XXDP

USER MANUAL

CZQXAK0

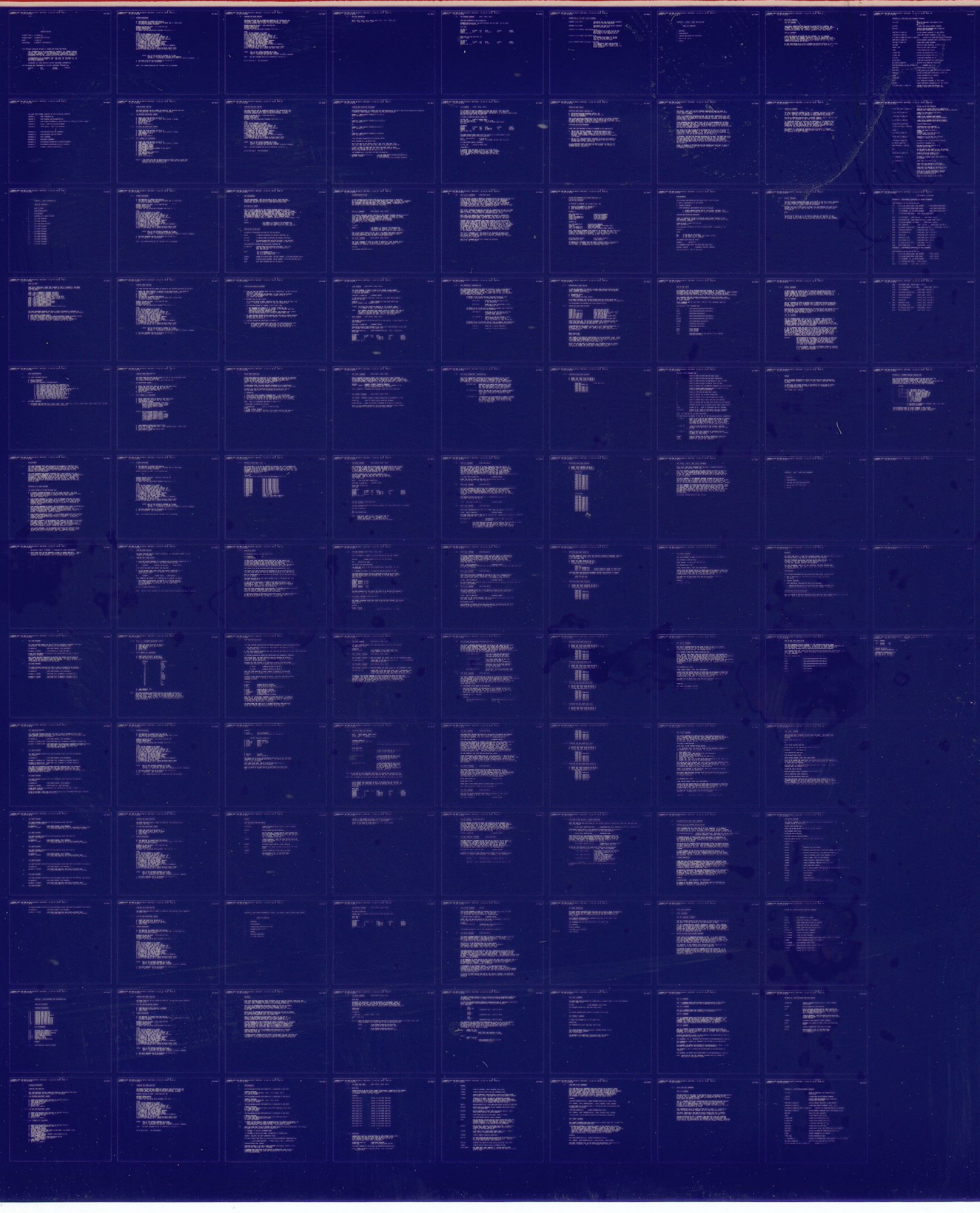
AH-9095K-MC

FICHE 1 OF 1

AUG 1978

digital

MADE IN USA



```
.REM    %
```

IDENTIFICATION

PRODUCT CODE: AC-9093K-MC

PRODUCT NAME: CZQXAKO XXDP USR MAN

DATE : 15 AUG 78

MAINTAINER: DIAGNOSTIC ENGINEERING

THIS MAINDEC REPLACES MAINDEC-11 DZQDO AND DZQDE AND DZQXA

THE XXDP USER MANUAL CONSISTS OF THE FOLLOWING CHAPTERS:

CHAPTER 1. XXDP INTRODUCTION

TABLE OF CONTENTS

1. WHAT IS XXDP

XXDP IS A "CATCH-ALL" NAME FOR A GROUP OF PDP-11 DIAGNOSTIC SOFTWARE PACKAGES AVAILABLE ON MULTIMEDIA, AND WHICH ARE PERIODICALLY UPDATED. XXDP INCLUDES:

TCDP - TC11 DIAGNOSTIC PACKAGE (DECTAPE).
RKDP - RK11 DIAGNOSTIC PACKAGE (DECPACK).
TMDP/THDP - TM11/TM02 DIAGNOSTIC PACKAGE
        9 TRACK CAN BE LOADED FROM TU10 OR TU16
TADP - TA11 DIAGNOSTIC PACKAGE (TM11 CASSETTES).
RXDP - RX11/RX01 DIAGNOSTIC PACKAGE (FLOPPY DISK).
RPDP - RP11 DIAGNOSTIC PACKAGE
RBDP - RH11/RP04 DIAGNOSTIC PACKAGE
RSDP - RH11/RS03 DIAGNOSTIC PACKAGE
RMDP - RK06 DIAGNOSTIC PACKAGE
RLDP - RL01 DIAGNOSTIC PACKAGE
RYDP - RX211/RX02 DIAGNOSTIC PACKAGE (FLOPPY DISK)

THE XXDP PACKAGES CONTAIN THE PDP-11 FAMILY DIAGNOSTIC PROGRAMS IN MEDIA OTHER THAN PAPER TAPE. XXDP PACKAGES HAVE THE FOLLOWING ADVANTAGES:

A. MORE COMPACT STORAGE MEDIA.
B. EASY AND CONVENIENT MEANS OF LOADING PROGRAMS UNDER KEYBOARD CONTROL
C. MEANS ARE PROVIDED FOR UPDATING AND MODIFYING PROGRAMS.
D. POSSIBLE TO SEQUENTIALLY RUN A SERIES OF PROGRAMS THROUGH USE OF THE "CHAIN MODE" FEATURE. (PROGRAMS MUST BE CHAINABLE).

2 XXDP REQUIREMENTS

2.1 ALL XXDP PACKAGES REQUIRE:

A. PDP-11 PROCESSOR
B. CONSOLE DEVICE
C. ONE OF THE DIAGNOSTIC PACKAGE MEDIA:

1. TC11 DECTAPE CONTROL AND TU56 TRANSPORT OR,
2. RK11 DISK CONTROL AND RK03 OR RK05 DRIVE OR,
3. TA11 CONTROL AND TU60 CASSETTE DRIVE OR,
4. RX11/RXV11 FLOPPY CONTROL UNIT AND RX01 FLOPPY DRIVE OR,
5. TM11 MAGTAPE CONTROL AND TU10 MAGTAPE DRIVE OR,
6. TM02 MAGTAPE CONTROL UNIT AND TU16 DRIVE OR,
7. RP11 DISK CONTROLLER AND RP03 DRIVE OR,
8. RH11/RP04 DISK CONTROLLER AND RP04 DRIVE OR,
9. RH11/RS03 DISK CONTROLLER AND RS03 DRIVE.
10. RK611 DISK CONTROLLER AND RK06 DRIVE.
11. RL11/RLV11 DISK CONTROLLER AND RL01 DRIVE.
12. RX211/RX02 FLOPPY CONTROL UNIT AND DRIVE

2.2 OPTIONAL HARDWARE:

A. BOOTSTRAP ROM FOR THE TC11, RK11, TA11, TM11, TM02, RX11, RXV11, RP03, RP04, RS03, RK06, RL01, OR RX0
   IT MAKES LOADING THE XXDP MONITOR MORE CONVENIENT.

3. DISCLAIMERS

3.1 THE XXDP PACKAGES HAVE BEEN DESIGNED FOR DIAGNOSTIC PURPOSES ONLY. THE XXDP SOFTWARE IS NOT INTENDED TO BE COMPATIBLE WITH ANY OTHER PDP-11 FAMILY SOFTWARE. ANY NON-DIAGNOSTIC USES OF THE SOFTWARE, OR USES OF THE SOFTWARE IN OTHER THAN THE MANNER DESCRIBED IN THIS DOCUMENT ARE NOT SUPPORTED.

