placed into the buffer, and bit 23 of the cell is set to one. No more data is placed into the buffer until bit 23 is set to zero. When the test program has accessed and processed the data, it sets this bit to zero.

Bit 22 of the cell is used to indicate that an error message has been received from the LVDC. This error message is not placed into the buffer area but into a special bank zero buffer for LVDC errors. When an error is received, bit 22 of the communication cell is set to one to indicate to the test program that the error has occurred. When the test program acknowledges the error and processes accordingly, the test program resets bit 22 to zero. Figure 4.1.4 shows this communication cell.

4.1.2.3 LVDC Input to Test Programs and Function Executor Routines

A test program or function executor routine must supply an input buffer for incoming LVDC data. Figure 4.1.9 shows the buffer format and a coding example of this input buffer.

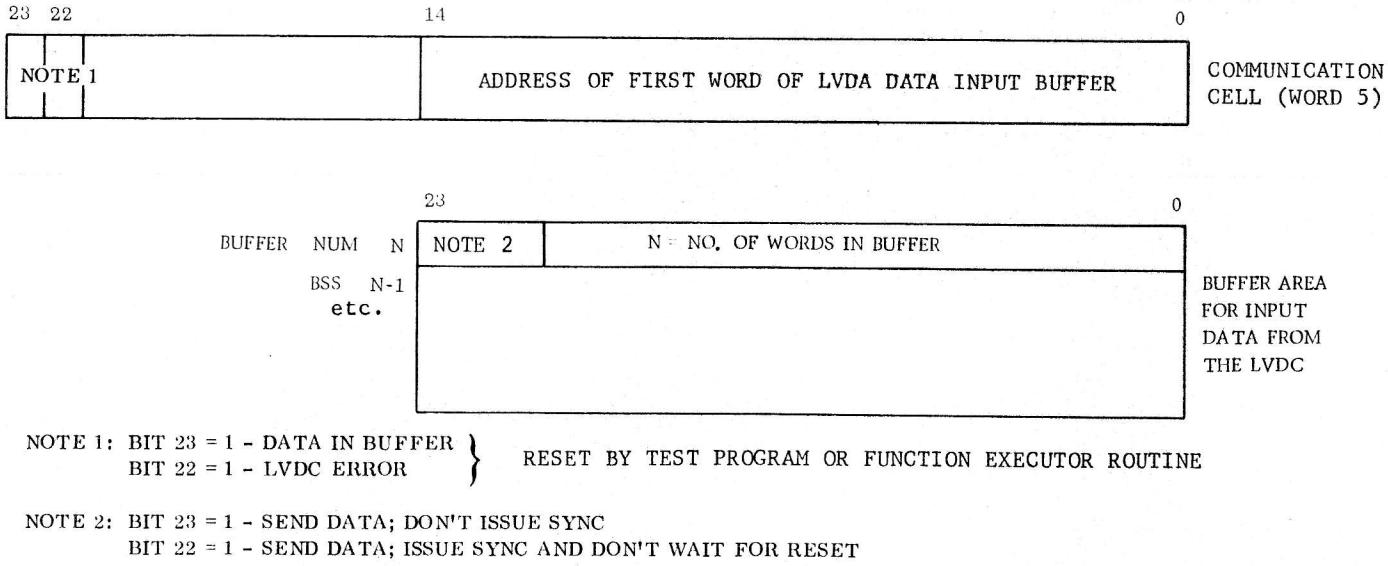


Figure 4.1.9 LVDC Input Data Communication Cell and Buffer Structure for Test Programs and Function Executor Routines

The test program or function executor reserves storage location words for LVDC input data. The first word of this buffer area contains the number of actual words that are needed to hold the input data. The size of the buffer area can be up to 512 words, and the address of the first word of the buffer is placed into the fifth word of the test program or function executor communication cell.

When the Operating System has data for the test program or function executor routine, it accesses the address of the buffer area from the communication cell. It then places the actual number of words required for the input data into the first word of the buffer area and places the data into the locations which follow the number of data words. Since the Operating System has determined the actual number of data words used, the requesting program can now determine how many words to process.

Another means to access data from the LVDC is to use the macro instruction READ. This macro allows the test program or function executor routine to access one LVDC data word. When one LVDC data word is accessed, two RCA-110A data words are generated. The first generated word contains 9 tag bits and the two least significant bits of the LVDC word. The second generated data word contains the 24 most significant bits of the LVDC data word. Figure 4.1.10 shows these word structures.

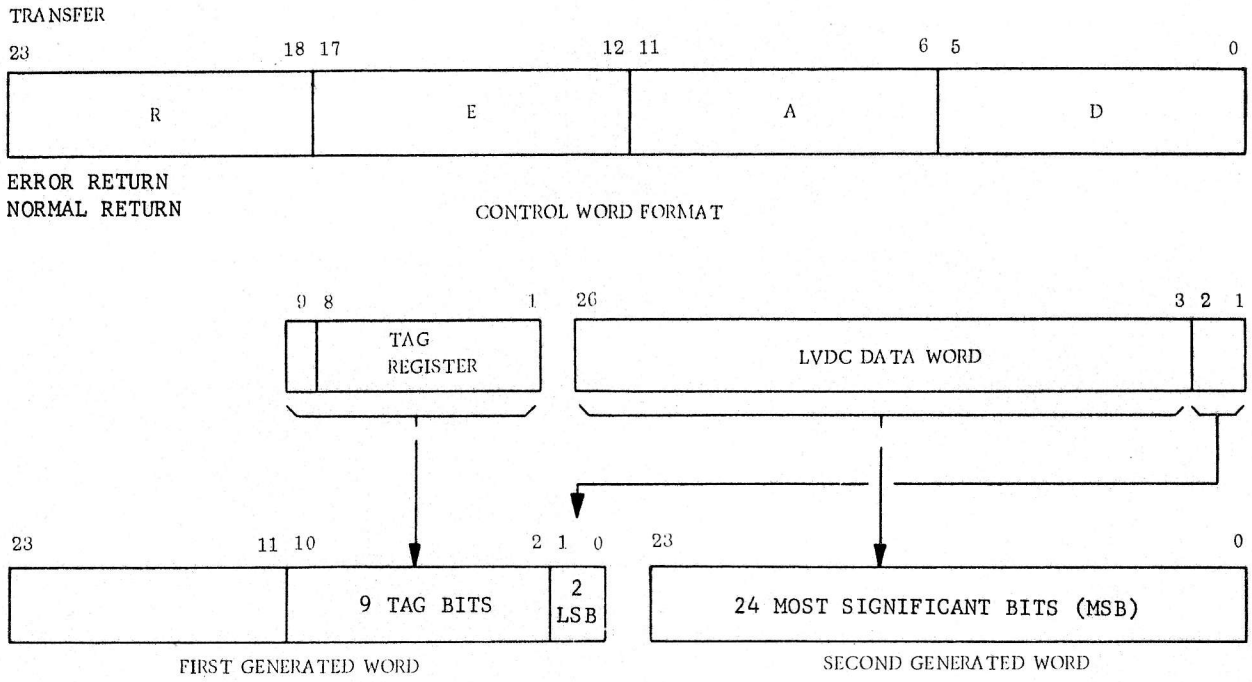


Figure 4.1.10 LVDC Data Word Conversion

The control word of the macro instruction contains the BCD word READ. This indicates the type of request to the LVDC Interface program. After the LVDC data word is accessed, the data is placed into the left and right accumulators. The left accumulator contains the second generated word and the right accumulator contains the first generated word.

4.1.2.3.1 LVDC Error Table: An error message may come from the LVDC and be placed in a bank zero buffer area instead of the test program or function executor routine input buffer area; in this way the LVDC program creates an error table. If a test programmer required data contained in the LVDC Error table he accesses this table by referring to its IVAR, CLFER in the LCCC, and SLVERR in the MLC. The size of this table is 14 data words; each contains specific error information. Figure 4.1.11 shows the error table format.

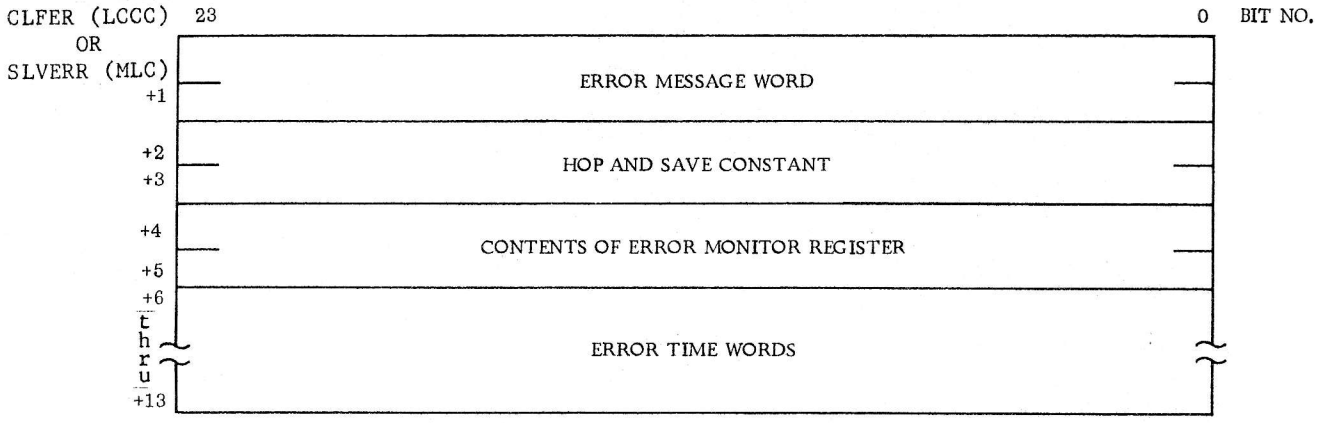


Figure 4.1.11 LVDC Error Table Format (LCCC and MLC)

4.1.2.3.2 LVDC History Table: The LVDC history table is a three-word storage area in bank zero of both the LCCC and the MLC. This storage area contains the status codes of the twelve previously performed LVDC routines.

A test program or function executor may determine which routines have been performed by accessing this table by its IVAR, CCELV. The format of this table is shown in Figure 4.1.12.

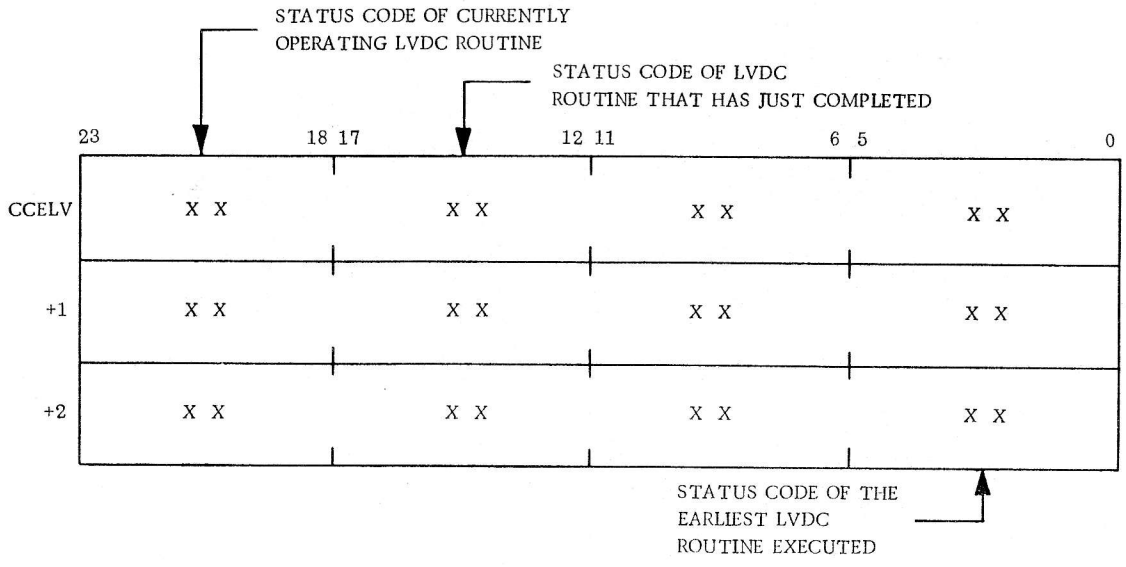


Figure 4.1.12 LVDC History Table Format (LCCC and MLC)

The status codes of the LVDC routines appear in chronological order. The status code of the routine in operation appears in the most significant six bits of CCELV, and so forth, with the status code of the first of the last twelve sequential routines in the least significant six bits of CCELV + 2.

4.1.2.4 Test Program Output to LVDC

The test programmer may send data to the LVDC by using the macro instruction SWRTE. The control word format and the coding structure for this request is shown in Figure 4.1.13. The calling parameter ADDRS indicates to the LVDC Interface program the storage location of the data area. This address is specified in bits 14 through 0. The first word of the data area contains the number of data words. This word is followed by the output data that is sent to the LVDC.

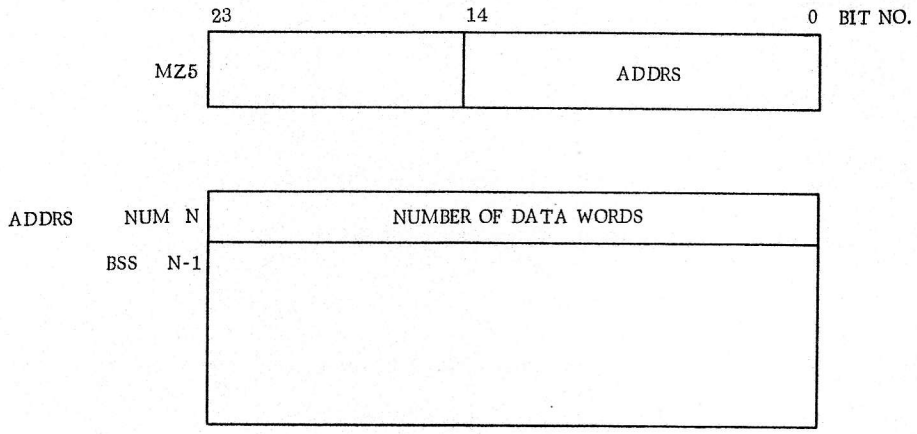


Figure 4.1.13 Output Storage for LVDC Data

4.1.2.5 Control Requests to LVDC

The control functions for commanding the LVDC to perform a specific routine and notifying it that data is available, are requested by the macro instructions SCMND and SYNC, SYCNB, and RSYNC. The control word format to request these functions is shown in Figure 4.1.14.

The control word requires input parameters and a request code. The input parameters are specified by the test program in bits 14 through 0 of the control word. The request codes which are assigned by the Operating System are specified in bits 17 through 15.

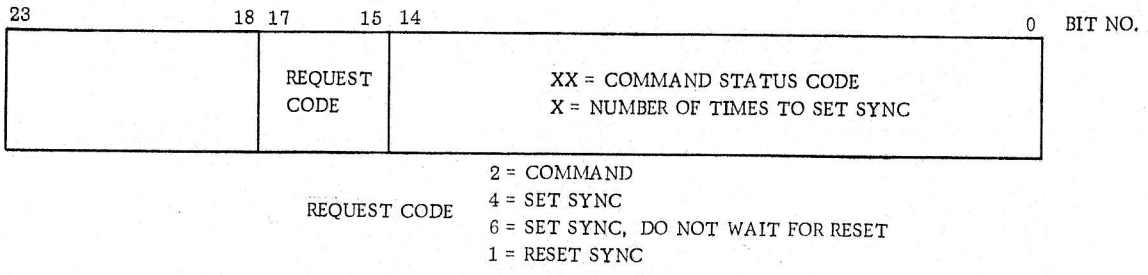


Figure 4.1.14 LVDC Command and Control Words (SYNC Example)

The macro instruction SCMND is used to request that the LVDC perform a specific routine. The request code must be a 2. The input parameter for this request is a two-digit octal code for the routine desired. Regular and special status codes for LVDC routines are given in Tables 4.1-3 and 4.1-4.

The status codes given in the following table are issued to command the specified LVDC routine. Some routines are assigned two codes -- an octal code and a complement code. For these routines, the status command reflects the octal code, whether the octal code or the complement code was used.

Table 4.1-3 LVDC Routine Regular Status Codes

Octal Code	Complement Code	Routine Name or Function
00		Mode Sort Routine Ground Entry
01	76	Program Continue Routine
02	75	Accelerometer Pulse Count Routine Entry
03	74	Gimbal Angle Monitor Routine Entry
04	73	Discrete Input Routine Entry
05	72	Discrete Output Routine Entry
06	71	G&C Steering Test Entry
07		DOM Test Routine Entry
10	67	Ladder Error Test Routine Entry
11		Prepare To Launch (Simulated Flight) Without A Platform
12	65	Prepare To Launch (Flight) With A Platform
13	64	Exercise Attitude Command Routine-Mode 1 Entry
14	63	Computer Interface Unit Read Routine Entry
15	62	Switch Selector Routine Entry
16		Sector Sum Check Routine Entry
17	60	Exercise Attitude Command Routine-Mode 2 Entry
20	57	Sector Dump Routine Entry
21		Verify and Prepare to Load Routine Entry
22	55	Memory Error Test
23	54	Load Routine Entry
24	53	Minor Loop, Switch Selector and Real Time Counter Tests Entry
25	52	Unassigned
30		Prepare to Launch (Simulated Flight) With A Platform
31		Repeatable Simulated Flight Linkage Routine
32		Telemetry Test Routine
35		Command Receiver Linkage Routine
40		Power Up and Initialize Routine
52		Unassigned

The status codes given in the following table are issued, but cannot be commanded. None of the routines have complement codes.

Table 4.1-4 LVDC Routine Special Status Codes

Octal Code	Routine Name or Function
26	Forced Begin and Initialize Routine Entry
34	Exercise Attitude Command Routine Error Entry - Initial Converter Still Not Selected
37	Reserved for Simulated Flight
42	Reserved for Flight
43	Exercise Attitude Command Test - Maximum Attained on One or More Axes
45	Read Input Data Subroutine - Overtime Message
46	Exercise Attitude Command Test Driving Towards Null
47	Error Monitor Subroutine Entry
51	Forced Begin Routine Error - EMR Non-Zero
56	Verify and Prepare to Load Routine - Try to Load Without Load of Previous Data (If Any)
66	Completion of Repeatable Simulated Flight Run
70	Reserved for Simulated Flight
77	Forces the EMR Subroutine to Return to Forced Begin FBDA
77	Issued to Mode Register Only When in Special Loader Mode Sort
25	Unassigned
27	Unassigned
33	Unassigned
41	Unassigned
44	Unassigned
50	Unassigned
61	Unassigned

The macro instruction SYNC is used to send a SYNC signal to the LVDC a specified number of times. The LVDC resets the signal to acknowledge receipt of the sync signal. This process is continued until all sync signals have been received by the LVDC. The number of times SYNC is set is specified by the test programmer. After the operation is complete, the number of times the SYNC signal has been set is returned in the left accumulator in bits 12 through 0. This SYNC count is made in order to check program performance. The request code for a set SYNC operation is a 4.

The macro instruction SYNCB is used when the LVDC is to be notified that data is available. The SYNC signal is only set once and the test program does not wait for the LVDC to reset it. There is no input parameter for this request. The request code for a SYNCB operation is a 6.

The macro instruction RSYNC is used to reset the SYNC signal, instead of having it reset by the LVDC. There is no input parameter for this request. The request code for a RSYNC operation is a 1.

4.1.3 Hardwire Analog Voltage Input

The Operating System provides the test programmer with the capability to read hardwire analog DC voltages. Two programs, the LCCC DC Analog Input Interface and the MLC DC Analog Input, provide this capability for the test programmer.

The LCCC DC Analog Input Interface program is used by test programs resident in the LCCC. This program allows the test programs to read DC hardwire analog voltages from the MLC.

The MLC Analog Input program is used by MLC function executor routines to read MLC DC hardwire analog voltages. This program also accepts requests to read DC analog voltages from test programs resident in the LCCC sent via Data Link from the LCCC DC Analog Input Interface program.

These Operating System analog programs provide macro instructions to initiate read operations. The macro names and their coding are given in Table A-5, Appendix A. The test programmer must provide the test point number in the calling parameters. When the operation is completed, return indicators are placed in the right and left accumulators.

There are no error indications returned by the MLC DC Analog Input program. The LCCC DC Analog Input Interface program returns an unsuccessful indicator if the Data Link failed during processing of the request.

4.1.4 Digital Data Acquisition System (DDAS)

DDAS provides communication capability for test programs to read DDAS measurements and calibrations from the vehicle. The Operating System provides test programs operating in the LCCC and function executor routines operating in the MLC with the capability to communicate with the DDAS via the Digital Data Acquisition System Interface program. A block diagram of DDAS hardware and software is depicted in Figure 4.1.1.

Data may be read from all stages of the vehicle in various modes. The control data required for reading of DDAS data by the Digital Data Acquisition System Interface program is specified in calling sequences control words and control C words. The control data is specified by the test programmer.

The Digital Data Acquisition System Interface program provides the test program with the capability to request DDAS read operations. It also provides the capability to request termination or deletion of a previous read request. These requests are initiated with macro instructions and their calling sequences as given in Table A-7, Appendix A.

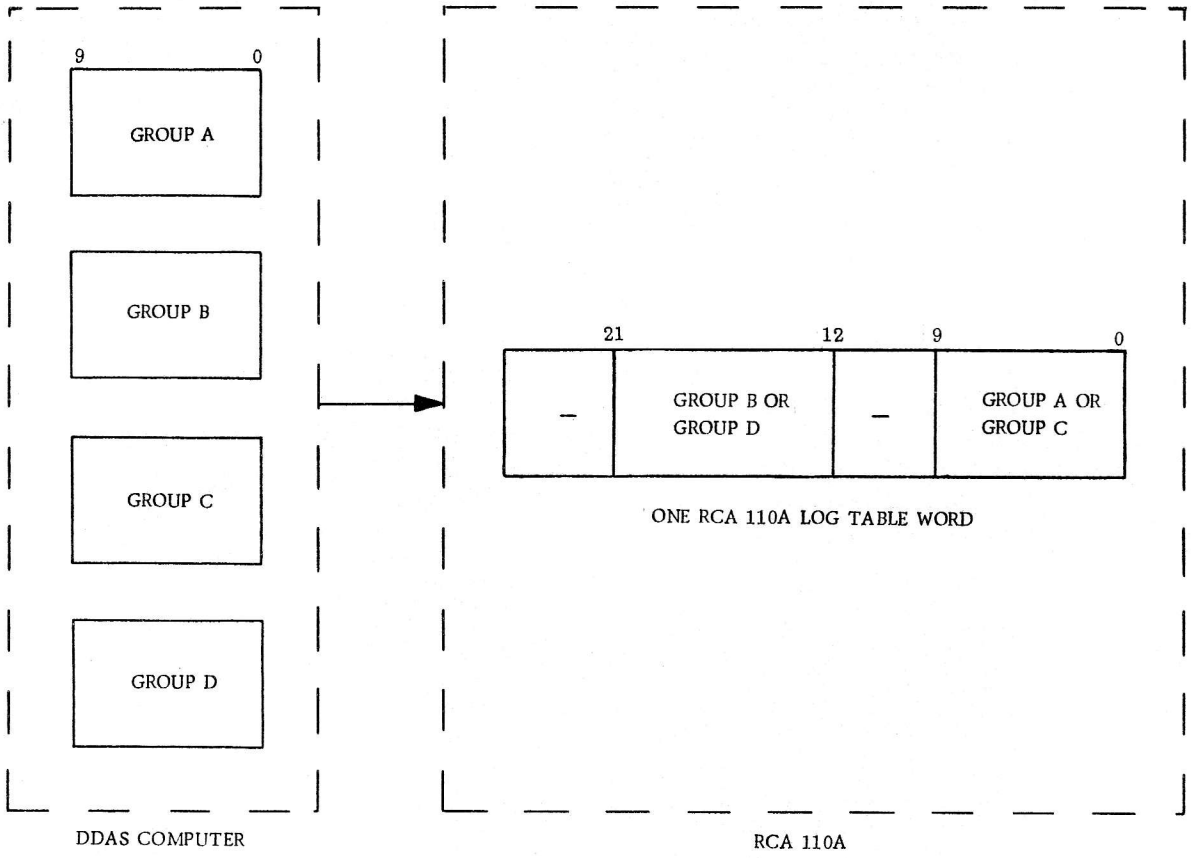
4.1.4.1 DDAS Log Table

The DDAS Log Table is the storage location which contains the DDAS data read from any or all of the vehicle stages. This data comes from the vehicle via the DDAS Computer Interface into the RCA-110A. The format of the words contained in the DDAS Computer Interface is either 40- or 20-bit words for each DDAS address. If the transfer mode transfers 40 bits, two RCA-110A data words are needed for the DDAS Log Table. If the transfer mode transfers 20 bits, then one RCA-110A word is needed. Figure 4.1.15 shows the format of the RCA-110A word for each DDAS group.

4.1.4.2 DDAS Control Words

When a test programmer requests a DDAS read operation, he must supply three or more control words for effective data transfer. These control words are identified as A, B, and C. Control word A contains information pertaining to the DDAS Log Table. Control word C contains information pertaining to each stage from which DDAS data is read. If DDAS data is to be read from more than one stage, the same number of C control words are specified as there are stages read. Control word B contains information pertaining to control word C. Since there may be more than one C control word specified, information pertaining to control word C must be specified in control word B. Specific information for each control word follows: Control word A contains the following:

- Total number of RCA-110A data words used for the DDAS Log Table (the number of words is the total number required for DDAS from all stages).
- DDAS Log Table address



NOTE:

1. GROUPS A THROUGH D CONTAIN 10 BITS OF INFORMATION EACH.
2. C AND D ENTER ONLY IF 40 BITS ARE BEING TRANSFERRED.

Figure 4.1.15 Digital Data Acquisition System / RCA-110A Interface Data Word Format

Control Word C contains:

- Particular stage number
- DDAS stage address (multiplex, frame, channel, group)
- DDAS submode
- Number of DDAS Computer Interface sequential data words for the particular stage specified

Control word B contains the following:

- Bank location of the DDAS Log Table
- Total number of C control words
- Total number of DDAS Computer Interface data words for all DDAS read from all stages

The format for these control words is shown in Figure 4.1.16. A more detailed description for coding is given in the RCA-110A Programmers Reference Manual.

4.1.4.3 DDAS Transfer Modes

A test programmer may read DDAS data in three types of modes: log, single scan, and special.

The log mode is a DDAS Input/Output Data Channel transfer mode identified as a "continuous update" mode. When the DDAS data has been updated, it is transferred from the specified DDAS address to the computer log table. Data updated for each DDAS address occurs every 250 milliseconds. When the last DDAS address (specified in the last C control word) has been read, control returns to the first C control word and additional data continues to be read into the computer log table. The DDAS read operation continues until a termination is requested by the test program or a hardware error occurs.

NOTE

To avoid difficulty in identifying data stored in the log table, log table length should be equal to the total number of words to be read, or a multiple thereof. The log table length must always be an even number of words for log mode transfer.

In the single scan mode, the IODC performs one complete cycle of operation and terminates. The difference between single scan and log mode is that in the log mode the IODC will receive DDAS data only when the specified word has been updated. The IODC will not terminate until the log table is full, half-full, or a terminate command is received.

The special mode transfers 40-bit words to the RCA-110A instead of the normal 20-bit words. This operation is identical to the log mode, except that data is read from only one DDAS address for each C control word. Data is transferred from this address as many times as specified by the word sequence count and stored in sequential locations in the log table.

The DDAS Interface program provides additional options with the above transfer modes. These options are:

- Sequential Transfer
- Updated Block Transfer
- Normal Sequence
- Log Mode Termination (when one log table is half-full)

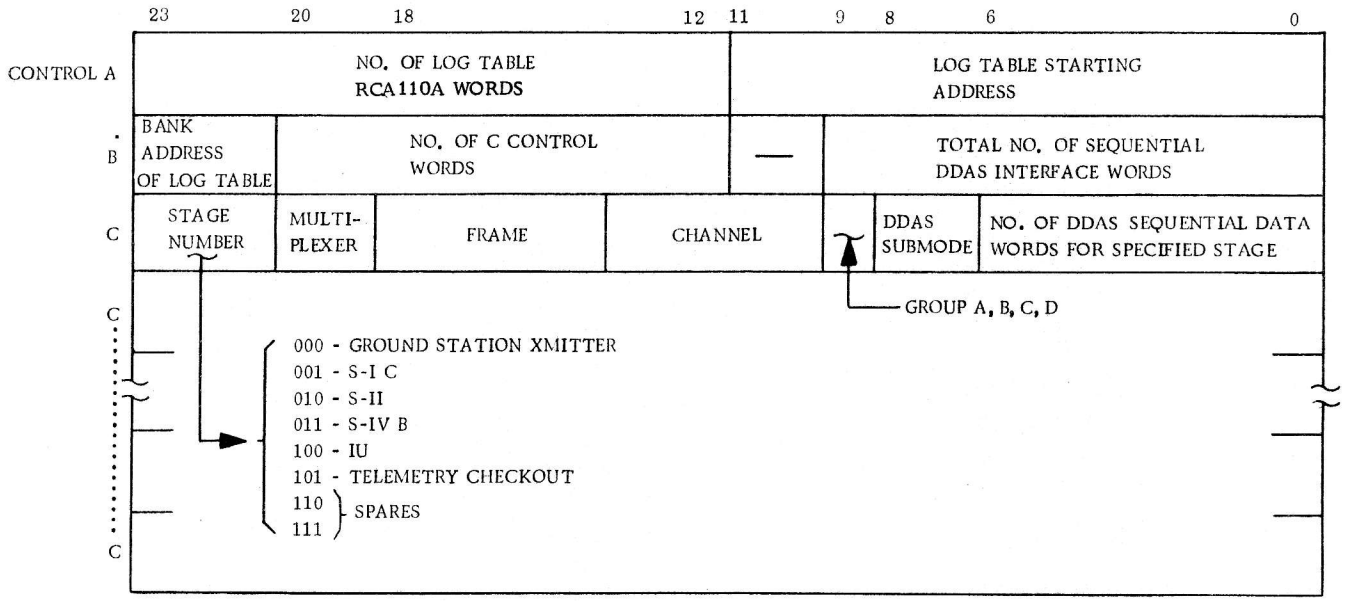


Figure 4.1.16 DDAS Control Word and Storage

Sequential transfer, update block transfer, and normal sequence are DDAS Computer Interface transfer modes which are further explained in the RCA-110A Programmer's Reference Manual.

