

**(Phase III)**

**Maintenance Operation Protocol  
Functional Specification  
(MOP)**

Order No. AA-K178A-TK  
Version 2.1.0

digital  
software



# DECnet DIGITAL Network Architecture (Phase III)

## Maintenance Operation Protocol Functional Specification (MOP)

Order No. AA-K178A-TK  
Version 2.1.0

October 1980

This document describes the Maintenance Operation Protocol (MOP) functions, interfaces, messages, and operation. MOP provides a set of messages and rules for loading and dumping computer memory and testing a data link. MOP operates within a data link control procedure that provides message framing and bit error detection. Although MOP is independent of any specific data link control procedure, within the DIGITAL Network Architecture, the Digital Data Communications Message Protocol (DDCMP) is used for that purpose.

This document provides technical information to assist those who implement the protocol. It also provides general information that will enable others to evaluate MOP. The document assumes a familiarity with computer communications, DDCMP, and DECnet. It is not intended to instruct those unfamiliar with these concepts.

To order additional copies of this document, contact your local  
Digital Equipment Corporation Sales Office.

digital equipment corporation · maynard, massachusetts

First Printing: October 1980

This material may be copied, in whole or in part, provided that the copyright notice below is included in each copy along with an acknowledgment that the copy describes the MOP protocol developed by Digital Equipment Corporation.

This material may be changed without notice by Digital Equipment Corporation, and Digital Equipment Corporation is not responsible for any errors which may appear herein.

Copyright © 1980 by Digital Equipment Corporation

The postage-prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DIGITAL	DECsystem-10	MASSBUS
DEC	DECtape	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EDUSYSTEM	PHA
UNIBUS	FLIP CHIP	RSTS
COMPUTER LABS	FOCAL	RSX
COMTEX	INDAC	TYPESET-8
DDT	LAB-8	TYPESET-11
DECCOMM	DECSYSTEM-20	TMS-11
ASSIST-11	RTS-8	ITPS-10
VAX	VMS	SBI
DECnet	IAS	PDT
DATATRIEVE	TRAX	

## CONTENTS

	Page
1.0 INTRODUCTION	1
2.0 FUNCTIONAL DESCRIPTION	3
2.1 Message Exchange	3
2.2 Storage Requirements	3
2.3 MOP Modes	4
2.4 Initiating Function Requests	4
2.5 Memory Loading	4
2.6 Memory Dumping	5
2.7 Link Testing	5
2.8 Unattended System Control	5
2.9 Link Control Procedure	6
3.0 INTERFACE	7
3.1 Commands to DDCMP	7
3.2 Responses from DDCMP	8
4.0 MESSAGES	9
4.1 Message Format Notation	9
4.2 MOP Message Formats	10
4.2.1 Memory Load with Transfer Address	10
4.2.2 Memory Load without Transfer Address	11
4.2.3 Request Memory Dump	12
4.2.4 Enter MOP Mode	12
4.2.5 Request Program	13
4.2.6 Request Memory Load	14
4.2.7 MOP Mode Running	14
4.2.8 Memory Dump Data	15
4.2.9 Reserved for Remote-11 Use	15
4.2.10 Reserved for Remote-11 Use	15
4.2.11 Parameter Load with Transfer Address	15
4.2.12 Loopback Test	17
4.2.13 Looped Data	17
5.0 OPERATION	18
5.1 MOP Primary and Secondary Mode Operation	19
5.2 Loading Computer Memory	20
5.3 Dumping Computer Memory	21
5.4 Link Testing	21
5.5 Unattended Control	22
6.0 MOP STATES, EVENTS, AND ACTIONS	23
APPENDIX A EXAMPLES	27
APPENDIX B DDCMP MAINTENANCE MODE FORMAT	28
APPENDIX C PROGRAM TYPES	29
APPENDIX D REVISION HISTORY	30

## TABLES

TABLE 1 Satellite State Table	24
TABLE 2 Host State Table	26



## 1.0 INTRODUCTION

This document describes the Maintenance Operation Protocol (MOP) functions, messages, and operation. MOP loads and dumps the memory of an adjacent remote computer system via a physical communications link, and tests that link. MOP allows control of unattended remote systems that are part of a DECnet network.

MOP allows selection and initiation of functions from either computer on the link. These computer systems are called the host and the satellite. The host implements the MOP functions; the satellite, the target of the host, receives the load, is the source of the dump, and loops the test messages back to the host. It is possible in a DECnet network for a host to get its load images from another node or for it to send the dump to another node for storage.

MOP is an optional component of the DIGITAL Network Architecture (DNA), the model upon which the DECnet products are built. DECnet is a family of hardware and software products that create distributed networks from DIGITAL computers and their interconnecting data links, permitting resource sharing and interprogram communication.

The DNA model comprises eight layers as follows, each with its distinct functions:

**The Physical Link Layer.** The Physical Link layer provides the hardware interfaces to specific system devices.

**The Data Link Layer.** The Data Link layer manages communications over a physical link, utilizing the Digital Data Communications Message Protocol (DDCMP).

**The Transport Layer.** The Transport layer routes messages between source and destination nodes.

**The Network Services Layer.** The Network Services layer manages logical data channels, using the Network Services Protocol (NSP).

**The Session Control Layer.** The Session Control layer provides the system-dependent aspects of logical link management, such as name-to-address mapping and access control.

**The Network Application Layer.** The Network Application layer provides I/O device and file access. It also supports other network functions used by the two higher layers.

**The Network Management Layer.** The Network Management layer provides user control over network parameters and counters. It also provides down-line loading, up-line dumping, and testing functions.

**The User Layer.** The User layer supports user services and programs.

MOP operates within the data field of a data link control procedure that provides message framing (locating the beginning and end of a message), and bit error detection. This is usually provided by DDCMP, but can be provided by hardware facilities in some implementations.

Related documents are as follows:

DNA Data Access Protocol (DAP) Functional Specification, Version 5.6.0, Order No. AA-K177A-TK

DNA Digital Data Communications Message Protocol (DDCMP) Functional Specification, Version 4.1.0, Order No. AA-K175A-TK

DNA Network Management Functional Specification, Version 2.0.0, Order No. AA-K181A-TK

DNA Network Services (NSP) Functional Specification, Version 3.2.0, Order No. AA-K176A-TK

DNA Session Control Functional Specification, Version 1.0.0, Order No. AA-K182A-TK

DNA Transport Functional Specification, Version 1.3.0, Order No. AA-K180A-TK

The following overview document for these specifications is an introduction to the structure and functions of DNA.