3.2 THE XXDP PACKAGES ARE BINARY PACKAGES ONLY. THEY PROVIDE THE PDP-11 FAMILY DIAGNOSTIC PROGRAMS IN THE VARIOUS MEDIA DESCRIBED. DOCUMENTATION FOR EACH OF THE PROGRAMS STORED IN A XXDP PACKAGE MUST BE OBTAINED SEPARATELY, FROM SOFTWARE DISTRIBUTION CENTER (SDC). HOWEVER, THIS DOCUMENTATION MUST BE OBTAINED AT THE SAME TIME AS THE PACKAGE, IN ORDER TO INSURE THAT THE DOCUMENTS AND THE PROGRAMS ARE AT THE SAME REVISION LEVEL.

4. CONTENTS OF A XXDP PACKAGE

THE BASIC PARTS OF A XXDP PACKAGE ARE:

A. A CONTROL PROGRAM REFERRED TO AS THE "XXDP MONITOR". THE XXDP MONITOR PROVIDES THE MEANS TO LOAD PROGRAMS UNDER KEYBOARD CONTROL, AND TO OBTAIN A DIRECTORY OF CONTENTS OF THE XXDP MEDIUM (DECTAPE, MAGTAPE, ETC).

B. XXDP UPDATE PROGRAM #1 (UPD1). THIS 4K PROGRAM PROVIDES THE BASIC MEANS FOR MODIFYING AND UPDATING THE PROGRAMS IN THE XXDP PACKAGE. HANDLES DEVICES LISTED ABOVE (SECTION 2.1C) TO AND INCLUDING RX11 BECAUSE THIS PROGRAM RELOCATES ITSELF IN MEMORY, IT REQUIRES AT LEAST 8K MEMORY.

C. XXDP UPDATE PROGRAM #2 (UPD2). AN 8K PROGRAM WITH A MORE COMPREHENSIVE SET OF COMMANDS THAT PROVIDE MORE CONVENIENCE AND EASE OF UPDATING THE XXDP PACKAGE. HANDLES DEVICES USED ABOVE (SECTION 2.1C) TO AND INCLUDING RK06. BECAUSE THIS PROGRAM RELOCATES ITSELF IN MEMORY, IT REQUIRES AT LEAST 16K MEMORY.

D. XXDP UPDATE PROGRAM #3 (UPD3). A 9K PROGRAM CONTAINING THE FEATURES FOUND IN UPD2 BUT HANDLES ALL DEVICES LISTED ABOVE IN SECTION 2.1C. BECAUSE THIS PROGRAM RELOCATES ITSELF IN MEMORY, IT REQUIRES AT LEAST 20K MEMORY.

E. XXDP COPY1 PROGRAM, AN 8K PROGRAM THAT ENABLES THE USER TO DUPLICATE THE XXDP MEDIUM. THIS PROGRAM ONLY COPIES XXDP SOFTWARE. IT IS NOT A GENERAL PURPOSE COPY UTILITY PROGRAM. HANDLES DEVICES LISTED ABOVE (SECTION 2.1C) TO AND INCLUDING TM02. THIS PROGRAM REQUIRES AT LEAST 8K MEMORY TO RUN; IT DOES NOT RELOCATE ITSELF.

F. XXDP COPY2 PROGRAM. AN 8K PROGRAM CONTAINING THE FEATURES FOUND IN COPY1 BUT SERVICES DIFFERENT DEVICES. HANDLES DEVICES LISTED ABOVE (SECTION 2.1C) FROM RP11 DOWN. BECAUSE THIS PROGRAM

RELOCATES ITSELF IN MEMORY, IT REQUIRES AT LEAST 16K MEMORY.

G. XTECO XXDP TEXT EDITOR PROGRAM IS USED TO CREATE AND EDIT ASCII TEXT FILES FOR USE IN XXDP, SUCH AS BATCH CONTROL FILES FOR UPD2 PROGRAM OR CHAIN SEQUENCE FILES TO BE RUN BY XXDP MONITOR.

5. THE TCDP PACKAGE

THE TCDP PACKAGE MAKES THE PDP-11 FAMILY DIAGNOSTIC PROGRAMS AVAILABLE ON DECTAPES. THE PACKAGE CONSISTS OF THE FOLLOWING ITEMS:

AC-9093K-MC	XXDP USER MANUAL (THIS DOCUMENT).
MAINDEC-11-DZZFA	TCDP DECTAPE #1. XXDP SOFTWARE.

OTHER TCDP DECTAPES (IN EXCESS OF 20) CONTAINING THE PDP-11 FAMILY DIAGNOSTIC PROGRAMS.

ONLY THOSE DECTAPES REQUIRED TO SUPPORT THE TARGET SYSTEM NEED BE ORDERED. DECTAPE #1 SHOULD ALWAYS BE ORDERED. THE PDP-11 MAINDEC INDEX LISTS THE CONTENTS OF EACH TCDP DECTAPE. IT SHOULD BE REFERENCED TO DETERMINE THE DECTAPES THAT ARE NEEDED.

6. THE RKDP PACKAGE

THE RKDP PACKAGE PROVIDES THE PDP-11 FAMILY DIAGNOSTICS ON DECPACK. IT CONSISTS OF THE FOLLOWING ITEMS THAT MUST BE ORDERED INDIVIDUALLY:

AC-9093K-MC	XXDP USER MANUAL (THIS DOCUMENT).
MAINDEC-11-DZZAA	XXDP-RKDP RK11 DIAGNOSTIC PACKAGE DISK 1
MAINDEC-11-DZZZB	XXDP-RKDP RK11 DIAGNOSTIC PACKAGE DISK 2
MAINDEC-11-DZZZC	XXDP-RKDP RK11 DIAGNOSTIC PACKAGE DISK 2

7. THE TMDP/THDP PACKAGE

THE TMDP/THDP PACKAGE PROVIDES THE PDP-11 FAMILY DIAGNOSTICS ON 7 OR 9 TRACK MAGTAPE (TU10/TU16/TS03). THE PACKAGE CONSISTS OF THE FOLLOWING ITEMS THAT MUST BE ORDERED INDIVIDUALLY:

AC-9093K-MC	XXDP USER MANUAL (THIS DOCUMENT).
MAINDEC-11-DZZAC-A-MB7	XXDP-TMDP/THDP TM11 DIAGNOSTIC PACKAGE (7 TRACK) OR,
MAINDEC-11-DZZAC-B-MB9	XXDP-TMDP/THDP TM11/TM02/TS03 DIAGNOSTIC PACKAGE (9 TRACK). THE TMDP/THDP 9 TRACK PACKAGE CONTAINS THE THDP AND TMDP MONITORS WHICH ENABLE THE SAME TAPE TO BE USED FOR EACH DRIVE.

8. THE TADP PACKAGE

THE TADP PACKAGE CONSISTS OF THE FOLLOWING ITEMS THAT MUST BE ORDERED INDIVIDUALLY:

AC-9093K-MC	XXDP USER MANUAL (THIS DOCUMENT).
MAINDEC-11-DZZDH-A-TB	XXDP-TADP TA11 DIAGNOSTIC PACKAGE CASSETTE.
MAINDEC-11-DZZTV-A-TB	XXDP-TADP TA11 DIAGNOSTIC PACKAGE CASSETTE CONTAINING COPY2 PLUS UPD3.

THE TADP TA11 DIAGNOSTIC PACKAGE CASSETTE CONTAINS ONLY THE PROGRAMS REQUIRED TO PROVIDE LOADING, COPYING, AND UPDATING FACILITIES. THE DIAGNOSTIC PROGRAMS ARE STORED IN STANDARD TA11 CASSETTES THAT MUST BE OBTAINED SEPARATELY. REFER TO MAINDEC INDEX FOR A LIST OF AVAILABLE TA11 CASSETTES AND THEIR CONTENTS.

9. THE RXDP PACKAGE

THE RXDP PACKAGE CONSISTS OF THE FOLLOWING ITEMS THAT MUST BE ORDERED INDIVIDUALLY:

AC-9093K-MC	XXDP USERS MANUAL (THIS MANUAL)
MAINDEC-11-DZZGA	RXDP FLOPPY #1. XXDP SOFTWARE.

OTHER RXDP FLOPPIES (IN EXCESS OF 20) CONTAINING THE PDP-11 FAMILY DIAGNOSTIC PROGRAMS.

REFER TO MAINDEC INDEX FOR UP-TO-DATE LIST OF RX11 DISKETTES AND THEIR CONTENTS, OR REFER TO PDP-11 SOFTWARE PRICE LIST.

10. THE RPDP PACKAGE

THE RPDP PACKAGE CONSISTS OF THE FOLLOWING ITEMS THAT MUST BE ORDERED INDIVIDUALLY:

AC-9093K-MC	XXDP USERS MANUAL (THIS MANUAL),
MAINDEC-11-DZQUN	RPDP-XXDP RP11/RP02/RP03 MONITOR. AVAILABLE ON PAPER TAPE, IT IS ALSO AVAILABLE AS A FILE IN OTHER XXDP PACKAGES.