The log mode termination capability will only interrupt log mode processing when half of the log table data words have been updated.

4.1.4.4 DDAS Calibration and Measurement Processing

Whenever a test program or function executor routine is using the DDAS Interface program for calibration and measurement of DDAS readings, it must set a bit in communication cell CCXCT to indicate to the DDAS Interface program which stage of the vehicle is being used for the calibration or measurement. If a test program or routine tries to read DDAS data from the same stage of the vehicle on which calibration and measurement is being performed, the DDAS data being read may be invalid. Prior to performing a read, the DDAS Interface program checks the communication cell CCXCT to see if the stage from which DDAS data is to be read is also being used for calibration or measurement. If so, the DDAS Interface program sets a bit in the read request control word to indicate the above condition.

The format of this communication cell is shown in Figure 4.1.17. One of the bits 7 through 0 is set to one if a particular stage is being used to calibrate or measure a DDAS reading. Bit position 0 is set to one if the DDAS Ground Station transmitter is being used. Bit position 1 is set to one if the S-IC stage is being used and so on.

NOTE

When a calibration or measurement test program or routine has completed, it must set the bit to zero.

4.1.4.5 Log Mode Interrupt Processing

A test program may request a log (continuous update) mode read operation. When this log table has one-half of its contents updated or is completely updated during the DDAS log mode read operation, a DDAS I/O Data Channel termination interrupt occurs. The Interrupt Processor of the Operating System has control and transfers control to the test program's Log Mode Interrupt Subroutine.

Any test program which uses log mode processing must contain a Log Mode Interrupt subroutine (refer to Figure 2.1.2). The address of this subroutine must be specified in the test programs fourth communication cell.

Prior to requesting a DDAS log mode read request, the test program must set bit 23 of the communication cell to a one (1). If a DDAS I/O Data Channel termination interrupt occurs, the Interrupt Processor checks bit 23 of the communication cell to see if log mode interrupt processing is being performed. If bit 23 is a one (1), the Interrupt Processor accesses the address of the test program Log Mode Interrupt subroutine in the communication cell and transfers control to the subroutine. When the subroutine performs the necessary housekeeping, it returns control to the Interrupt Processor. If a DDAS I/O Data Channel termination interrupt occurs and bit 23 is not a 1, an error has occurred and a message is displayed and printed.

NOTE

The DDAS Log Mode Interrupt Subroutine must be of short duration since most interrupts are inhibited during the subroutine.

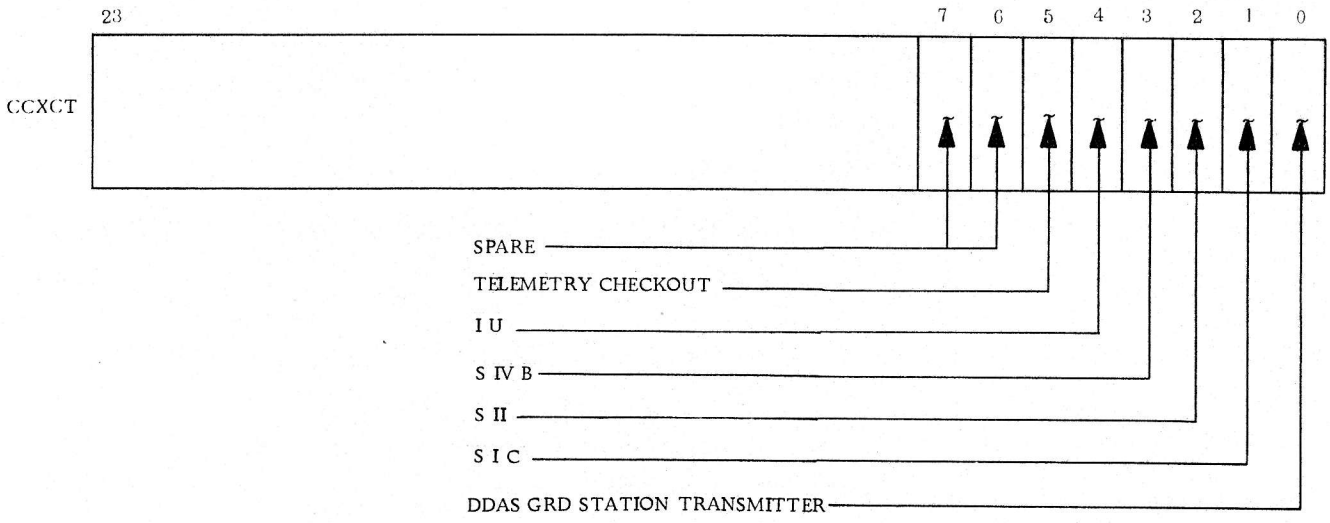


Figure 4.1.17 Calibration In Progress Communication Cell

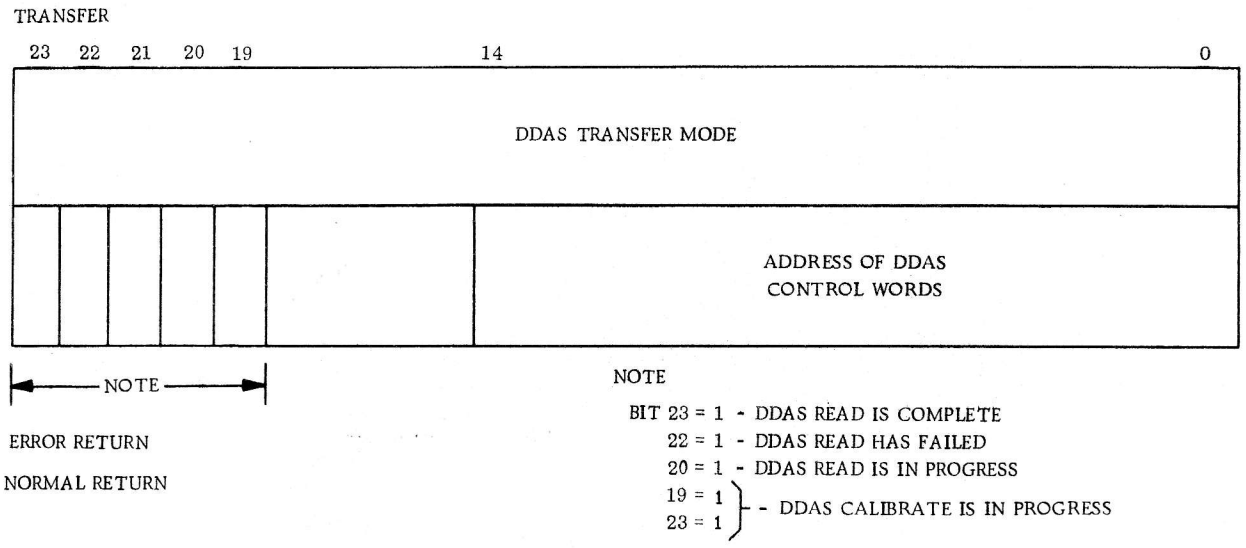


Figure 4.1.18 DDAS Read Request Word Format

4.1.4.6 DDAS Read Requests

The DDAS Interface program provides test programs and Function Executor routines the capability to request a DDAS read operation in various DDAS transfer modes. There are two types of read requests. One stacks the read requests and returns control to the calling program immediately; the other read request performs the DDAS read operation before returning control to the test program.

The DDAS read request which is stacked may be performed by the MACRO instruction XDDS and its calling sequence as given in Table A-7 of Appendix A. When this MACRO is coded, the test programmer must specify two calling parameters. These calling parameters are assembled into the calling sequence control words as shown in Figure 4.1.18. The parameters are as follows:

- DDAS Transfer Mode Code
- Address of DDAS Control Word Table

The DDAS transfer mode code indicates to the Operating System which transfer mode the test programmer desires to read DDAS. There are 8 possible modes and the code for each is contained in the first control word. The codes and corresponding mode are as follows.

0 = Single Scan-sequential transfer with no remote DDAS capability
2 = Log Mode - updated block transfer
3 = Special Mode
4 = Log Mode - updated block transfer with termination when the log table is 1/2 full
5 = Special mode with termination when the log table is 1/2 full
10 = Log Mode - updated block transfer with a normal sequence submode
12 = Log Mode - updated block transfer with a normal sequence submode and a termination when the log table is 1/2 full

The control word address is the core location of control word A and is specified in the second control word. When the stack read request is made, the request may or may not be stacked. There are two return locations provided for each possibility. They are identified as the NORMAL and ERROR return. When the request has been stacked, then the DDAS Interface program returns to the NORMAL return. Within the LCCC, when the request has not been stacked, return is made to the ERROR return. The possible error condition is Stack Full. Within the MLC, returns to the ERROR return may be caused by any of the following:

- Improper Scan Mode specified
- Incorrect Log Table length specified
- Illegal number of control C words used
- Data Link Failure

When a return is made to one of the return locations, bits are set in the control word containing the mode code. The bits and their significance are:

Bit 23=1 DDAS read is complete
22=1 DDAS read has failed
20=1 Requested read in Progress
19=1, 23=1 DDAS calibrate test is in progress. Data may be bad.

The Read and Hold request initiates a DDAS read operation in the single scan-sequential transfer mode only. Control is not returned until the entire read operation is complete. Request for this operation is made with an XDDRH MACRO instruction and its calling sequence.

The test program or Function Executor routine links to the DDAS Interface program with TSPO SXRH, 7. The only calling parameters required for this request is the address of the DDAS control word table. This address is specified in the first control word of the read and hold calling sequence.

The return locations and parameters are similar to those as in the stack read request.

4.1.4.7 Terminate or Delete a Read Request

The DDAS Interface program provides the test programmer the capability to

- Delete a previous read request that has been stacked.
- Terminate a current DDAS read operation.

The terminate or delete request may be performed by specifying the XDDAT MACRO instruction and its calling sequence. The test program or Function Executor routine links to the interface program with a TSPO SXAT, 7.

The calling parameter that must be supplied by the test programmer is the address of the first control word of the original read request. This address may be symbolic or absolute and is specified in the control word as shown in Figure 4.1.19.

When the request has terminated the read operation or has deleted the original read request that has been stacked, control is returned to the core location identified as the normal return. This return location follows the control word. Bits are set in the control word to indicate the results of the request. If bit 22 is set to 1, then the original read request that had been stacked has been deleted. If bit 23 is set to 1, then a read request was in operation and has been terminated.

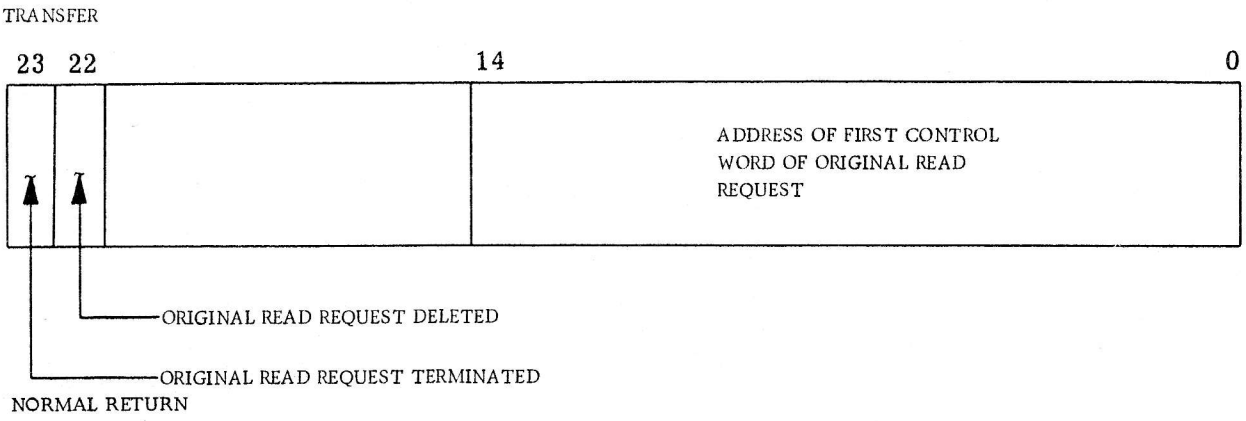


Figure 4.1.19 Terminate or Delete Control Word

4.1.4.8 MLC Remote DDAS Requests

Requests for DDAS data from the MLC are processed by the Remote DDAS/Count-Clock routine. A request is validated and transmitted to the LCCC, where the DDAS reading is initiated. Control is returned to the caller when the request has been transmitted to the LCCC successfully. Complete bits are not set for the caller until the requested DDAS data has been returned from the LCCC. Refer to Table A-7, Appendix A, for a description of calling sequences.

The restrictions on MLC DDAS requests are as follows:

- These requests are limited to two DDAS modes: Single Scan Sequential Transfer and Log Mode Terminate on Half Full.
- Control A, B, and C words must be contiguous in core.
- A maximum of 8 Control C words and 50 DDAS data words are allowed. The data word input buffer must be 7 words larger than the number of expected data words to allow for storage of GMT in binary, and for value of countdown clock in BCD and in binary at the time of DDAS read.

The format of the DDAS data table is as shown below.

23 LSB GMT (BINARY)	Word 1
4 MSB GMT (BINARY)	2
T-DD (BCD)	3
HHM (BCD)	4
M.SS (BCD)	5
23 LSB CDC (BINARY)	6
HOLD INDICATOR / 4 MSB CDC (BINARY)	7
DDAS DATA WORDS (MAXIMUM OF 50)	8 (Address indicated in Control A word)
	etc.

NOTE

The address specified in the Control A word is the starting address of the DDAS data words in the buffer table. The information contained in the first 7 words of this table precedes the DDAS data words.

4.2 DISPLAY SYSTEM INPUT/OUTPUT

The Saturn Operational Display System provides the test programmer with the capability to communicate with the system's fifteen display consoles through the Display Control Computer (DCC) in the Launch Control Center. The Saturn Operating System provides three programs which are the communication link between the test program and the test engineer. These programs are:

- Display Data Link (DLSL)
- Display Service (SDSR)
- Remote Display Service (SRDY)

The Display Data Link program is resident in the LCCC. It accomplishes two things for the test programmer: it accepts test engineer parameter input and routes this input data to the test program via Test Program Control (ET01); and it accepts messages from the test program via the Display Service Program and sends them to the DCC.

The Display Service program, resident in the LCCC, serves the test programmer by placing binary and BCD data from the test program into acceptable format for the DCC and the display console. This program supplies X and Y console coordinates, refresh memory addresses, and routes all test program messages to the proper console (these three functions may be specified by the test programmer if he so chooses). The data and control information is then sent to the Display Data Link program.

The Remote Display Service program is resident in the MLC and serves as a communication link between the Operating System programs, function executor routines, and the display console via the Display Service and Display Data Link programs. The Remote Display Service program transfers messages via the Data Link to the Display Service program.

Figure 4.2.1 shows a block diagram of the major elements of data flow and communication links between the test programs, function executor routines, and the display console.

4.2.1 Operational Display System

The display consoles of the Saturn Operational Display System provide an interface between the test program and the test engineer.

The display portion of each console consists of a cathode ray tube (CRT) capable of display on a 12 by 12 inch area.

4.2.2 Types of Displayed Data

Data to be displayed upon the console may be classified as variable or background data.

4.2.2.1 Variable Data

Variable data is defined as information which is dynamic in nature and is continuously being scanned for update. Variable data is generated and continuously updated during execution of a test program or function executor routine in the LCCC or the MLC. The test programmer may provide that this data coded BCD or preformatted (in refresh memory format) within the test program or function executor routine, transmitted to the DCC, and processed as necessary for display. The test programmer may also choose to code, for the Display Description tape, a display description containing a format table. The format table serves to convert raw binary variable data generated or retrieved by the test

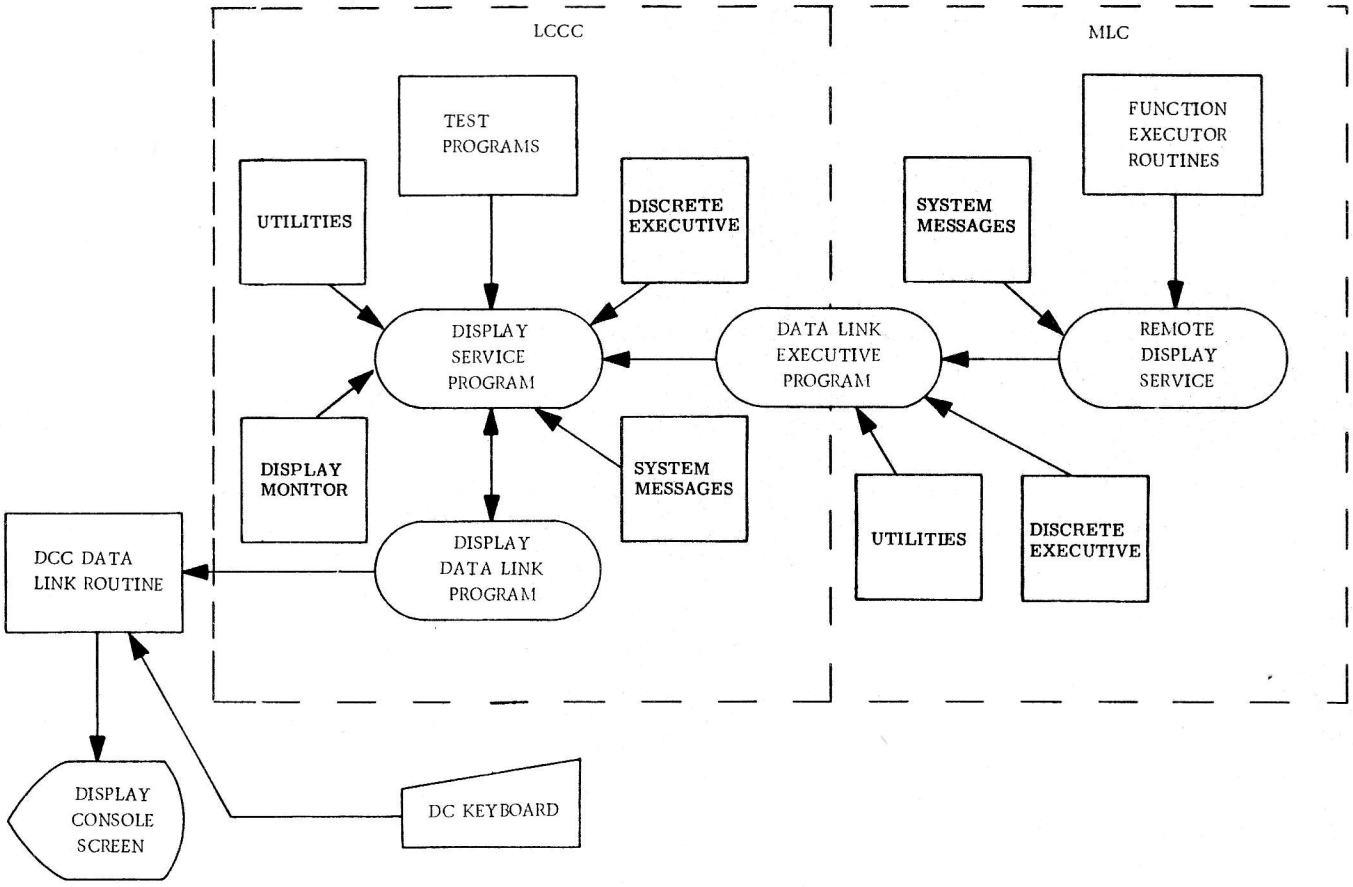


Figure 4.2.1 Display System Input/Output Data Flow

program or function executor routine into a form suitable for display (i.e., in refresh memory (RM) format). Refer to paragraphs 4.2.7.5, 4.2.7.6, and 4.2.7.7 for further explanation of variable data.

4.2.2.2 Background Data

Background data is defined as information which is static in nature and is not subject to change by the Operational Display System itself. This data is used to provide reference information for variable data which is continuously being updated. Background data may be provided by only one of the following for each set of variable data: by the precoded data (which is preformatted or in RM format) in a Display Description tape record, or by a 35mm film chip projected by the Operational Display System's Random Access Slide Projector (RASP). Refer to paragraphs 4.2.7.3 and 4.2.7.4 for further explanation of background data.

4.2.3 Digital Data and Slide Data

Variable data and the precoded background data stored within a display description are commonly termed digital data; these types of data are generated for display via the LCC and the DC computers.

Data provided by a projected 35mm film chip is termed slide data; this data becomes an image generated for display via TV camera.

4.2.4 Digital Data Symbols

The test programmer may utilize digital data symbols of two main types--alphanumeric characters and vectors. Alphanumeric characters consist of alphabetic, numeric, and special characters, each of which may appear in a large or small configuration, and may be displayed either sequentially or randomly.

Vectors may be used to provide the axes of graphs, as symbols indicating magnitude and direction or a relay closure, or simply as lines to separate columns of alphanumeric characters.

Data to be prepared for generation via slide may consist of any type of data, but will mainly consist of alphanumeric characters and vectors (refer to paragraph 4.2.6.3 for further explanation of display of slide data or slide data mixed with variable digital data).

4.2.5 Display Screen Configuration

The CRT display area is designed so that all displayed information may be referenced in terms of its 12-inch square X/Y coordinate system. The 12-inch square has been defined in terms of a 512 by 512 digital unit (dit) grid matrix. The zero reference point (X/Y coordinate 000/000) is at the upper left corner of the matrix. The coordinate 000/511 is the lower left corner reference point; the coordinate 511/000 is the upper right corner reference point; and the 511/511 coordinate is the lower right corner reference point.

This grid system yields 262,144 intersection points which the test programmer may uniquely reference with an X/Y coordinate. However, areas at the top and bottom of the CRT display area are reserved for the display of special information by the Operating System and Display System, and should not be referenced by the test programmer. (Exception: when digital data is mixed with slide data, the Special Slide Flag bit provided in the format of a display description may be coded to disable both reserved areas from their usual use, and to allow variable digital data to be displayed within these areas.)

Information displayed in the area at the top of the screen (termed status area) is as follows: test program or ATOLL procedure status; Operating System status; operator number and console page; GMT in hours, minutes, and seconds; and additional information pertaining to display console operation. The status area locations are 000 through 022 in the Y direction, and 000 through 511 in the X direction.

The area at the bottom of the screen is used to display information as follows. The upper half of this area is used by the Operating System to display information to the test engineer; the lower half (termed scratch pad area) is used to display information entered on the keyboard by the test engineer (which he may choose to retain for future reference). This area of the console covers grid locations 455 through 511 in the Y direction, and 000 through 511 in the X direction.

The test programmer may then reference all data for display in locations 000 through 511 in the X direction, and in the respective Y locations for the various types of data given below:

- For vectors: Y locations 023 through 454.
- For small characters: Y locations 031 or 035 through 454.
  If characters are referenced at location 031, they will appear when displayed immediately adjacent to any data displayed in lower status area; if characters are referenced at location 035, they will be displayed appropriately spaced below any data displayed in lower status area.
- For large characters: Y locations 039 or 048 through 454.
  If characters are referenced at location 039, they will appear when displayed immediately adjacent to any data displayed in lower status area; if characters are referenced at location 048, they will be displayed appropriately spaced below any data displayed in the lower portion of the status area.

Further requirements for referencing data are given in paragraph 4.2.6.

4.2.6 Referencing of Displayed Data

The test programmer may reference each displayed character or vector according to the display console grid scheme described in paragraph 4.2.5. The test programmer must reference data according to the type and combinations of data to be displayed, as follows:

- Display of digital data (variable data, variable data with precoded data as background, or precoded data alone). Refer to paragraphs 4.2.6.1 and 4.2.6.2 for details of referencing.
- Display of digital (variable) data mixed with slide data as background, or slide data alone. Refer to paragraph 4.2.6.3 for details of referencing.

4.2.6.1 Referencing Alphanumeric Digital Data

The test programmer references all alphanumeric digital data by specifying the X/Y console coordinates at the lower left corner of the character. A small character may be specified for display in each 10 by 13 dit (along the X and Y axes, respectively) area of the allocated display area. A maximum of 51 small characters may be specified for display in each row along the X axis; and a maximum of 39 small characters for display in each column along the Y axis. A large character may be coded for each 20 by 26 dit area. A maximum of 25 large characters may be specified for display in each row along the X axis; and a maximum of 19 for display in each column along the Y axis.

4.2.6.2 Referencing Vectorial Digital Data

The test programmer defines a resultant vector with an X/Y set of coordinates for both the origin and the termination points; for example, 020/030 and 040/060 are the X/Y coordinates for the origin and termination points, respectively, of the vector thus defined. (A resultant vector is any vector which can be defined in terms of two (X and Y axis) component vectors).

The vector must then be defined in terms of its X and Y component vectors. The maximum length of an unconnected vector is two inches. If the test programmer wishes to display a resultant vector of more than two inches, he must first divide it into a number of vector segments, each with a length of two inches or less. Each vector segment would then be broken down into its X and Y component vectors, and the number of digital unit (dit) increments for each component determined. The dit increments for the X and Y components are then added to the origin coordinates of the first vector segment to determine its termination point (which is also the origin point for the second vector segment). The number of X and Y increments for a 2-inch vector is determined as follows:

a. Determine the inch length of the X and Y component vectors by measuring them off.

b. Calculate the number of increments along each component with the ratio formula:

$$\frac{X}{12} = \frac{I(X)}{512} \quad \text{and} \quad \frac{Y}{12} = \frac{I(Y)}{512}$$

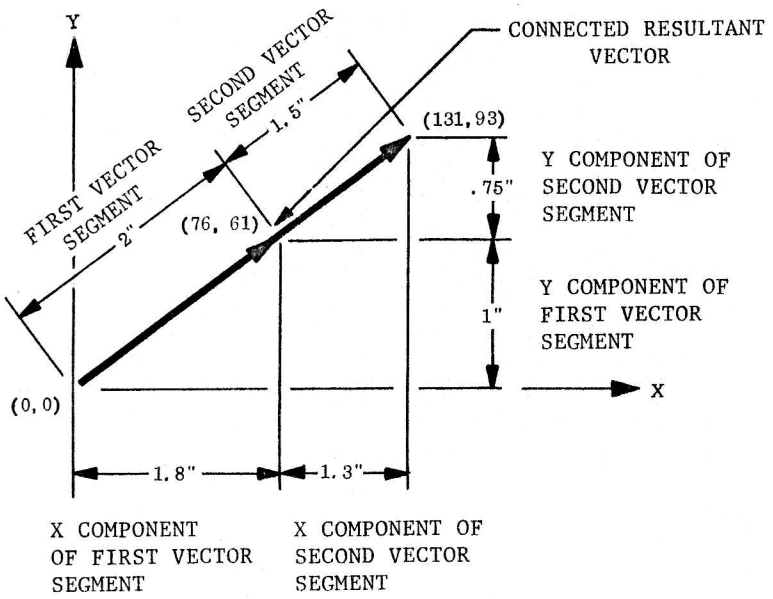
where:

I(X) = Number of increments along X component, in dits
I(Y) = Number of increments along Y component, in dits
12 = Length and width of CRT display area, in inches
512 = Number of increments along each dimension of CRT display area, in dits

c. Add the calculated number of increments to the X and Y coordinates of the origin point to determine the X and Y coordinate of the termination point.

For a number of connected vector segments, the test programmer must specify an origin point for the first connected vector segment, and termination points for the remaining connected vector segments.

The figure below illustrates the method for determining the two termination points of vector segments making up a connected resultant vector, and the data word coding formats to represent them. The resultant connected vector of 3.5 inches is divided into two connected vector segments of 2 and 1.5 inches. The X and Y component vectors are determined to be 1.8 inches and 1 inch, respectively, for the first vector; and 1.3 inches and .75 inch, respectively, for the second vector. According to the ratio formula given above, the number of increments is determined to be 76 by 61 dits for the X and Y components of the first vector segment, and 55 by 32 dits for the X and Y components of the second vector segment. The termination point of the first vector segment (also the origin point of the second vector segment) is then determined by adding the first set of increments to the origin point of the first segment. The termination point of the second vector segment is determined by adding the second set of increments to the termination point of the first vector segment. The coordinates of each vector are shown in the refresh memory data word formats of the figure.



RATIO EQUATION FOR FIRST VECTOR SEGMENT

$\frac{X_1}{12} = \frac{I(X_1)}{512}$ $\quad \frac{Y_1}{12} = \frac{I(Y_1)}{512}$

$I(X_1) = 76.8 = 76$ $\quad I(Y_1) = 61.5$

RATION EQUATION FOR SECOND VECTOR SEGMENT

$\frac{X_2}{12} = \frac{I(X_2)}{512}$ $\quad \frac{Y_2}{12} = \frac{I(Y_2)}{512}$

$I(X_2) = 55.5 = 55$ $\quad I(Y_2) = 32$

23	22 ... 14	13	12	11	10 ... 2	1	0	BIT NO.
P	X = 0				Y = 0	0	1	

POSITION WORD (VECTOR ORIGIN)

23	22 ... 14	13	12	11	10 ... 2	1	0	BIT NO.
P	X = 76	0			Y = 61	1	0	

FIRST VECTOR TERMINATION POINT

23	22	13	12	11	10 ... 2	1	0	BIT NO.
P	X = 76 + 55 = 131	0			Y = 61 + 32 = 93	1	0	

SECOND VECTOR TERMINATION POINT

Figure 4.2.2 Vector Segment Calculation and Refresh Memory Word Formats

4.2.6.3 Referencing Digital Data Mixed with Slide Data

Figure 4.2.3 illustrates the display console format defined for display of digital data mixed with slide data or slide data alone. Although displayed information can originate on any grid point, this format has been developed to provide a standard set of grid coordinates for digital data and to be consistent with character origin grid locations specified in the Sanders Associates "Saturn V Operational Display System Program Specifications," dated 2 April 1965. Refer also to the "Saturn V Operational Display System Slide Specification, " MSFC No. III-6-602-64, IBM No. 67-F11-0009, dated 15 September 1967, for further information.