DNA General Description, Order No. AA-K179A-TK



## 2.0 FUNCTIONAL DESCRIPTION

MOP provides five functions:

1. Load memory. MOP provides the capability for down-line loading the memory of a computer system. The object of the load is the satellite system; the source of data is the host system.
2. Dump memory. MOP is used to dump the contents of the host system, usually upon a failure.
3. Test link. MOP tests the integrity of the data link and/or its hardware components (modems and interfaces). The host initiates the link test, and the satellite loops back the test data.
4. Force entry into the MOP mode. MOP is used to halt current satellite operation and put the satellite into MOP mode. This provides a means for the host to control an unattended satellite system.
5. Transfer control to programs resident in memory. MOP transfers control to programs regardless of whether they are loaded by MOP or by other means.

### 2.1 Message Exchange

Host and satellite systems exchange messages over a single data link in alternate half-duplex mode. Messages travel within the Digital Data Communications Message Protocol (DDCMP) message framing link control procedure envelope, which provides data integrity and error detection. Only messages with no bit errors pass to MOP. MOP messages handle message acknowledgments and retransmission.

### 2.2 Storage Requirements

Whereas hosts must implement all MOP functions, satellites implement only what they are capable of performing. Basic functions require very little code and may be coded into read-only memories resident in the satellite system. A primary mode enables the satellite with limited amounts of read-only-memory to bootstrap itself up to a full-function secondary mode.

In a DECnet network, a host may get its down-line load images from another node or send its dump to another node for storage via DECnet functions. Thus loading and dumping functions can extend to nodes other than those directly connected to the satellite.

### 2.3 MOP Modes

MOP operates in two modes as follows:

1. **Primary mode.** Primary mode is used only with satellites that contain limited amounts of read-only memory. A ROM containing only MOP primary code down-line loads the secondary code into the system, bootstrapping itself up to secondary mode.

The primary code sends a Request Program Message to the host and expects in return a single Program Load with Transfer Address Message containing the entire secondary code.

The secondary code is then loaded into memory and started.

2. **Secondary mode.** Most functions execute in secondary mode. A satellite may start in secondary mode either by using a large ROM containing the secondary code or by loading the secondary code via a local load device such as a cassette.

Some functions may require an even larger program than can be handled by the secondary code (which must be loaded in a single message). In these cases, the secondary level program is used to load the required program for the required functions (as is done for any load memory request).

### 2.4 Initiating Function Requests

Either the host or the satellite can initiate function requests. The satellite initiates requests by sending request messages to the host (for example, the Request Program message and Request Memory Load message). The host initiates requests, after the satellite has sent a MOP Mode Running Message, by sending an appropriate request command to the satellite (for example, the Memory Load message and the Request Memory Dump message). All message exchanges operate one message at a time alternately between host and satellite.

### 2.5 Memory Loading

Memory loading can be initiated in either of the following ways. The choice depends on the initiating computer and the previous function performed.

1. The satellite sends a Request Program message to the host; the host responds with a Memory Load message.

or

2. The host sends a Memory Load message to the satellite without having received such a request.

Each load is numbered. The satellite acknowledges each load with a Request Memory Load message requesting the next load. The final load is either a Memory Load with Transfer Address message or a Parameter Load with Transfer Address message. This transfers execution to the loaded program after loading any necessary running parameters.

After transfer, the new program may do one of the following:

1. Continue to run MOP.
2. Run another protocol or data link control procedure on the link.
3. Choose not to use the link at all and run stand alone.

Thus, MOP may be exited following a transfer to a loaded program.

## 2.6 Memory Dumping

The host always initiates memory dumping by sending Request Memory Dump message requests to the satellite. The satellite responds with Memory Dump Data messages containing the requested memory image. This process usually continues until the host has completed the dump, at which time it will usually start a memory load operation with a Memory Load message (described above).

## 2.7 Link Testing

The Loopback Test message tests the link. The active side sends this message out on the link, and waits for its return. The passive side returns a Looped Data message if the message is looped from a computer. If the message is looped from a loopback plug, modem facility, or other unintelligent loopback device, the passive side returns a Loopback Test message. By moving the loopback point and isolating components, the user can diagnose problems. Upon receipt of the returned message, the active side, knowing that it originally sent the message, does not return it. Use of the Looped Data message prevents a situation in which two computers passively loop to one another due to the timing of a test abort.

## 2.8 Unattended System Control

The Enter MOP Mode message, together with the appropriate hardware, enables an unattended satellite computer to halt current operation and begin either MOP primary or secondary mode operation. Control transfers to a resident MOP program or bootstrap. The hardware recognizes the Enter MOP Mode message and forces the computer system to transfer control to the MOP program. The password in this message protects the system from being controlled and loaded by an unauthorized host.

Section 5 describes the MOP operation including error recovery.

## 2.9 Link Control Procedure

MOP operates within the envelope of a data link control procedure that provides error-free transmission and reception on a data link connecting the host and satellite systems. This procedure provides the following functions:

1. **Message framing.** Message framing involves locating the beginning and end of a message at the receiving end of a link. This procedure involves bit, byte, and message synchronization. MOP only sends and receives messages that are multiples of 8-bit bytes.
2. **Bit error detecting.** Bit error detection is the detecting of one or more bit errors introduced by the communication medium. Messages must be completely received in memory by the link control procedure and checked for errors before they are passed to MOP. Only good messages (without bit errors) may be passed to MOP. To speed up error recovery by avoiding MOP waiting for timeouts, data bit errors can be flagged to MOP.

### NOTE

It is important to have a data link control receive buffer and to place it so that it will not interfere with MOP operation. If a receive buffer is not used, a MOP load message will be directly loaded into its specified address and then checked for bit errors. If the address is incorrect, the image will be loaded in the wrong place, destroying other information.

3. **Turning around half-duplex links, and selecting and addressing multipoint stations.** The data link control procedure provides any timers necessary for these operations. The timers are transparent to MOP. MOP transmit and receive commands (Section 3) trigger selection of a station and turning around of the link. MOP has no specific link control commands. However, MOP may optionally handle modem control signals directly, if they are passed through the MOP interface by the link control procedure.
4. **Error recording.** The link control procedure may record errors, but is not required to pass them to MOP.

The link control procedure typically has a special MOP mode that provides the above functions. There must be ways of entering and exiting this mode, and the mode can operate more simply than other link control modes at the cost of performance. The DDCMP Functional Specification provides a detailed description of how the Digital Data Communications Message Protocol provides these functions.

Within DECnet, MOP operates in the data link layer of the Digital Network Architecture (DNA). The data link control procedure provided in DNA is the Digital Data Communications Message Protocol (DDCMP).