11. THE RBDP PACKAGE

THE RPDP PACKAGE CONSISTS OF THE FOLLOWING ITEMS THAT MUST BE ORDERED INDIVIDUALLY:

AC-9093K-MC	XXDP USERS MANUAL (THIS MANUAL),
MAINDEC-11-DZQUO	RBDP-XXDP RP04 MONITOR, AVAILABLE ON PAPER TAPE, IT IS ALSO AVAILABLE AS A FILE ON OTHER XXDP PACKAGES.

12. THE RSDP PACKAGE

THE RSDP PACKAGE CONSISTS OF THE FOLLOWING ITEMS THAT MUST BE ORDERED INDIVIDUALLY:

AC-9093K-MC	XXDP USERS MANUAL (THIS MANUAL),
MAINDEC-11-DZQUP	RSDP-XXDP RS04 MONITOR, AVAILABLE ON PAPER TAPE, IT IS ALSO AVAILABLE AS A FILE ON OTHER XXDP PACKAGES.

13. THE RMDP PACKAGE

THE RMDP PACKAGE CONSISTS OF THE FOLLOWING ITEMS THAT MUST BE ORDERED INDIVIDUALLY:

AC-9093K-MC	XXDP USER MANUAL (THIS MANUAL).
MAINDEC-11-DZZRA	XXDP RMDP RK06 MONITOR, AVAILABLE ON PAPER TAPE. IT IS ALSO AVAILABLE AS A FILE ON OTHER XXDP PACKAGES.

14. THE RLDP PACKAGE

THE RLDP PACKAGE CONSISTS OF THE FOLLOWING ITEMS THAT MUST BE ORDERED INDIVIDUALLY:

AC-9093K-MC	XXDP USER MANUAL (THIS MANUAL),
MAINDEC-11-DZQUZ	XXDP RLDP RL01 MONITOR, AVAILABLE ON PAPER TAPE. IT IS ALSO AVAILABLE AS A FILE ON OTHER XXDP PACKAGES.

15. THE RYDP PACKAGE

THE RYDP PACKAGE CONSISTS OF THE FOLLOWING ITEMS THAT MUST BE ORDERED INDIVIDUALLY.

AC-9093K-MC	XXDP USER MANUAL (THIS MANUAL)
MAINDEC-11-CZQU5	XXDP RYDP RX02 MONITOR, AVAILABLE ON PAPER TAPE. IT IS ALSO AVAILABLE AS A FILE ON OTHER XXDP PACKAGES.

CHAPTER 2. XXDP GENERAL USE DOCUMENTATION

TABLE OF CONTENTS

1. LOADING PROCEDURES

1.1 LOADING TCDP MONITOR

THE TCDP MONITOR CAN BE LOADED BY MEANS OF THE BM792YB ROM BOOT, MR11-DB ROM BOOT, OR VIA A "TOGGLE-IN" PROCEDURE.

1.1.1 VIA BM792YB BOOTSTRAP LOADER

A. MOUNT THE DESIRED TCDP DECTAPE ON DECTAPE DRIVE 0.
B. MAKE DRIVE READY AND WRITE LOCK IT.
C. LOAD ADDRESS 173100
D. SET SR TO 177344
E. PRESS START
F. GO TO 1.1.4 STEP A.

1.1.2 VIA MR11-DB BOOTSTRAP LOADER

A. MOUNT THE DESIRED TCDP DECTAPE ON DFCTAPE DRIVE 0.
B. MAKE DRIVE READY AND WRITE LOCK IT.
C. LOAD ADDRESS 173120
D. PRESS START
E. GO TO 1.1.4 STEP A.

1.1.3 VIA "TOGGLE-IN" PROCEDURE

A. MOUNT THE DESIRED TCDP DECTAPE ON DECTAPE DRIVE 0.
B. MAKE DRIVE READY AND WRITE LOCK IT.
C. LOAD ADDRESS 177342
D. DEPOSIT VALUE 004003
E. DECTAPE WILL REWIND AND STOP IN END ZONE. THE REMOTE LIGHT ON DRIVE SHOULD REMAIN LIT.
F. PRESS EXAMINE KEY.
G. DEPOSIT VALUE 000001. REMOTE LIGHT SHOULD GO OUT.
H. LOAD ADDRESS 000216
I. DEPOSIT SEQUENTIALLY THE FOLLOWING VALUES:
   012737,000005,177342,000777
J. LOAD ADDRESS 000216
K. PRESS START
L. GO TO 1.1.4 STEP A.

1.1.4 COMMON PROCEDURE

A. THE MONITOR IS LOADED FROM MEDIUM
B. THE MONITOR TYPES THE FOLLOWING MESSAGE AND IS THEN READY
   TO ACCEPT KEYBOARD COMMANDS.

```
DZQUC-E 21-JUL-76 TCDP - TC11 MONITOR NNK
RESTART ADDR: XXXXXX
BOOTED VIA UNIT#: 0
TO ABORT THE FOLLOWING HELP MESSAGE TYPE CTRL C ( C)

TYPE:
F<CR> TO SET CONSOLE FILL COUNT
D<CR> FOR DIRECTORY ON CONSOLE, OR
D/F<CR> FOR SHORT DIRECTORY ON CONSOLE, OR
D/L<CR> FOR DIRECTORY ON LINE PRINTER, OR
D/L/F<CR> FOR SHORT DIRECTORY ON LINE PRINTER,
R COPY1<CR> TO RUN COPY1 PROGRAM
R FILENAME<CR> TO RUN ANY OTHER  PROGRAM
L FILENAME<CR> TO LOAD A PROGRAM ONLY
S<CR> TO START THE PROGRAM JUST LOADED,
S ADDR<CR> TO START THE PROGRAM AT SPECIFIC ADDRESS
C FILENAME<CR> TO RUN A CHAIN,
C FILENAME/QV<CR> TO RUN A CHAIN IN QUICK VERIFY MODE.
REFER TO XXDP MANUAL MD-11-DZQXA FOR ADDITIONAL HELP.
.
```

WHERE: NNK IS THE SYSTEM'S STORAGE UP TO 28K,
XXXXXX IS THE MONITOR'S RESTART ADDR: ADDRESS.
THE DOT (.) INDICATES THE MONITOR IS READY TO ACCEPT COMMANDS.

C. THE HELP FILE MAY BE ELIMINATED BY TYPING CTL C.
D. GO TO SECTION 2. USE PROCEDURES.

NOTE: <CR> MEANS PRESSING THE "RETURN" KEY ON KEYBOARD

1.2 LOADING RKDP MONITOR

THE RKDP MONITOR CAN BE LOADED BY MEANS OF THE BM792YB ROM BOOT, MR11-DB ROM BOOT, OR VIA A "TOGGLE-IN" PROCEDURE.

1.2.1 VIA BM792YB BOOTSTRAP LOADER

A. MOUNT THE RKDP DECPACK ON DRIVE 0.
B. LOAD DRIVE. WRITE LOCK IT. WAIT UNTIL DRIVE IS READY.
C. LOAD ADDRESS 173100
D. SET SR TO 177406
E. PRESS START
F. GO TO 1.2.4 STEP A

1.2.2 VIA MR11-DB BOOTSTRAP LOADER

A. MOUNT THE RKDP DECPACK ON DRIVE 0.
B. LOAD DRIVE. WRITE LOCK IT. WAIT UNTIL DRIVE IS READY.
C. LOAD ADDRESS 173110
D. PRESS START
E. GO TO 1.2.4 STEP A

1.2.3 VIA "TOGGLE-IN" PROCEDURE

A. MOUNT THE RKDP DECPACK ON DRIVE 0.
B. LOAD DRIVE. WRITE LOCK IT. WAIT UNTIL DRIVE IS READY.
C. LOAD ADDRESS 177404
D. DEPOSIT VALUE 000001
E. LOAD ADDRESS 010000
F. DEPOSIT VALUES 012737,000005,177404,000777
G. LOAD ADDRESS 010000
H. PRESS START
I. WAIT ONE SECOND. PRESS HALT.
J. LOAD ADDRESS 000000
K. PRESS START
L. GO TO 1.2.4 STEP A

NOTE!!!! THE RKDP DISK MAY BE BOOTED AND RUN FROM A DRIVE OTHER THAN DRIVE 0 (ANY DRIVE BETWEEN 0 AND 7), PROVIDED THE ROM USED SUPPORTS MULTIPLE DRIVE BOOTING.