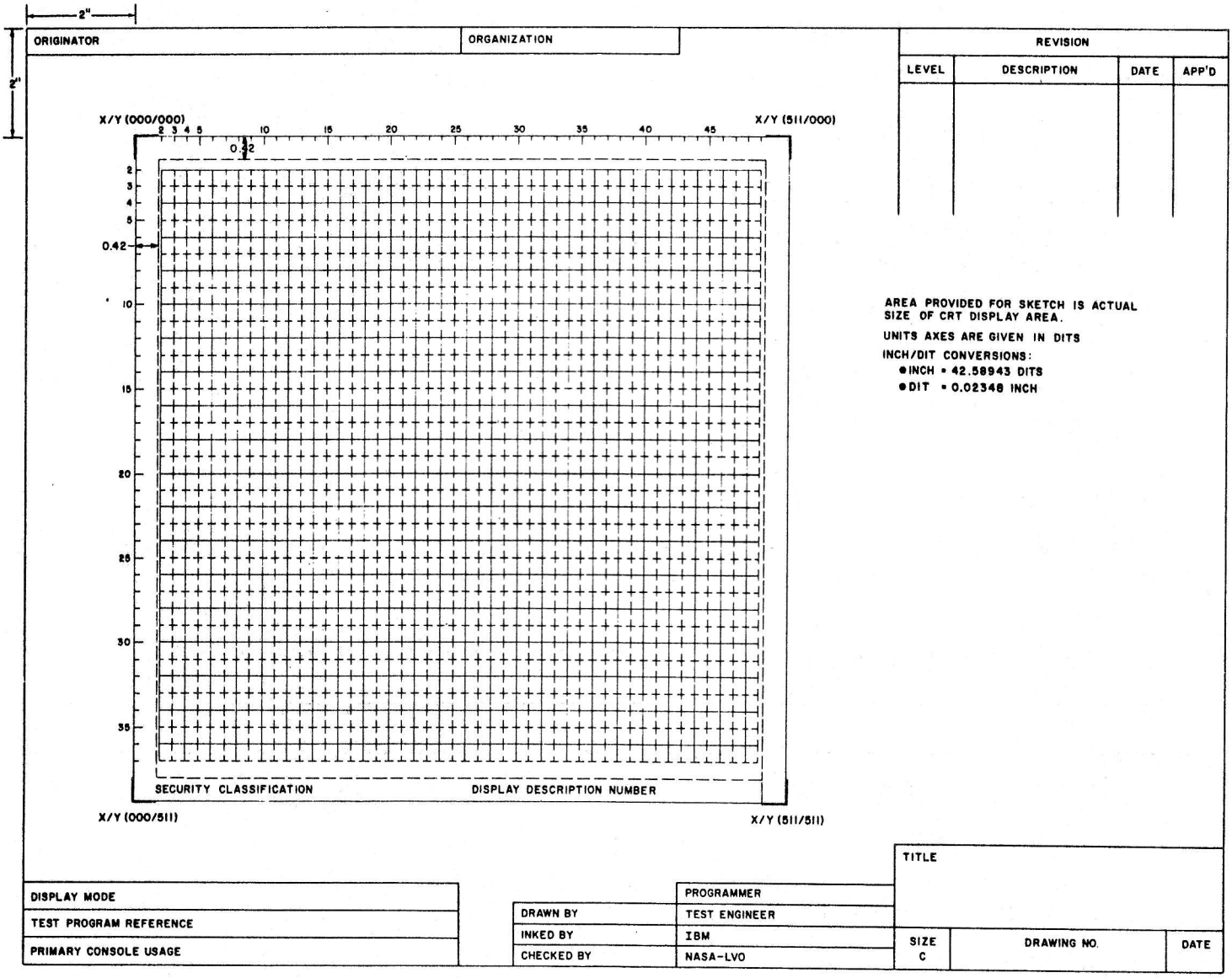


Figure 4.2.3 Display Console Grid Format for Display of Digital Data Mixed with Slide Data or Slide Data Alone.

The grid scheme serves three purposes:

a. Provides a 1/2 inch border within the 12-inch square display area to allow for slide/digital worst-case compatability conditions.

b. Provides grid reference units (boxes) for digital data so that the characters, when displayed under worst-case conditions, will not appear outside the area allowed by the boxes. (A small character can be contained in one small grid box, and a large character in four small grid boxes).

c. Facilitates positioning of slide data relative to digital data.

4.2.6.3.1 Requirements for Digital Data

For display with slide data, digital alphanumeric character sizes are specified as follows:

- Small characters are 0.2 inch in height by 0.15 inch in width.
- Large characters are 0.4 inch in height by 0.30 inch in width.

The display console format allows for display of:

- 51 small characters per row
- 25 large characters per row
- 39 rows of small characters
- 19 rows of large characters

4.2.6.3.2 Requirements for Slide Data

Slide data is permitted to be of any type and maximum size, but must conform to the minimum requirements given in the table below when it is to be used to group digital data.

Table 4.2-1 Minimum Requirements for Slide Data Used to Group Digital Data

Item	Requirement
Distance between any two distinct groups of data	0.06 inch
Height of slide boxes (used to group digital alphanumeric data)	
a. For small character digital data	1.16 inch
b. For large character digital data	1.36 inch

Table 4.2-1 Minimum Requirements for Slide Data Used to Group Digital Data (Continued)

Item	Requirement
Width of slide boxes	Determined by the area occupied by the sequence of digital characters, summed with both of the following: a. The required margin at left of the character sequence--0.48 inch for small and large characters-- and, b. The required margin at the right of the sequence--0.40 inch for small and 0.31 inch for large characters.

4.2.7 Display Description

A display description (DD) is a tape record of data to be displayed as requested during SLCC programming system operations. A test programmer may specify any one or all of the following types of data in a display description:

- Background (static) data
- Format tables (for use in formatting variable data for display)
- Addresses of items to be DMON-monitored for variable data

A display description must also contain the console reference locations for background and/or variable data to be displayed.

Background data may consist of a predrawn slide or of precoded background data (alphanumeric data, vector data, or both). In any single display description, the test programmer may specify either slide or any type of precoded background data, but never both.

Format tables are data sets which contain conversion coefficients for transformation of any variable data in binary form into engineering units for display. Variable binary data may be displayed in the graphic or tabular modes (graphic mode data consists of analog data; tabular mode data may consist of analog or discrete data). Three types of format tables are required to convert graphic and tabular binary data into a form suitable for display:

- Graphic analog
- Tabular analog
- Tabular discrete

Analog data describes a magnitude (in binary) for either graphic and tabular mode display. These data are considered binary in form when the number of bits of data is specified in the associated format table. Graphic analog data produces numeric data for display.

Tabular discrete data is considered binary in form when only one data bit of each 24-bit word transmitted is interrogated, and this bit is specified in the associated format table. Related to this bit in the format table are two one-word messages to be displayed. One message is displayed if the bit checked is set, and the other is displayed if the bit checked is not set.

A display description may be considered standard or non-standard in format. As shown in Table 4.2-2, 10 numbers (513-522) have been reserved for DD's in standard format. These DD's have not been implemented at present, but are represented by dummy DD's on the Display Description tape (dummies also exist on tape for other DD's not yet in use). The standard DD may contain precoded background data, which would be placed in the Display System's spare refresh memory (one spare accessible by all of the system's present configuration of 15 consoles); format tables for variable data; and/or addresses of predefined DMON items.

The remaining DD's (included under the three categories in Table 4.2-2 called Slides, Operations Lists, and Non-Standard Formats) may be considered non-standard in format; that is, they are display descriptions each specified by the test programmer to meet a unique display requirement. DD's in these categories may contain precoded background data (or, for those in the Slides category, a reference number to a slide); format tables, and/or addresses of predefined DMON items.

4.2.7.1 Display Description Numbers

Each display description has a display description number, running from 1 through 1022. These numbers are assigned to four different display description categories as given in Table 4.2-2.

Table 4.2-2 Display Description Numbers By DD Category

STANDARD DISPLAY DESCRIPTIONS	
DD Numbers by Category	Data Combinations
Standard Formats 513-522	• Variable data alone • Variable data with precoded background data • Precoded background data alone
NON-STANDARD DISPLAY DESCRIPTIONS	
DD Numbers by Category	Data Combinations
Slides 1-512 (DD numbers 11-35 contain predefined DMON items)	• Slide data alone • Slide data with variable data

Table 4.2-2 Display Descriptions Numbers by DD Category (continued)

NON-STANDARD DISPLAY DESCRIPTIONS (continued)	
DD Numbers by Category	Data Combinations
Operations Lists 523-542	• Precoded background data alone
Non-Standard Formats 543-1022	• Varable data alone • Varable data with precoded background data • Precoded background data alone

4.2.7.2 Display Description Categories and Data Types

The following figure classifies display descriptions in two major categories, and gives a breakdown of data types within these two categories.

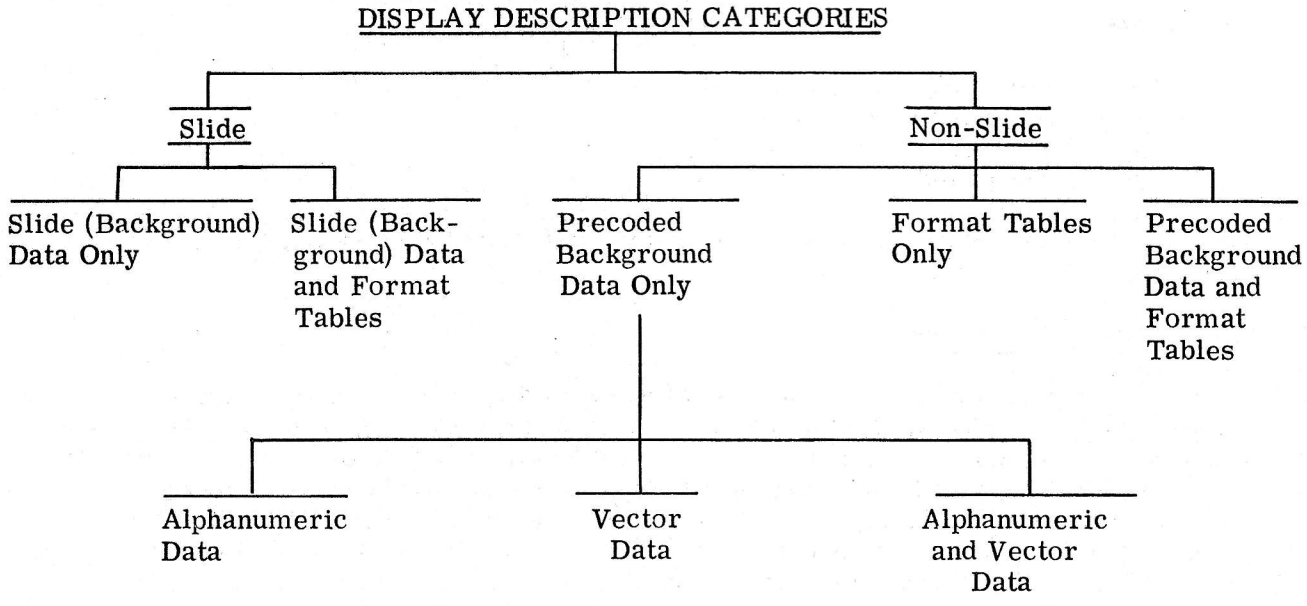


Figure 4.2.4 Display Description Categories and Data Types

Display descriptions are contained on a binary tape generated by the Display Description Tape Generator (DDTG). If a test programmer wishes to access a display description (to use in formatting variable data for display, and perhaps also to provide background information for this data), he specifies a display description number which the test program calls via the Display Service routine (SDSR). After SDSR has retrieved the display description from tape, it sends the format tables to DCC memory, and any precoded background data to DCC refresh memory for display.

Only then may the binary variable data be sent to the DCC for formatting and display.

4.2.7.3 Slide Background Data

Data to be used to provide slide background data is drawn as an engineering master and photographed to provide a 35mm film chip. The film chip is then mounted in a slide mount compatible with the Saturn Operational Display System equipment. The completed slide is placed in one of two available slide projector banks which are accessible by any display console.

Each slide is assigned the DD number of the display description with which it is associated. Slides are grouped according to the two slide projector banks, identified as banks A and B, in which the slides are placed. Bank A is capable of containing 256 slides (DD numbers 1 through 256), and bank B is capable of containing 256 slides (DD numbers 257 through 512).

4.2.7.4 Precoded Background Data

Instead of a reference to a slide, a test programmer may specify precoded data as background information to variable data. Alphanumeric characters coded in BCD, vectors coded in refresh memory format, or both may be specified. Precoded background data displays may consist of any single type or combination of the following data types:

- Small characters (up to 396)
- Large characters (up to 198)
- Connected vectors (up to 198)
- Unconnected vectors (up to 98)

In each display description, the test programmer is allocated a 250-word area in which to place precoded background data.

4.2.7.5 Graphic Mode Data

The display of data in the graphic mode is defined as a display of a general curve which relates a variable parameter to time. In this mode, the test programmer may code for display a curve which best fits the engineering data. A graphic mode curve may be displayed only if the required conversion data is in specified format tables. The curve is calculated using the general form of a fourth-degree polynomial: $a_0 + a_1 x^1 + a_2 x^2 + a_3 x^3 + a_4 x^4$. The variable "x" represents a number (i.e., engineering data such as an analog voltage); the letter "a" represents the calibration coefficients used in conversion of binary data for display in the graphic mode. The calculated value of the polynomial expression yields the Y coordinate for the graph.

The Display Control Computer creates, as an artificial time value in seconds, the value for the X coordinate of the graph. To generate this value, the DCC assumes one second of elapsed time between transmission of each message of binary data (transmission of binary data entails transmission of one item of information at a time; any additional data transmitted at that time is ignored). In the graphic mode, use of this technique restricts the plotting of a curve to that of a curve of one variable.

The following restrictions for the graphic mode must also be considered by the test programmer:

- The Y axis must begin at zero and the computed functions must be positive.
- The Y axis must not be defined as negative. It must extend from zero to some positive value.

4.2.7.6 Tabular Mode Data

Display of data in the tabular mode is defined as a display of alphanumeric characters; for example: a binary representation of discrete information (alphabetics) or a binary expression of an analog measurement (numerics). Tabular mode format tables (analog or discrete) must be placed in DCC memory before variable data is transmitted to the DCC.

When discrete data is sent to the DCC, a prespecified bit is checked and a prespecified message displayed, depending on the bit value. If analog information is to be displayed in the tabular mode, the numeric value is calculated using the general polynomial. Desired accuracy of this value is determined by the number of coefficients prespecified. Thus, if a linear conversion is sufficient, only the first two coefficients must be specified. Only four significant digits of a numeric value calculated in the tabular mode are displayed (if computed value is +8171.009, +8171 appears on the display screen).

4.2.7.7 Format Tables

Format tables contain information which allows the DCC to format variable data for display. The variable data must be a binary representation of the parameter being monitored, which may be an analog or a discrete measurement. When the variable data is to be displayed, the data from the format tables is accessed to convert the data into the tabular or the graphic mode for console display. Tabular displays may come from either analog or discrete input data; graphic displays may come only from analog input data.

Because format tables must be defined for graphic and tabular mode displays, three different kinds of format tables are defined: graphic analog, tabular analog (linear and non-linear), and tabular discrete.

Each of these kinds of tables contains data unique to its function and is assigned an appropriate DD number. The format table data is coded, punched on cards, placed on tape, and accessed when required. The format tables for DD's are arranged into groups which are numbered 0 through 7. The purpose of this group number assignment is to permit flexibility of binary data acquisition; that is, an entire set of raw binary data need not be transmitted for its associated DD, but rather only that binary data related to a group within the defined set of format tables. Group 7 is reserved for graphic analog tables, and group 0 is reserved for predefined DMON items. Groups 1 through 6 are used by test programs for tabular analog and tabular discrete tables.

The graphic analog format table contains the data required to generate graphic curve representation of a series of binary values of analog voltages with respect to time, as follows.

- X and Y coordinate location of the graph's origin.
- Number of data bits in the field to be converted to display format.
- Number of shifts required to left-justify the data field in a 24-bit word.
- Code to indicate whether a negative value is represented in one's or two's complement form.
- Graph range as a function of time.
- Calibration coefficients (from two to five).
- Maximum Y value indicating the maximum range in which the converted data may be graphed on the display console.
- Number of increments in the X and Y direction required for the entire graph.

The tabular analog format table contains the data required to compute and display the numeric value of a binary analog measurement, as follows.

- X and Y coordinate location to display the first digit of the computed value.
- Code to indicate whether large or small characters are desired.
- Number of bits in the data field to be converted to the numeric value.
- Number of shifts required to left-justify the data field in a 24-bit word.
- Code to indicate whether a negative value is represented in one's or two's complement form.
- The upper and lower limits between which the computed value lies.
- Calibration coefficients (from two to five).

The tabular discrete format table contains data to detect the desired discrete information and display information about the bit value, as follows.

- X and Y console coordinate indicating where the discrete message is to begin.
- Code to indicate whether the message is to be displayed in large or small characters.
- Number of shifts required to left-justify the data bit to be checked in a 24-bit word.
- Two four-character messages: one to be displayed if checked bit value is one; the other if checked bit value is zero.

4.2.7.8 Refresh Memory Space Allocations for Display Description Data

Figure 4.2.5 shows the refresh memory space allocations for display description data. For display of variable data (alone or with precoded background data) or precoded background data alone, the test programmer must code for the locations and contents shown in the top half of the figure. When he wishes to display a slide (alone or with variable data), he must code for the locations and contents shown in the bottom half of the figure.

<table>
<tr><td rowspan="2">Variable Data (alone or with Background Data) or Background Data alone on Display Screen</td><td>Locations 4-111</td><td>Locations 112-223</td><td>Locations 224-735</td><td>Locations 736-1647</td><td>Locations 1648-2047</td></tr>
<tr><td>Status Line Data</td><td>Scratch Pad Data</td><td>Background Data, If Any</td><td>EDIT EXECUTE (Keyboard) Data</td><td>Variable (Real-Time) Data</td></tr>
<tr><td rowspan="2">Projected Slide (alone or with Variable Data) on Display Screen</td><td>Locations 4-111</td><td>Locations 112-223</td><td>Locations 224-330</td><td colspan="2" rowspan="2">NOTES
1. Only the first 330 refresh memory locations are displayed when slide and variable data are displayed together.
2. Use of special slide flag (in a display description) enables use of locations 4 through 330 for storage of variable data.</td></tr>
<tr><td>Status Line Data</td><td>Scratch Pad Data</td><td>Variable (Real-Time) Data</td></tr>
</table>

Figure 4.2.5 Refresh Memory Space Allocations for Display Description Data

4.2.8 DD Symbolic and Binary Tape Input

All predefined non-standard display descriptions reside on a binary display description tape. Magnetic Tape Station (MTS) 6, IODC 2, used for display descriptions, is on-line to the LCCC. A test programmer or a test engineer can request a display description by sending certain control words with the display description number to the Display Service program (SDSR). This program searches the display description tape, accesses the desired description, and then sends the control and data words for the DD to the DCC.

The DD tape contains 1 file of 65 records and 15 files of 64 records (actual or dummy), for a total of 1025 records. Each file is closed out by an End-of-File (EOF) mark. The first record of the first file is a 12-word header ID record. Each of the other records on tape contains the information for one display description, or is a dummy record (contains all-zero words, except for the DD number in word 2). Within a file on tape, DD's are placed in ascending numerical order by DD number. Figure 4.2.6 provides a graphic representation of the format of the data as it appears on the binary DD tape.

The binary DD tape is generated via the RCA-110A Display Description Tape Generator (DDTG), from three possible sources of input: symbolic cards, symbolic tape (created via ATOLL), and an old binary master DD tape. The following input combinations may be used.

a. Complete symbolic DD card deck
b. Symbolic update cards and symbolic tape
c. Symbolic update cards, symbolic tape, and old binary DD tape
d. Symbolic tape
e. Symbolic tape and old binary DD tape
f. Symbolic update cards and old binary DD tape

The symbolic DD tape is composed of a variable number of records, each one a 21-word card image in binary packed mode. Each DD is preceded by a 20 word record containing a PROG card image and followed by an END card image, an EOF mark, or both. A double EOF designates the end of the symbolic tape.

Figure 4.2.7 illustrates the system flow in generation or update of a display description tape.

4.2.8.1 DD Data Cards

Display description data is first coded and then punched on cards. These cards are used as input to DDTG to generate a new binary DD tape, or to the ATOLL Compiler to generate a symbolic tape to be used as input to DDTG. There are five types of data cards: A, B, D, 3, and 4. Card type A is used to define linear tabular analog data. Card type B is used to define precoded background data. Card type D is used for tabular discrete data. Card types 3 and 4 are used together to define non-linear tabular analog data or graphic analog data.

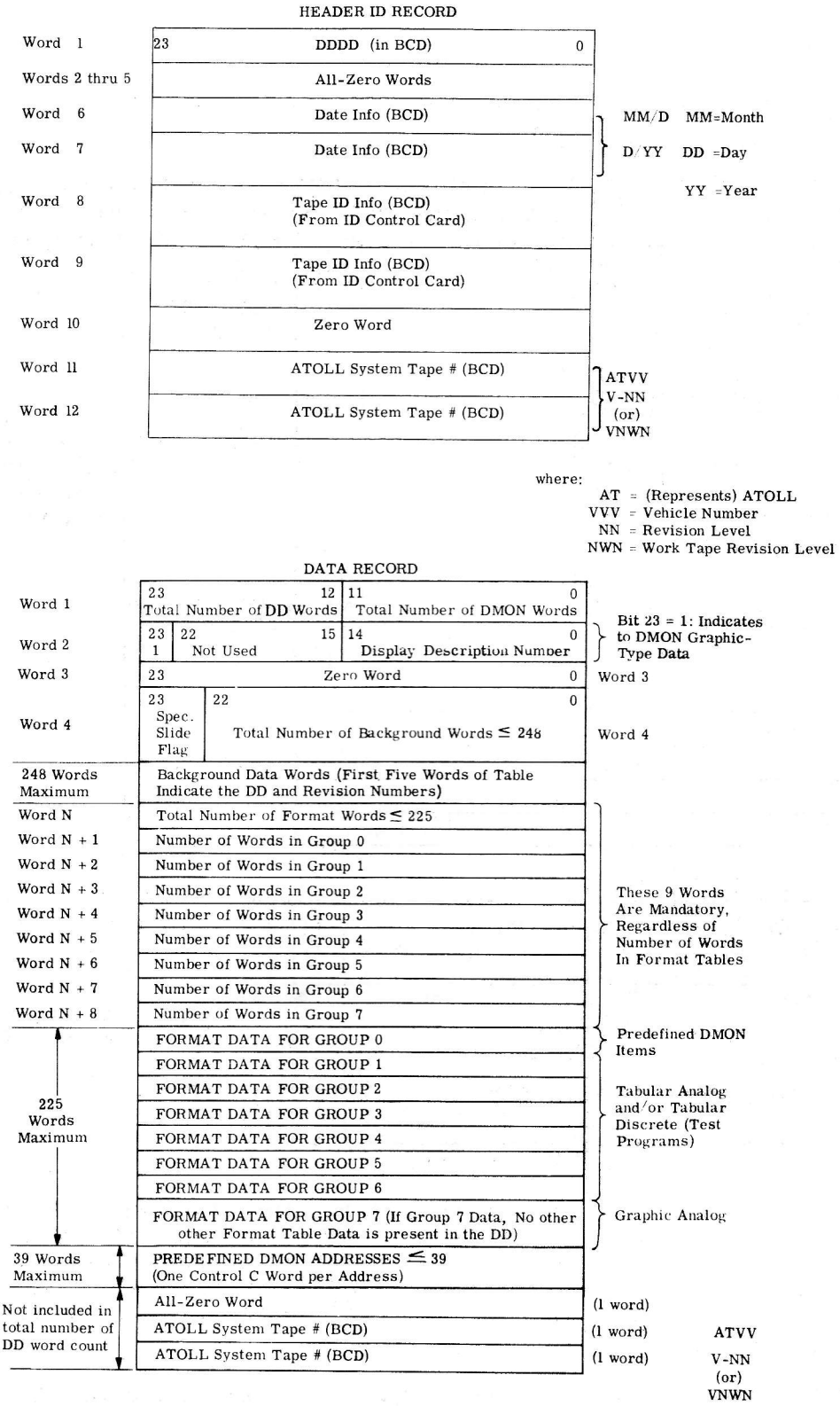
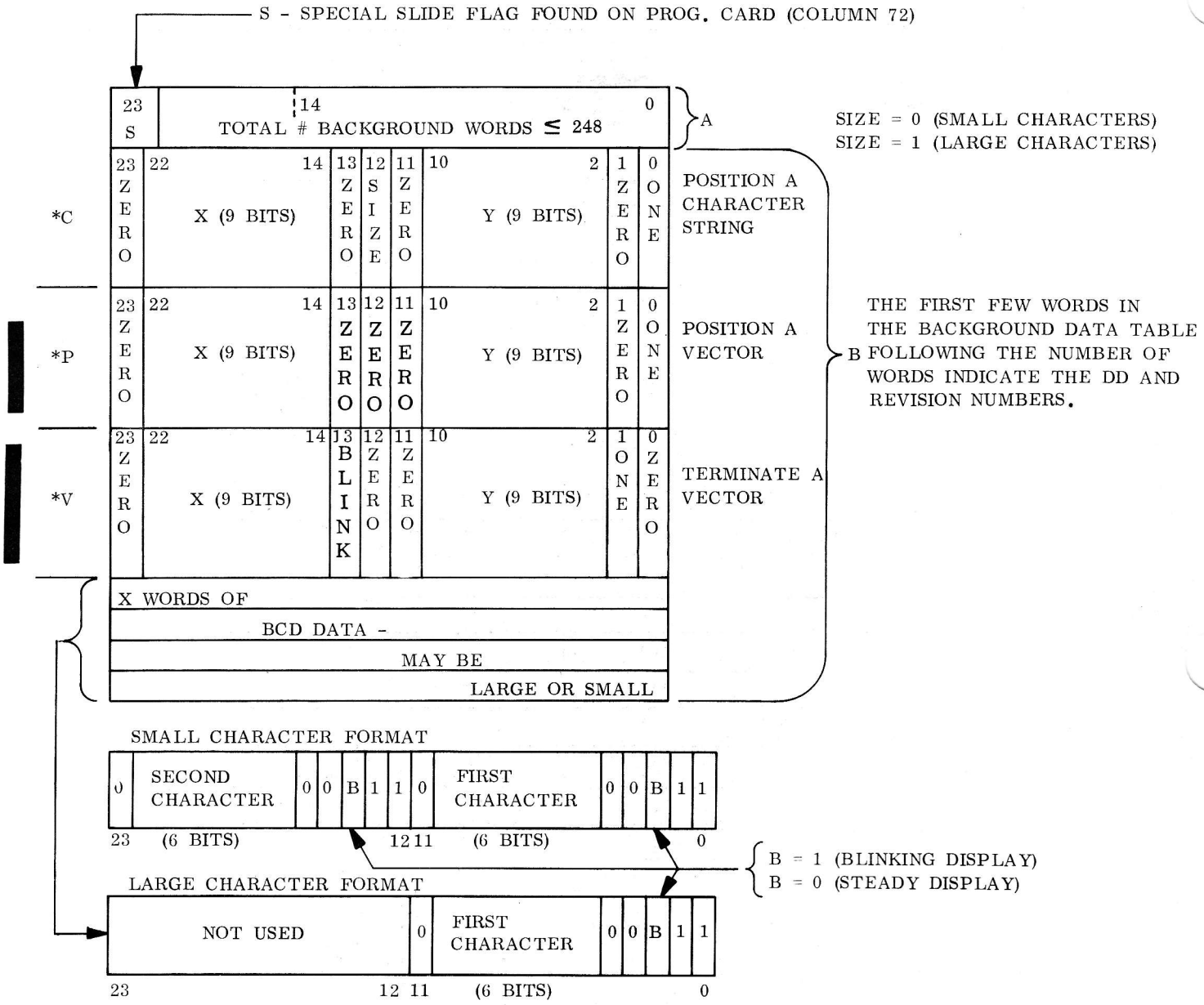


Figure 4.2.6 Display Description Binary Tape Format - Header and DD Data Record (Sheet AA)



(C, P, or V) *ALL 3 OF THESE MAY NOT BE IN SEQUENTIAL ORDER, BUT THEY MAY ALL APPEAR IN A SINGLE DD. C MUST BE FOLLOWED BY BCD CHARACTERS, NOT P AND/OR V. P AND/OR V DO NOT REQUIRE BCD CHARACTERS AND MAY BE ADJACENT IN A DD.