### 3.0 INTERFACE

This section describes how MOP interfaces to the data link control procedure used for framing and error detection. It utilizes DDCMP as an example of such a data link procedure. References to DDCMP imply a reference to the actual data link procedure used.

The interface between MOP and DDCMP consists of a number of commands to DDCMP and responses from DDCMP used to transmit and receive MOP messages to and from the data link. The actual interface depends heavily on the features and capabilities within the operating systems running MOP and DDCMP. Mechanisms used for passing commands and receiving responses might include shared tables, calls with parameter lists, I/O registers, and/or interrupt mechanisms.

The information passed between MOP and DDCMP are messages. Its description usually consists of a starting buffer address and a length or byte count, or a chain of addresses and counts.

#### 3.1 Commands to DDCMP

##### ENTER MAINTENANCE MODE

This command initializes DDCMP into the maintenance mode before transmitting or receiving maintenance mode messages. This is the special mode used for MOP operation. If a MOP message is received while not in this mode the protocol will halt, inform the user that such a message was received, and allow the user to restart the protocol in this mode via the Enter Maintenance Mode command.

##### TRANSMIT MESSAGE

This command gives a message to DDCMP for transmission in the maintenance mode.

##### RECEIVE MESSAGE

This command gives an empty buffer to DDCMP for reception of the next maintenance message. Alternately, MOP might supply DDCMP with a pool of buffers and have the protocol select one. In this mode there will be a command to return empty buffers to the pool for reassignment by DDCMP.

##### HALT PROTOCOL

This command stops transmission and reception by DDCMP. Optionally, there may be a way to control the modem signals and force the modem to hang up in the dial case. The protocol may also halt by receiving a maintenance mode message while in normal mode or by receiving a normal mode message while in the maintenance mode. After halting, the protocol may be started again in the maintenance mode or in the normal mode.

### 3.2 Responses from DDCMP

#### MESSAGE TRANSMITTED

Response to the Transmit Message command. The message has been sent out on the link. The response is returned after actual transmission is completed and includes any delay due to selection, link turnaround, and transmission time.

#### MESSAGE RECEIVED

The next maintenance message has been received. Either MOP supplied a buffer in a Receive Message command, or a buffer will be taken from a pool previously supplied to DDCMP. In some implementations, if DDCMP has not been initialized into the maintenance mode and a maintenance mode message is received, there may be a separate response indicating that the protocol at the other end of the link is in maintenance mode. At that point, the protocol will halt, and MOP will have to initialize DDCMP into the maintenance mode prior to transmitting and receiving maintenance messages. In these implementations, the original received MOP message may be lost and not actually passed to MOP.

#### MESSAGE RECEIVED IN ERROR

An optional reply to MOP indicating that a maintenance message was received with a data CRC (block check) error. DDCMP was able to frame the message, but it had one or more bit errors in the data field. This response is useful in reducing the length of time MOP will wait for a reply to a previously sent message. Without this response MOP will wait for a MOP timeout interval. The length of this interval is implementation specific, but must include processing delays and transmission time. Typically, the value would be the same as the select or response timer value used by the data link control procedure.

#### START RECEIVED

While in maintenance mode, the remote side sent a normal DDCMP start message. When the local side is providing passive loopback service, this occurrence notifies it that the line is to be restarted in DDCMP normal mode.

## 4.0 MESSAGES

### 4.1 Message Format Notation

All MOP messages are sent embedded within a physical link control protocol. Within DECnet, the DDCMP maintenance mode envelope is generally used for this purpose. This section presents the general format of the MOP messages sent within the data field of that envelope. Appendix B specifies the DDCMP Maintenance Mode format.

The following notation is used to describe the MOP messages:

FIELD (LENGTH): CODING Description of field

Where:

FIELD Is the name of the field being described.  
LENGTH Is the length of the field expressed as one of the following:

1. The number of 8-bit bytes
2. The letters "I-n" meaning image field with n being a number that is the maximum length in 8-bit bytes of the image. The image is preceded by a 1-byte count of the length of the remainder of the field. Image fields are variable length and may be null (count=0). All 8 bits of each byte are used as information bits. If the length is "R" the field may extend the remainder of the message (maximum length is determined from maximum message length).

CODING Is the representation type used, as follows:

B Binary  
BM Bit map (each bit has independent meaning)  
A ASCII  
Null Interpretation depends on data representation

Notes:

1. All numeric values are shown in decimal representation unless otherwise noted.
2. All fields are transmitted low-order or least-significant bit first on the data links unless otherwise specified.
3. Bits in a MOP field are numbered from 0 to n where 0 is the low-order or least-significant bit.
4. Fields that refer to memory on specific computers are numbered according to the conventions on that computer system.
5. The same names used in fields in separate messages have the same meaning and format.

## 4.2 MOP Message Formats

The general format of MOP messages is:

CODE	OPERAND
------	---------

Where:

CODE Is the MOP message type code (1 byte).

OPERAND Is the operand information specific to each message type.

### 4.2.1 Memory Load with Transfer Address

This message causes the contents of the image data to be loaded (deposited) into memory at the load address and the system to be started at (the PC set to) the transfer address.

CODE	LOADNUM	LOADADDR	IMAGE DATA	TRANSFER	ADDR
------	---------	----------	------------	----------	------

Where:

CODE (1):B 0

LOADNUM (1):B Is the load number for multiple load images. This message may be preceded by Memory Load without Transfer Address messages. The load number starts at zero and is incremented for each load message sent in a loading sequence. A load number of zero is always valid and resets the expected load number. Zero should not be used for all load numbers in a sequence of load messages because it nullifies the sequence checking of the MOP protocol. LOADNUM is modulo 256. After load number 255 is load number 0.

LOADADDR (4):B Is the memory load address for storage of the data image.

IMAGE DATA Is the image to be stored into computer memory. The form sent will be machine dependent and will vary with the type and word length of the system:

PDP-11 Each byte represents 1 memory byte.

VAX-11/780 Each byte represents 1 memory byte.

PDP-8 Each 3 bytes represents 2 memory words:

Byte	Meaning
1	Low 8 bits of memory word 1 (4-11)
2	Low 8 bits of memory word 2 (4-11)
3	Low 4 bits of byte are high 4 bits of word 1 (0-3), high 4-bits of byte are high 4-bits of word 2 (0-3).



DECSYSTEM 10/20 Each 5 bytes represents one 36-bit word.

Byte	Meaning
1	Highest numbered (low-order) 8 bits of word (28-35)
2	Next 8 bits (20-27)
3	Next 8 bits (12-19)
4	Next 8 bits (4-11)
5	Low 4 bits of byte are highest order 4 bits of word (0-3); highest 4 bits of byte are unused and set to zero.