1.2.4 COMMON PROCEDURE

A. THE MONITOR IS LOADED FROM MEDIUM
B. THE MONITOR TYPES THE FOLLOWING MESSAGE AND IS THEN READY
   TO ACCEPT KEYBOARD COMMANDS.

```
DZQUD-F 22-OCT-77 RKDP - RK11 MONITOR NNK
RESTART ADDR: XXXXXX
BOOTED VIA UNIT#: 0
TO ABORT THE FOLLOWING HELP MESSAGE TYPE CTRL C ( C)

TYPE:
F<CR> TO SET CONSOLE FILL COUNT
D<CR> FOR DIRECTORY ON CONSOLE, OR
D/F<CR> FOR SHORT DIRECTORY ON CONSOLE, OR
D/L<CR> FOR DIRECTORY ON LINE PRINTER, OR
D/L/F<CR> FOR SHORT DIRECTORY ON LINE PRINTER,
R COPY1<CR> TO RUN COPY1 PROGRAM,
R FILENAME<CR> TO RUN ANY OTHER  PROGRAM
L FILENAME<CR> TO LOAD A PROGRAM ONLY
S<CR> TO START THE PROGRAM JUST LOADED,
S ADDR<CR> TO START THE PROGRAM AT SPECIFIC ADDRESS
C FILENAME<CR> TO RUN A CHAIN,
C FILENAME/QV<CR> TO RUN A CHAIN IN QUICK VERIFY MODE.
REFER TO XXDP MANUAL MD-11-DZQXA FOR ADDITIONAL HELP.
.

        WHERE: NNK IS THE SYSTEM'S STORAGE UP TO 28K,
               XXXXXX IS THE MONITOR'S RESTART ADDRESS.
               THE DOT (.) INDICATES THE MONITOR IS READY TO ACCEPT COMMANDS.
```

C. THE HELP MESSAGE MAY BE ELIMINATED BY TYPING CTL C.
D. GO TO CHAPTER 2. USE PROCEDURES.

NOTE: <CR> MEANS PRESSING THE "RETURN" KEY ON KEYBOARD

1.3 LOADING TADP MONITOR

THE TADP MONITOR CAN BE LOADED BY MEANS OF THE BM792YH ROM BOOT AS FOLLOWS:

A. MOUNT THE TADP CASSETTE IN DRIVE 0 (LEFT HAND DRIVE). THE CASSETTE SHOULD BE WRITE-LOCKED TO PREVENT ACCIDENTALLY WRITING ON IT.
B. LOAD ADDRESS 173300
C. PRESS START
D. THE MONITOR IS LOADED FROM MEDIUM
E. THE MONITOR TYPES THE FOLLOWING MESSAGE AND IS THEN READY TO ACCEPT KEYBOARD COMMANDS.

```
DZQUE-F 21-JUL-76 TADP - TA11 MONITOR NNK
RESTART ADDR: XXXXXX
BOOTED VIA UNIT#: 0
TO ABORT THE FOLLOWING HELP MESSAGE TYPE CTRL C ( C)

TYPE:
F<CR> TO SET CONSOLE FILL COUNT
D<CR> FOR DIRECTORY ON CONSOLE, OR
D/F<CR> FOR SHORT DIRECTORY ON CONSOLE, OR
D/L<CR> FOR DIRECTORY ON LINE PRINTER, OR
D/L/F<CR> FOR SHORT DIRECTORY ON LINE PRINTER,
R COPY1<CR> TO RUN COPY1 PROGRAM
R FILENAME<CR> TO RUN ANY OTHER  PROGRAM
L FILENAME<CR> TO LOAD A PROGRAM ONLY
S<CR> TO START THE PROGRAM JUST LOADED,
S ADDR<CR> TO START THE PROGRAM AT SPECIFIC ADDRESS
C FILENAME<CR> TO RUN A CHAIN,
C FILENAME/QV<CR> TO RUN A CHAIN IN QUICK VERIFY MODE
REFER TO XXDP MANUAL MD-11-DZQXA FOR ADDITIONAL HELP.
.
```

WHERE: NNK IS THE SYSTEM'S STORAGE UP TO 28K,
XXXXXX IS THE MONITOR'S RESTART ADDRESS.
THE DOT (.) INDICATES THE MONITOR IS READY TO ACCEPT COMMANDS.

F. THE HELP MESSAGE MAY BE ELIMINATED BY TYPING CTL C.
G. GO TO CHAPTER 2. USE PROCEDURES.

1.4 LOADING TMDP/THDP MONITOR

THE TMDP/THDP MONITOR CAN BE LOADED BY ANY OF THE ACCEPTABLE ROMS, OR VIA A "TOGGLE-IN" PROCEDURE.
THE TOGGLE-IN PROCEDURE IS ONLY VALID FOR THE TM11.

1.4.1 VIA BOOTSTRAP LOADER

A. MOUNT THE TMDP/THDP TAPE ON DRIVE 0 AND MAKE READY.
B. REWIND DRIVE 0 TO "BOT" AND SET "ON-LINE"
C. LOAD THE PROPER ADDRESS CORRESPONDING TO THE ROM/TAPE DRIVE CONFIGURATION.
D. PRESS START
E. GO TO 1.4.3 STEP A.

1.4.2 VIA "TOGGLE-IN" PROCEDURE

A. MOUNT TMDP/THDP TAPE ON DRIVE 0 AND MAKE READY.
B. REWIND DRIVE 0 TO "BOT" AND SET "ON-LINE".
C. LOAD ADDRESS 010000
D. DEPOSIT THE FOLLOWING VALUES: (FOR TM11)
005137,172524,012737,060011,172522
000777,012737,060003,172522,105737
172522,100375,000137,000000
GO TO STEP E

DEPOSIT THE FOLLOWING VALUES: (FOR TM02)
012737,001300,172472,012737
177777,172446,012737,000031,172440
105737,172462,100375,012737,177400
172442,005037,172444,042737,000007
172452,012737,000071,172440,105737
172440,000375,000137,000000
GO TO STEP H

E. LOAD ADDRESS 010000 AND PRESS START.
F. AFTER ONE SECOND DEPRESS HALT. LOAD ADDRESS 010014, PRESS START
G. GO TO 1.4.3 STEP A.
H. LOAD ADDRESS 10000 AND PRESS START.
I. GO TO 1.4.3 STEP A.

1.4.3 COMMON PROCEDURE

A. THE MONITOR IS LOADED FROM MEDIUM
B. THE MONITOR TYPES THE FOLLOWING MESSAGE AND IS THEN READY
   TO ACCEPT KEYBOARD COMMANDS.

```
DZQUF-G 22-OCT-77 TMDP - TM11 MONITOR NNK

        OR


DZQUH-D 22-OCT-77 TMDP - TM02/TU16 MONITOR NNK
RESTART ADDR: XXXXXX
BOOTED VIA UNIT#: 0
TO ABORT THE FOLLOWING HELP MESSAGE TYPE CTRL C ( C)

TYPE:
F<CR> TO SET CONSOLE FILL COUNT
D<CR> FOR DIRECTORY ON CONSOLE, OR
D/F<CR> FOR SHORT DIRECTORY ON CONSOLE, OR
D/L<CR> FOR DIRECTORY ON LINE PRINTER, OR
D/L/F<CR> FOR SHORT DIRECTORY ON LINE PRINTER,
R COPY1<CR> TO RUN COPY1 PROGRAM
R FILENAME<CR> TO RUN ANY OTHER  PROGRAM
L FILENAME<CR> TO LOAD A PROGRAM ONLY
S<CR> TO START THE PROGRAM JUST LOADED,
S ADDR<CR> TO START THE PROGRAM AT SPECIFIC ADDRESS
C FILENAME<CR> TO RUN A CHAIN,
C FILENAME/QV<CR> TO RUN A CHAIN IN QUICK VERIFY MODE.
REFER TO XXDP MANUAL MD-11-DZQXA FOR ADDITIONAL HELP.
.

        WHERE: NNK IS THE SYSTEM'S STORAGE UP TO 28K,
               XXXXXX IS THE MONITOR'S RESTART ADDRESS.
               THE DOT (.) INDICATES THE MONITOR IS READY TO ACCEPT COMMANDS
```

C. THE HELP MESSAGE MAY BE ELIMINATED BY TYPING CTL C.
D. GO TO CHAPTER 2. USE PROCEDURES.

NOTE: <CR> MEANS PRESSING THE "RETURN" KEY ON KEYBOARD

1.5 LOADING RXDP MONITOR

THE RXDP MONITOR CAN BE LOADED BY MEANS OF THE BOOTSTRAP LOADER, OR VIA "TOGGLE-IN" PROCEDURE.

1.5.1 LOADING THE 11V03 SYSTEM

A. WITH THE SYSTEM POWERED UP IN ENABLE MODE THE PROMPT CHARACTER ($) IS TYPED AND THE USER TYPES IN THE DEVICE CODE AND <CR>;

$ (PROMPT) DX0<CR> OR DX1<CR> (USER ENTRY)

THIS BOOTS THE RXDP AND STARTS THE MONITOR.