Figure 4.2.6 Display Description Binary Tape Format - Background Data Words (Sheet AB)

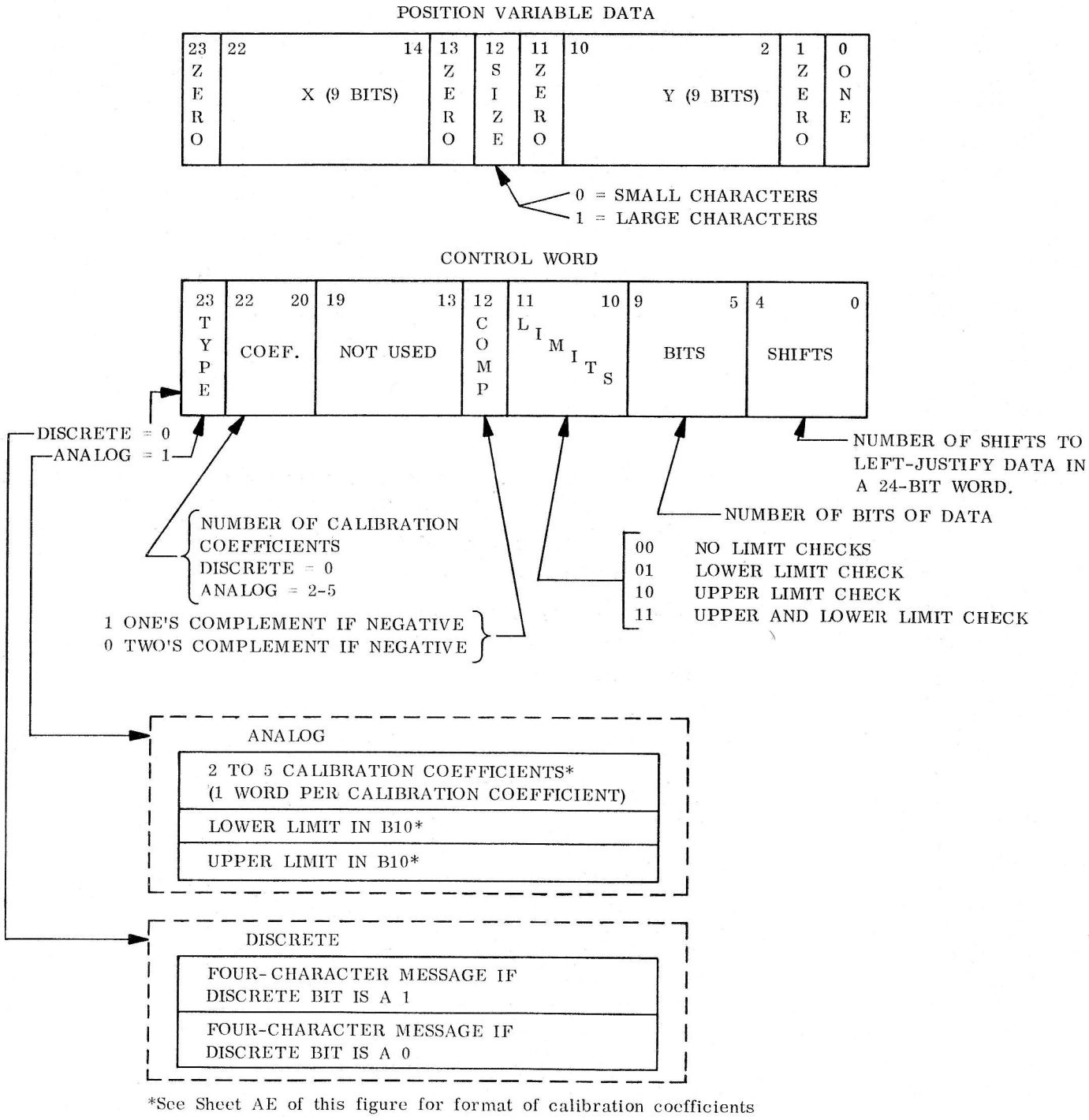


Figure 4.2.6 Display Description Binary Tape Format - Groups 0 through 6 Format Table Data Words (Sheet AC)

POSITION GRAPH ORIGIN

23 — 18	17 — 9	8 — 0
ZEROES	X (9 BITS)	Y (9 BITS)

CONTROL WORD

23	22 — 20	19 — 13	12	11 — 10	9 — 5	4 — 0
TYPE	COEF.	NOT USED	COMP	LIMITS	BITS	SHIFTS

{ NUMBER OF SHIFTS TO LEFT-JUSTIFY DATA IN A 24-BIT WORD

{ NUMBER OF BITS OF DATA

{ NO MEANING FOR GROUP 7

{ 1-ONE'S COMPLEMENT IF NEGATIVE
0-TWO'S COMPLEMENT IF NEGATIVE

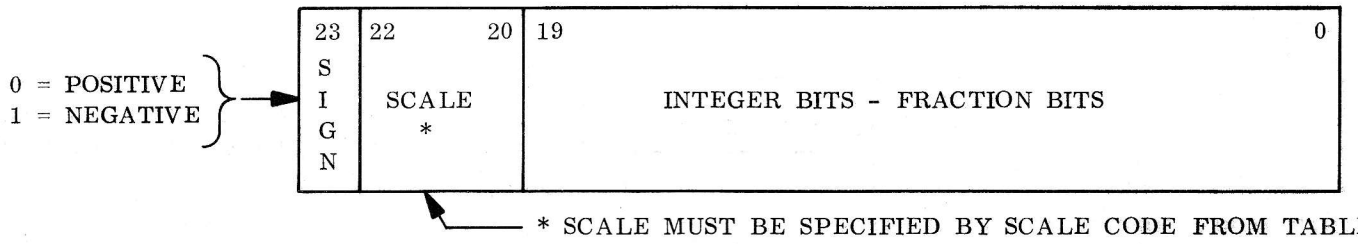
{ NUMBER OF CALIBRATION COEFFICIENTS (2-5)

{ 1 = ANALOG. DISCRETE HAS NO MEANING FOR GROUP 7

23 — 18	17 — 9	8 — 0
23 — 0: 2 TO 5 CALIBRATION COEFFICIENTS* (1 WORD PER CALIBRATION COEFFICIENT)		
23 — 0: GRAPH RANGE AS A FUNCTION OF Y IN B10*		
ZEROES	GRAPH SIZE X (9 BITS)	GRAPH SIZE Y (9 BITS)
23 — 0: GRAPH RANGE AS A FUNCTION OF TIME IN B0*		

*See Sheet AE of this figure for format of calibration coefficients.

Figure 4.2.6 Display Description Binary Tape Format - Group 7 Format Table Data Words (Sheet AD)



SCALE CODE TABLE

B-X BINARY PT. OF COEFF.	SCALE CODE	NO. INTEGRAL BITS	NO. FRACTION BITS
B 7	000	13	7
B 9	001	11	9
B 11	010	9	11
B 13	011	7	13
B 15	100	5	15
B 17	101	3	17
B 19	110	1	19
B 20**	111	0	20
B 10	N/A	13	10
B 0	N/A	23	0

** ZERO COEFFICIENTS MUST HAVE SCALE CODE OF B 20

Figure 4.2.6 Display Description Binary Tape Format - Calibration Coefficients for Format Table Data (Sheet AE)

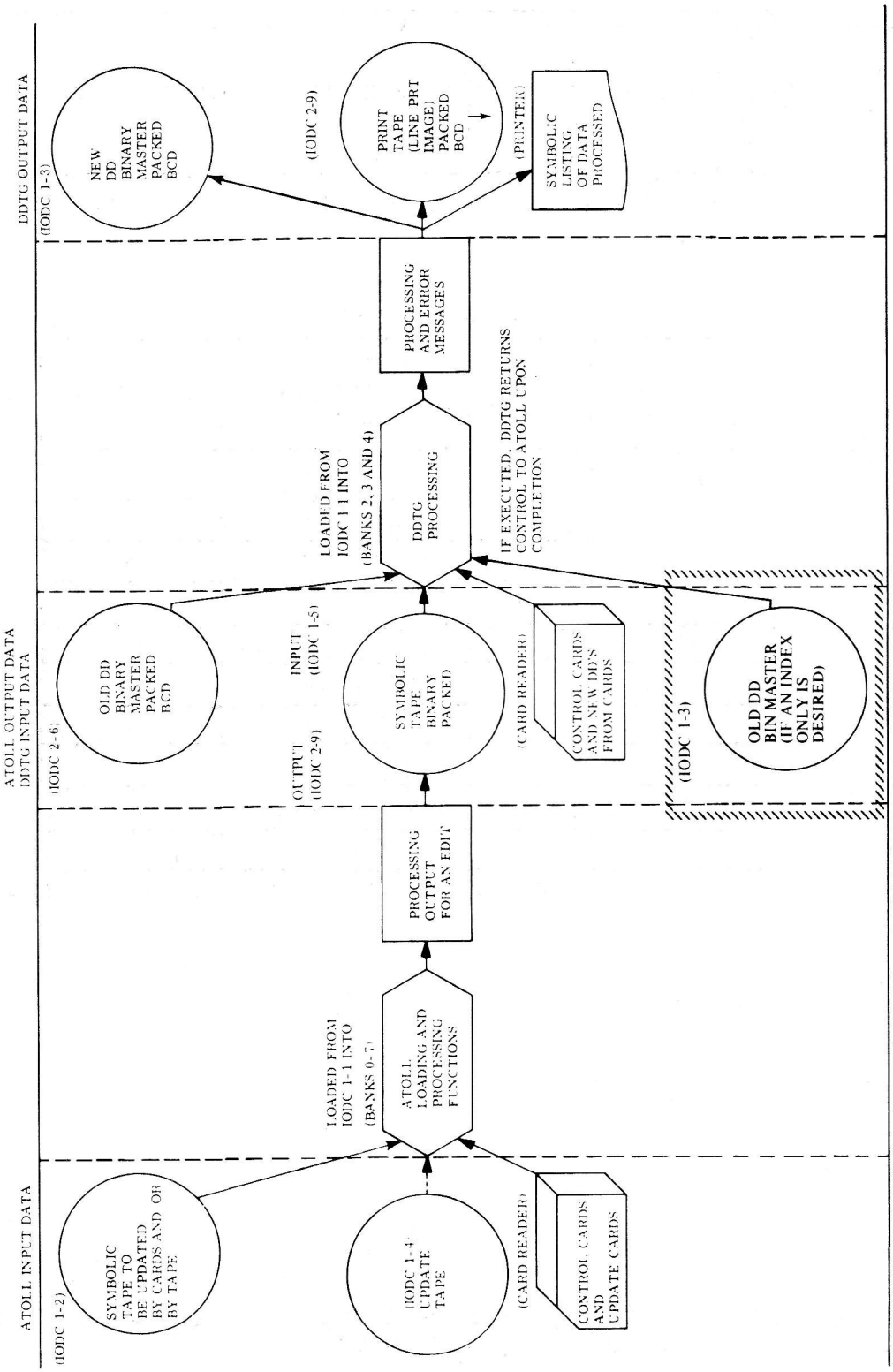


Figure 4.2.7 Display Description Tape Generation/Update System Flow

Refer to Appendix F of this volume for a description of equations and relationships to be used in calculating unknowns for analogs and discretes, and of legal values for DDAS analogs and discretes.

Background table data is found on a type B card only. Format table data may be found on type A, D, and 3 and 4 cards. There are eight groups of format table data with group 0 being used for DMON data, group 7 used for graphic analog data, and groups 1-6 used for tabular analog and tabular discrete data (test programs). DMON format table data may be found on type A, D, and 3 cards. If group 7 format table data is contained in a DD, no other group's format table data can be contained in that DD.

The following tables define field contents for each DD data card type.

Card Type A (Linear Tabular Analog Data)

Column(s)	Contents	Comments
1	A	Defines card type as A.
3-5	X Coordinate	Defines position of variable data to be displayed. May be any value within the range of 0-511 (X coordinate), and 0-438 (Y coordinate).
7-9	Y Coordinate	
11	Character Size	Defines character size, as follows. S = Small characters L = Large characters Blank = Small characters are assumed
13	Group Number	Must be groups 0-6 for tabular analog data. Group 0 for DMON, or from group 1 through group 6 for test programs. Group 0 is assumed if a group number is not specified. If group 0 is specified, columns 17-30 must contain an analog address.
15	One's or Two's Complement if Negative	1 or 2 for one's or two's complement, respectively. If blank, 2 is assumed.
17-30	DMON Address	Left-justified (specified only if group 0 was given in column 13).
33-40	Lower Signal Range	Right-justified. May be any value within the range of ± 8191.99.
41-48	Upper Signal Range	
49-56	Lower Limit	May be any value within the range of ±8191.99, right-justified; or either or both fields may be left blank. If displayed value exceeds either limit specified, the value blinks. A blank in either field is assumed to indicate the limit requires no limit-checking.
57-64	Upper Limit	

Card Type A (continued)

Column(s)	Contents	Comments
67-68	Number of Bits of Data	Must be 11. If blank, 11 is assumed.
70-71	Number of Shifts to Left-Justify Data in a 24-Bit Word	This value is 1 for a group B DDAS measurement, or 13 for a group A DDAS measurement. If field is left blank, DDTG can calculate the number of shifts from the analog address specified in columns 17-30. These columns and columns 17-30 must not both be blank.
73-80	Sequence Number	First two characters are alphabetic, the remaining 6 characters are numeric (beginning with 000000, 000100, 000200, etc).

Card Type B (Background Data)

Column(s)	Contents	Comments
1	B	Defines card type as B.
3-5	X Coordinate	Same as for a type A card.
7-9	Y Coordinate	
11	Character Size or Vector Coordinate	S designates small character for BCD background data. S is assumed if L, P, or V is not specified, or if column 11 is blank. L designates large characters for BCD background data. P designates the position point of a vector at X-Y coordinates. V designates the termination of a vector at X-Y coordinates. For vectors, a position point must be specified before a termination point. Several termination points may be specified in sequence; in this case, the position point is assumed to be the last termination point.
13-71	Background Data	Data to be displayed is left-justified. If column 11 is a V or P, columns 13-71 must be blank. This field is scanned from right to left to determine amount of background data present.
73-80	Sequence Number	Same as for a type A card.

Card Type D (Tabular Discrete Data)

Column(s)	Contents	Comments
1	D	Defines card type as D.
3-5	X Coordinate	Same as for type A card.
7-9	Y Coordinate	
11	Character Size	Same as for a type A card.
13	Group Number	From 0 to 7. Must be group 0 if a discrete address is present in columns 17-30.
17-30	DDAS Discrete or DI Address	Left-justified. If a DDAS address is present, must be group 0. If a DI is specified, the DI number must range from 0-3023.
33-36	Message Displayed if Discrete Bit is a 0.	If blank and columns 41-44 are blank, DDTG inserts the message OFF. If blank and columns 41-44 are not blank, DDTG leaves this field blank. If a message is specified, it must not be the same as a message specified in columns 41-44.
41-44	Message Displayed if Discrete Bit is a 1.	If blank and columns 33-36 are blank, DDTG inserts the message ON. If blank and columns 33-36 are not blank, DDTG leaves this field blank. If a message is specified, it must not be the same as a message specified in columns 33-36.
47-48	Number of Shifts to Left-Justify Data in a 24-Bit Word	If specified, must be right-justified. For DI, must be in range of 0-23. If blank and DDAS or DI address is present in columns 17-30, DDTG calculates the number of shifts. These columns and columns 17-30 must not both be blank.
73-80	Sequence Number	Same as for a type A card.

Card Type 3 (Non-Linear Tabular Analog Data)

Column(s)	Contents	Comments
1	3	Defines card type as 3.
3-5	X Coordinate	Right-justified. Defines beginning point of first character. May range from 0-511 (X coordinate), and from 0-438 (Y coordinate).
7-9	Y Coordinate	
11	Character Size	Same as for type A card.

Card Type 3 (continued)

Column(s)	Contents	Comments
13	Group Number	Must be group 0-6 for non-linear tabular analog data. If group 0, must have DMON address in columns 35-48.
17-18	Number of Bits of Data	Must be 11, right-justified.
20-21	Number of Shifts to Left-Justify Data in a 24-Bit Word	The value is 1 for a group B DDAS measurement, or 13 for a group A DDAS measurement, both right-justified.
23	One's or Two's Complement if Negative	1 or 2 for one's or two's complement, respectively. If blank, 2 is assumed.
35-48	DMON Address	Must contain a DMON address, left-justified, if group 0 is specified in column 13.
73-80	Sequence Number	Same as for a type A card.

Card Type 4

Column(s)	Contents	Comments
1	4	Defines type as 4.
3-10	Coefficient 0	May be any value within the range of ± 8191.99, right-justified. Must not be blank.
13-20	Coefficient 1	May be any value within the range of ± 8191.99, right-justified. Must not be blank.
23-30	Coefficient 2	May be any value within the range of ± 8191.99, right-justified. If blank, coefficients 3 and 4 are assumed blank also.
33-40	Coefficient 3	May be any value within the range of ± 8191.99, right-justified. If blank, coefficient 4 assumed blank also.
43-50	Coefficient 4	May be any value within the range of ± 8191.99, right-justified.
73-80	Sequence Number	Same as for a type A card.

Card Type 3 (Graphic Analog Data)

Column(s)	Contents	Comments
1	3	Defines card type as 3.
3-5	X Coordinate	Defines beginning point of vector. May range from 0-511 (X coordinate), and from 0-438 (Y coordinate), both right-justified.
7-9	Y Coordinate	
11	Blank	Must be blank.
13	Group Number	Must be group 7 for graphic analog data.
17-18	Number of Bits of Data	Must be 11, right-justified.
20-21	Number of Shifts to Left-Justify Data in a 24-Bit Word	Must be 1 for a group B DDAS measurement, or 13 for a group A DDAS measurement, both right-justified.
23	One or Two's Complement if Negative	1 or 2 for one's or two's complement, respectively. If blank, 2 is assumed.
28-30	Graph Range as a Function of Time	May be any value from 0-450 dits, right-justified.
35-48	DMON Address	Must be specified, left-justified.
73-80	Sequence Number	Same as for a type A card.

Card Type 4 (Graphic Analog Data - Format 1)

Column(s)	Contents	Comments
1	4	Defines card type as 4.
3-10	Coefficient 0	May be any value within the range of $\pm$ 8191.99, right-justified.
13-20	Coefficient 1	May be any value within the range of $\pm$ 8191.99, right-justified.
23-30	Coefficient 2	May be any value within the range of $\pm$ 8191.99, right-justified. If left blank, coefficients 3 and 4 are assumed blank also.
33-40	Coefficient 3	May be any value within the range of $\pm$ 8191.99, right-justified. If left blank, coefficient 4 is assumed blank also.

Card Type 4 (Format 1) continued

Column(s)	Contents	Comments
43-50	Coefficient 4	May be any value within the range of ± 8199.99, right-justified, or blank.
53-60	Graph Range Along Y Axis	May be any value within the range of ± 8199.99, right-justified.
63-65	Graph Size in X Direction	May be any value from 0-511 dits, right-justified.
67-69	Graph Size in Y Direction	May be any value from 0-438 dits, right-justified.
73-80	Sequence Number	Same as for a type A card.

or

Card Type 4 (Graphic Analog Data - Format 2)

Column(s)	Contents	Comments
1	4	Defines card type as 4.
3	R	Signals DDTG to calculate the two coefficients for a first-degree polynomial from the upper and lower signal ranges in the next two fields.
13-20	Lower Signal Range	Both signal ranges must be values within ± 8191.99, right-justified.
23-30	Upper Signal Range	
53-60	Graph Range Along Y Axis	May be any value within the range of ± 8199.99, right-justified.
63-65	Graph Size in X Direction	May be any value from 0-511 dits, right-justified.
67-69	Graph Size in Y Direction	May be any value from 0-438 dits, right-justified.
73-80	Sequence Number	Same as for a type A card.

4.2.8.2 ATOLL and DDTG Control Cards

Besides DD data cards, certain control cards must be supplied to the ATOLL Compiler to cause generation or update of a DD symbolic tape and/or processing of DD data to generate a binary DD tape (DDTG). Some of the information to be placed on control cards must be obtained from the most current DDTG assembly listing.

Six ATOLL control cards (JOB, OPTN, EDIT, PROG, PROC, and EOJ) are used in DD symbolic tape generation, and in calling up DDTG to process DD data for binary tape generation. Refer to Appendix A of this volume for control card formats; and to paragraph 4.2.8.3 for sample control card deck set-ups.

When called by ATOLL to process DD's, DDTG recognizes six control cards which specify the type of input and order of processing. The format for each of these cards is as follows.

a. OPT card

Column(s)	Contents	Comments
8-10	OPT	Identifies card.
16-20	BTAPE	Specifies use of an old binary master tape on IODC 2-6 as input.
21	Comma	
22-26	PTAPE	Specifies output of a print tape (packed BCD, LIEF format of one line per record) on IODC 1-5.

If no OPT card is present, input from an old binary master tape is not expected, and a print tape is not generated. In this case DDTG looks to the PROG card for an indication of whether the information for the DD to be processed is on cards or on symbolic tape.

b. Header card

Column(s)	Contents	Comments
1-2	ID	Identifies card.
*		
14-21	Tape Identification	User may place data of his choice in this field.

c. PROG card

Column(s)	Contents	Comments
1-6	/ϕ PROG	Identifies card.
8-11	AXXX	The letter A is replaced by the letters S or D which signify that the DD is a slide or a display. The letters XXX are replaced by the number of the DD, from 0 through 511 for S and 0 through 510 for D.

(Continued on next page)

* Date information on the ID card is no longer valid and will not be used, since the date is now taken from the ATOLL /ϕJOB control card.

c. PROG card (continued)

Column(s)	Contents	Comments
13-15	Series Number	Must be three characters (alphanumeric or numeric).
71	T	Indicates to DDTG that the DD number specified in columns 9-11 is on symbolic tape on IODC 2-9.
72	F	Indicates use of the special slide flag in the DD.

d. END card

Column(s)	Contents	Comments
8-10	END	Indicates to DDTG the end of information for a DD. DDTG then processes the DD and writes it to tape.

e. EOD card

Column(s)	Contents	Comments
8-10	EOD	Indicates to DDTG the end of information for all DD's. DDTG then completes all processing and returns control to ATOLL.

f. INDEX card

Column(s)	Contents	Comments
1-5	INDEX	Indicates to DDTG that indexing of an old binary master DD tape on IODC 1-3 is desired.

If an INDEX card is specified it must be the only DDTG control card in the deck; once it is read no other DDTG control will be processed. However, if it is found embedded in the control card deck it will cause DDTG to terminate.

4.2.8.3 Sample ATOLL and DDTG Control Card Decks for Generating Symbolic and Binary DD Tapes

With regard to display description data, ATOLL provides the capability to generate a DD symbolic tape by processing cards, or cards and symbolic tape. If requested (PROC card), ATOLL also calls DDTG into execution to create a binary DD tape using any combination of the three sources of DD data (cards, symbolic tape, binary master tape). The following are sample control card decks for typical DD tape generation tasks.

a. To edit symbolic cards on to a symbolic tape (ATOLL only), the following deck set-up is used.

NOTE

In the following card samples, for JOB x DATA the X=IB or V for the S-IB or S-V system. When the IB system is used, the card will be punched as follows: CC 6=ϕ, CC's 7 & 8=IB, CC 9=ϕ, CC's 10 thru 13=DATE. For the S-V system CC 7=V.

```
0         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890

/ JOB X DATE (See Note, Page 4-62)
/ OPTN  CARD
/ EDIT
/ PROG AXXX SER NEW PF                                                      TF

   SYMBOLIC DD DATA CARDS TO BE PLACED ON TAPE

/ EOJ
```

b. To edit symbolic cards on to a symbolic tape and then to generate a binary DD tape and an index listing of the binary DD tape (ATOLL and DDTG), the following deck set-up is used.

```
0         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890

/ JOB X DATE (See Note, Page 4-62)
/ OPTN  CARD
/ EDIT
/ PROG AXXX SER NEW PF                                                      TF

** SYMBOLIC DD DATA CARDS TO BE PLACED ON TAPE

/ PROC

** CARD DECK SET-UP AS ILLUSTRATED IN ITEM F OF THIS PARAGRAPH.

/ EOJ
```

c. To edit symbolic tape and cards on to a symbolic tape (ATOLL only), the following deck set-up is used.

```
0         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890

/ JOB X DATE (See Note, Page 4-62)
/ EDIT
                      ADD
/ PROG AXXX SER REVN DROP PF                                                TF
                      PF

** SYMBOLIC DD DATA UPDATE CARDS

/ EOJ
```

d. To edit symbolic tape and cards on to a symbolic tape, and then to generate a binary DD tape and an index listing of the binary DD tape (ATOLL and DDTG), and print revision history data, the following deck set-up is used.

```
0        1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890

/ JOB X DATE (See Note, Page 4-62)

/bOPTNbHIST

/ EDIT
                        ADD
/ PROG AXXX SER REVN DROP PF                                              TF
                        PF

** UPDATE CARDS

/ PROC

** CARD DECK SET-UP AS ILLUSTRATED IN ITEM F OF THIS PARAGRAPH

/ EOJ
```

e. To generate a binary tape (DDTG only) from the symbolic DD data input combination specified on the OPT and/or PROG cards and an index listing of the binary DD tape, the following card deck set-up is used.

```
0        1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890

/ JOB X DATE (See Note, Page 4-62)
/ OPTN CARD
/ EDIT
/ PROC

** CARD DECK SET-UP AS ILLUSTRATED IN ITEM F OF THIS PARAGRAPH

/ EOJ
```

f. For processing of DD input by DDTG to generate a binary tape, the following cards are used in conjunction with decks illustrated in items b, d, and e.

```
0        1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890

*       OPT     BTAPE,PTAPE
ID      (HEADER CARD - CONTAINS INFORMATION FOR VARIABLE HEADER RECORD)
/ PROG AXXX SER                                                           TF
        (DD DATA CARDS FOR EACH PROG CARD ON WHICH A "T" WAS NOT PLACED
        IN COLUMN 71 TO INDICATE THE DD IS ON SYMBOLIC TAPE)

*       (COMMENT CARDS - MAY APPEAR ANYWHERE IN A DECK)

        END
        EOD
```

g. The following cards are used to generate an index listing of a previously generated binary DD tape on IODC 1-3.

```
0         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890

/bJOBbDATE (See Note, Page 4-62)
/bOPTNbCARD
/bEDIT
/bPROC



INDEX
```

4.2.8.4 DDTG Program Output

In addition to generating a binary master tape, an index listing of the binary master, and a print tape (if requested), DDTG also prints out the following data on the line printer.

a. Initial option messages (followed by a program halt). One of the following initial messages will be printed depending on the use of the DDTG INDEX control card.

NO INDEX CONTROL CARD READ

Message	Comments	M-Register Event Code
$DDTG PROCESSING - TAPE ASSIGNMENTS SYMBOLIC INPUT ON 1-5 BINARY MASTER INPUT ON 2-6 BINARY MASTER OUTPUT ON 1-3 PRINT TAPE OUTPUT ON 2-9	This message describes the IODC tape station assignments required and the use of SSW-8 before the program can continue execution.	1111_8
SSW-8 ON WILL INHIBIT PRINTING OF AN INDEX LISTING OF NEW BINARY MASTER		
PRESS START TO CONTINUE		

INDEX CONTROL CARD READ

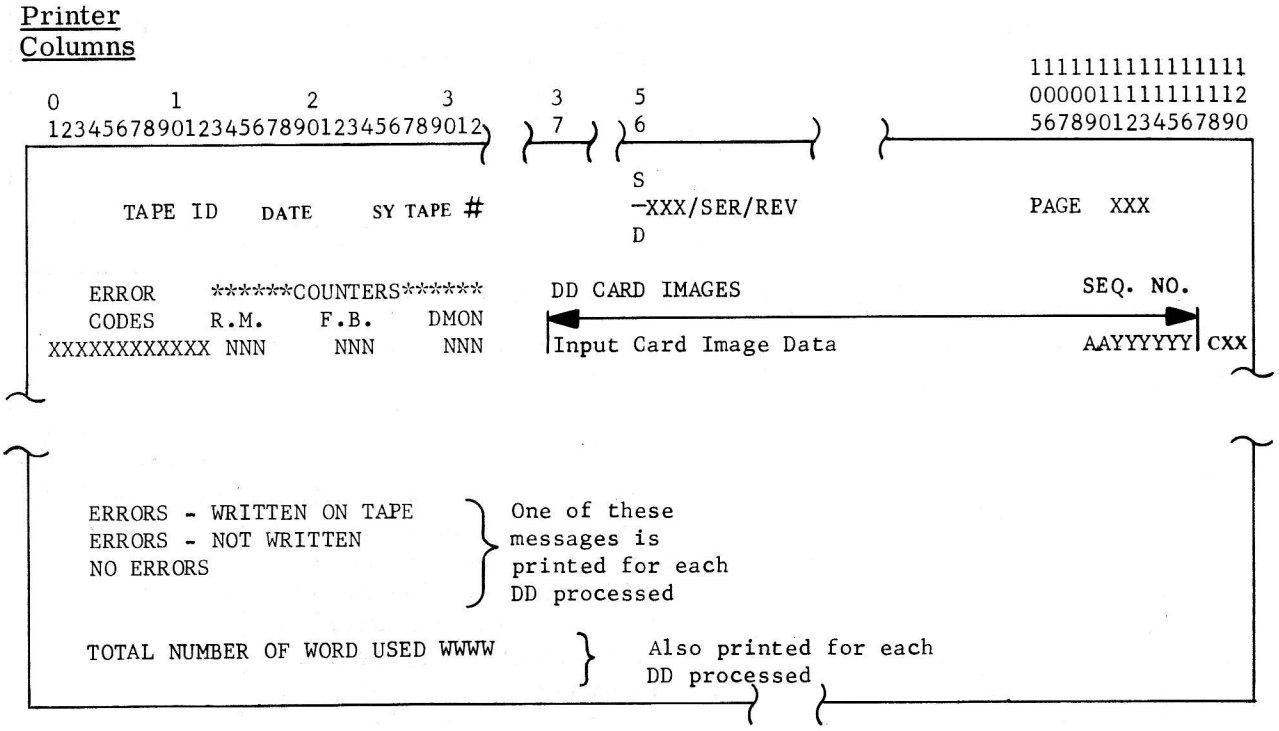
INDEX LISTING REQUESTED-BINARY MASTER TO INDEX ON 1-3 ** PRESS START	This message describes the IODC tape assignment before the program can continue execution.	11110000_8

b. Error messages (each followed by a program halt)

Message	Comments	M-Register Event Code
INVALID OPTION CARD	Options not punched properly. Insert new option card and press START.	11_8
ILLEGAL CARD. PROG CARD EXPECTED	Place correct card in reader and press START.	33_8
PROG CARD NOT IN SEQUENCE	DD number not greater than last DD number, or slide card encountered after displays are being processed. Replace with correct card and press START.	22_8
INVALID PROG CARD	The character T or a blank was not coded in column 71, or the character F or a blank was not coded in column 72. Replace with correct card and press START.	44_8
LOGICAL EOT ENCOUNTERED ON SYMBOLIC IN TAPE	Double EOF found while searching for PROG card. The DDTG being searched for and subsequent DD's are dummied, copied from the binary master tape, or processed from symbolic cards, whichever is applicable.	66_8
DESIRED PROG PRECEDES NEXT PROG ON SYMBOLIC IN TAPE	PROG card images on tape are out of order or missing. Press START to read next PROG card from reader.	55_8
UNABLE TO EXECUTE OPERATION FOR OCTAL ID XX	The letters XX are replaced by the octal code of the SIOC operation attempted. The following codes may appear in this message: 31, 32, 35, 36, 40, 41, 43, 42, 26, 27, and 10 (refer to Appendix A of Volume I, Table A-3 for a definition of these codes).	10101010_8

Message	Comments	M-Register Event Code
UNABLE TO USE MTS X-Y, DDTG (TERMINATED OR CONTINUING)	The letters X-Y are replaced by the number of the IODC and magnetic tape station which is inoperative.	70707070_8 (if the program is terminated)
PROCESSING COMPLETE DDTG TERMINATED		7777_8 (if the line printer is inoperative)
THE DATE IS NOW TAKEN FROM THE ATOLL /ϕJOB CONTROL CARD	This message is to inform the user that date information is no longer valid on an ID control card. (This will not prevent further processing.)	11111111_8
$DDTG TERMINATED - AN INDEX CARD WAS ILLEGALLY READ	An INDEX control card can not be embedded in a DDTG control card deck.	N/A

c. DD processing information - The following information is printed on each printer page for DD's processed by DDTG.