TRANSFER ADDR (4):B Is the starting address of the image just loaded.

NOTE

IMAGE DATA or LOADADDR and IMAGE DATA may be omitted. Valid MOP message lengths (DDCMP count values) will be 6 (LOADADDR and IMAGE DATA omitted), 10 (IMAGE DATA omitted), or greater than 10.

4.2.2 Memory Load without Transfer Address

This message causes the contents of the image data to be loaded (deposited) into memory at the load address.

CODE	LOADNUM	LOADADDR	IMAGE DATA
------	---------	----------	------------

Where:

CODE (1):B            2  
 LOADNUM              Is described in Section 4.2.1.  
 LOADADDR             Is described in Section 4.2.1.  
 IMAGE DATA          Is described in Section 4.2.1.

NOTE

IMAGE DATA may be omitted. Valid MOP message lengths may be 6 (IMAGE DATA omitted), or greater than 6. Messages without IMAGE DATA cause nothing to be loaded, however, the LOADNUM value is still incremented for the next load.

#### 4.2.3 Request Memory Dump

This message requests a dump (examining) of a portion of memory to be returned in a memory data dump message.

CODE	MEMADDR	NUMLOCS
------	---------	---------

Where:

CODE (1):B            4  
MEMADDR (4):B        Is the starting memory address for the dump.  
NUMLOCS (2):B        Is the number of locations to dump.

where:

PDP-11                1 location is 1 byte.  
VAX-11/780            1 location is 1 byte.  
PDP-8                 1 location is 1 word.  
DECSYSTEM-10/20      1 location is 1 word.

#### NOTE

This request results in a single Memory Dump Data message. A dump should not be requested for more data than can be reliably sent in a single reply on the link type used. The maximum link control procedure message length limits the maximum length for a given link.

#### 4.2.4 Enter MOP Mode

This message causes a system not in the MOP mode to enter the MOP mode if the password matches. This usually means transferring control on the satellite to a MOP program.

CODE	PASSWORD
------	----------

Where:

CODE(1):B            6  
PASSWORD (4):B        Is a password that must match before the receiving station can enter the MOP mode. If the password is sent over a DDCMP link in the maintenance mode, the link will enter the DDCMP maintenance mode (due to the maintenance mode envelope), but the node will only enter the MOP mode (for example, respond to load requests) if the PASSWORD matches.

#### 4.2.5 Request Program

This message requests a program to be sent in some number of memory load messages.

CODE	DEVTYPE	MOPVER	PGMTYPE	SOFTID
------	---------	--------	---------	--------

Where:

CODE (1): B            8

DEVTYPE (1): B        Is the device type at the requesting system. Used to cause the proper requested program to be loaded if it is device specific. Device types include:

Value	Meaning
0	DP (DP-DA SYNCHRONOUS LINE INTERFACE)
2	DU (DU-DA SYNCHRONOUS LINE INTERFACE)
4	DL (DL-C AND -E ASYNCHRONOUS SERIAL LINE INTERFACE)
6	DQ (DQ-DA SYNCHRONOUS SERIAL LINE INTERFACE)
8	DA (DA-B OR -AL UNIBUS LINKS)
10	DUP (DUP-DA SYNCHRONOUS LINE INTERFACE)
12	DMC (DMC-DA/AR, -MA/AL, -FA/AR INTERPROCESSOR LINKS)
14	DN (DN-BA OR -AA AUTOMATIC CALLING UNIT)
16	DLV (DLV-E ASYNCHRONOUS LINE INTERFACE)
18	DMP
20	DTE (DTE20-11)
22	DV (DV-AA/BA SYNCHRONOUS LINE MULTIPLEXER)
24	DZ (DZ-A OR -8 ASYNCHRONOUS SERIAL LINE INTERFACE)
28	KDP (KMC/DUP-DA SYNCHRONOUS LINE MULTIPLEXER)
30	KDZ (KMC/DZ-A ASYNCHRONOUS LINE MULTIPLEXER)
32	KL (KL8J)

MOPVER (1):B        Is the MOP format version number. For specifications version 1 and 2 the format version number is 1.

PGMTYPE (1):B        Is the generic type of program requested. Appendix C describes these types. The program type numbers are as follows:

Value	Meaning
0 (or omitted)	Secondary loader
1	Tertiary loader
2	Operating system

SOFTID (I-R): A        Identifies the software being requested. Specified as an image field of ASCII characters. This information is not needed if PGMTYPE is enough to specify the requested program. Also omitted if PGMTYPE is omitted.

#### 4.2.6 Request Memory Load

This message requests the next load in a loading sequence and provides error status on the previous load.

CODE	LOADNUM	ERROR
------	---------	-------

Where:

CODE (1): B        10

LOADNUM (1):B     Is the number of the load being requested.

ERROR (1): B      Is the error indicator for previous load numbered segment, as follows:

Value	Meaning
0 (or omitted)	No error.
1	Memory Load message image data not properly loaded (for example, because of a memory boundary problem).

#### 4.2.7 MOP Mode Running

This message indicates to a host that the system is in the MOP mode and supports the features indicated.

CODE	DEVTYPE	MOPVER	MEMSIZE	FEATURES
------	---------	--------	---------	----------

Where:

CODE (1):B        12

DEVTYPE            Is described in Section 4.2.5.

MOPVER             Is described in Section 4.2.5.

MEMSIZE (4):B     Is the size of physical machine memory in number of locations. Section 4.2.3 contains the location definition.

FEATURES (1):BM   Features available at this station according to the bit(s) set.

Bit	Feature
0	Multiblock load. Accepts the following messages: <ol style="list-style-type: none"> <li>1. Memory Load With Transfer Address message</li> <li>2. Memory Load Without Transfer Address message</li> <li>3. Parameter Load With Transfer Address message</li> </ol>
1	Dump. Accepts the Request Memory Dump message.
2	Loopback. Accepts the Loopback Test message.
3-7	Reserved.

#### 4.2.8 Memory Dump Data

The Memory Dump Data message, replying to a Request Memory Dump message, sends back the requested memory image.

CODE	MEMADDR	IMAGE DATA
------	---------	------------

Where:

CODE (1):B            14

MEMADDR (4):B        Is the starting address of the dump.

IMAGE DATA            Is the dump of memory in the same form as IMAGE DATA in Section 4.2.1.

#### 4.2.9 Reserved for Remote-11 Use

Remote-11, a networking product using RT-11 uses this message type.