B. IN THE CONSOLE HALT MODE (ODT) THE PROMPT CHARACTER (@) IS TYPED AND THE ANSWER IS THE ADDRESS OF THE BOOT ROM AND A G (GO) <CR>.

@ (PROMPT) 173000 G<CR> (USER ENTRY)

THIS ENABLES THE PROMPT "$", PROCEED AS IN CHAPTER "A" ABOVE.

C. THE TOGGLE-IN PROCEDURE IN CHAPTER 1.5.3 MAY BE ENTERED AFTER THE PROMPT "@", USING THE <CR> AS A TERMINATOR AND <LF> AS AN ADVANCE TO THE NEXT LINE. TO START ENTER ADDRESS AND "G".

D. GO TO COMMON PROCEDURE 1.5.4

NOTE: ON ALL 11V03 SYSTEMS THE LINE CLOCK MUST BE DISABLED VIA THE LTC SWITCH.

1 5 2 VIA ------ PRIMARY BOOTSTRAP LOADER

A. MOUNT THE RXDP DISKETTE ON DRIVE 0.
B. LOAD AND WAIT UNTIL DRIVE READY.
C. LOAD ADDRESS ---------
D. PRESS START.
E. GO TO 1.5.3.

1.5.3 VIA "TOGGLE-IN" PROCEDURE.

A. MOUNT RXDP DISKETTE ON DRIVE 0.
B. LOAD AND WAIT UNTIL DRIVE READY.
C. LOAD FOLLOWING PRIMARY BOOTSTRAP:

ADDRESS	CONTENTS
1000	5000
2	12701
4	177170
6	105711
10	1776
12	12711
14	3
16	5711
20	1776
22	100405
24	105711
26	100004
30	116120
32	2
34	770
36	0
40	5000
42	110

D. LOAD ADDRESS 1000
E. PRESS START

NOTE: THE RXDP SECONDARY BOOTSTRAP EXPECTS THE PRIMARY BOOTSTRAP TO LEAVE THE DRIVE NUMBER IN R0 AND THE BUS ADDRESS OF THE RXCS REGISTER IN R1. IF A "TOGGLE-IN" PRIMARY BOOTSTRAP DIFFERENT FROM THAT GIVEN ABOVE IS USED, THIS CONDITION MUST BE SATISFIED. IF R0 CONTAINS A VALUE NOT 0 OR 1, THE SECONDARY BOOTSTRAP WILL DEFAULT TO DRIVE 1.

1.5.4 COMMON PROCEDURE

A. THE MONITOR IS LOADED FROM THE MEDIUM.
B. THE MONITOR TYPES THE FOLLOWING MESSAGE AND IS THEN
   READY TO ACCEPT KEYBOARD COMMANDS.

```
DZQUJ-D 22-OCT-77 RXDP - RX11/RX01 MONITOR NNK
RESTART ADDR: XXXXXX
BOOTED VIA UNIT#: 0
TO ABORT THE FOLLOWING HELP MESSAGE TYPE CTRL C ( C)

TYPE:
F<CR> TO SET CONSOLE FILL COUNT
D<CR> FOR DIRECTORY ON CONSOLE, OR
D/F<CR> FOR SHORT DIRECTORY ON CONSOLE, OR
D/L<CR> FOR DIRECTORY ON LINE PRINTER, OR
D/L/F<CR> FOR SHORT DIRECTORY ON LINE PRINTER,
R COPY1<CR> TO RUN COPY1 PROGRAM,
R FILENAME<CR> TO RUN ANY OTHER  PROGRAM
L FILENAME<CR> TO LOAD A PROGRAM ONLY
S<CR> TO START THE PROGRAM JUST LOADED,
S ADDR<CR> TO START THE PROGRAM AT SPECIFIC ADDRESS
C FILENAME<CR> TO RUN A CHAIN,
C FILENAME/QV<CR> TO RUN A CHAIN IN QUICK VERIFY MODE.
REFER TO XXDP MANUAL MD-11-DZQXA FOR ADDITIONAL HELP.
.

WHERE:  NNK IS THE SYSTEMS STORAGE UP TO 28K.
        XXXXXX IS THE MONITORS RESTART ADDRESS.
        THE DOT (.) INDICATES THE MONITOR IS READY TO ACCEPT COMMANDS
```

C. THE HELP MESSAGE MAY BE ELIMINATED BY TYPING CTL C.
D. GO TO CHAPTER 2. USE PROCEDURES.

NOTE: <CR> MEANS PRESSING THE "RETURN" KEY ON THE KEYBOARD.

1.6 LOADING THE RPDP MONITOR

THE RPDP MONITOR CAN BE LOADED BY MEANS OF THE ROM BOOT WHICH SUPPORTS THE RP03.

1.6.1 VIA THE ROM BOOTSTRAP LOADER

A. MOUNT THE RPDP DISK ON DRIVE 0.
B. LOAD AND WAIT UNTIL DRIVE IS READY.
C. LOAD PROPER ROM ADDRESS FOR RP03.
D. PRESS START.

1.6.2 COMMON PROCEDURE

A. THE MONITOR IS LOADED FROM THE MEDIUM.
B. THE MONITOR TYPES THE FOLLOWING MESSAGE AND IS THEN READY TO ACCEPT KEYBOARD COMMANDS.

```
DZQUN-B 21-JUL-76 RPDP - RP11 MONITOR NNK
RESTART ADDR: XXXXXX
BOOTED VIA UNIT#:  0
TO ABORT THE FOLLOWING HELP MESSAGE TYPE CTRL C ( C)

TYPE:
F<CR> TO SET CONSOLE FILL COUNT
D<CR> FOR DIRECTORY ON CONSOLE, OR
D/F<CR> FOR SHORT DIRECTORY ON CONSOLE, OR
D/L<CR> FOR DIRECTORY ON LINE PRINTER, OR
D/L/F<CR> FOR SHORT DIRECTORY ON LINE PRINTER,
R COPY1<CR> TO RUN COPY1 PROGRAM,
R FILENAME<CR> TO RUN ANY OTHER  PROGRAM
L FILENAME<CR> TO LOAD A PROGRAM ONLY
S<CR> TO START THE PROGRAM JUST LOADED,
S ADDR<CR> TO START THE PROGRAM AT SPECIFIC ADDRESS
C FILENAME<CR> TO RUN A CHAIN,
C FILENAME/QV<CR> TO RUN A CHAIN IN QUICK VERIFY MODE.
REFER TO XXDP MANUAL MD-11-DZQXA FOR ADDITIONAL HELP.
.

WHERE:  NNK IS THE SYSTEMS STORAGE UP TO 28K.
        XXXXXX IS THE MONITORS RESTART ADDRESS.
        THE (.) INDICATES THE MONITOR IS READY TO ACCEPT COMMANDS.
```

C. THE HELP MESSAGE MAY BE ELIMINATED BY TYPING CTL C.
D. GO TO SECTION 2. USE PROCEDURES.

1.7 LOADING THE RBDP MONITOR

THE RBDP MONITOR CAN BE LOADED BY MEANS OF THE ROM BOOT WHICH SUPPORTS THE RP04 DISK.

1.7.1 VIA THE ROM BOOTSTRAP LOADER

A. MOUNT THE RBDP DISK ON DRIVE 0.
B. LOAD AND WAIT UNTIL DRIVE IS READY.
C. LOAD PROPER ROM ADDRESS FOR THE RP04.
D. PRESS START.

1.7.2 COMMON PROCEDURE

A. THE MONITOR IS LOADED FROM THE MEDIUM.
B. THE MONITOR TYPES THE FOLLOWING MESSAGE AND IS THEN READY TO ACCEPT KEYBOARD COMMANDS:

```
DZQUO-B 21-JUL-76 RBDP - RP04 MONITOR NNK
RESTART ADDR: XXXXXX
BOOTED VIA UNIT#: 0
TO ABORT THE FOLLOWING HELP MESSAGE TYPE CTRL C ( C)

TYPE:
F<CR> TO SET CONSOLE FILL COUNT
D<CR> FOR DIRECTORY ON CONSOLE, OR
D/F<CR> FOR SHORT DIRECTORY ON CONSOLE, OR
D/L<CR> FOR DIRECTORY ON LINE PRINTER, OR
D/L/F<CR> FOR SHORT DIRECTORY ON LINE PRINTER,
R COPY1<CR> TO RUN COPY1 PROGRAM,
R FILENAME<CR> TO RUN ANY OTHER PROGRAM
L FILENAME<CR> TO LOAD A PROGRAM ONLY
S<CR> TO START THE PROGRAM JUST LOADED,
S ADDR<CR> TO START THE PROGRAM AT SPECIFIC ADDRESS
C FILENAME<CR> TO RUN A CHAIN,
C FILENAME/QV<CR> TO RUN A CHAIN IN QUICK VERIFY MODE.
REFER TO XXDP MANUAL MD-11-DZQXA FOR ADDITIONAL HELP.
.

WHERE:  NNK IS THE SYSTEMS STORAGE UP TO 28K.
        XXXXXX IS THE MONITORS RESTART ADDRESS.
        THE (.) INDICATES THE MONITOR IS READY TO ACCEPT COMMANDS.
```

C. THE HELP MESSAGE MAY BE ELIMINATED BY TYPING CTL C.
D. GO TO CHAPTER 2. USE PROCEDURES.

1.8 LOADING THE RSDP MONITOR

THE RSDP MONITOR CAN BE LOADED BY MEANS OF THE ROM BOOT WHICH SUPPORTS THE RS03 DISK.