R.M. = Refresh Memory counter, which provides a running count of the number of words (NNN) used by refresh memory (background data).

F.B. = Format Buffer counter, which provides a running count of the number of words (NNN) used for format data.

DMON = Display Monitor counter, which is a running counter of the number of words (NNN) required for DMON addresses.

CXX = Update History

DD CARD IMAGES = Each DD card image is printed as it is punched, or as it appears on symbolic tape.

TAPE ID = Up to 8 characters of tape ID information (taken from ID header card, or from old binary tape).

DATE = 8 Character date taken from the ATOLL /ϕJOB control card. (Format is MM/DD/YY where MM = Month, DD = day, and YY = year)

SY TAPE # = 8 Character ATOLL compiler ID supplied via an ATOLL bank zero cell. (This data is built by STAG while generating an ATOLL compiler system tape.)

UPDATE HISTORY = The update history of a specific DD card image on symbolic tape if it has been changed and the revision number when it was changed.

$\frac{S}{D}$XXX/SER/REV = S or D indicates "slide" or "display", respectively; XXX indicates the DD number (0-511); SER indicates the series number of the DD; and REV indicates the revision number of the tape. The first three items are obtained from the PROG card; and the last from the symbolic input tape (or, if no symbolic input tape is used, also from the PROG card).

PAGE XXX = Page number, which is a running count of the number of printer pages used for each DD. This number is reset to 1 each time a new PROG card is encountered.

ERROR CODES XXXXXXXXXXXX = Up to 12 error codes, which denote DD data card errors found by DDTG during card validation. An explanation of each of these codes is as follows.

WWWW = Total number of words used in the DD.

Error Code	Cause of Error	Program Action
A	Illegal DMON address, S-IB stages used for S-V system, or S-V stages used for S-IB system.	DDTG inserts a standard discrete (AP1B0-19R03-01) or analog (AP1B0-19-03-00) address and continues processing.
B	Background data exceeds limit	DDTG continues to validate type B cards but not to process them. Other card types are not affected.
C	No coefficients present and no upper and lower ranges to calculate them from, or limits are invalid or are not right justified.	DDTG inserts a standard upper and lower range of 5-0, respectively, and continues processing.

Error Code	Cause of Error	Program Action
D	Illegal size indicator on card being processed, or two's complement field is in error.	A small character size is assumed, and a two's complement assumed.
E	Invalid card type (only A, B, D, 3, 4, and * are acceptable), or data is coded in the Background Data field on a type B card when a P or V has been specified in the Size field.	DDTG prints this card only, and continues processing with the next card.
F	Format data exceeds table size	DDTG validates the rest of this card and other cards with format data only, and processes all other cards.
I	Illegal card: no DMON address present and no group number present, or bad group number	DDTG inserts the standard DMON analog address (AP1B0-19-03-00) and a group number equal to zero and continues processing.
M	Identical entries on type D card in fields entitled If Discrete Bit Is a 1 and If Discrete Bit is a 0.	DDTG continues processing, and inserts the entry as it appears on the card.
O	Card out of place: type 4 card preceded type 3 card.	DDTG stops processing, validates the rest of the cards in the DD, and dummies the DD being processed.
P	Illegal card deck set-up: no type 4 card follows a type 3 card, or there is an incorrect format on a type 4 card which follows a type 3 card.	
R	Number of Bits of Data and Number of Shifts to Left-Justify fields on a type 3 or type A card do not agree, or an illegal number is specified in the Number of Shifts field on a type D card.	DDTG inserts a standard value and continues processing. For type 3 or A cards, the value is 11 bits, and shifts for group A or B DDAS measurements are made to equal 13 or 1, respectively. For type D cards, the standard value is 1 bit and 0 shifts for discretes.
T	DMON data exceeds table size	DDTG continues to process only type B cards and to validate all other cards in this DD.
X	Invalid X-Y coordinates	DDTG continues to process DD, inserting a standard of X=010 Y=010 for each error of this type found.

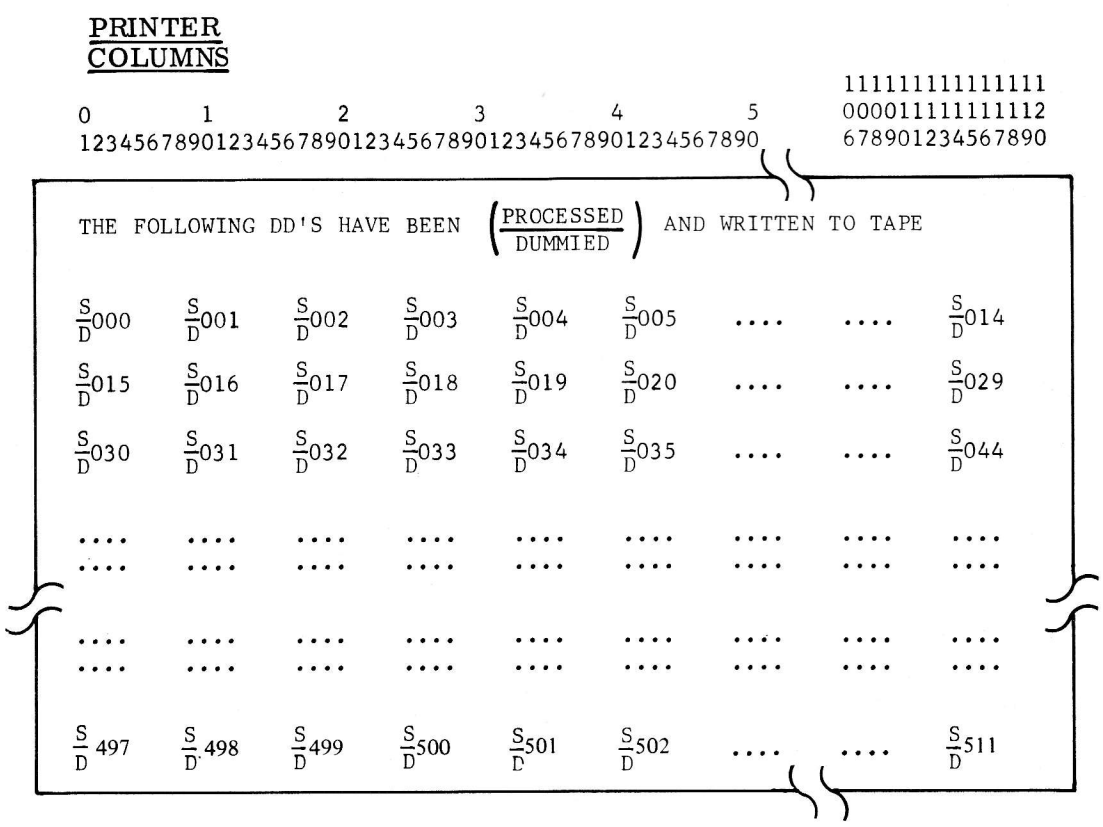
Error Code	Cause of Error	Program Action
G	One of the two required coefficients on the type 4 card is not present or is invalid, or one of the non-required coefficients is invalid	DDTG continues processing. A standard value of 5 is inserted for required coefficients, while non-required coefficients are ignored.
N	Graph range as a function of time not present or invalid	DDTG continues processing and inserts a standard value of 6.
J	Graph range as a function of Y not present or invalid	DDTG continues processing and inserts a standard value of 5.
K	Graph size in X direction not present or invalid	DDTG continues processing and inserts a standard value of 250.
L	A non-required coefficient is missing after another has been found	DDTG uses only the two required coefficients, if available, and validates the other cards for the DD.
H	Graph size in Y direction not present or invalid	DDTG continues processing and inserts a standard value of 320.
*	More than 12 errors occurred on one card (only room for 12 error codes)	DDTG continues processing, making any changes necessary to continue processing this DD.
W	A card containing graphic analog format table data for DMON has been read, and other cards read for that DD did not all contain graphic data (group 7).	DDTG continues processing, inserting a standard address (as shown for A error code), and using a standard group number of 0.
Z	The DD exceeds the maximum allowed word size of 550 words.	DDTG stops processing, validates the cards in the DD, and then dummies the DD being processed.
1-9 or +	DDTG has detected a mispunched card column. The number in the error code column denotes the number of mispunched columns. A number of punch errors greater than nine (9) is indicated by the + symbol.	DDTG continues processing.

At the end of each job, DDTG prints the following diagnostic and post-processing information.

THERE ARE $\left(\frac{\text{NO}}{\text{NNN}}\right)$ DD's IN EXPLICIT ERROR

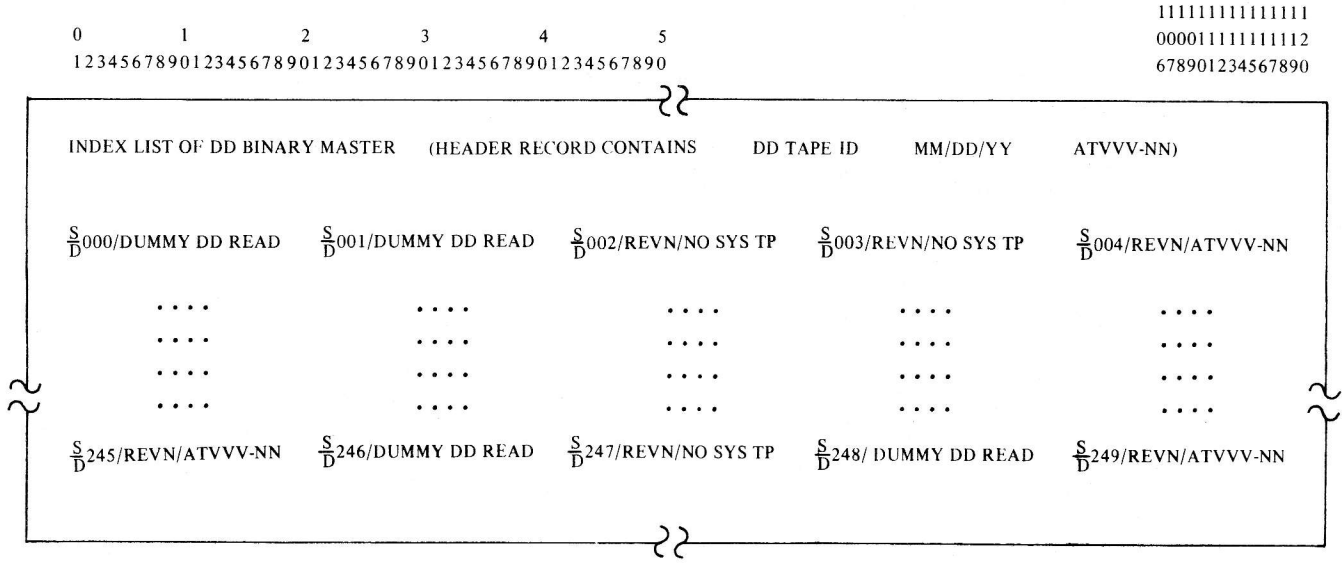
In this diagnostic message, NNN is replaced by the number of DD's in explicit error.

Post-processing data for DD's processed is printed by DD number, with a maximum of 555 DD's per page (all post-processing data may be contained on two pages), as follows.



The following is an example of the index data printed by DDTG. A maximum of 250 indexed DD's can be printed per page. Each contains data to inform the user that the DD is actual or a dummy, provide the DD revision number, and provide the ATOLL Compiler ID used to generate each specific DD. Each page contains a header message to inform the user of the type of data being printed.

PRINTER
COLUMNS



HEADER MESSAGE LEGEND

- DD TAPE ID = The tape ID data taken from the ID Header Control Card.
- MM/DD/YY = The date taken from the ATOLL /ϕJOB Control Card.
- ATVVV-NN = The ATOLL compiler ID information.

MESSAGE TEXT LEGEND

- S/DXXX/REVN/ATVVV-NN = DD generated using an ATOLL tape containing the system tape number.
- S/DXXX/N-Aϕ/NO SYS TP = DD generated or copied which contained no ATOLL system tape number.
- S/DXXX/DUMMY DD READ = Dummy DD generated.

4.2.9 Display Requests

There are two calling sequences used by test programs and function executor routines for requesting a display of variable data. The test programs resident in the LCCC use the macro instruction DISP and link to the Display Service routine with the instruction TSPO SDSR, 7. The function executor routines resident in the MLC link to the Remote Display Service routine with the instruction TSPO SRDY, 7. The calling sequence for the macro instruction DISP (LCCC) contains a control word which follows the transfer instruction. This calling sequence control word contains the address of the control words for the data type and the data to be displayed. The address is specified in bits 14 through 0. Bits 23 and 22 of the first control word contain the codes for the status of the display request operation. Figure 4.2.8 shows the format of this control word. The MLC calling sequence control word contains the codes for the status of the request. The status code is determined after the attempt to transmit the request to the LCCC. If the display fails in the LCCC, the message is automatically printed.

When the test program gives control to the Operating System, the request is stacked at priority level zero. The Operating System then requests level 1/2 processing and returns control to one of two core locations. These core locations are busy and normal return, and follow, respectively, the calling sequence control word. The Operating System returns control to the busy return if the request has been stacked. For the normal return, bits 23 and 22 in the first control word are set to a status code indicating whether the data was successfully or unsuccessfully displayed.

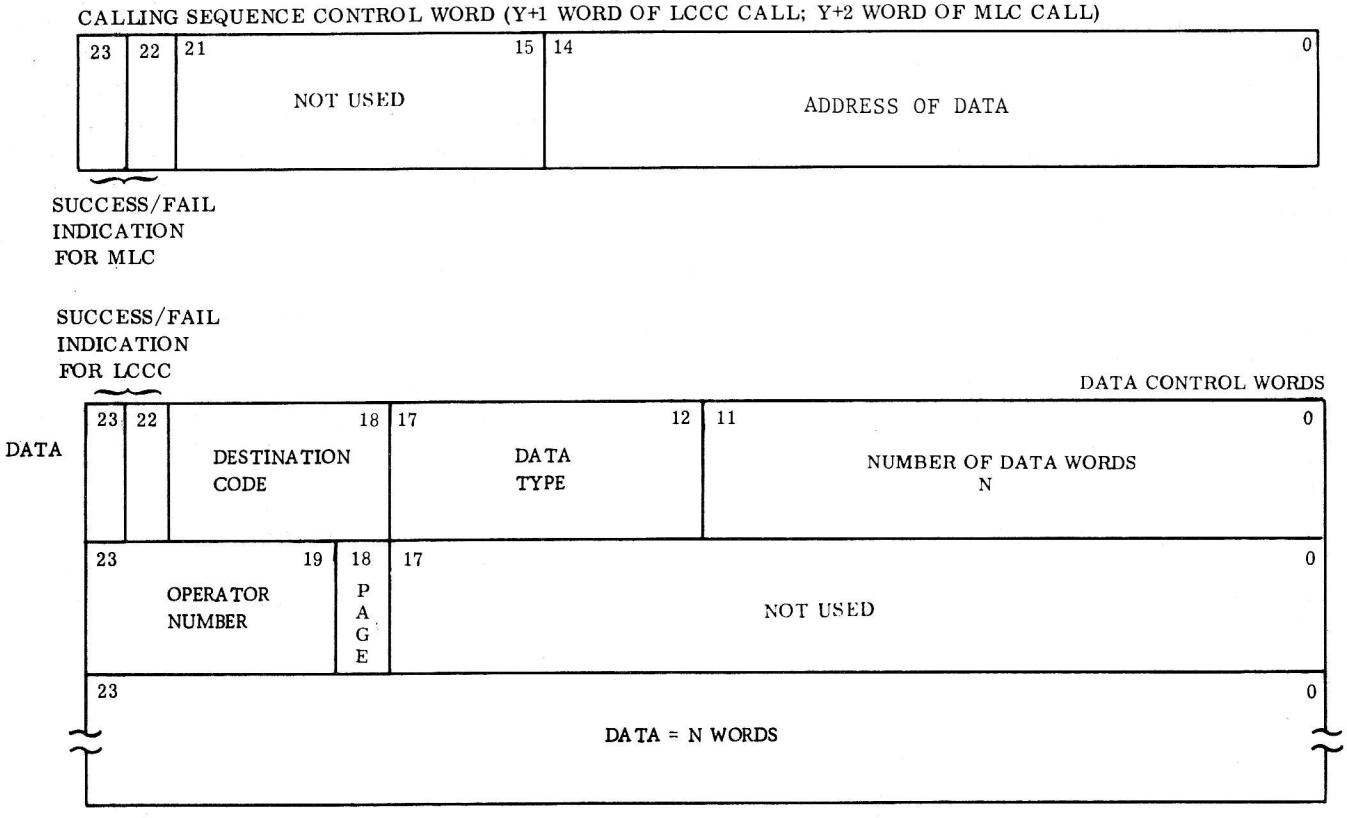


Figure 4.2.8 Display Request Control Word Format

The codes and status in bits 23 and 22 of the data control word sent to the LCCC are as follows:

Bits 23 and 22 = 01 - Data not displayed
= 10 - Data displayed

The codes and status in bits 23 and 22 of the calling sequence control word from the MLC are as follows:

Bits 23 and 22 = 01 - Data Link failure prevented transmission to LCCC
= 10 - Request was transmitted to LCCC

The test programmer may code this return location as required.

Refer to Table A-4 of Appendix A for a description of LCCC and MLC display request calling sequences.

4.2.9.1 Message Types

Six types of data have been classified:

- Binary
- Packed BCD
- Preformatted
- Memory
- Display Description
- TV Transmit, Operator Select, and Command Function Clear Select

Each type is transferred to the Operational Display System and identified by a certain message type code. Each message contains control and data words. Binary messages may contain either analog or discrete data, and are formatted for display with the format tables of a display description. Packed BCD messages are used to send BCD variable data to the Display System. Preformatted messages (i.e., in refresh memory format) contain data to display either BCD or vectors. Memory messages contain control information for the Operating System to access 16 words of data from either core or drum and to display them. A display description number is used by the Operating System to access the display description from the tape.

4.2.9.2 Data Control Word Format

When a test program or function executor routine requests a display of any of the six types of data, it must indicate to the Operating System which data type is to be displayed. The code for the data type, and other control information is specified in control words. These control words precede the data words, and are transferred with the data to the Operating System. The address of the first control word is given to the Operating System in the calling parameters of the display macro instructions.

The general format of these control words is given in Figure 4.2.8 and remains the same for the six data types. The contents of the words differ for each type of data and are coded by the test programmer as specified in subsequent paragraphs. The first control word contains three fields. They are:

- Destination
- Data Type
- Number of Data Words

The Destination field is contained in bits 23 through 18; this field contains either a 01 or 02 octal code. These codes are used by the DCC as control information. The Data Type field is contained in bits 17 through 12; this field contains different octal codes for the different types of data and indicates the type of data to be sent.

The Number of Data Words field is contained in bits 11 through 0. It indicates to the Operating System the number of data words following the two control words. This field is specified by the test programmer after calculating the total number of data words needed to contain all the information to be displayed.

The second control word contains two control fields. They are:

- Operator Number
- Page

The Operator Number field is contained in bits 23 through 19, and identifies the display console operator to which the data is to be sent. Since this field contains five bits, it can contain numbers from 0 through 31 (the Saturn Launch Computer Complex has at present a configuration of 15 consoles).

The Page field is contained in bit 18. There are two possible storage areas of refresh memory which can be displayed upon the console; these areas are referred to as page A and page B. This field indicates which of the two storage areas is to contain the data. If bit 18 is a 0, the data is stored in the area allocated for page A. If bit 18 is set to 1, the data is stored in the area allocated for page B.

If a test program is operating and the two fields of the second control word are empty, the Operating System assigns to them the operator number and page that requested the test program. If a test program is not operating, the Operating System sends the data to the master console.

As previously mentioned, the contents of some of these control fields differ for each type of data. Table 4.2-3 shows the contents of each field for each data type.

4.2.9.3 Binary Control and Data Word Format

The complete binary control and data word format is shown in Figure 4.2.9. Two additional fields are added to the second control word. These are:

- Display Description Number
- Group Number

Table 4.2-3 Data Control Word Codes

Data Type	Field				
	Destination Code	Data Type Code	Number of Data Words	Operator No.	Page
Binary	$(01)_8$	$(03)_8$	N	0-15	A = 0 B = 1
Packed BCD	$(01)_8$	$(07)_8$	N	0-15	A = 0 B = 1
Pre-formatted	$(01)_8$	$(10)_8$	N	0-15	A = 0 B = 1
Memory	$(01)_8$	$(11)_8$	$N = (20)_8 = 16_{10}$	0-15	A = 0 B = 1
Display Description	$(02)_8$	$(01)_8$ - Keybd Req. $(02)_8$ - Test Prog. Req.	0	0-15	A = 0 B = 1
TV Transmit & Operator Select	$(01)_8$	$(16)_8$	$(10)_8$	0-15	1

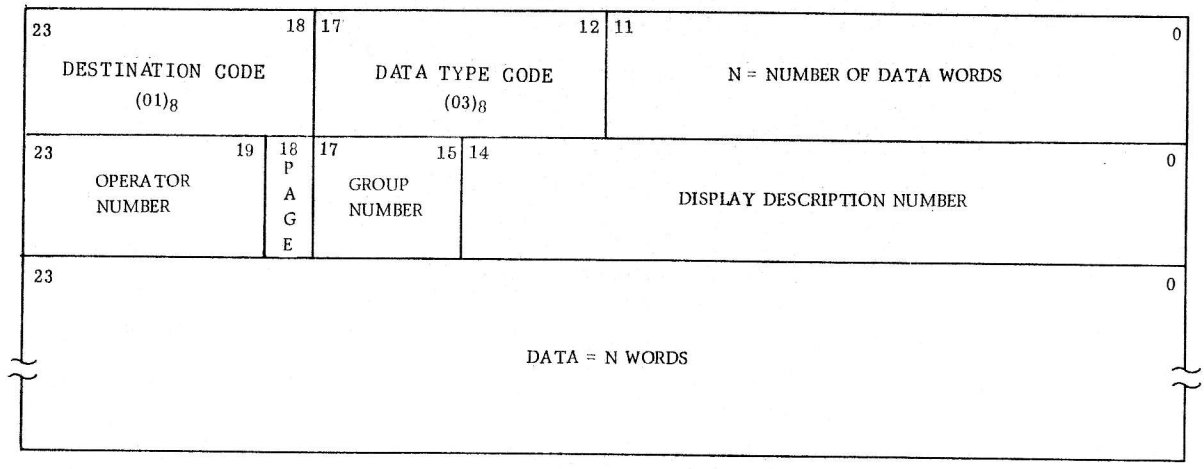


Figure 4.2.9 Binary Control and Data Word Format

The Display Description Number field is contained in bits 14 through 0. The test programmer must ensure that the format tables in a display description have been transmitted to DCC memory prior to sending the variable binary data. This field is used to specify the number of the desired display description. The Operating System extracts this description from tape and sends it to the DCC prior to sending the binary data.

The Group Number field is contained in bits 17 through 15. It is also required for display of binary data. Since format tables are needed to convert the binary data to the appropriate format, the group number specifying the desired format table is also specified by the test programmer.

4.2.9.4 Packed Binary Coded Decimal (BCD) Control and Data Word Format

The control and data word formats for packed BCD data are defined in the following figure.

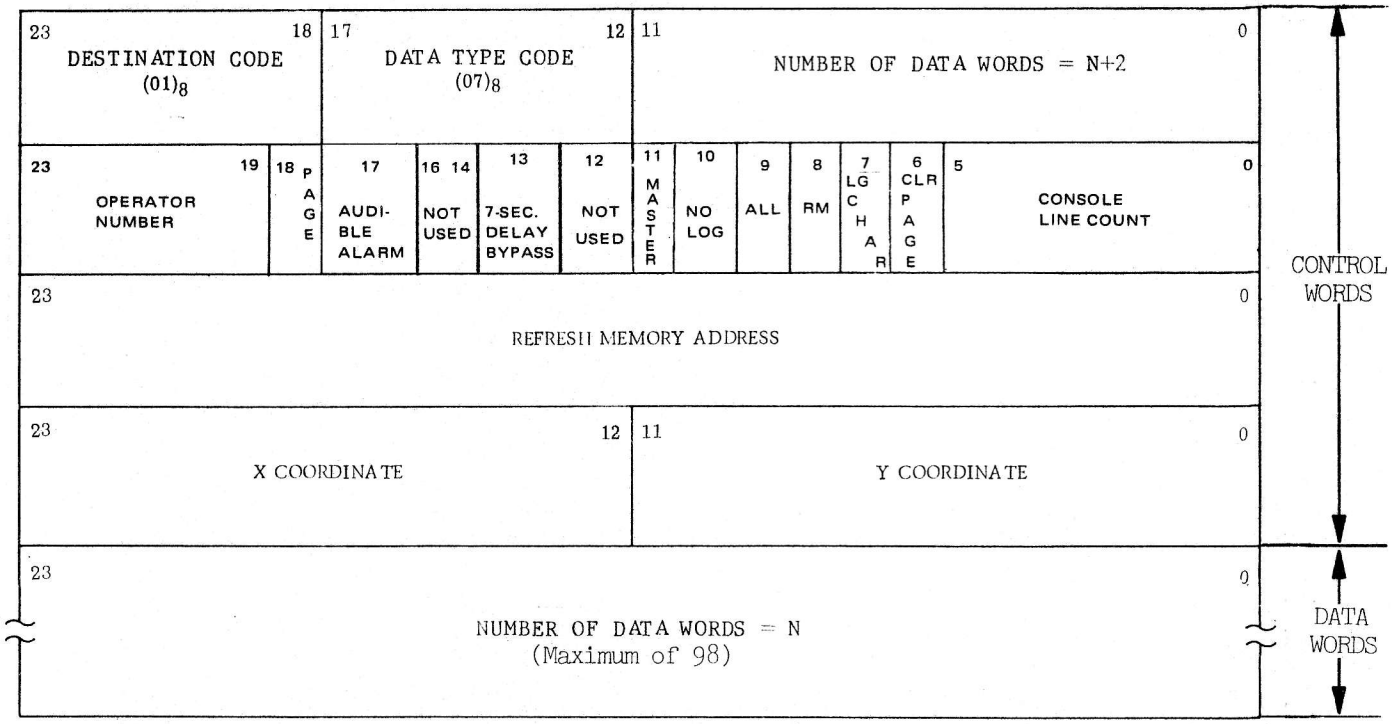


Figure 4.2.10 Packed BCD Control and Data Word Format

The special bits of the second control word are defined in this and the following paragraphs. If the message is to be displayed on page A of all three master consoles, the test programmer sets all bits of the Operator Number and Page fields to ones (the Operating System sets the MASTER bit to one). If the message is to be displayed on the maintenance console (1A) only, the test programmer sets the bits of the Operator Number field to all ones and the PAGE bit to zero. The MASTER control bit (bit 11) should always be coded zero by the test programmer. The 7-SEC DELAY BYPASS bit (bit 13) can be coded as one to allow bypass of the 7-second minimum display time for the most recently displayed message before a console page clear can occur. If bit 13 is coded as zero, the 7-second minimum display time remains in effect.

NOTE

Bit 13 is checked only if the console page is less than full, and the CLR PAGE bit (bit 6 is set)

The NO LOG control bit (bit 10) should always be coded zero by the test programmer, i.e., all test program message requests are logged on log tape. This bit is set to one by the Operating System to bypass normal logging of certain message requests. If the ALL control bit (bit 9) is set to one and the operator number is zero, the message is displayed to the operator whose number is specified in the Operator Number field.

If the refresh memory (RM) control bit (bit 8) is set to one, it indicates to the Operating System that the test programmer has specified his own refresh memory address and X/Y console coordinates. If this bit is zero, the Operating System supplies this data.

NOTE

If the test programmer uses this feature, he must keep track of RPM storage locations for any existing data and is responsible for proper housekeeping.

The LG CHAR bit (bit 7) can be set to one by the test programmer to cause the message to be displayed in large characters, or to zero to cause the message to be displayed in small characters. The CLR PAGE control bit is bit 6. If this bit is set to one, the entire console page is to be cleared prior to display of the current data, unless a display description is on the page. The CLR PAGE bit being set will not clear a display description. A display description can be cleared under program control with a request for DD1023 (1777_8). This DD contains no data; and when displayed, causes a blank display screen.