CODE	MESSAGE
------	---------

Where:

CODE (1):B            16

MESSAGE                Refer to Remote-11 documentation.

#### 4.2.10 Reserved for Remote-11 Use

Remote-11, a networking product using RT-11 uses this message type.

CODE	MESSAGE
------	---------

Where:

CODE (1):B            18

MESSAGE                Refer to Remote-11 documentation.

#### 4.2.11 Parameter Load with Transfer Address

This message is used instead of Memory Load with Transfer Address (as the last load) to load system parameters before transferring control to the loaded program.

CODE	LOADNUM	PARAMETERS	TRANSFER ADDR
------	---------	------------	---------------

Where:

CODE (1):B            20

LOADNUM                Refer to Section 4.2.1.

PARAMETERS            ENTRY, ENTRY, ... ,ENDMARK

ENTRY                    PARTYPE,    PARLENGTH,    PARVALUE  
(ENTRY fields are optional)

E (1):B      Parameter type number  
 PARLENGTH (1):B      Number of bytes in PARVALUE  
 PARVALUE      Parameter value according to  
    PARTYPE and PARLENGTH, as follows:

PARTYPE	PARLENGTH	PARVALUE
1	1 to 6	ASCII node name target node is to use for itself.
2	1 to 2	Binary node number target node is to use for itself.
3	1 to 6	ASCII name of host assigned to node (for example, control host for task loading of core-only systems).
4	1 to 2	Binary node number of host assigned to node.

ENDMARK (1):B      0

TRANSFER ADDR      Refer to Section 4.2.1

#### NOTE

The parameter message, if used, is the last in a multiblock load. The minimum MOP message length is 7. The following techniques pass the parameters to the operating systems:

PDP-8      Currently unspecified.

PDP-11      Store the parameters at the top of physical memory in the locations listed below.

"Top of memory" refers to the first even address beyond available memory. Its value is the same as the value returned in the MEMSIZE field (Section 4.2.7).

Top of memory - 64. bytes =  
 a 2-byte node number. If none, then zeros.

Top of memory - 62. bytes =  
 a 6-byte node name padded with spaces. If none, then first 2 bytes are zeros.

Top of memory - 56. bytes =  
 a 6-byte host node name padded with spaces. If none, then first 2 bytes are zeros.

Top of memory - 50. bytes =  
 all bytes through top of memory reserved.

The registers are set as follows:

R0 Load device CSR address  
R1 -1  
R2 -1  
R3 Unit number of load device  
R4 2-byte ASCII load device

mnemonic:

XL DL11E, DLV11E  
XU DU11, DUV11  
XW DUPl1  
XM DMC11  
XB DAl1B  
XP DP11  
XQ DQ11

VAX-11/780 Currently unspecified.

DECSYSTEM-10/20 Currently unspecified.

#### 4.2.12 Loopback Test

The Loopback Test message tests a link by echoing the message sent. The active side sends the message; the passive side echoes the message. Upon receiving the echoed message, the original sender does not echo it again. (That would cause an infinite loop.) If the passive side is a computer, it changes the function code to Looped Data before echoing the message. This prevents an infinite loop where the active side aborts the test, receives an echoed Loopback Test message from a passive computer, and echoes the message. If the passive side is unintelligent, it echoes the Loopback Test Message unchanged. In this case, the aborted active side eventually receives a Looped Data Message and terminates the passive loopback.

CODE	LOOPDATA
------	----------

Where:

CODE (1):B 24

LOOP DATA Is the data to be looped back. This message is returned exactly as sent. The maximum length is limited by the maximum link control procedure message length for a given link.

#### 4.2.13 Looped Data

The Looped Data message is returned by an intelligent passive looper in response to a Loopback Test message.

CODE	LOOPDATA
------	----------

Where:

CODE (1):B 26

LOOPDATA Is the data received in the Loopback Test message.

## 5.0 OPERATION

MOP messages are sent within a link control procedure that provides the functions of message framing and bit error detection. MOP operates between two computer systems directly connected by a data link, which is dedicated to the MOP operation. On multipoint links, the subchannel connecting the two systems is dedicated to MOP operation. MOP messages are sent alternately between the computers. One system is designated the host and the other, the satellite. The host controls MOP operation and usually initiates the error recovery procedures. The host provides the data for the memory load, receives the data from the memory dump, and performs the link test function. The satellite is the object of the load, responds with data for the dump, and echoes messages for the link testing function. All message acknowledgment, timeout, and retransmission functions are handled at the MOP level via MOP messages and functions. The link control procedure only provides a bit error check. Section 3 describes the link control procedure functions.

The DNA-defined link control procedure is the DDCMP maintenance mode. The DDCMP Functional Specification contains a complete description of DDCMP.

It is possible to add to MOP the capability of obtaining load images from and of storing dump images at other nodes rather than using local host storage. Thus remote nodes not directly connected to the satellite can control loading and dumping functions. Regardless of where the host gets or sends the satellite data, the operation between the host and satellite is the same.

In general, the link control procedure should buffer incoming messages, check them for bit errors, and then pass them to MOP. The buffer area should be chosen so that it does not conflict with areas of memory requiring MOP access (for example, loading or dumping). Load images should not be loaded directly into the specified load address in load messages. Messages with bit errors might have load addresses in error, which would cause them to be loaded into the wrong area of memory, possibly destroying important information. The only exception to this restriction is the loading of the secondary loader, which is known in advance (by specification) to be loaded in a particular memory area.

MOP uses timeouts to recover from errors. That is, when a command is sent, a timer is started. If a response is not received within a timeout interval, the timer expires and error recovery procedures are initiated. The host system usually handles the timeout function. The exception to this is MOP primary mode initialization where the satellite is in the MOP mode and the host is not. In this case, timeouts and retransmission occur from the satellite. The link control procedure passes messages without bit errors to MOP and ignores all others. MOP's timeout mechanism recovers the messages that the link control procedure ignored due to bit errors. Optionally, the link control procedure may notify MOP that a message was received with one or more bit errors. In this case, MOP may initiate error recovery procedures prior to the expiration of the response timeout. This may improve the performance of MOP on links with high error rates. The error recovery procedures are the same, whether initiated in response to a timeout or notification from the link control procedure of message reception with bit errors.