1.8.1 VIA THE ROM BOOTSTRAP LOADER

A. LOAD AND WAIT UNTIL DRIVE 0 IS READY.
B. LOAD PROPER ROM ADDRESS FOR THE RS03.
C. PRESS START.

1.8.2 COMMON PROCEDURE

A. THE MONITOR IS LOADED FROM THE MEDIUM.
B. THE MONITOR TYPES THE FOLLOWING MESSAGE AND IS THEN READY TO ACCEPT KEYBOARD COMMANDS.

```
DZQUP-B 21-JUL-76 RSDP - RS04 MONITOR NNK
RESTART ADDR: XXXXXX
BOOTED VIA UNIT#: 0
TO ABORT THE FOLLOWING HELP MESSAGE TYPE CTRL C ( C)

TYPE:
F<CR> TO SET CONSOLE FILL COUNT
D<CR> FOR DIRECTORY ON CONSOLE, OR
D/F<CR> FOR SHORT DIRECTORY ON CONSOLE, OR
D/L<CR> FOR DIRECTORY ON LINE PRINTER, OR
D/L/F<CR> FOR SHORT DIRECTORY ON LINE PRINTER,
R COPY1<CR> TO RUN COPY1 PROGRAM,
R FILENAME<CR> TO RUN ANY OTHER PROGRAM
L FILENAME<CR> TO LOAD A PROGRAM ONLY
S<CR> TO START THE PROGRAM JUST LOADED,
S ADDR<CR> TO START THE PROGRAM AT SPECIFIC ADDRESS
C FILENAME<CR> TO RUN A CHAIN,
C FILENAME/QV<CR> TO RUN A CHAIN IN QUICK VERIFY MODE.
REFER TO XXDP MANUAL MD-11-DZQXA FOR ADDITIONAL HELP.
.
```

WHERE: NNK IS THE SYSTEM STORAGE UP TO 28K.
XXXXXX IS THE MONITORS RESTART ADDRESS.
THE (.) INDICATES THE MONITOR IS READY TO ACCEPT COMMANDS.

C. THE HELP MESSAGE MAY BE ELIMINATED BY TYPING CTL C.
D. GO TO SECTION 2. USE PROCEDURES.

1.9 LOADING THE RMDP MONITOR

THE RMDP MONITOR MAY BE LOADED BY MEANS OF THE APPROPRIATE ROM BOOTSTRAP. ONCE THE RMDP MONITOR HAS BEEN BOOTED, IT TYPES THE FOLLOWING MESSAGE AND IS READY TO ACCEPT KEYBOARD COMMANDS.

```
DZQUT-B 22-OCT-77 RMDP - RK06 MONITOR NNK

RESTART ADDR: XXXXXX
BOOTED VIA UNIT#: 0
TO ABORT THE FOLLOWING HELP MESSAGE TYPE CTRL C ( C)

TYPE:
F<CR> TO SET CONSOLE FILL COUNT
D<CR> FOR DIRECTORY ON CONSOLE, OR
D/F<CR> FOR SHORT DIRECTORY ON CONSOLE, OR
D/L<CR> FOR DIRECTORY ON LINE PRINTER, OR
D/L/F<CR> FOR SHORT DIRECTORY ON LINE PRINTER,
R COPY1<CR> TO RUN COPY1 PROGRAM,
R FILENAME<CR> TO RUN ANY OTHER PROGRAM
L FILENAME<CR> TO LOAD A PROGRAM ONLY
S<CR> TO START THE PROGRAM JUST LOADED,
S ADDR<CR> TO START THE PROGRAM AT SPECIFIC ADDRESS
C FILENAME<CR> TO RUN A CHAIN,
C FILENAME/QV<CR> TO RUN A CHAIN IN QUICK VERIFY MODE.
REFER TO XXDP MANUAL MD-11-DZQXA FOR ADDITIONAL HELP.
.
```

WHERE: NNK IS THE SYSTEM STORAGE UP TO 28K.
XXXXXX IS THE MONITORS RESTART ADDRESS.
THE (.) INDICATES THE MONITOR IS READY TO ACCEPT COMMANDS.

NOTE. THE HELP MESSAGE MAY BE ELIMINATED BY TYPING CTL C.

GO TO SECTION 2. USE PROCEDURES.

1.10 LOADING THE RLDP MONITOR

THE RLDP MONITOR MAY BE LOADED BY MEANS OF THE APPROPIATE ROM BOOTSTRAP. ONCE RLDP MONITOR HAS BEEN BOOTED, IT TYPES THE FOLLOWING MESSAGE AND IS READY TO ACCEPT KEYBOARD COMMANDS

```
DZQUZ-A 22-OCT-77 RLDP - RL01 MONITOR NNK
RESTART ADDR: XXXXXX
BOOTED VIA UNIT#: 0
TO ABORT THE FOLLOWING HELP MESSAGE TYPE CTRL C ( C)

TYPE:
F<CR> TO SET CONSOLE FILL COUNT
D<CR> FOR DIRECTORY ON CONSOLE, OR
D/F<CR> FOR SHORT DIRECTORY ON CONSOLE, OR
D/L<CR> FOR DIRECTORY ON LINE PRINTER, OR
D/L/F<CR> FOR SHORT DIRECTORY ON LINE PRINTER,
R COPY1<CR> TO RUN COPY1 PROGRAM,
R FILENAME<CR> TO RUN ANY OTHER PROGRAM
L FILENAME<CR> TO LOAD A PROGRAM ONLY
S<CR> TO START THE PROGRAM JUST LOADED,
S ADDR<CR> TO START THE PROGRAM AT SPECIFIC ADDRESS
C FILENAME<CR> TO RUN A CHAIN,
C FILENAME/QV<CR> TO RUN A CHAIN IN QUICK VERIFY MODE.
REFER TO XXDP MANUAL MD-11-DZQXA FOR ADDITIONAL HELP.
.
```

WHERE: NNK IS THE SYSTEM STORAGE UP TO 28K.
XXXXXX IS THE MONITORS RESTART ADDRESS.
THE (.) INDICATES THE MONITOR IS READY TO ACCEPT COMMANDS.

NOTE. THE HELP MESSAGE MAY BE ELIMINATED BY TYPING CTL C.

GO TO SECTION 2. USE PROCEDURES.

1.11 LOADING THE RYDP MONITOR

THE RYDP MONITOR MAY BE LOADED BY MEANS OF THE APPROPRIATE ROM BOOTSTRAP. ONCE THE RYDP MONITOR HAS BEEN BOOTED, IT TYPES THE FOLLOWING MESSAGE AND IS READY TO ACCEPT KEYBOARD COMMANDS.

```
CZQUS-A 15-AUG-78 RYDP - RX02 MONITOR NNK
RESTART ADDR: XXXXXX
BOOTED VIA UNIT#: 0
TO ABORT THE FOLLOWING HELP MESSAGE TYPE CTRL C ( C)

TYPE:
F<CR> TO SET CONSOLE FILL COUNT
D<CR> FOR DIRECTORY ON CONSOLE, OR
D/F<CR> FOR SHORT DIRECTORY ON CONSOLE, OR
D/L<CR> FOR DIRECTORY ON LINE PRINTER, OR
D/L/F<CR> FOR SHORT DIRECTORY ON LINE PRINTER,
R COPY1<CR> TO RUN COPY1 PROGRAM,
R FILENAME<CR> TO RUN ANY OTHER  PROGRAM
L FILENAME<CR> TO LOAD A PROGRAM ONLY
S<CR> TO START THE PROGRAM JUST LOADED,
S ADDR<CR> TO START THE PROGRAM AT SPECIFIC ADDRESS
C FILENAME<CR> TO RUN A CHAIN,
C FILENAME/QV<CR> TO RUN A CHAIN IN QUICK VERIFY MODE
REFER TO XXDP MANUAL MD-11-DZQXA FOR ADDITIONAL HELP
.

WHERE:  NNK IS THE SYSTEM STORAGE UP TO 28K.
        XXXXXX IS THE MONITORS RESTART ADDRESS.
        THE (.) INDICATES THE MONITOR IS READY TO ACCEPT COMMANDS

NOTE.   THE HELP MESSAGE MAY BE ELIMINATED BY TYPING CTL C
```

GO TO SECTION 2. USE PROCEDURES.