NOTE

If the console page is less than full when a message is requested and the CLR PAGE bit for the message is set to one, the most recently displayed message may be held for display a minimum of 7 seconds before the page is cleared, depending on the condition of the 7-SEC. DELAY BYPASS bit (bit 13). If this delay is requested, the message request is stacked for display. If the delay is bypassed, the page is immediately cleared, and the requested message is displayed at the top of the console page.

The LINE COUNT control bits are bits 5 through 0. These bits are coded to contain the number of lines in which the message is to be displayed. A maximum of 63 lines can be specified, although the display console can only display a maximum of 39 lines of data. Six lines of the display area are reserved for display of special information, which leaves a maximum of 33 lines in which a message can be displayed.

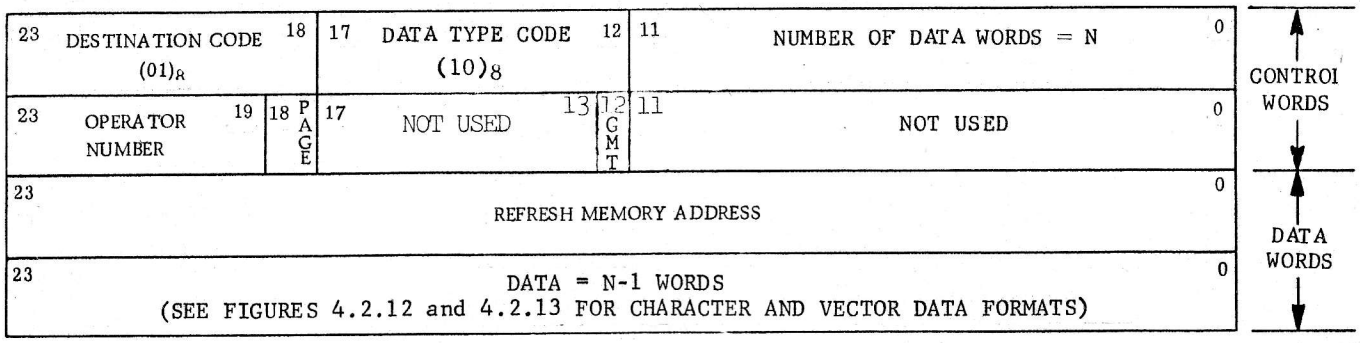
NOTE

An incorrect line count may result in an overwrite on the display console.

The two control words for the refresh memory address and the X/Y console coordinates are used in conjunction with the RM control bit. The refresh memory address specifies the starting location in which the data is to be stored. The X/Y coordinates specify the starting location on the display console for the message to be displayed.

4.2.9.5 Preformatted Control and Data Word Format

The format of the message control and data words to transmit preformatted data (characters or vectors) to the Display System is shown in the following figure.



NOTE

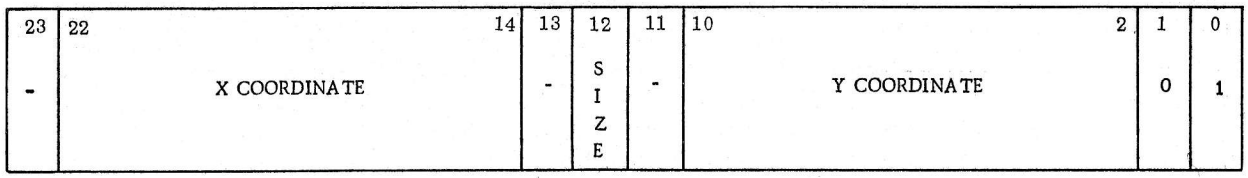
GMT BIT = 1 - GMT VALUE CONTAINED IN
LAST DATA WORD OF DISPLAY REQUEST

Figure 4.2.11 Preformatted Control and Data Word Format

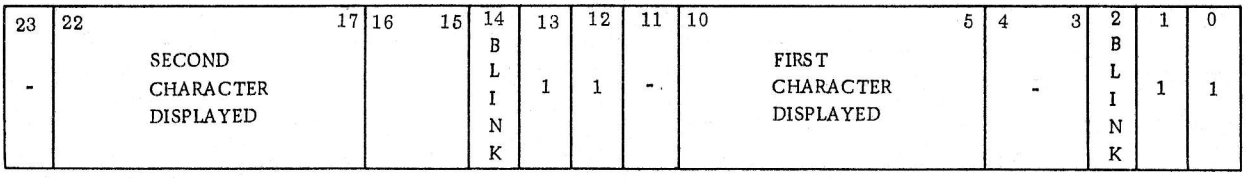
The first two control words contain the standard control information, including the GMT bit (bit 12) set 1 by DMON if GMT is to be displayed in the message. The refresh memory address is the location in refresh memory in which the preformatted data is to be stored, and must be specified by the test programmer. The data words which follow this address word contain the preformatted data, which consists of characters and/or vectors.

4.2.9.5.1 Preformatted Character Display

The required refresh memory format to request display of two characters is shown in the following figure.



CHARACTER POSITION WORD (SEE NOTE 1)



CHARACTER DATA WORD (SEE NOTE 2)

Figure 4.2.12 Refresh Memory Format for Character Display

Figure 4.2.12 Refresh Memory Format for Character Display (continued)

NOTE 1

MAY OR MAY NOT BE SPECIFIED IN CONJUNCTION WITH CHARACTER DATA WORDS. IF NOT SPECIFIED, CHARACTER DATA POSITIONING IS DETERMINED BY REFRESH MEMORY ADDRESS IN THIRD CONTROL WORD OF MESSAGE REQUEST.

NOTE 2

SIZE BIT (First Word)	= 1 - LARGE CHARACTERS = 0 - SMALL CHARACTERS
BITS 0 & 1 (Both Words)	= 01 - CHARACTER POSITION WORD = 11 - CHARACTER DATA WORD
BLINK BIT (Second Word)	= 1 - DISPLAYED DATA IS BLINKING = 0 - DISPLAYED DATA IS NOT BLINKING

Two data words are used: one contains X/Y coordinates for the displayed information, and the other contains the characters to be displayed.

The first data word contains the X/Y coordinates for the first character to be displayed on the display screen (these coordinates specify the lower left-hand corner console position for the character). This word also contains a Size field and Data Word Code field. The Size field is specified in bit 12, and indicates whether the two characters are to be displayed in large or small configurations. If the characters are to be large, bit 12 is set to 1; if the characters are to be small, bit 12 is set to 0. The Code field is specified in bits 1 and 0 in both words. The code 01 in these bit positions indicates to the Operating System that this word is a character position word; the code 11 indicates this word is a character data word.

The test programmer specifies the two characters to be displayed in bit positions 22 through 17 and 10 through 5 of the second data word. The character specified in bit positions 10 through 5 is the first character to be displayed. Associated with each character is the Blink field and the Data Word Code field. The Blink field is set to one if the character is to blink when displayed; the bit is set to zero if the displayed character is not to blink. The Data Word Code field is specified in bits 1 and 0; the code 11 in these bit positions to indicate that this data word contains the characters to be displayed.

If a character or a group of characters is to be displayed in a random fashion, a position data word is required with each character or group of characters. If a string of sequential characters is to be displayed, the test programmer need only specify one position word indicating the location of the first character.

4.2.9.5.2 Preformatted Vector Display

The required refresh memory format to display a vector segment is shown in Figure 4.2.13. Display of a vector segment requires two words: one word contains the X/Y coordinates for the origin point; the other contains the X/Y coordinates for the termination point. The X and Y coordinates for each set of origin and termination points are given in bits 22 through 14 and bits 10 through 2, respectively.

23	22 — 14	13	12	11	10 — 2	1	0
–	X COORDINATE FOR VECTOR ORIGIN POINT	–	–	–	Y COORDINATE FOR VECTOR ORIGIN POINT	0	1

VECTOR ORIGIN COORDINATE (POSITION WORD)

23	22 — 14	13	12 — 11	10 — 2	1	0
–	X COORDINATE FOR VECTOR TERMINATION POINT	BLINK	–	Y COORDINATE FOR VECTOR TERMINATION POINT	1	0

VECTOR TERMINATION COORDINATE (VECTOR WORD)

NOTES

BLINK BIT (Second Word) = 1 - DISPLAYED DATA IS BLINKING
= 0 - DISPLAYED DATA IS STEADY

BITS 1 & 0 (Both Words) = 01 - VECTOR ORIGIN X/Y COORDINATE
= 10 - VECTOR TERMINATION X/Y COORDINATE

Figure 4.2.13 Refresh Memory Format for Vector Display

The first data word contains the origin coordinates and is considered the position word. Bit positions 1 and 0 contain the code 01 to indicate to the Operating System that this data word contains the vector origin point.

The second data word contains the X/Y coordinates of the termination point, the Blink field, and the Data Word Code field. The Data Word Code field, in bit positions 1 and 0, contains the code 10 to indicate to the Operating System that this data word contains the vector termination point.

If connected vectors are to be displayed, only one position word is necessary; the remaining data words specify the termination point of succeeding connected vectors. If unconnected vectors are displayed, a position word and words specifying a termination point are required for each unconnected vector.

4.2.9.6 Memory Control and Data Word Format

The control word format for displaying memory data is shown in **Figure 4.2.14.** An additional field, shown in control word 2, contains the address of the core or drum storage area in which the data is located (the address specified can be the IVAR of the first word of the data). This control word format is used to display only 16 data words; therfore, the Number of Data Words field is coded with the constant 16.

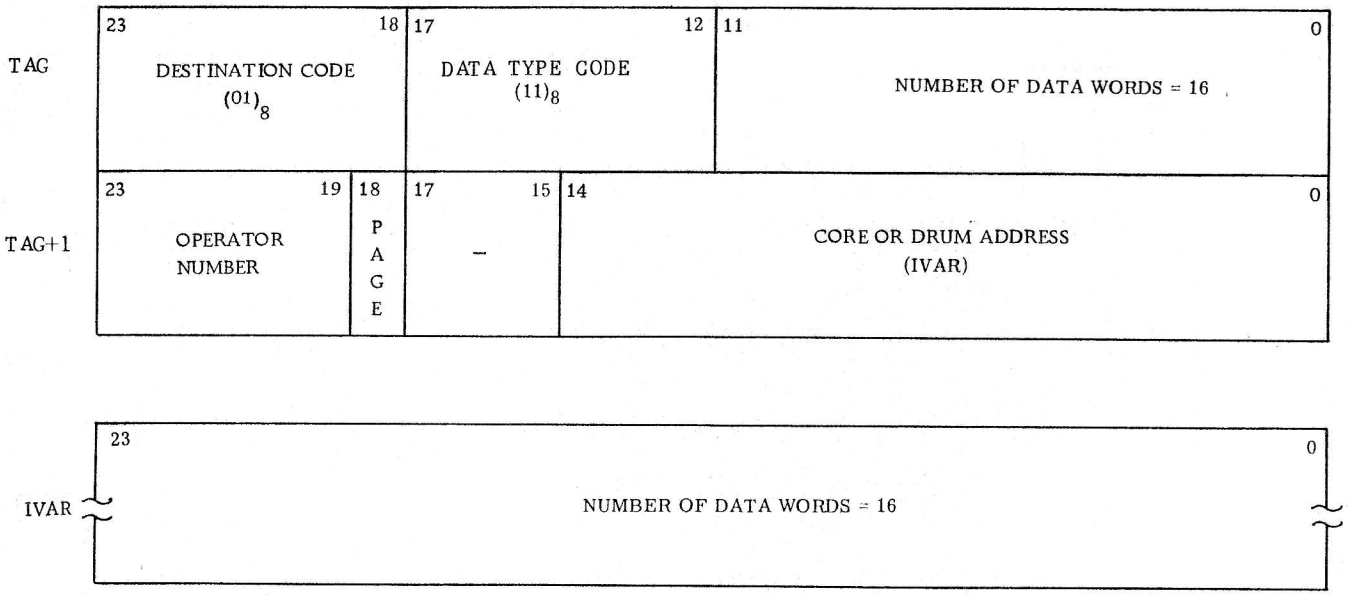


Figure 4.2.14 Display Memory (Drum or Core) Control and Data Word Format

4.2.9.7 Format for Display Description Control Words

The control word format for a display description is shown in Figure 4.2.15. The additional control field is the display description number. To utilize the contents of a display description, the test programmer specifies the display description number in bits 14 through 0 of the second control word. It should be noted that this format does not indicate any data. This data exists on tape and is accessed by the Operating System. Since no data is required, there are no data word fields provided.

	23 — 18	17 — 12	11 — 0
DATA	DESTINATION CODE $(02)_8$	DATA TYPE CODE $(01)_8$ - Keyboard Request $(02)_8$ - Test Program Request	—

	23 — 19	18	17 — 15	14 — 0
+1	OPERATOR NUMBER	PAGE	—	DISPLAY DESCRIPTION NUMBER

Figure 4.2.15 Display Description Control Word Format

4.2.9.8 Format for TV Transmit, Operator Select, and Command Function Clear Select Control and Data Words

The control words for an Operator Select, Command Function Clear Select, and TV Transmit are shown in Figure 4.2.16. The format for the TV Transmit and Command Function Clear Select are similar to the Operator Select, with the exception being the second control word.

Operator Select

23 — 19	18	17 — 12	11 — 0
0	1	16	0 0 1 0
Op. No.	1	14	0 0 0 X
0	0	1st. Char.	0 0 0 0
0	0	2nd. Char.	0 0 0 0

Command Function Clear Select

23 — 19	18	17 — 12	11 — 0
Op. No.	0	0	0

TV Transmit

23 — 19	18	17 — 12	11 — 0
Op. No.	1	42	0 0 0 X

Where X = 0 for Command Function Clear Select Off
1 for Command Function Clear Select On

Figure 4.2.16 Operator Select; TV Transmit Control Word Format

The Command Function Clear Select capability allows ATOLL programs, executing test programs and function executor routines to clear a selected command function. Command functions which may be cleared from a selected console and the affect of the clear select are described in the following table:

Command Function	Affect of Clear Function
Type	Present Mode Terminated, Request Mode Initiated.
Vector	Present Mode Terminated. Requested Mode Initiated.
Operator Select	Transfers display to its own refresh memory.
Slide Select	Present Mode is terminated. Transfers display console to its own refresh memory. The requested mode is initiated.
CCTV Select	Identical with Slide Select
Format Select	Identical with Slide Select
Display Test	Identical with Slide Select
Hard Copy	Hard Copy request is cancelled and requested mode is initiated.

As illustrated in Figure 4.2.16, the following events would occur as a result of setting bit zero (X) of the second control word.

- Operator Select - The existing command function on the console (Op. No.) represented by bits 19-23 would be cleared and its display would transfer to its own refresh memory. Subsequently an Operator Select would be performed on the console specified in bits 12-17 of control words 3 and 4.

- TV Transmit - With bit zero set, the existing command function on the console (Op. No.) represented by bits 19-23 would be cleared and a TV Transmit function would be performed on the refresh memory of the console represented in bits 12-17 of control words 3 and 4.

- Command Function Clear Select - When the second control word is coded as illustrated, the command function existing on the console (Op. No.) will be cleared.

4.3 PERIPHERAL DEVICES INPUT/OUTPUT

The Input/Output Control for Data Channels One and Two (SIOC) program provides communication for test programs with the magnetic tape stations, magnetic drums, card readers, and line printers at the LCCC, and for function executor routines with magnetic drum at the MLC. These peripheral devices and their IODC assignments are as shown in Figure 4.3.1. The basic operations are read, write, and control. The program that performs these operations (SIOC) may be accessed by use of the calling sequences given in Table A-3, Appendix A.

There are a number of different calling sequences for read, write, and control operations. Each calling sequence contains the appropriate input parameters required for the particular operation. The control word formats and bit positions are indicated in the following paragraphs of this section. These formats are used with the coding examples given in the calling sequences.

Each calling sequence also contains a transfer instruction. This instruction is a TSPO SIOC, 7, which links the test program or function executor routine to the SIOC program. The number of control words in the calling sequences varies according to the type of operation desired. The two core locations which follow the control words are reserved for Operating System returns to the test program or function executor routine. The Operating System returns control to the first of these two core locations if the SIOC request stack is full. This condition is indicated as the WAIT return. This location must be coded to transfer to the routine which processes this stack full condition. The core location which follows the WAIT location is the normal return. The Operating System returns control to this location on three conditions: request stacked, operation failed, and operation completed. The Operating System sets bits 23 and 22 with certain codes to indicate that one of these three conditions has occurred. These bits are known as the operation status bits. The codes and corresponding conditions are as follows:

00 - Request Stacked - Bit 23 = 0, Bit 22 = 0
01 - Operation Failed - Bit 23 = 0, Bit 22 = 1
10 - Operation Succeeded - Bit 23 = 1, Bit 22 = 0

This location should be coded to check the type of condition that occurs and to then process accordingly.

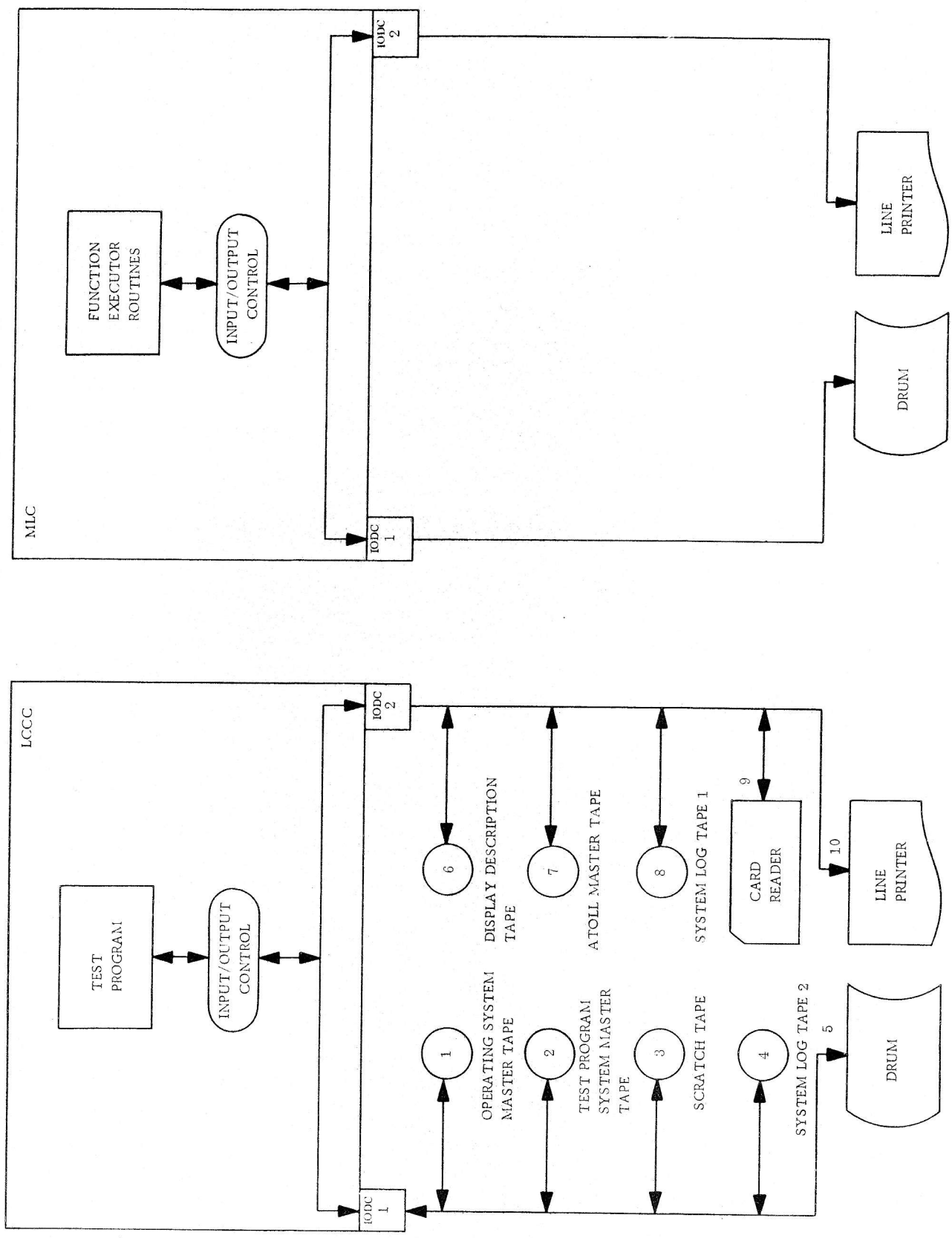


Figure 4.3.1 Saturn Peripheral Device Assignments

4.3.1 Magnetic Tape Input/Output

The Operating System provides a variety of tape operations that may be performed by the test programs. These operations are read, write, backspace, skip, write end of file, and rewind tape. Read/write tape operations have priority over the backspace and skip tape operations; and thus, the test programmer should not attempt a read/write operation until the backspace or skip tape operation has been completed. The test programmer may check the operation status bits as a precaution against this occurrence.

The test programmer must supply input parameters for these tape operations in a three-word format (see figure 4.3.2). These parameters are the operation code, logical tape number, data address, and number of words. The Operation Code is a two-digit octal code indicating the type of operation. The logical tape number is the number assigned to access a particular tape unit. The data address is the address of the first core location where data is accessed for a write operation or where data may be placed on a read tape operation. The number of words to be written on tape is given with a maximum of 4095 words per data block.

NOTE

The only tape accessible by test programs is the scratch tape, assigned logical number 3 of IODC 1, at the LCCC.

TRANSFER

23 22	21	20 19	18 10	9 6	5 0
		NOT USED	NUMBER OF FILES/ RECORDS TO BE SKIPPED	TAPE NUMBER	TAPE OPERATOR CODE

23 0
HSM DATA ADDRESS WHERE DATA IS TO BE PLACED ON READ OR ACCESSED FOR WRITE

23 0
NUMBER OF WORDS TO BE READ FROM OR WRITTEN ON TAPE

WAIT

RETURN

NOTE

- BITS 23, 22: OPERATION STATUS
- BIT 21: END OF FILE (EOF) DETECTED ON READ

Figure 4.3.2 Word Formats for Tape Operations

4.3.1.1 Read Tape

There are four read tape routines: read packed BCD data, read unpacked BCD data, read packed binary data, and read unpacked binary data. The macro name and operation code for each is indicated in Table A-3, Appendix A. Figure 4.3.2 shows the control word format for these operations. The operation code is placed in bits 5 through 0 of the first control word. The tape number for the specific operation is placed in bits 6 through 9 of the same control word. The core address for data to be placed after the tape read operation is specified in bits 14 through 0 in the second control word. The number of words to be read is contained in the third control word, bits 11-0, with a maximum read of 4095 words per data block.

NOTE

When the Operating System completes an End-of-File read, bit 21 of control word is set to a one, and one computer word (17000000) is read into core.

4.3.1.2 Write Tape

There are two write tape routines. One is for writing packed BCD data; its macro name is WTDP, and its operation code is 035. The other routine writes packed binary data; its macro name is WTBP and its operation code is 036. Figure 4.3.2 shows the control word format for these operations. It is similar to the read tape operation as explained in the preceding paragraph except for the data address. The data address is the first core location where the data is to be accessed for a write tape operation.

4.3.1.3 Backspace Tape

This routine backspaces any number of records or files as specified by the test program. The specific operation is specified with the appropriate code. To backspace records, the macro name is SKRB and its operation code is 040. To backspace files, the macro name is SKRB and the operation code is 041. Figure 4.3.2 shows the control word format for the input parameters. The operation code is specified in bits 5 through 0. The logical tape number is specified in bits 6 through 9. The number of records or files to be backspaced is specified in bits 10 through 18. The last two control words shown are not used for this operation.

NOTE

When a backspace file command is given the tape is moved backward passed the EOF and then a skip record forward is initiated. Therefore, the tape is positioned in front of the EOF.

4.3.1.4 Skip Tape Forward

This routine skips tape forward for any number of records or files. The number of records or files is specified by the test programmer. He specifies the specific operation with the appropriate code. To skip a specified number of files forward, the macro name is SKFF, and its operation code is 043. To skip a specified number of records forward, the macro name is SKFF and the operation code is 042. The control word format for these operations is the same as the backspace operation, and is shown in Figure 4.3.2.

4.3.1.5 Write End of File

This routine writes an End of File (EOF) on tape. The macro name is WEOF and its operation code is 026. Figure 4.3.2 shows the control word format for this operation. The operation code is specified in bits 5 through 0 and the tape number is specified in bits 6 through 9.

4.3.1.6 Rewind Tape

This routine rewinds tape. The macro name is REWD and the operation code is 027. Figure 4.3.2 also shows the format for the input parameters. This format is the same as for a Write End-of-File operation as explained in the preceding paragraph.

4.3.1.7 Re-Allocate LCCC Tape Station

LCCC Magnetic Tape Stations (MTS's) are initially assigned as follows: MTS's 1 through 5 are on IODC 1, and MTS's 6 through 9 are on IODC 2. This routine provides a display of current MTS assignments if $ULTA is entered via the display console keyboard. It may also be used to assign an MTS to another IODC by entering $ULTA X-Y (X and Y are replaced by the new IODC and MTS numbers, respectively).

4.3.2 Drum Input/Output

The Operating System provides read or write operations for communication with the magnetic drum. There are two macro instructions and calling sequences used for these operations. The test program or function executor routine must supply input parameters with these calling sequences. The parameters are the operation number, the core address, the number of words, the drum IVAR and the drum increment from that IVAR.

The control word format for these parameters is shown in Figure 4.3.3. The operation code indicates whether a drum read or write operation is desired. The core address is the location where the data is to be accessed from for a write or where data is to be placed on a read. This address may be symbolic or absolute. The number of words indicates the number of data words expected to be read or written. There may be a maximum of 4095 words for one operation. The drum address is the drum location from which data is to be accessed, or where data is to be placed. This address may be a four-character BCD IVAR name for either the read or write operation. The drum increment address is the absolute amount added to the drum IVAR address so that the resultant address is the drum location of data to be read or written.

NOTE

Test programs may access drum areas SSSS, SSST, and SSSU.

The read drum operation accesses a specified number of data words from drum and places them in core. The macro name is RDRM and the operation code is 01. Figure 4.3.3 shows the control word format for this operation. The operation code is specified in bits 5 through 0 of the first control word. The core address is specified in bits 14 through 0 of the second word. The number of words read or written is specified by bits 11-0 in the third word. The drum address for read or write is specified by all bits in the fourth word. The last control word contains the drum increment address.

A write drum operation accesses a specified number of data words from core and writes them on drum. The macro name is WDRM and its operation code is 02. Figure 4.3.3 shows the control word format.

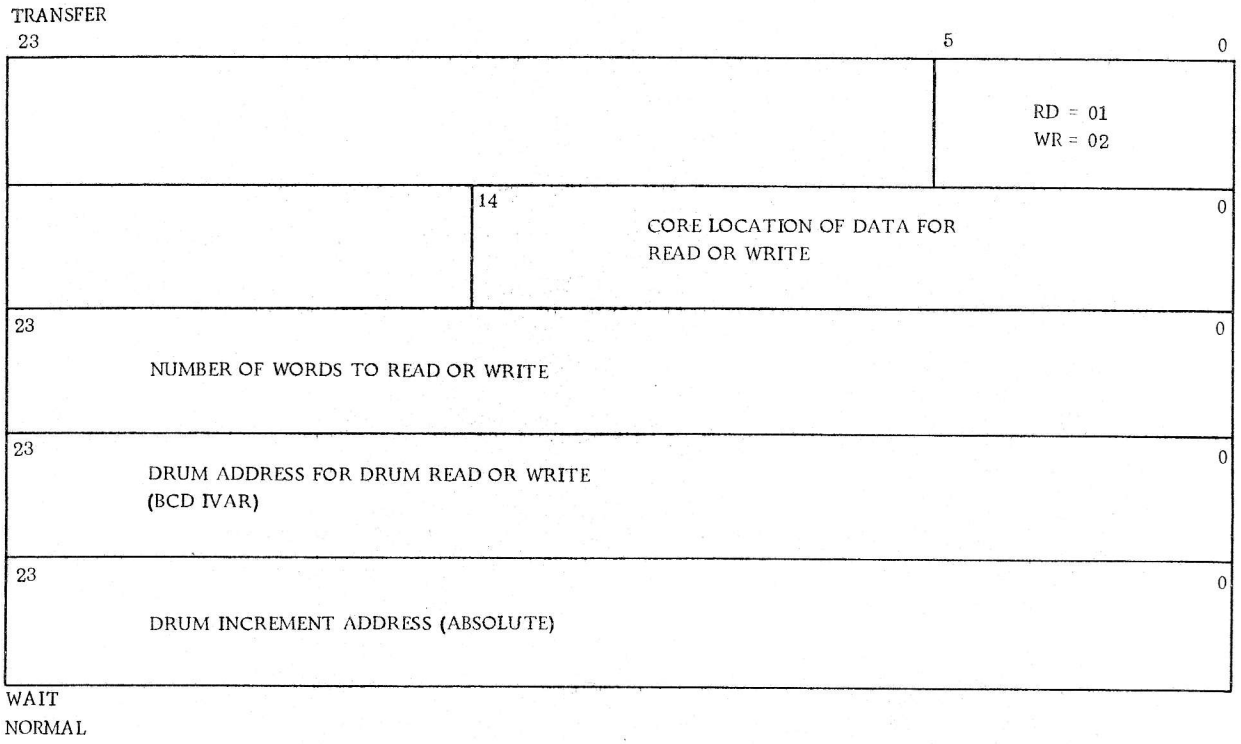


Figure 4.3.3 Drum Read and Write Control Words

4.3.3 Card Reader Input - LCCC

The Operating System does not permit the test program to punch out data on cards; it only permits the reading of data. This data may be read from cards either in binary or BCD form. The mode for reading BCD data is multiple character Input. This mode places two or four card column BCD characters into one data word.