## 5.1 MOP Primary and Secondary Mode Operation

Program loading, image dumping, and link testing usually occur in MOP secondary mode. If there is enough space within the satellite to contain the entire secondary program (either within ROM, or loaded from a local load device), the satellite starts in the MOP secondary mode. Otherwise, a minimum program, primary mode, bootstraps the satellite system to the secondary mode. The operation to achieve secondary mode is:

1. If the primary program is running, it sends a Request Program message to the host system requesting the secondary loader.
2. When the host receives this message, it sends the entire secondary program in the form of a single Memory Load with Transfer Address message.
3. If this is in error (no message received), the primary program sends another Request Program message after a timeout period.
4. Once the secondary loader is loaded and running, the secondary loader may send one of the following:
  - a. The Request Program message to request a special function program (for example, tertiary loader) or an operating system
  - b. The MOP Mode Running message to tell the host what functions are available and wait for the host to initiate action

The choice of message depends on the capabilities of the specific secondary program that was loaded. If it is an advanced loader program only (for example, tertiary loader) then the host sends a Request Program message. If the loaded program is a more general-purpose secondary program capable of multiple functions (for example, load, dump, test), then the host sends the MOP Mode Running message. This flexibility allows control requests to come from either the satellite or the host depending on the operating requirements.

If the secondary loader already exists in the satellite system, the dialogue starts at Step 4 above. The action taken in Step 4 determines whether (1) the satellite wants to initiate action (for example, load memory) or (2) the host selects the MOP function (the Satellite tells the host that the MOP mode is running).

There may be many secondary loaders with different capabilities. The specific network and host determine which is the appropriate one to use. Once the secondary loader is loaded, it may load other loaders using the loading memory function.

Each DECnet implementation requires a system-specific primary program. However, all primary programs must exchange the same protocol messages according to the same rules, even though the programs are different internally. The required operation of the primary program for each computer family is as follows:

PDP-8. Currently unspecified

PDP-11. The primary program sends a Request Program message for the secondary loader (PGMTYPE and SOFTID in message are omitted). It expects the secondary loader to be returned in a single Memory Load with Transfer Address message. In this message, the fields must have the following values:

LOADNUM = 0  
LOADADDR = 6  
TRANSFER ADDR = 6

The MOP message (from CODE to TRANSFER ADDR) is loaded at location 0. That is, the secondary loader (IMAGE DATA field) is loaded in a single message block, starting at location 6 in physical memory with a starting transfer address to location 6. Upon transfer, the PDP-11 registers have the following values:

R1 = load device CSR address

The primary loader starts a timer while waiting for the Memory Load message. If the secondary loader is received in error, if no loader is received or if any other message is received, the primary program times out and restarts, retransmitting the Request Program message again. If this cycle occurs some threshold number of times, the satellite assumes the host is not answering and may disconnect the link (for example, "hang up" a switched circuit).

VAX-11/780. Currently unspecified

DECsystem-10/20. Currently unspecified

## 5.2 Loading Computer Memory

Memory loading occurs in the MOP secondary mode. Either the satellite secondary program or the host may initiate loading as described in Step 4 of Section 5.1. In either case, loading occurs by the exchange of Memory Load messages from the host and Request Memory Load responses from the satellite. Loading starts with load number 0 and is incremented for each successive load. The host sends a Memory Load with a load number of 0. If the Memory Load message has no bit errors, it is loaded into memory, and a Request Memory Load for load number 1 is returned by the satellite. This both acknowledges load 0 and requests the next load (load 1). Any error occurring at the satellite in load 0 is reported in this message as well.

If the load message is in error, the satellite does nothing and waits for the host to timeout and retransmit the load. If the wrong load number is received, the satellite responds with a request for the expected load number. Alternately, if the satellite is confused (for example, wrong load received too many times) it may go back to its initial state and send either a MOP Mode Running message or a Request Program message back to the host. The host is then responsible for restarting the load from load 0 again. In general, the host is capable of more intelligence than the satellite, and thus the host has burden of the error recovery. If the host receives a message with bit errors, it should send its last message again, either upon a timeout or optionally, on notification of the error from the link control procedure.

The final load causes the secondary mode loader to transfer to the designated transfer address after, optionally, acknowledging that last load with a Request Memory Load message for the next expected load

(the load after the last one). The acknowledgment is omitted when it is known that the loaded program will itself be sending MOP messages as in the case of the loading of a higher level MOP function. If the program requested was a loader, the final load is acknowledged, since the loader sends a Request Program message when it starts, acknowledging its proper operation. The final load, including the transfer address, may be either a Program Load with Transfer Address message or a Parameter Load with Transfer Address message.

If the host receives a MOP Mode Running or Request Program message following the sending of the message with a transfer address, the received message should be examined to determine if it was sent by the program to which the host intended to transfer control or was sent by the MOP loader to which the transfer message was sent. This determination distinguishes between the initial load request of a new loader and the restarting of the current loader. This determination involves comparing the program request or features fields of the received message against the current loading parameters to determine if they came from the old or new program. If they came from the old program, then loading should start again from the beginning. If they came from the new program, then successful loading has occurred and new MOP operation may commence.

Loads that exceed memory limits are not loaded and cause the next load to be requested with an error code (ERROR field) included in that request. In some cases, the secondary loader requests an advanced loader (a tertiary loader), which then loads the operating system. One MOP program may load another via the load memory procedures and thus transfer from one MOP program to another in a series of MOP operations.

### 5.3 Dumping Computer Memory

The host system initiates the dumping function by sending Request Memory Dump messages to the satellite system. The satellite then responds with a Memory Dump Data message. If no reply is returned, the host times out and repeats the request. If the satellite receives a message with bit errors, it may respond in one of two ways:

1. Ignore them and let the host time out.
2. Return a MOP Mode Running message to cause retransmission of the request at the host prior to timer expiration.

Requests outside memory bounds result in transmission of a Memory Dump Data message with zeros for image data. Requests for more than can be sent in a single block result in transmission of the maximum that can be sent in a block. The remainder of the request is ignored.

### 5.4 Link Testing

The Loopback Test message tests the condition of a data link. The message is sent by the host, turned around, and returned on the data link. The actual looping back may be done by a program in the satellite or by a loopback box or switch placed somewhere between the host and satellite systems. With this technique, problems can be pinpointed by moving the loopback point and isolating components.

## 5.5 Unattended Control

The Enter MOP Mode message controls unattended systems. If sent to a running satellite that implements this message, it causes the satellite to transfer control to MOP primary or secondary mode programs in the satellite. If implemented in hardware, the hardware searches for the proper Enter MOP Mode message string. Upon a match, the hardware causes a transfer to the MOP program, usually residing in read only memories. This allows a system that is in a loop or that is halted to be remotely controlled and rebooted.

A password, part of the Enter MOP Mode message, avoids improper or accidental use. Only a message with a matching password is considered valid and processed. All others are ignored. Upon entering MOP mode, the primary or secondary program executes as described in Section 5.1.