2. USE PROCEDURES

THE USE PROCEDURES THAT FOLLOW APPLY TO ALL XXDP MONITORS EXCEPT FOR THE TADP MONITOR WHICH PROVIDES SLIGHTLY MODIFIED OPERATIONS, REFER TO SECTION 2.5 TADP MONITOR EXCEPTIONS.

2.1 SET THE FILL COUNT

THE TTY OUTPUT ROUTINE OF THE UPDATE PROGRAM NORMALLY OUTPUTS 14(8) FILLER CHARACTERS AFTER A CARRIAGE RETURN, IN ORDER TO INSURE THAT THE LA30S TERMINAL PRINTS CORRECTLY, HOWEVER, ON TERMINALS OTHER THAN THE LA30S THE FILLER CHARACTERS ARE NOT REQUIRED AND ARE TIME CONSUMING AND ANNOYING. THE NUMBER OF FILLER CHARACTERS OUTPUT CAN BE CHANGED BY MEANS OF THE "F" COMMAND. THE F COMMAND SHOULD BE THE FIRST COMMAND ISSUED IN ORDER TO PROPERLY SET UP THE CONSOLE. TYPE:

```
F<CR>

000014  1               ;THE 000014 IS TYPED BY THE PROGRAM AND
                        ;INDICATES THE CURRENT FILLER COUNT.  THE  1
                        ;INDICATES THE USER TYPED A FILLER COUNT OF 1.
```

2.2 OBTAINING A DIRECTORY

TO OBTAIN A DIRECTORY TYPE ONE OF THE FOLLOWING:

D<CR>	TO OBTAIN DIRECTORY ON CONSOLE TERMINAL, OR
D/F<CR>	TO OBTAIN SHORT DIRECTORY ON CONSOLE TERMINAL,
D/L<CR>	TO OBTAIN DIRECTORY ON LINE PRINTER. LINE PRINTER MUST BE PRESENT ON SYSTEM NO CHECK IS MADE FOR IT.

THE DIRECTORY CONTAINS THE FOLLOWING INFORMATION:

FILNAM.EXT	PROGRAM NAME AND EXTENSION ASSIGNED. .BIN, .BIC, AND .SAV, ARE THE ONLY VALID EXTENSIONS FOR XXDP MONITOR USE.
NOTE:	.BIN IS A BINARY FILE .BIC IS A CHAINABLE BINARY FILE .SAV IS A CORE IMAGE FILE.
LENGTH	NUMBER OF BLOCKS USED. DECIMAL NUMBER. (DISK AND DECTAPE ONLY).
START	STARTING BLOCK NUMBER. OCTAL NUMBER. (DISK AND DECTAPE ONLY).
DATE	DATE WHEN PROGRAM WAS PUT ON MEDIUM.

2.3 LOADING AND RUNNING PROGRAMS

A. TYPE "R" AND THE PROGRAM NAME (UP TO 6 CHARACTERS). DO NOT TYPE THE EXTENSION (.BIN,.BIC,).
THIS WILL LOAD AND RUN THE PROGRAM. TO JUST LOAD THE PROGRAM TYPE "L" AND THE PROGRAM NAME. ONCE LOADED TYPING A "S" WILL START THE PROGRAM.

B. DEPRESS THE CTL AND C KEYS.

IF A TYPING ERROR IS MADE, DEPRESS THE CTRL AND C KEYS AT SAME TIME. A DOT (.) WILL BE TYPED. RETYPE "R" AND THE PROGRAM NAME.

C. THE DESIRED PROGRAM IS LOADED, A DOT TYPED, AND,

1. THE PROGRAM SELF STARTS IF IT IS SELF STARTING, OR
2. THE PROGRAM IS STARTED AT LOC 000200 IF THE PROGRAM NAME WAS ENDED WITH AN ALTMODE CHARACTER, OR
3. THE MONITOR WAITS FOR ANOTHER COMMAND. THE PROGRAM JUST LOADED MUST BE STARTED MANUALLY BY TYPING S PROGRAM NAME <CR>.

D. TO LOAD ANOTHER PROGRAM AFTER RUNNING THE PREVIOUSLY LOADED PROGRAM, RESTART THE MONITOR AT THE RESTART ADDRESS, OR RELOAD THE MONITOR AS DESCRIBED IN CHAPTER 1.

E. POSSIBLE ERRORS ARE DESCRIBED IN CHAPTER 3.

CAUTION: WHEN LOADING DIAGNOSTICS THAT TEST THE XXDP MEDIUM CARE MUST BE TAKEN TO INSURE THAT THE MEDIUM IS NOT ACCIDENTALLY DESTROYED. THAT IS THE REASON THAT THE MEDIUM MUST BE WRITE-LOCKED. REMOVE IT IF IT IS DESIRED TO TEST THAT DRIVE.

2.4 CHAIN MODE OPERATION

CHAIN MODE OPERATION CONSISTS OF THE SEQUENTIAL EXECUTION OF PROGRAMS WITHOUT OPERATOR INTERVENTION. ONLY PROGRAMS THAT HAVE BEEN MODIFIED TO RUN IN CHAIN MODE CAN BE CHAINED. CHAINABLE PROGRAMS ARE IDENTIFIED IN THE DIRECTORY BY THE EXTENSION .BIC.
NOTE: .BIC IS A CHAINABLE BINARY FILE.

TO RUN CHAIN MODE, THE XXDP MONITOR REQUIRES A FILE INDICATING THE PROGRAMS TO RUN AND THE NUMBER OF PASSES EACH PROGRAM SHOULD RUN.

A CHAIN FILE MAY BE GENERATED BY USING THE XTECO TEXT EDITOR, AND THE USER MUST PUT A .CCC EXTENSION ON THE CHAIN FILE.

TO SUMMARIZE:

1. CHAIN MODE RUNS CHAINABLE PROGRAMS ONLY. (.BIC EXTENSIONS).
2. A CHAIN FILE INDICATES THE PROGRAMS TO RUN AND THEIR PASS COUNTS.
3. ONLY PROGRAMS RESIDENT ON THE SAME MEDIUM DRIVE CAN BE CHAINED
4. THE CHAIN FILE MUST BE ON THE SAME MEDIUM WITH A .CCC EXTENSION.

NOTE: THE .CCC EXTENSION INDICATES A CHAIN FILE

CHAIN MODE IS ENTERED BY TYPING:

C FILENAME<CR> (WHILE IN MONITOR MODE).
WHERE:
C IS THE "CHAIN" COMMAND
FILENAME IS THE VALUE OF THE ASCII FILE THAT CONTAINS THE MONITOR COMMANDS TO BE EXECUTED. THE FILE MUST HAVE A " CCC"EXTENSION

2.4.1 MAKING A CHAIN ASCII FILE

THE CHAIN ASCII FILE MAY BE CREATED BY RUNNING THE XTECO PROGRAM AND USING THE TEXT EDITOR TO CREATE THE ASCII CHAIN FILE. THE CHAIN FILE MAY CONTAIN ANY OF THE COMMANDS SUPPORTED UNDER THE XXDP MONITOR. THE COMMANDS IN THE ASCII FILE ARE EXECUTED IN THE ORDER IN WHICH THEY ARE ENTERED AND RUN AS A BATCH MODE.
EXAMPLE OF A CHAIN FILE;

```
;CPU.CCC
;THIS CHAIN FILE EXERCISES THE XYZ PROCESSOR WITH T1-T13..
;
R DOAA/1000              ;RUN T1 1000 TIMES<CR>
R DOBA/1000              ;RUN T2 1000 TIMES<CR>
R DOCA/1000              ;RUN T3 1000 TIMES<CR>
R DODA/1000              ;RUN T4 1000 TIMES<CR>
R DOEA/1000              ;RUN T5 1000 TIMES<CR>
R DOFA/1000              ;RUN T6 1000 TIMES<CR>
R DOGA/1000              ;RUN T7 1000 TIMES<CR>
R DOHA/1000              ;RUN T8 1000 TIMES<CR>
R DOJA/1000              ;RUN T9 1000 TIMES<CR>
R DOKA/1000              ;RUN T10 1000 TIMES<CR>
R DOLA/1000              ;RUN T11 1000 TIMES<CR>
R DOMA/1000              ;RUN T12 1000 TIMES<CR>
L DONA                   ;LOAD T13<CR>
S/1000<CR>               ;START IT, RUN 1000 TIMES<CR>
C CPU                    ;RESUBMIT CHAIN FILE AGAIN.
```

2.4.2 RUNNING A CHAIN

TO EXECUTE A CHAIN FILE THE USER TYPES;

C FILNAM<CR> OR
C FILNAM/QV<CR>

IN THE FIRST CASE THE PASS COUNT SPECIFIED IN THE CHAIN FILE IS USED BY THE XXDP MONITOR TO DETERMINE THE NUMBER OF PASSES TO EXECUTE EACH PROGRAM. IN THE SECOND CASE THE PASS COUNT IS NOT USED AND EACH PROGRAM IS EXECUTED ONLY ONCE. THE /QV SWITCH PROVIDES A SINGLE EXECUTION MODE OF OPERATION OR "QUICK VERIFY".