The Operating System provides two macro instructions and their calling sequences for these operations. They allow the test program to link to the two Operating System routines which perform card read operations. The test program must supply three parameters for the Operating System. They are the operation code, the data word address, and the number of data words. The operation code indicates the type of card read operation to be performed. The data word address is the core location where the first data word is to be placed. This address may be symbolic or absolute. The number of words indicates the total number of data words required to contain the input data.

If the card format is BCD, 4 columns are required for each word allowing a maximum of 20 words per card. For binary data, one word is accommodated in every two columns to allow a maximum of 40 words per card.

4.3.3.1 Read Binary Card Data - LCCC

This routine reads data from cards punched in a binary format. The macro name is RBIN and the operation code 011. The format for the control words is shown in Figure 4.3.4. The operation code is placed in the first control word. The address of the first core location where data is to be placed is specified in bit positions 14 through 0 of the second control word. The number of data words required for the input data is specified in the third control word bits 11-0.

4.3.3.2 Read BCD Card Data - LCCC

This routine reads data from cards punched in a BCD format. The macro name is RBCD and the operation code 010, respectively. The control word format for the specified parameters is shown in Figure 4.3.4. This format is the same as the Read Binary Card Data operation.

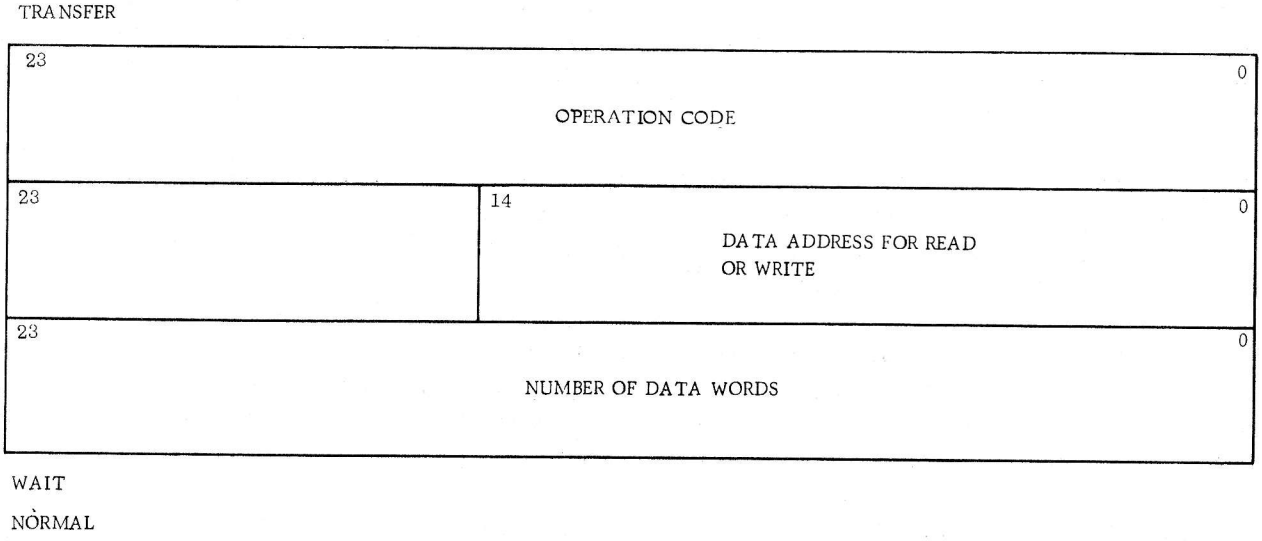


Figure 4.3.4 Read Card, Write Printer Control Words

4.3.4 Line Printer Output

Direct use of the line printer by the test programmer is allowed only at the LCCC. The MLC printer is used by the Operating System only in the event of a Data Link failure; if this occurs, all messages intended for the LCCC printer are generated out instead at the MLC printer. Thus, function executor messages normally printed at the LCCC via the Remote Display and Print Service routine of the System Message program are printed instead at the MLC. In the case of general messages, only the message number, GMT, and variable data words are printed. When error messages or system messages are generated, the line printer immediately skips eight lines on the printed page. This action tends to cause all error messages to be more noticeable. Additionally, GMT and CDC are printed within 30 seconds of any line printer activity.

Whenever a print operation is performed, the multiple character mode is used. This mode accepts data to be printed in a four character per word format. Data will be printed only for the two following conditions:

- When sufficient data is transferred to fill the 120 character (30 word) printer buffer.
- When the data transferred contains a carriage return code.

For multi-line printing, the buffer will continue to print as long as a carriage return code is not found, or a full line of 30 words is put into the print buffer. During the multi-line printing, no provision is made for double- or triple-spacing; however, the output data may contain the carriage return for activating line spacing. The programmer may position the printer up 16 lines by means of setting bit 6 of the Sense Control Panel.

The test program may perform two types of operations with the printer. They are print data and position the page. There are three macro instructions with their calling sequences used to perform these operations. They are print data, print data and position to the top of the next page, and position the printer page to a desired location.

The Operating System requires three parameters to perform these operations. They are the data word address, the number of data words, and the page position code. The data word address is the core location of the first data word and indicates from where the data may be accessed for printing. The address may be either symbolic or absolute.

The number of data words indicates to the Operating System the number of data words to print. The page position code indicates to the Operating System where to position the page for printing. These codes are shown in Table 4.3.1.

Table 4.3-1 Page Position Codes

Printer Code	Control Loop Channel Number	Page Position
1	2	Top of Page
2	3	Triple Space
3	4	1/4 Page Space
4	5	Double Space
5	6	1/2 Page
6	7	Single Space
7	8	End of Page

4.3.4.1 Print Data

This routine prints a specified number of data words. The MACRO name is WRPT, and the operation code is 014. The format for the three parameters are shown in Figure 4.3.4, paragraph 4.3.3.2. The operation code is specified in the first control word; the data address is specified in bits 14 through 0 of the second control word; and the number of words to be printed is specified in the third control word.

4.3.4.2 Print Data and Position Page

This routine prints a specified number of data words and then positions the top of the next page for further printing. The operation code is 015. The parameters are specified as explained in the preceding paragraph. Figure 4.3.4 shows the format of these parameters.

4.3.4.3 Position Printer Page

This routine positions the printer form to the desired location as specified by the test program. Figure 4.3.5 shows the control word format for this operation and indicates the two required parameters specified by the test program. The operation code is contained in bits 5 through 0 and is in octal. The page position code is contained in bits 13 through 10. This code is an octal number 1 through 7 (Table 4.3-1).

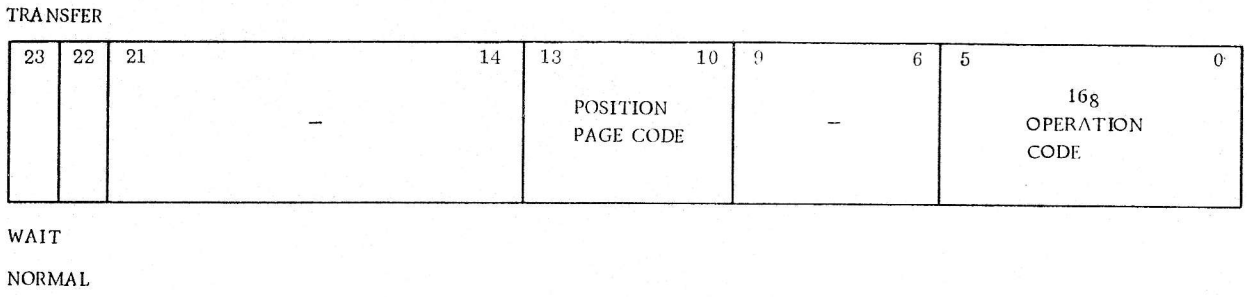


Figure 4.3.5 Page Position Control Word Format

Appendices

APPENDIX A
CALLING SEQUENCES

Table A-1. Discrete Status Check Calling Sequence

Function	Macro Call and Expansion		Calling Parameters	Return Parameters	
	OP Field	Variable Field		Left Accumulator	Right Accumulator
Single MDI in LCCC and MLC	SMDI — — TSPO * WORD 1	X, S — — — — SAEA, 7 S, O, O, X NORMAL RETURN	X = MDI Number, Bits 11-0 S = MDI Expected Status, Bit 23 (0=OFF, 1=ON)	+ = Status as Expected - = Status not as Expected 0 = Data Link Failed	
Single LDI in LCCC	SLDI — — TSPO * WORD 1	X, S — — — — SAGA, 7 S, O, O, X NORMAL RETURN	X = LDI Number, Bits 11-0 S = LDI Expected Status, Bit 23 (0=OFF, 1=ON)	+ = Status as Expected - = Status not as Expected 0 = Data Link Failed 40000010 = Parameter input error	
Single MDO in LCCC and MLC	CMDO — — TSPO NUM	X — — — — — SANA, 7 X NORMAL RETURN	X = MDO Number To Check, Bits 11-0	+ = DO is ON - = DO is OFFF 0 = Data Link Failed 40000010 = Parameter input error	
Single LDO in LCCC	CLDO — — TSPO NUM	X — — — — — SAMA, 7 X NORMAL RETURN	X = LDO Number To Check, Bits 11-0	+ = DO is ON - = DO is OFF 40000010 = Parameter input error	
Multiple MDI in LCCC and MLC	MMDI — — TSPO * WORD 2 NUM	X, Y, MM — — — — SAFA, 7 X, Y MM NORMAL RETURN	X = Low Number of MDI's, Bits 23-12 Y = High Number of MDI's, Bits 11-0 MM = Mask Word of Expected MDI Status (Octal and Right Justified)	+ = Status as Expected - = Status not as Expected 0 = Data Link Failed	Mask Word of Unexpected Discretes (Octal and Right Justified)
Multiple LDI in LCCC	MLDI — — TSPO * WORD 2 NUM	X, Y, MM — — — — SAHA, 7 X, Y MM NORMAL RETURN	X = Low Number of LDI's, Bits 23-12 Y = High Number of LDI's, Bits 11-0 MM = Mask Word of Expected LDI Status (Octal and Right Justified)	+ = Status as Expected - = Status not as Expected 0 = Data Link Failed 40000010 = Parameter input error	

*WORD 1 FORM 1, 1, 10, 12
WORD 2 FORM 12, 12

NOTE 1: The current status table (MDI, LDI, MDO, and LDO) is used for the status check.
NOTE 2: The discrete status check routines are not protected beyond level zero and level 1/2.

Table A-2. Issue Discrete Calling Sequences

Function	Macro Call and Expansion		Calling Parameters	Return Parameters	
	OP Field	Variable Field		Left Accumulator	Right Accumulator
Issue Single MDO from LCC and MLC	SMDO TSPO WORD 1	X, S SACA, 7 S, 0, 0, X NORMAL RETURN	X = MDO Number in Binary, Bits 11-0 S = MDO State to Issue, Bit 23 (1=ON, 0=OFF)	+ = Operation Successful - = Operation Failed Error Code in Bits 9-0 01 = Inhibited 02 = Disallowed 03 = Data Link Failure 04 = System in Idle Mode 05 = DO Parity, Do Not Issue 06 = MTOAD Failure 10 = Parameter Input Error	Mask Word of Inhibited or Disallowed DO's, (Right Justified)
Issue Multiple MDO from LCCC and MLC	MMDO TSPO WORD 2 NUM NUM	X, Y, MM, II SADA, 7 X, Y MM II NORMAL RETURN	X = Lowest MDO Number of Group, Bits 23-12 Y = Highest MDO Number of Group, Bits 11-0 MM = Mask Word of Discrete State (Octal and Right Justified) II = Mask Word of Discretes to Issue (Octal and Right Justified). If Mask=0, All Discretes designated by Word 2 are Issued.		
Issue Single LDO from LCCC	SLDO TSPO WORD 1	X, S SAPA, 7 S, 0, 0, X NORMAL RETURN	X = LDO Number, Bits 11-0 S = LDO State to Issue, Bit 23 (1=ON, 0=OFF)		
Issue Switch Selector Function from LCCC and MLC	SWSEL TSPO WORD 4	X, Y, R, C SAKA, 7 0, C, R, Y, X NORMAL RETURN	X = Switch Selector Channel Number, Bits 7-0 Y = Stage Select Code, Bits 12-8 (01=S-IC, 02=S-II, 04=S-IVB, 10=S-IB, 20=IU) For S-IB system operation both S-IC and S-II stage select codes are disallowed. R = Read Command, Bit 13 (1=Issue, 0=Do not Issue) C = Complement Channel, Bit 14 (1=Issue Complement, 0=Issue True Channel Number)	+ = Command Issued - = Operation Failed Due to Hardware 0 = Operation Failed Due to Environment (See Error Codes in Right Accumulator)	Switch Selector Error Codes Bits 8-0 01 = Calling parameter input error 02 = Stage inhibited by switch 03 = Stage power off 04 = Stage zero signal is not present 05 = Stage is hardware inhibited 06 = Manual panel using switch selector 07 = Interface cannot be cleared; read command not issued 10 = MDO switch inhibited during issued 11 = MDO ESE switch inhibited during read 12 = IU power off 13 = LVDC using switch selector 14 = Data link failure 15 = System in idle mode 16 = No vehicle read verification 17 = Interface not clear - Read Issued 20 = ESE/LVDA Switch in LVDA position

*WORD 1 FORM 1, 1, 10, 12
WORD 2 FORM 12, 12
WORD 4 FORM 9, 1, 1, 5, 8

NOTE: The issue discrete routines are not protected beyond level zero and level 1/2.

Table A-2. Issue Discrete Calling Sequences (Continued)

Function	Macro Call and Expansion		Calling Parameters	Return Parameters	
	OP Field	Variable Field		Left Accumulator	Right Accumulator
Issue Switch Selector Function (Continued)					Switch Selector Error Codes (Cont'd) Bits 8-0 17 = Interface cannot be cleared; read command issued 20 = ESE/LVDA switch in the LVDA position
Issue Pulsed MDO and Check MDI Response (LCCC) * *	PDOCK TSPO WORD 3 WORD 5	X, W, Y, Z, T, XLS SAJA, 7 W, X, Y XLS, Z, 0, T NORMAL RETURN	X = MDO number, Bits 22-12 W = Initial State to Set the MDO, Bit 23. Y = MDI Number to Check, Bits 11-0. If = 0, No MDI Check is Made. Z = MDI Expected State, Bit 22. T = Pulse Time in Milliseconds, Bits 11-0. XLS = Final State to Set MDO After Expiration of Pulse Time, Bit 23.	+ = The Interval Between the Time the MDO was Issued and the MDI Response. 0 = MDO Issued With No MDI Check Requested. - = MDO Inhibited or Disallowed, or Improper MDI Response During Requested Check. Bits 8-0 Contain One of the Following Error Codes: 001 = Inhibited MDO 002 = Disallowed MDO 003 = Data Link Failure 004 = System in Idle Mode 005 = MDO Parity Error 006 = MTOAD Failure 010 = Parameter Input Error	Left Accumulator Positive or Zero: Value of Time Overrun During Which MDO Remained in Initial State After Pulse Time Expired. Value May Be Zero. Left Accumulator Negative: Bits 5-0 Contain One of the Following Error Codes: 01 = MDO Inhibited on Attempt to Issue Initial State 02 = MDO Disallowed on Attempt to Issue Initial State 03 = MDO Inhibited on Attempt to Issue Final State 04 = MDI Did Not Respond Properly Within Pulse Time
*WORD 3 FORM 1, 11, 12 WORD 5 FORM 1, 1, 7, 15					

NOTE: The issue discrete routines are not interrupt-protected beyond level zero and level 1/2.

Table A-3. Input-Output Control (SIOC) Calling Sequences for LCCC and MLC

I/O Device	Function	Macro Call XVAR	Octal ID KK		Macro Call and Expansion OP Field	Macro Call and Expansion Variable Field	Calling Parameters	Return Parameters
Tape (LCCC)	**Read Packed BCD Data **Read Packed Binary Data **Read Unpacked BCD Data **Read Unpacked Binary Data Write Packed BCD Data Write Packed Binary Data	RTDP RTBP RTDU RTBU WTDP WTBP	31 32 33 34 35 36		XXXX TSPO RFORM PZ5 NUM TRA	TN, HSMAD, NWDS, WAIT SIOC, 7 M, TN, KK HSMAD NWDS WAIT NORMAL RETURN	XXXX - Macro Call TN - Tape Unit No., Bits 9-6 HSMAD - Starting Core Address, Bits 11-0 NWDS - No. of Words Transferred, Bits 11-0 (Maximum of 4095_{10}) WAIT - Name of Routine to Return to if SIOC Request Stack is Full SKP - No. of Files (Records) to Skip or Backspace, Bits 18-10 (Maximum of 777_8 or 512_{10}) DRMAD - Four Character BCD Name of Data IVAR for Read Drum - Four Character Symbolic Address of Data IVAR for Write Drum DRINC - Absolute Increment Address for Drum Operation KK - Octal ID for Macro Call, Bits 5-0	**Bit 21 = 1 of Y+1 Word Indicates an EOF Was Read During a Read Tape Operation Bits 23 & 22 of Y + 1 Indicates Status of Operation 10 - Completed 01 - Failed 00 - Request Stacked Bits 23 - 12 of Y+3 Contains the Number of Words Read.
	Backspace Record(s) Backspace File(s) Skip File(s) Forward Skip Record(s) Forward	SKRB SKFB SKFF SKRF	40 41 43 42	*	XXXX TSPO KFORM TRA	SKP, TN, WAIT SIOC, 7 M, SKP, TN, KK WAIT (NOTE 1) NORMAL RETURN		
	Write End of File Rewind Tape	WEOF REWD	26 27	*	XXXX TSPO RFORM TRA	TN, WAIT SIOC, 7 M, TN, KK WAIT NORMAL RETURN		
Drum (LCCC and MLC)	Read Drum Write Drum	RDRM WDRM	01 02		XXXX TSPO NUM PZ5 NUM BCD NUM TRA	HSMAD, NWDS, DRMAD, DRINC, WAIT SIOC, 7 KK HSMAD NWDS 1, DRMAD DRINC WAIT NORMAL RETURN		

*KFORM FORM 5, 9, 4, 6
*RFORM FORM 14, 4, 6

NOTE 1: Tape must be pre-positioned for correct data transfer. Since Read/Write has priority over tape positioning, the complete bit for position tape must be received before initiating Read/Write.

Table A-3. Input-Output Control (SIOC) Calling Sequences for LCCC and MLC (Continued)

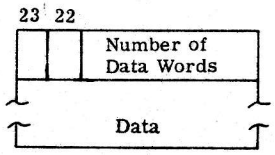
I/O Device	Function	Macro Call XVAR	Octal ID KK	Macro Call and Expansion OP Field	Macro Call and Expansion Variable Field	Calling Parameters	Return Parameters
Card & Printer	Read Card Binary Data-LCCC Read Card BCD Data-LCCC Write Printer-LCCC & MLC Write Printer & Position to Top of Page-LCCC	RBIN RBCD WRPT WPNP	11 10 14 15	XXXX TSPO NUM PZ5 NUM TRA	HSMAD, NWDS, WAIT SIOC, 7 KK HSMAD NWDS WAIT NORMAL RETURN	ADV - Code to Position Printer Page, Bits 13-10 1 = Top of Next Page 2 = Triple Space 3 = Quarter Page 4 = Double Space 5 = Half Page 6 = Single Page 7 = End of Page M, N } Operating System Control Fields	
Printer (LCCC Only)	Position Printer Page	APFM	16	XXXX TSPO * PFORM TRA	ADV, WAIT SIOC, 7 M, ADV, N, 016 WAIT NORMAL RETURN		

*PFORM FORM 10, 4, 4, 6

Table A-4. LVDC Interface and Display Service Calling Sequence

Operating System Programs	Function	Macro Call and Expansion		Calling Parameters	Return Parameters	
		OP Field	Variable Field		Left Accumulator	Right Accumulator
LVDC Interface (LCCC)	Command LVDC to Status XX	SCMND TSPO PZ5 TRA	XX, ERROR SFLT, 7 XX, 2 ERROR NORMAL RETURN	XX = Status of LVDC ERROR = Address of Error Recovery Routine	Status of LVDC, Bits 12-0 Error Return, Bit 23 = 1 Unable to Execute Bit 23 = 0 Data Link Failed NOTE: For LCCC, If L = 0, No Reply Received from MLC.	
	Send Data to LVDC	SWRTE TSPO MZ5 TRA	ADDRS, ERROR SFLT, 7 ADDRS ERROR NORMAL RETURN	ADDRS = Address of First Data Word (See NOTE 1)		
	Read 1 LVDC WORD	READ TSPO BCD TRA	ERROR SFLT, 7 1, READ ERROR NORMAL RETURN	ERROR = Address of Error Recovery Routine	Data Word 1, Bits 23-0 Error Return, Bit 23 = 1 Unable to Execute Bit 23 0 Data Link Failed NOTE: For LCCC, If L = 0, No Reply Received from MLC.	Bits 10-0 Data Word 2
	Set SYNC X Times	SYNC TSPO PZ5 TRA	X, ERROR SFLT, 7 X, 4 ERROR NORMAL RETURN	X = Number of times to set Sync. (Program waits for LVDC to reset it each time.) ERROR = Address of Error Recovery Routine	Number of times sync is set, Bits 12-0 Error Return, Bit 23 = 1 Unable to Execute Bit 23 0 Data Link Failed NOTE: For LCCC, If L = 0, No Reply Received from MLC.	
	Set Sync One Time, Do Not Wait for Reset	SYNCB TSPO PZ5 TRA	ERROR SFLT, 7 0, 6 ERROR NORMAL RETURN			
	Reset Sync	RSYNC TSPO NUM TRA	ERROR SFLT, 7 0100000 ERROR NORMAL RETURN			

NOTE 1: ADDRS



Bit 23 = 1 Do not issue sync after data is sent
Bit 22 = 1 Issue sync, but do not wait for sync reset after data is sent

Table A-4. Remote Display and Display Service Calling Sequences (Continued)

Operating System Programs	Function	Macro Call and Expansion		Calling Parameters	Return Parameters
		OP Field	Variable Field		
System Message-Remote Display Requests	Request Display of Raw Binary, Packed BCD, Memory, Pre-Formatted, Display Description, or Print of BCD Data (MLC)	RDAP	OM, NW, DATAR	DATAR = 15-bit symbolic address of first word of data. OM = Output mode (bits 23-21): 000 = Display only 001 = Print only 010 = Display and print NW = Number of words in message including control words (bits 20-0); 102 words maximum. DATA = 15-bit symbolic address of first control word of data to be displayed (NOTE 2). BUSY = Location to transfer to when request cannot be accepted.	Operation Status, Bits 23 and 22 of Y + 2 Word 01 - Transmission Failed 10 - Transmission Successful
		TSPO TFORM* PZ5 TRA	SRDY, 7 OM, NW DATAR BUSY NORMAL RETURN		
Display Service-Local Display Requests	Request Display of Raw Binary, Packed BCD, Memory, Pre-Formatted, or Display Description Data (LCCC)	DISP	DATA, BUSY		Operation Status, Bits 23 and 22 of Control Word 1 00 - Request Stacked 01 - Data Not Displayed 10 - Data Displayed
		TSPO PZ5 TRA	SDSR, 7 DATA BUSY NORMAL RETURN		

*TFORM 3, 21

NOTE 2: DATA

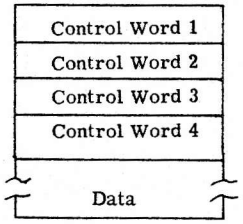


Table A-5. Analog I/O Program Calling Sequences

	Function	Macro Call and Expansion		Calling Parameters	Return Parameters	
		OP Field	Variable Field		Bank Zero, Word 2	Bank Zero, Word 1
LCCC DC Analog Input Interface and MLC Analog Input Read	Read an Analog Voltage	DALIN TSPO NUM	A, C SDA1, 7 A	A = Test Point Number to Be Read	Left Accumulator Bit 0-9 = Value of analog voltage Bit 10 = Sign of voltage Bit 23 ON = Analog has not been read.	Right Accumulator 23 LSB of GMT when analog was read

Table A-6. Test Program Control, Loader, and Function Executor Calling Sequences

Operating System Programs	Function	Macro Call and Expansion		Calling Parameters
		OP Field	Variable Field	
Test Program Control (LCCC)	Display Options Specified By Test Program	OPTION TSPO PZ5 *SFORM	HEADR, A, B, C, D, TBL ST11, 7 HEADR A, B, C, D,, TBL	HEADR = Address of Header Message, Bits 14-0 A, B, C, D = 0 - Not to Display Options = 1 - To Display Options TBL = Address of Table, Bits 14-0 Note - The text for each option entry is limited to a maximum of 28 characters (7 words)
	Display Standard Options	NOGO TSPO PZ5 BCD	ADDRS, XX ST12, 7 HEADR 1, XX	HEADR = Address of Header Message to be Displayed, Bits 14-0 XX = Option Selected, Either C, or D, or both **
	ATOLL call for Machine Language Test Program	TSPO BCD NUM	ST15, 7 1, NAME 0XX000000 NORMAL RETURN	NAME = Test Program to be Called XX = Console Card Key Left Accumulator Upon Return: - = Card Code Invalid + = Unable to Load/Execute
	Test Program Self-Terminate and Execute a Test Program	CALL TSPO BCD	XXXX ST16, 7 1, XXXX	XXXX = Name of Test Program
	Display Message to Request Parameters	DSPLY TSPO PZ5	ADRS ST17, 7 ADRS NORMAL RETURN	ADRS = Address of First Word of Message to Be Displayed
	Return to Op. Sys. After Normal Completion	RTRN TSPO	ST18, 7	---
	Return to Op. Sys. if Abnormal Termination	TERM TSPO	ST19, 7	---
Test Program Loader (LCCC)	Load a Subroutine from Tape to Core or Preposition the Test Program Tape	LOAD TSPO BCD TRA	XXXX, E SUBR, 7 1, XXXX ERROR NORMAL RETURN	XXXX = Name of Test Subroutine ERROR = Location to Return to if Subroutine is Not Loaded * If bank Ø cell CLTTPP is non zero this sequence prepositions Test Program Tape to desired file.

*SFORM FORM 1, 1, 1, 1, 5, 15

** When standard options are displayed via ST12, the 100 word limit restriction on the programmer supplied header message is reduced by the number of words required for Standard Options A, B, C and D. The word count of these options are:

Option A & B	19 words
Option C	5 words
Option D	5 words

Descriptions of options selected and resultant program operation is provided in Section II of this document.

Table A-6. Test Program Control, Loader, and Function Executor Calling Sequences (Continued)

Operating System Programs	Function	Macro Call and Expansion		Calling Parameters	Return Parameters Left Accumulator
		OP Field	Variable Field		
Function Executor (MLC)	Call a Function Executor Routine	FUNC	XXXX, SEC, PCT, PARM, RTN	XXXX = BCD IVAR ID of Routine SEC = Time Interval of Routine ($010_{10} \leq$ sec $\leq 999_{10}$) (NOTE 1) PCNT = No. of Parameter Words PARM = Address of Parameters (0 If no parameters) RTN = 1 - Return after request has been transferred to MLC = 0 - Return after MLC Function Executor Routine has been given control	+ = Cannot Execute - = Routine has Control in MLC 0 = Request Transferred to MLC
		TSPO BCD BCD BCD NUM PZ5	SMFX, 7 1, FUNC 1, XXXX 1, /SEC PCNT PARM, RTN NORMAL RETURN		
	Terminate a Function Executor Routine	FUNT	XXXX	XXXX = BCD IVAR of Routine to Terminate	+ = Operation Failed - = Operation Successful
		TSPO BCD BCD	SMFX, 7 1, FUNT 1, XXXX NORMAL RETURN		

NOTE 1: SEC = 000 if routine is to be executed once.

Table A-7. DDAS, Logging, and Remote Core-to-Core Calling Sequences

Operating System Programs	Function	Macro Call and Expansion		Calling Parameters	Return Parameters
		OP Field	Variable Field		
DDAS Executive (LCCC and MLC)	Request DDAS Read Operation (LCCC and MLC)	XDDS	MODE, TABLE, ERRET	MODE = 0 - Single Scan, Sequential Transfer (MLC DDAS Allowed)** = 1 - Single Scan, Sequential Transfer = 2 - Log Mode, Updated Block Transfer = 3 - Special Mode = 4 - Log Mode, Updated Block Transfer, Terminate Read when Log Table is 1/2 Full (MLC DDAS Allowed)** = 5 - Special Mode, Terminate Read when Log Table is 1/2 Full. = 7 - Special MLC Mode - Provide GMT (2 Words); and Count Clock Value and Count Clock Status (5 Words) from LCCC = 10 - Log Mode Updated Block Transfer (DDAS Normal Sequence Sub-Mode)* = 12 - Log Mode Updated Block Transfer (DDAS Normal Sequence Sub-Mode)* Terminate Read when Log Table is 1/2 Full. TABLE = Starting Address (Symbolic or Absolute) of DDAS Control Word Table ERRET = Location to Return when Unable to Process Request	Operation Status, Bits 23, 22 Successful - 10 Failed (Error) - 01 DDAS Read is in Progress, Bit 20 = 1 DDAS Calibrate Test is in Progress, Bit 19 = 1 (Word Y + 1)
		TSPO NUM PZ5 TRA	SXRS, 7 MODE TABLE ERRET NORMAL RETURN		
	Request DDAS Read with Delayed Return (LCCC)	XDDRH	TABLE, ERRET		
		TSPO PZ5 TRA	SXRH, 7 TABLE ERRET NORMAL RETURN		
	This sequence requests the DDAS Read and Hold Routine in the Single Scan, Sequential Transfer Mode on Level 0 only.				
	Terminate A Log Mode Read or Delete a DDAS Read Request (LCCC)**	XDDAT	ADDR	ADDR = Address (Symbolic or Absolute) or Original Read Request to be Terminated or Deleted.	Terminate Read Bit 23 = 1 Delete Request Bit 22 = 1 (Y + 1)
		TSPO PZ5	SXAT, 7 ADDR + 1 NORMAL RETURN		
System Logging	Transfer of Data to Log Tape (LCCC, MLC)	WLOGG	IDENT, VARBL, NWDS, HSMAD, FULL	IDENT = BCD 4 Character Name of Program Requesting Data to be Logged VARBL = A Word Available for Requesting Program to Further Identify its Data NWDS = Number of Words to be Logged HSMAD = 15 Bit or Symbolic Address of Data to be Logged FULL = Location to Transfer when Logging Request cannot be Accepted	Request Status Stacked Bit 23 = 0 Serviced Bit 23 = 1 (Word Y + 4)
		TSPO BCD NUM NUM PZ5 TRA	SLOG, 7 1, IDENT VARBL NWDS HSMAD FULL NORMAL RETURN		

*Normal sequence sub-mode is specified by calling program via control C words.