## 6.0 MOP STATES, EVENTS, AND ACTIONS

This section presents the possible states, events, and actions of host and satellite systems running the MOP protocol. In general, events are receptions of MOP messages from the other system or timeouts. Actions are steps to be taken upon the occurrence of an event. The particular action to be taken depends on the state of the host or satellite. The tables in this section specify these relationships.

The possible satellite states are as follows:

1. non-MOP            The satellite is not in MOP mode.
2. PRIMARY MODE     The satellite is running the basic primary mode program and trying to load the secondary program from the Host.
3. SECONDARY MODE   The satellite is running the secondary mode program and waiting for a command from the Host.
4. LOADING MODE     The satellite is running in the secondary loading mode where it is requesting loads from the host and loading them into memory.

The satellite performs the following actions, classified according to state. The numbers preceding each action are used for reference in the satellite state table (Table 1) that follows this list of actions.

- 1.1 If primary code is available in the satellite, then send Request Program message for secondary code and enter Primary state. If secondary code is available, then send one of the following:
  - (a) The Request Program message for a tertiary loader or operating system (if secondary is only a loader) and enter Loader State.
  - (b) The MOP Mode Running message (if other functions are supported) and enter secondary state.
- 2.1 Valid Memory Load is per requirements of primary code for each computer family. If loaded then take action as for secondary code under 1.1.
- 2.2 Send Request Program message again and wait for secondary code.
- 2.3 Take action as in 2.2 until some threshold is exceeded, then reset, hanging up link if a dial-type connection.
- 3.1 If load number 0, then handle as if message had been received in Loading state.
- 3.2 Send Memory Dump Data response.
- 3.3 Change function code to Looped Data and echo message.
- 3.4 Either ignore or send messages as described for secondary under 1.1.
- 4.1 Valid Memory Load per requirements of computer family, LOADNUM field must be as expected (or zero). If valid then load into memory, send a Request Memory Load for received LOADNUM + 1.

- 4.2 Either send Request Memory Load for expected LOADNUM or ignore.
- 4.3 If Memory Load with Transfer, first load memory part. If Parameter Load with Transfer, set up parameter list. If the transfer is to an operating system (or other non-MOP program), send Request Memory Load for LOADNUM + 1 and then transfer to TRANSFER ADDR.
- 4.4 Either send Request Memory Load for expected LOADNUM or return to Secondary state and take action as under 1.1 for secondary code.

Table 1  
Satellite State Table

State	Event	New State	Action
non-MOP	Receive valid Enter MOP Mode	Primary, Secondary, or Loading	1.1
	User requests MOP mode	Primary or Secondary	1.1
Primary	Receive valid Memory load with Transfer address	Secondary or Loading	2.1
	Receive other message	Primary	2.2
	Timeout	Primary	2.3
Secondary	Receive Memory Load	Loading	3.1
	Receive Transfer	Non-Mop	3.1
	Receive Dump request	Secondary	3.2
	Receive Loopback Test Message	Secondary	3.3
	Receive other message	Secondary	3.4
Loading	Receive valid Memory Load	Loading	4.1
	Receive invalid Memory Load	Loading	4.2
	Receive Transfer	non-MOP	4.3
	Receive other message	Loading or Secondary	4.4

The host states are as follows:

non-MOP	The host is not in MOP mode with respect to the connected satellite.
LOADING MODE	The host is loading the memory of the satellite.
DUMPING MODE	The host is dumping the memory of the satellite.
LINK TEST MODE	The host is testing the data link.

The host performs the following actions, classified according to state. The numbers preceding the actions are used for reference in the host state table (Table 2) following this list.

- 1.1 Send the first block of the program requested, start timer, enter Loading state. If sending secondary code as a single message then do not start timer and stay in non-MOP state.
- 1.2 Determine function to be performed. If loading, take action as in 1.1; if dumping, send a Request Memory Dump request, start timer, and enter Dumping state; or if link testing, send a Loopback Test message, start timer, and enter Link test state.
  - 2.1 If request for either last load number sent or next load, send:
    - (a) Memory Load message either with or without transfer address or
    - (b) send Parameter Load with Transfer Address if last load and parameters are required. If request for another load, restart loading from load 0 again. Restart timer when sending response. If received Request Load is in response to transfer message (last load) then enter non-MOP state and do not start timer.
  - 2.2 If the MOP Mode Running Message indicates that the desired features are not present, restart loading from load 0. This could occur if the satellite received an invalid message and desired to reinitialize.

If the desired features are present, then the action described in 1.2 should be taken.
- 2.3 Send last load again, restart timer.
- 3.1 If more memory is required to be dumped then send another Request Memory Dump, start timer. If loading is to be started, then take action as described under 1.2. If link testing is to be started, then take action as described under 1.2.
- 3.2 Send Request Memory Dump again and restart timer.
- 4.1 If more link testing is to be done, send another Loopback Test message and start timer. For other functions, take action as described under 1.2.
- 4.2 Resend Loopback Test and restart timer.

Table 2  
Host State Table

State	Event	New State	Action
non-MOP	Receive valid Request Program	Loading	1.1
	Receive valid MOP Mode Running	Loading, dumping, or link test	1.2
Loading	Receive valid Request Memory Load	Loading or non-MOP	2.1
	Receive MOP Mode Running	Loading, Dumping or Link Test	2.2
	Timeout	Loading	2.3
	Receive valid Request Program	Loading	1.1
Dumping	Receive Memory Dump Data	Dumping, loading, or Link Test	3.1
	Timeout	Dumping	3.2
Link test	Receive Looptest	Link Test	4.1
	Receive Looped Data	Link Test	4.1
	Timeout	Link Test	4.2



**APPENDIX A**

**EXAMPLES**

A.1 Satellite running ROM bootstrap primary program wants to be loaded:

Events	
Step 1	Satellite primary program sends Request Program Message for secondary loader.
Step 2	Host sends Memory Load with Transfer Address of secondary program image.
Step 3	Satellite sends Request Program for operating system.
Step 4	Host sends Memory Load without Transfer Address load 0.
Step 5	Satellite sends Request Memory Load 1. Repeat Steps 4 and 5 for each successive load, eventually the last load (number n) is received with transfer address.
Step 6	Satellite sends Request Memory Load for load number n + 1 and transfers to program.
Notes:	<ol style="list-style-type: none"> <li>1. Last request is used only to ACK.</li> <li>2. Satellite may request tertiary loader (after Step 2) which would then request the operating system. In this case the last Request Memory Load Message is used and the ACK is omitted.</li> </ol>

A.2 Host wants to examine the memory of a satellite (secondary program running)

Events	
Step 1	Satellite sends MOP Mode Running with dump function supported.
Step 2	Host sends Request Memory Dump.
Step 3	Satellite sends Memory Dump Data.

APPENDIX B

DDCMP MAINTENANCE MODE FORMAT

In the DDCMP maintenance mode, the link is dedicated to performing MOP functions. The DDCMP message provides a CRC block check on the MOP data, but does not provide any acknowledgment of receipt or sequence check. The DDCMP maintenance mode envelope is similar in format to a DDCMP numbered data message. See the DDCMP specification for details on this format and operation. All numeric values are shown in decimal representation. All bytes are 8 bits long. DDCMP maintenance messages have the following format:

SYNSEQ	DLE	COUNT	Q	S	FILL	FILL	ADDR	CRC1	MAINTMSG	CRC2
--------	-----	-------	---	---	------	------	------	------	----------	------

Where:

- SYNSEQ      The DDCMP sync sequence:  
Synchronous links - 4 or more bytes; =150. (226 octal)  
Asynchronous links - none
- DLE          The maintenance message header start byte; =144. (220 octal)
- COUNT        The count field (14 bits); number of bytes in the MAINTMSG data field.
- Q            The QSYNC flag (1 bit); = 1.
- S            The SELECT flag (1 bit); = 1.
- FILL         A fill byte; = 0.
- FILL         A fill byte; = 0.
- ADDR         The tributary station address byte; for point-to-point operation = 1; for multipoint operation = 1-255.
- CRC1         The header CRC computed on DLE through ADDR (16-bit CRC-16).
- MAINTMSG     The maintenance message data field. Contains the MOP messages (COUNT bytes in length).
- CRC2         The data CRC computed on the MAINTMSG data field only (16-bit CRC-16).

The MOP messages are sent within the MAINTMSG field within the DDCMP maintenance mode envelopes.

APPENDIX C  
PROGRAM TYPES

In the Request Program Message subfield PGMTYPE (see Section 4.2.5) the generic type of program must be specified. This may be: 0 (or omitted) for the secondary loader; 1 for the tertiary loader; or 2 for the operating system. A brief description of the PDP-11 secondary and tertiary loaders is provided below.

Type 0 (secondary loader) - The secondary loader is sent in a single Memory Load with Transfer Address Message. The secondary loader must, therefore, fit into a single message block. It is loaded at location 6. Current secondary loaders are between 400 and 600 bytes in length, depending upon the device type used. They use the stack address set up by the primary loader. For current loaders this will be between 17400 and 17776. The secondary loader assigns its buffer space below the stack. The secondary loader accepts Memory Load with and without Transfer Address Messages. It is, therefore, capable of doing multi-block loads into absolute addresses without memory management. It requests a tertiary boot to be loaded.

Type 1 (tertiary loader) - The tertiary loader is loaded by the secondary in a multi-block load starting at location 10000. It will run with memory management on if it exists on the system. The tertiary loader moves itself to the top of physical memory and assigns its stack and buffer space just below itself. It is, therefore, capable of multi-block loads from location 0 up to its buffer address, usually the last 1-2K of physical memory. It requests the operating system to be loaded. The current tertiary loaders do not specify any specific operating system. The choice of system to send is established by prior agreement and by command at the host system.

APPENDIX D  
REVISION HISTORY

This Appendix provides a list of changes that have been made to MOP Version 1.1 (dated January, 1976) and to MOP Version 2.0 (dated March, 1978).

D.1 Changes made to MOP Version 1.1 updated it to Version 2.0. These changes:

1. Removed all references to MOP being used directly to non-adjacent systems over DECnet links. The NICE protocol performs MOP-like functions within DECnet, using actual MOP protocol only over a physical link.
2. Decoupled MOP from DDCMP maintenance mode. The protocol specifies the requirements of a link control procedure to be used with MOP. DDCMP maintenance mode is one such procedure.
3. Deleted the following messages not needed in MOP. These are now handled by NICE. Code 20 - Examine data by name, Code 22 - clear data by name, Code 26 - Examined data by name.
4. Clarified the description of the fields in all MOP messages. Clarified and expanded the operational details of MOP and added a state table for operation.
5. Added a detailed description of the requirements of the data link control procedure to be used by MOP and a detailed description of the interface, set of commands and responses, to that procedure.
6. Added VAX and DECSYSTEM 10/20 information in message formats where necessary.
7. Changed message 8 - (Request MOP secondary mode program) to Request Program. It is now used to request all program loads in MOP, not just the secondary program. The STADDR field is removed and replaced by a MOP version number field. Added DTE20 to DEVTYPE field. PGMTYPE field is changed and SOFTID is added.
8. Changed message 10 - Request memory load to remove NODE and SOFTID, function now part of message 8 described above. Added an ERROR field to return any errors on previous load.
9. Changed message 12 - to MOP mode running from secondary mode running. Removed STADDR and replaced with MOP version number. Added a FEATURES field to describe the MOP features a node supports.

10. Added a new message code 20 - Parameter load with transfer addr. This message is used to load a parameter block before transferring control to a just loaded program.
11. Added a detailed description of primary mode and the operation of loading the secondary program.

D.2 Changes made to MOP Version 2.0 updated it to Version 2.1. These changes:

1. Added Looped Data Message as response to a Loopback Test Message.
2. Added host node number parameter to Parameter Load with Transfer Address Message.
3. Added notification from DDCMP that a start was received while in maintenance mode.



READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

---

---

---

---

---

---

---

---

---

---

---

---

Did you find errors in this manual? If so, specify the error and the page number.

---

---

---

---

---

---

---

---

---

---

---

---

Please indicate the type of reader that you most nearly represent.

- Assembly language programmer
- Higher-level language programmer
- Occasional programmer (experienced)
- User with little programming experience
- Student programmer
- Other (please specify) \_\_\_\_\_

Name \_\_\_\_\_ Date \_\_\_\_\_

Organization \_\_\_\_\_

Street \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip Code \_\_\_\_\_  
or  
Country

Please cut along this line.

-- Do Not Tear - Fold Here and Tape --

**digital**



No Postage  
Necessary  
if Mailed in the  
United States



**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

**SOFTWARE DOCUMENTATION**  
146 MAIN STREET ML 5-5/E39  
MAYNARD, MASSACHUSETTS 01754

-- Do Not Tear - Fold Here and Tape --

Cut Along Dotted Line





\*\*\* 362 \*\*\*

AA-K178A-TK  
DIGITAL NETWORK ARCH III MOP  
DECUS PRICE - \$ 14.00