**MLC DDAS Request Restrictions

1. MLC DDAS request limited to two modes: Single Scan Sequential Transfer or Log Mode Terminate on Half Full
2. Control A, B, and C words must be contiguous in core.
3. Maximum of 8 Control C words and 50 DDAS data words. Data word input buffer must be 7 words larger than number of expected data words for storage of GMT in binary, and value of countdown clock in BCD and in binary at time of DDAS read. Refer to paragraph 4.1.4.8 of Volume II for the format of the DDAS log table.

Table A-7. DDAS, Logging, and Remote Core-to-Core Calling Sequences (Continued)

Operating System Programs		Function	Macro Call and Expansion		Calling Parameters	Return Parameters
			OP Field	Variable Field		
DATA LINK EXECUTIVE	Remote Core-to-Core Transfer Subroutine	Transfer Core Contents from Remote to Local Computer	RCOR	HSMAD, NWDS, RCMAD, WAIT	HSMAD = Starting Core Address in Local Computer (Symbolic or Absolute). Must Be in Octal When Absolute. NWDS = Number of Words of Data to Transfer (Maximum of 102410) RCMAD = 15-Bit Absolute Address of Starting Core Location in Remote Computer WAIT = Location to Return when Remote Core-to-Core Transfer Subroutine is Busy or Data Link Request Stack is Full	Operation Status, Bits 23, 22 Successful = 10 Failed = 01 (Word Y + 1)
			TSPO NUM PZ5 NUM PZ5 TRA	SCTP, 7 01 HSMAD NWDS RCMAD WAIT NORMAL RETURN		
		Transfer Core Contents from Local to Remote Computer	WCOR	HSMAD, NWDS, RCMAD, WAIT		
			TSPO NUM PZ5 NUM PZ5 TRA	SCTP, 7 02 HSMAD NWDS RCMAD WAIT NORMAL RETURN		
		Operating System MLC-LCCC Communication	RDLT	HSMAD, NWDS, DEST, FULL	HSMAD = Starting Core Address in Local Computer (Symbolic or Absolute) NWDS = Number of Words of Data to Transfer (Maximum of 511_{10}) DEST = Destination Code of Program Receiving Data Transmission FULL = Location to Return when Data Link Executive Stack is Full	Bit 23 = 1 Successful Bit 22 = 1 Unsuccessful Bit 21 = 1 No Read Area Available for Receiving Program (Word Y + 3)
			TSPO PZ5 NUM NUM TRA	SRDL, 7 HSMAD NWDS DEST FULL		

Table A-8. Utility Calling Sequences for LCCC and MLC

Routines	Function	Macro Expansion		Calling Parameters		Return Parameters	
		OP Field	Variable Field	Left Accumulator	Right Accumulator	Left Accumulator	Right Accumulator
Print Data (LCCC)	Deleted						
	Print Core Data	TSPO	SUDC, 7 NORMAL RETURN	First Data Word Address or XVAR	Last Data Word Address or XVAR	0 = Successful ± = Unsuccessful	---
Convert Data	Convert Octal to Binary	TSPO	SUOB, 7 NORMAL RETURN	4 LSD	4 MSD		Binary Number
	Convert Binary to Octal	TSPO	SUBO, 7 NORMAL RETURN	Binary Number		4 LSD	4 MSD
	Convert Binary to BCD	TSPO	SUBD, 7 NORMAL RETURN	Double Precision (Q of 23)		Dec. Pt. & 3 Fraction Digits	Sign & 3 Integer Digits
Read Relative GMT Count-Down Time	Read Relative Time (LCCC and MLC)	TSPO	SRRT, 7 NORMAL RETURN	---	---	23 LSB Relative Time	
	Read Relative Time with a Delay Check (LCCC and MLC)	TSPO NUM *	SRDC, 7 INCR NORMAL RETURN		Base Reference Relative Time	Accum.= (Base + Incr.) - Present If negative, time interval has elapsed. If positive, time interval has not elapsed, and contents of accumulator is equal to the time before expiration.	
	Read GMT and Convert to Binary	TSPO	SREB, 7 NORMAL RETURN	---	---	4 MSB	23 LSB
	Read GMT and Convert from Grey to BCD (LCCC and MLC)	TSPO BCD BCD BCD	SRED, 7 HHMM /SS. XXXℓ NORMAL RETURN			HHMM = Hours and Minutes (Y + 1) /SS. = Seconds (Y + 2) XXXℓ = Milliseconds (Y + 3)	
	Convert GMT from Binary to BCD	TSPO BCD BCD BCD	SRCN, 7 HHMM /SS. XXXℓ NORMAL RETURN	Binary GMT 4 MSB	Binary GMT 23 LSB		

*INCR - Increment Time in Milliseconds

Table A-8. Utility Calling Sequences for LCCC and MLC (Continued)

Routines	Function	Macro Expansion		Calling Parameters		Return Parameters	
		OP Field	Variable Field	Left Accumulator	Right Accumulator	Left Accumulator	Right Accumulator
Read Relative GMT Count-Down Time	Read Count-Down Clock and Convert to Binary and BCD (LCCC)**	TSPO BCD BCD BCD NUM NUM	SCDC, 7 T-DD H/M/S 23LSB 4MSB NORMAL RETURN			T-DD = Days (Y + 1) HHM = (Y + 2) M.SS = (Y + 3) 23 LSB = (Y + 4) 4 MSB = (Y + 5)	Binary Representation of Time
Save Register Data	Save L, R, Index Registers	TSPO PZ5	SSXR, 7 ADRS NORMAL RETURN	**ADRS - Core Address of Data, Bits 14-0			
	Restore L, R, Index Registers	TSPO PZ5	SRXR, 7 ADRS NORMAL RETURN				
Drum IVAR Search	Locate Address of Drum IVAR	TSPO	SLSV, 7		BCD IVAR	Address of IVAR on Drum L=Negative Error	Number of Words

** See DDAS Calling Sequence for MLC Requests to Read Countdown Clock Time

***Location	Contents
ADRS	R Accumulator
ADRS +1	L Accumulator
ADRS +2	Index Register No. 1
ADRS +3	Index Register No. 2
ADRS +4	Index Register No. 3
ADRS +5	Index Register No. 4
ADRS +6	Index Register No. 5
ADRS +7	Index Register No. 6

Table A-9. Display Console Keyboard Inputs to Utility Routines

Function	Keyboard Input	Parameters
Alter LCCC or MLC Memory	$UZCAYYYYY, XXXXXXXX	Z - Specifies the computer = M(MLC) = L(LCCC) YYYYY - Specifies the 15-bit address to alter XXXXXXXX - Specifies the new contents of memory address or drum address
Alter LCCC or MLC Drum	$UZDAIIII±CCCC, XXXXXXXX	IIII - Specifies the drum IVAR CCCC - Specifies the increment to be added or subtracted to the drum address XXXXXXXX - Specifies the new contents of drum address
Display LCCC or MLC Memory	$UZCDXXXXX	Z - Same as above XXXXX - Specifies the location to be displayed (15 bit address)
Display LCCC or MLC Drum	#UZDDIIII±CCCC	IIII and CCCC - Same as above
NOTE: Memory and drum displays will always result in a display of the requested word plus the succeeding words 15.		
Print LCCC or MLC Memory	$UZCPYYYYY, XXXXX	Z - Same as above YYYYY - Specifies the starting location to print (15 bit address) XXXXX - Specifies the ending location to print (15 bit address) IIII and CCCC - Same as above
Print LCCC or MLC Drum	$UZDPIIII±CCCC, NNNN	NNNN - Specifies the number of words to print
NOTE: The drum and memory print will be printed at the LCCC line printer. Index registers will only be printed with the memory print.		
Display LCCC or MLC Error Counters	$UZDE	Z - Same as above
Update Log Tape Label Record 1. Display and Print Current Label Record	$UDAY	
2. Update System Date	$UDAY MM/DD/YY	MM - Specifies the month DD - Specifies the day YY - Specifies the year TTTTTTTT - Specifies the test ID
3. Update System Date and Test ID.	$UDAY MM/DD/YY TTTTTTTT	
4. Update System Date, Test ID, and Vehicle Number	$UDAY MM/DD/YYTT TTTTTTSVVV	SVVV - Specifies the vehicle number
NOTE: The label record will be printed when $UDAY is entered, but $UDAY does not permit updating.		
Data Link Switchover	$UDLS	
DCC Memory Print	$UDCCXXXXX	XXXXX - Specifies the 15-bit starting address of print (prints 400_8 words)
LCCC Magnetic Tape Station Re-Allocation	$ULTA X-Y	X - Specifies the new IODC number Y - Specifies the new MTS number
NOTE: A table of current assignments will be displayed when $ULTA is entered.		
Reset Relative Time Circuitry	$UZRR	Z - Same as above

Table A-9. Display Console Keyboard Inputs to Utility Routines (Continued)

Function	Keyboard Input	Parameters
Close Out System Log Tape	$ULTC	None
Display Masked LDI Groups	$ULREDALL	This entry causes all LDI masked groups to be displayed and printed.
Mask an LDI Group	$ULREX-YYLMSK	This entry will mask the specified LDI group. X = 5 or 7 for IODC 5 or IODC 7 YY = LDI Group Number 1 through 63
Release LDI Masked Group	$ULREX-YYLREL	This entry will release the specified LDI group. X = 5 or 7 for IODC 5 or IODC 7 YY = LDI Group Number 1 through 63

APPENDIX B

OPERATING SYSTEM ROUTINE LINKAGE

APPENDIX B

TEST PROGRAM AND FUNCTION EXECUTOR ROUTINE LINKAGE
WITH OPERATING SYSTEM

LCCC	MLC	FUNCTION
Name	Name	
SACA	SACA	Issue Single MDO
SADA	SADA	Issue Multiple MDO
SAEA	SAEA	Check Single MDI Status
SAFA	SAFA	Check Multiple MDI Status
SAGA		Check Single LDI
SAHA		Check Multiple LDI
SAJA		Issue Pulsed MDO Check MDI
SAKA	SAKA	Issue Switch Selector Channel
SAMA		Check Status of LDO
SANA	SANA	Check Status of MDO
SAPA		Issue Single LDO
SCDC	SCDC	Read Countdown Clock
SCTP	SCTP	Remote Core-to-Core of Data Link Executive
SIOC	SIOC	System I/O Control IODC 1 and 2
SLOG	SLOG	System Logging Program
SRCN	SRCN	Convert GMT to BCD
SRDC	SRDC	Determine Relative Time Interval
SREB	SREB	Read GMT Return in Binary
SRED	SRED	Read GMT Return in BCD
SRRT	SRRT	Read Relative Time
SRXR	SRXR	Restore Indexes
SSXR	SSXR	Save Indexes
SUBD	SUBD	Binary to BCD Conversion
SUBO	SUBO	Binary to Octal Conversion
SUOB	SUOB	Octal to Binary Conversion
SUDC		Memory Dump Routine
SFLT	SFLT	LVDC Interface
SDSR	SRDY	Display Request
SDA1	DAL1	Read Analog
SDA2		Issue Analog
ST11		Display Programmer Options
ST12		Display Standard Options
ST16		Self-Terminate and Execute
ST17		Request Parameters
ST18		Test Program Normal Completion
ST19		Abnormal Termination Return
SUBR		Locate a test subroutine
SMFX		Call or Terminate a Function Executor Routine
SMFR		Returns to Function Executor
SXRS	SXRS	Request DDAS Read
SXRH	SXRH	Request DDAS Read and Delay Return
SXAT	SXAT	Delete DDAS Read and Terminate Log Mode Read
SLSG		Linkage to Load Test Program Segment

APPENDIX C

COMMON STORAGE AREAS

APPENDIX C

STANDARD STORAGE AREAS

ADDRESS	IVAR	OPER	CONTENTS	FUNCTION
0430	CCKM09	NUM	-9	Constant Pool
0431	CCKM08	NUM	-8	Constant Pool
0432	CCKM07	NUM	-7	Constant Pool
0433	CCKM06	NUM	-6	Constant Pool
0434	CCKM05	NUM	-5	Constant Pool
0435	CCKM04	NUM	-4	Constant Pool
0436	CCKM03	NUM	-3	Constant Pool
0437	CCKM02	NUM	-2	Constant Pool
0440	CCKM01	NUM	-1	Constant Pool
0441	CCKZER	NUM	0	Constant Pool
0442	CCKP01	NUM	1	Constant Pool
0443	CCKP02	NUM	2	Constant Pool
0444	CCKP03	NUM	3	Constant Pool
0445	CCKP04	NUM	4	Constant Pool
0446	CCKP05	NUM	5	Constant Pool
0447	CCKP06	NUM	6	Constant Pool
0450	CCKP07	NUM	7	Constant Pool
0451	CCKP08	NUM	8	Constant Pool
0452	CCKP09	NUM	9	Constant Pool
0453	CLFER	BSS	14	LVDC Error Table
0471	CLEPM	NUM	0	Test Program Parameter
		BSS	21	Input from T. P. C.
0517	CCEDS1	NUM	0	EDS Program (Cape) start
0520	CCEDS2	NUM	0	EDS Program (Cape) Reset
0521	CCFAZM	NUM	0	Platform Firing Azimuth
0522	CCAZCL	NUM	0	Platform
0523	CCTHE0	NUM	0	Platform Theadolite Azimuth
0524	CCFLAZ	NUM	0	Platform
0525	CCFINI	NUM	0	Platform
0526	CCPZCL	NUM	0	Platform
0527	CLTIME	NUM	0	LVDC Time Constant
0530		NUM	0	
0531	CLSIMF	NUM	0	LVDC Simulator Flight Flag
0532	CLVPWR	NUM	0	LVDC Power Flag
0533	CLVMEM	NUM	0	LVDC Memory Protect Flag
0534	CLVGC0	NUM	0	LVDC GCO Line Status
0535	CLVLDR	NUM	0	LVDC Special Loader Flag
0536	CLVMAP	NUM	0	LVDC Core Map Status
0537	DRMP	NUM	0	DDAS/RACS
0540	CSPTTG	NUM	0	Tower Test Table Indicator
0541	CLVTLC	NUM	0	Initial Verify Bypass Indicator
0542	CCAVTO	NUM	0	Auto Null Check-No Run Flag
0543		NUM	0	Permanent Storage for PTP4
0544		NUM		

APPENDIX D

ABBREVIATIONS AND ACRONYMS

APPENDIX D

GLOSSARY OF ABBREVIATIONS AND ACRONYMS

ABC	Airborne Computer
ACE	Automatic Checkout Equipment
ACS	Airborne Command Simulator
A/D	Analog To Digital
AGCSC	Automatic Ground Control System Computer
AMP	Accelerometer Monitor Program
APS	Auxiliary Propulsion System
ATOLL	Acceptance, Test, Or Launch Language
AVP	Address Verification Pulse
BCD	Binary Coded Decimal
BKUP	Backup
BO	Breakout
BPI	Bits Per Inch
BZA	Bank Zero Address
CALIPS	Calibrated Pressure Switches
CCIS	Control Computer Input Substitute
CCW	Counter Clockwise
CDC	Countdown Clock
CHAPS	Computer Hybrid Assembly Program
CIF	Central Instrumentation Facility
CIU	Computer Interface Unit
CM	Command Module
CMR	Command Module Repeater

C/O	Checkout
COD	Crossover Detector
CRG	Control Rate Gyro
CRP	Computer Reset Pulse
CRT	Cathode Ray Tube
CTL	Control
CW	Clockwise
D/A	Digital To Analog
DC	Display Console
DCC	Display Control Computer
DCE	Display Control Equipment (Brown Racks)
DCS	Digital Command System
DD	Display Description
DDAS	Digital Data Acquisition System
DEE	Discrete Events Evaluator
DI	Discrete In
DIN	LVDC Discrete In
DIT	Data Input Table
DMON	Display Monitor Program
DO	Discrete Out
DOM	Data Output Multiplexer
DOV	Discrete Out Vehicle
ECS	Environmental Control System
EDS	Emergency Detection System
EMR	Error Monitor Register

EMW	Error Message Word
ENBL	Enable
ESE	Electrical Support Equipment
EST	Eastern Standard Time
ETW	Error Time Word
G & C or G and C	Guidance and Control
GCO	Ground Checkout
GCP	Guidance Computer
GMT	Greenwich Mean Time
GND	Ground
GN2 or GN_2	Gaseous Nitrogen
GRR	Guidance Reference Release
GRRA	Guidance Reference Release Alert
GS	Ground Stage
GSE	Ground Support Equipment
HSM	High Speed Memory (RCA 110A Core Memory)
ICS	Inhibit Command System
ICSN	Inhibit Command System Not
IDB	Inertial Data Box
IND	Indication
INH or INHB	Inhibit

INI	Inhibit Interrupt
INT	LVDC Interrupt
I/O	Input/Output
IOA	Input/Output Address Line
IOAR	Input/Output Address Register
IOCS	Input/Output Control System
IODC	Input/Output Data Channel
IOR	Input/Output Register
IOS	Input/Output Sense Line
IU	Instrument Unit
IUED	Instrument Unit And Electrical Support Equipment
IVAR	Internal Variable
JR	Priority Request Register (Interrupt Requests)
JS	Priority Status Register
KSC	Kennedy Space Center
L	Left Accumulator
LCC	Launch Control Center
LCCC	Launch Control Center Computer
LDI	Discrete In To The Launch Control Center Computer
LDO	Discrete Out From The Launch Control Center Computer
LET	Launch Escape Tower
LIEF	Launch Information Exchange Facility
LH, LH2, or LH_2	Liquid Hydrogen
LO	Liftoff

LOX	Liquid Oxygen
LSB	Least Significant Bit
LSD	Least Significant Digit
LV	Launch Vehicle
LVDA	Launch Vehicle Data Adapter
LVDC	Launch Vehicle Digital Computer
MCD	Measurement Calibration Data
MDI	Discrete In To The Mobile Launcher Computer
MDO	Discrete Out From The Mobile Launcher Computer
MLC	Mobile Launcher Computer
MLF	Mobile Launch Facility
MLSR	Machine Language Subroutine
MLTP	Machine Language Test Program
MSB	Most Significant Bit
MSFC	Marshall Space Flight Center
MTOAD	More than One Address Detected
MTS	Magnetic Tape Station
N/C	Normally Closed
OAT	Overall Test
OCR	Orbital Checkout Ready
PAM	Pulse Amplitude Modulation
PARM	Parameter
PCM	Pulse Code Modulation
PGM	Program

PIT	Programmable Interval Timer
PS	Power Supply
PSC	Predefined Sequence Control
PTCS	Propellant Tanking Computer System
PU	Propellant Utilization
R	Right Accumulator
RACS	Remote Automatic Calibration System
RG	Rate Gyro
RM	Refresh Memory
RTNE	Routine
RSCR	Range Safety Command Receiver
RT	Real Time
RTIP	RACS Test In Progress
S/C	Spacecraft
SDF	Saturn Development Facility
SIMP	Simulated Plus Time
SLA	Service Lunar Adapter
SLAP	Symbolic Language Assembly Program
SLCC	Saturn Launch Computer Complex
SM	Service Module
SOAR	Saturn Oriented Assembly Routine
SPR	Spare
SS or SSw	Sense Switch

SSB	Single Sideband
SSF	Switch Selector Function
STAG	System Tape Arranging, Generating, And Editing Program
STAQ	Station Acquisition
SVDF	Saturn V Development Facility
SWSEL or SSEL	Switch Selector
SYL	Syllable
TAME	Tape Maintenance And Assembly System Program
TEMPO	Tape Maintenance Assembly Program
TLC	Total Loss Of Control
TM	Telemetry
TMR	Triple Modular Redundancy
TOK	Thrust OK
TP	Test Procedure
TPC	Test Program Control
VAB	Vertical Assembly Building
XVAR	External Variable

APPENDIX E

REFERENCES

A number of reference documents contain specific or detailed information supplementing this manual. These documents are:

- RCA 110A Programmer's Reference Manual
- SLAPS 2 Assembly Language Programmer Reference Manual
- Astrionics System Handbook
- Saturn V Operational Display System Software Program Specifications
- Specification for the Operating System for the Saturn Launch Computer Complex - Revision 2 MSFC-DRL-008A, Line Item No. 103; IBM No. 66-232-0001

APPENDIX F

DD TG CARD DATA CALCULATIONS

Mathematical Equations and Relationships Used to Calculate Unknowns for Analog and Discrete Signals

I. Hardwire Discrete - Any LDI or MDI from 0-3023.

1) Calculate number of shifts to left-justify data.

Number of shifts = 23-(R)

R = The remainder generated when the specified discrete number (in decimal) is divided by 24.

i.e. MDI 2872 - 119 + $\frac{16}{24}$

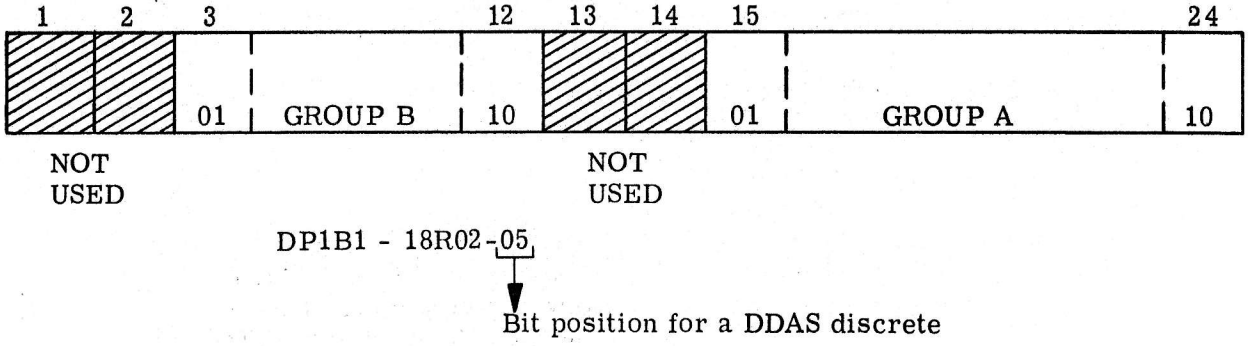
R = 16

Number of shifts = 23 - (16) = 7

2) Calculate number of bits of data: Predefined as follows. For a discrete, number of bits = 1

II. DDAS Discrete Signal

DP1B1-01R02-00 is an illegal DDAS discrete since a bit position of 00 is not allowed for DDAS discretes (13-14th characters). The 4th character determines the group (A or B) in the discrete DDAS data formatted as follows.



1) Calculate number of shifts to left-justify data:

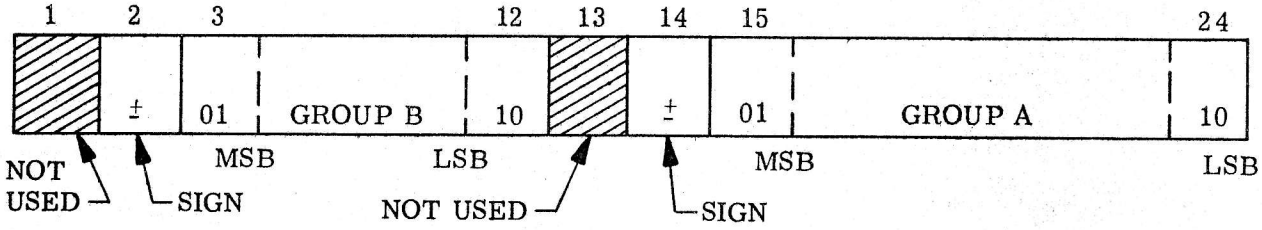
For Group A = (13 + bit position value)
i.e. DP1A1 - 18R02-05
Number of shifts for Group A = (13 + 5) = 18

For Group B = (1 + bit position value)
i.e. DP1B0-15R03-07
Number of shifts for Group B = (1 + 7) = 8

2) Calculate number of bits of data: Predefined - for a discrete, number of bits = 1.

III. DDAS Analog

DP1A1-05-01-02 is an illegal DDAS analog address since a bit position other than 00 is not allowed for a DDAS analog (13-14th character). The 4th character determines the group (A or B) in the DDAS analog data, which is formatted as follows.



1) Calculate number of shifts to left-justify data: Predefined as follows.

For Group A = Shift left 13 positions
For Group B = Shift left 1 position

2) Calculate number of bits of data = Predefined as follows.

Number of bits = 11

IV. Calculate Coefficients

$$A_0 = EB - I_p + L_s$$

$$A_1 = \frac{U_s - L_s}{975_{10}}$$

where: U_s = Upper Signal Range Value ± 8191.99
L_s = Lower Signal Range Value ± 8191.99
EB = Entered Bias
$I_p = 24_{10} \times A_1$

i.e. DP1A1-05-00

Upper Signal Range = 27.3
Lower Signal Range = 13.6
Entered Bias = 0 (for DDTG)

$$A_1 = \frac{27.3 - 13.7}{975} = \frac{13.6}{975} = \underline{0.014}$$

$$I_p - 24 \times 0.014 = 0.336$$

$$A = 0 - 0.336 + 13.7 = \underline{13.364}$$

Legal Values for DDAS Analog and Discrete Signals

Stage	Telemeter	Group	Major Time Slot		Channel	Multiplexer	Frame		Bit
A B C D E	P1	A or B	0 1 2 3	or V	01 to 30_{10}	R, G J, K L, H -, or U	00 to 10_{10}	-	00_{10} to 10_{10}
1	2-3	4	5	6	7-8	9	10-11	12	13-14

Character No. or Column

A = S-IC
B = S-II
C = S-IVB
D = Inst. Unit
E = ESE Monitor

C, D, F, G, J, K Discrete
M, N, R, S, T, U Y Signal
-, L, H = Analog Signal

The telemetry PCM channel assignment for a measurement is defined by an alphanumeric code which requires 14 columns. This code is defined as follows.

1. PCM/FM Telemetry Channel Assignments:

   a. Column 1

      Column 1 contains a letter that designates the stage of the vehicle from which the measurement is telemetered as follows:

      A S-IC
      B S-II
      C S-IVB
      D Instrument Unit
      E ESE

   b. Columns 2 and 3

      Columns 2 and 3 contain a letter and a number which serves to identify the RF data link as well as the type and sequential number of the primary telemeter which utilizes the link. The letter P1 designates a PCM/FM type telemeter.

c. Columns 4 and 5

Columns 4 and 5 contain a letter and number which serve the dual purpose of defining a multiplexer (physically) and a part of the PCM programming format (address).

(1) Physical Relationship

A multiplexer is an A0 multiplexer physically if it is defined as A0 in the PCM programming format in the normal flight mode. A given multiplexer can be A0, A1, A2, A3, B0, B2, or B3 depending on the way in which the multiplexers are programmed in the normal flight mode.

(2) PCM Programming Format

The letters A and B each define one-half of the PCM wavetrain (group). A and B operate at a basic rate of 3600 words/second and are interlaced for 7200 words/second in the PCM format. The numbers 0, 1, 2, and 3 are used with the letters A and B to define a part of the PCM programming format as follows.

A0	12	or	120 samples/second
B0	12	or	120 samples/second
A1	4	or	40 samples/second
A2	4	or	40 samples/second
A3	4	or	40 samples/second

d. Column 6

Column 6 contains a letter (V) or dash (-) to indicate whether or not the assigned measurement is at any time switched to another channel or address. A dash (-) indicates that no switching occurs; the letter V indicates that switching does occur. The remarks column and/or General Notes Section of the IP&CL defines the exact switching condition. These switching conditions include, but are not limited to: FM/FM to DDAS for prelaunch checkout, inflight switching including flight period switching, power transfers, and PCM mode shift.

e. Column 7 and 8

Column 7 and 8 contain the numbers 01 through 30 which identifies a main channel of a Model 270 multiplexer or the equivalent PCM address for digital data.

f. Column 9

Column 9 contains a letter or dash (-) to identify the type multiplexing or measurement routing as follows:

G - Discrete measurement routed direct to PCM/DDAS Model 301 Assembly.
J - Discrete measurement routed to Number 1, Model 410 multiplexer.
K - Discrete measurement routed to number 2, Model 410 multiplexer.
L - Analog measurement routed to a remote low-level sub-multiplexer.

R - Discrete measurement routed to a remote digital sub-multiplexer.
H - Analog measurement routed to a remote high-level sub-multiplexer.
Dash - Analog measurement routed directly to a Model 270 multiplexer.
*U - Discrete measurement routed directly to a Model 270 multiplexer.

g. Columns 10 and 11

Columns 10 and 11 contain the numbers 00 through 10 which identify Model 270 multiplexer sub-channels or the equivalent PCM address for digital data. The 00 indicates all ten sub-channels or frames are used.

h. Column 12

Column 12 contains a dash (-) for separation.

i. Columns 13 and 14

Columns 13 and 14 contain the numbers 00 through 10, which identify a particular digital bit. The 00 indicates all 10 bits are used.

NOTE

The numbers 01 through 10 correspond to the Most Significant Bit (MSB) (2^9) through the Least Significant Bit (2^0), i.e., MSB = Bit 01, LSB = Bit 10.

In this case, bit identification is contained in columns 13 and *14. This includes, but is not restricted to, measurements routed through the bilevel summing device.



Federal Systems Division
Electronics Systems Center
Huntsville, Alabama

Printed in USA



Apollo/Saturn Team