THE CHAIN FILE TO BE EXECUTED MUST HAVE AN EXTENSION OF . CCC.

THE CHAIN FILE AND THE OBJECTIVE PROGRAMS TO BE RUN MUST RESIDE IN THE SAME XXDP MEDIUM AND MUST BE MOUNTED ON DRIVE 0 OF XXDP DEVICE

WHEN IN CHAIN MODE SWITCH REGISTER OR SOFTWARE SWITCH REGISTER SHOULD BE SET TO 000000.

THE XXDP MONITOR WILL TYPE EACH COMMAND THAT IT EVALUATES AND THEN PROCEED TO EXECUTE IT.

IF THE MONITOR ENCOUNTERS A PROGRAM THAT DOES NOT HAVE A . BIC EXTENSION IT TYPES "NEXFIL". THEN IF THE ERROR RESULTED FROM A R (RUN COMMAND) ONLY, IT WILL CONTINUE WITH THE CHAIN FILE COMMAND, OTHERWISE IT TERMINATES THE CHAIN OPERATION.

WHEN THE LAST COMMAND OTHER THAN ANOTHER "C" COMMAND HAS BEEN EXECUTED THE XXDP MONITOR TERMINATES CHAIN MODE AND TYPES A DOT(.), READY TO ACCEPT ANOTHER COMMAND FROM THE CONSOLE.

IF THE USER WISHES TO TERMINATE CHAIN MODE BEFORE ITS NORMAL TERMINATION HE MAY DO SO BY REPEATEDLY TYPING CTL C (C) AT THE CONSOLE UNTIL THE MONITOR ACCEPTS IT AT THE END OF A PROGRAM PASS.

2.5 TADP MONITOR EXCEPTIONS

THE TADP PACKAGE CASSETTES ARE PACKAGED ACCORDING TO THE FOLLOWING SCHEMES.

1. ONE TADP CASSETTE CONTAINS THE TADP MONITOR AND XXDP UTILITIES
   (UPD1, UPD2, ETC)

2. ONE TADP CASSETTE CONTAINS THE TADP MONITOR AND XXDP UTILITIES (COPY2 PLUS UPD3)
FOR 16K CORE REQUIREMENTS.

3. SEVERAL DIAGNOSTIC CASSETTES CONTAINING THE DIAGNOSTIC PROGRAMS.

WHEN USING TADP, THE TADP CASSETTE MUST BE MOUNTED ON DRIVE 0
(LEFT HAND DRIVE) OF THE TA11; THE DIAGNOSTIC CASSETTE IS MOUNTED
ON DRIVE 1 (RIGHT HAND DRIVE).

BECAUSE THE TADP PACKAGE IS A TWO DRIVE SYSTEM, TWO ADDITIONAL
COMMANDS ARE PROVIDED THAT CONTROL THE DRIVE THAT IS TO BE ACCESSED..;

```
E 0<CR>          ;ENABLES ACCESS TO DRIVE 0.

E 1<CR>          ;ENABLES ACCESS TO DRIVE 1.
```

WHEN THE TADP MONITOR IS FIRST LOADED IT DEFAULTS TO DRIVE 0.
AT THAT POINT ALL COMMANDS GIVEN TO THE MONITOR APPLY TO DRIVE
0 ONLY.

TYPING E 1<CR> ENABLES ACCESS TO DRIVE 1 WITH ALL MONITOR COMMANDS
APPLYING TO DRIVE 1. TO RETURN TO ACCESS DRIVE 0 THE E 0<CR>
COMMAND IS GIVEN.

EXAMPLES;

```
E 0<CR>          ;ENABLES DRIVE 0 ACCESS.
D<CR>            ;OBTAINS DRIVE 0 DIRECTORY.
R UPD2           ;RUNS UPD2 AFTER LOADING FROM DRIVE 0.

E 1<CR>          ;ENABLES DRIVE 1 ACCESS.
D/F<CR>          ;FAST DIRECTORY FROM DRIVE 1.
L ZTCAA0<CR>     ;LOADS ZTCAA0 FROM DRIVE 1.
S 200<CR>        ;STARTS ZTCAA0.
E 0<CR>          ;RE-ENABLES DRIVE 0 ACCESS.
```

WHEN THE "D" (DIRECTORY) COMMAND IS GIVEN AND DRIVE 1 IS ENABLED
DRIVE 0 WILL BE ACCESSED FIRST IN ORDER TO LOAD THE NON-RESIDENT
DIRECTORY ROUTINE FROM THE TADP MONITOR ON DRIVE 0. THEN DRIVE 1
IS ACCESSED TO OBTAIN DRIVE 1 DIRECTORY.

IN CHAIN MODE THE CHAIN FILE IS ALWAYS ACCESSED FROM WHATEVER
DRIVE WAS ENABLED WHEN THE "C" COMMAND WAS GIVEN, EVEN IF THE
CHAIN FILE ITSELF CAUSES ANOTHER DRIVE TO BE ASSIGNED.

```
EXAMPLE;

E 0<CR>          ;DRIVE 0 ENABLED.
C CHAIN<CR>      ;RUN CHAIN FROM CHAIN.CCC (DRIVE 0).


        ASSUME CHAIN.CCC CONTAINS;

E 1<CR>          ;ENABLE DRIVE 1.
R T1/10<CR>      ;RUN T1 10 TIMES.
R T2/10<CR>      ;RUN T2 10 TIMES.
R T3<CR>         ;RUN T3
R T4<CR>         ;RUN T4
R T5<CR>         ;RUN T5
"
"
"
"
"
R T90<CR>        ;RUN T90
E 0<CR>          ;ENABLE DRIVE 0.

THE CHAIN.CCC FILE WILL BE ACCESSED FROM DRIVE 0.  ALL THE TEST
PROGRAMS WILL BE ACCESSED FROM DRIVE 1.  AT COMPLETION OF CHAIN
DRIVE 0 WILL BE ENABLED.

NOTE THAT WITH TADP, CHAIN FILES DO NOT HAVE TO BE IN
THE SAME CASSETTE AS THE TEST PROGRAMS.

WHEN IN DOUBT AS TO WHAT DRIVE IS AVAILABLE THE USER JUST
HAS TO GIVE THE COMMAND THAT ENABLES THE DRIVE HE WISHES TO
USE.
```

3. ERRORS

3.1 XXDP RESIDENT MONITOR ERRORS

INVCMD/SW	INVALID COMMAND AND/OR SWITCH. CHECK COMMAND JUST GIVEN.
DEVERR	DEVICE ERROR ON INPUT DEVICE.
EOM	END OF MEDIUM. OCCURS DURING INPUT OPERATIONS WHEN THE PROGRAM ATTEMPTS TO INPUT AND THE FILE IS AT AN END. SERIOUS PROBLEM. FILE IN STORAGE IS PROBABLY WIPED OUT.
INVADR	INVALID ADDRESS. MUST BE EVEN WITHIN EXISTING LOCORE AND HICORE LIMITS, AND MUST NOT BE WITHIN UPDATE PROGRAM.
CKSMER	CHECKSUM ERROR DURING "LOAD" COMMAND.
POFLO	PROGRAM TOO LARGE TO LOAD WITHIN EXISTING CORE SPACE.
INVNAM	INVALID CHARACTER TYPED FOR FILE NAME.
NEXFIL	NON-EXISTENT FILE. IF IN CHAIN MODE THE PROGRAM TO BE RUN DOES NOT HAVE .BIC EXTENSION.

CHAPTER 3. XXDP UPDATE PROGRAMS #1 (UPD1) , #2 (UPD2), AND #3 (UPD3) AND (UPD3R)

TABLE OF CONTENTS

1. ABSTRACT

EACH XXDP PACKAGE CONTAINS THREE PROGRAMS CALLED UPD1.BIN, UPD1A, UPD2.BIN, AND UPD3.BIN THESE PROGRAMS ARE USED TO ADD, DELETE, RENAME, OR PATCH PROGRAMS ON XXDP PACKAGES, AND IN GENERAL, PROVIDE FILE MAINTENANCE SERVICES.

UPD1 IS A 4K PROGRAM THAT RELOCATES ITSELF TO THE TOP OF MEMORY, TO LEAVE LOWER STORAGE FREE FOR OTHER PROGRAMS. UPD1 CAN OF PERFORM OPERATIONS ON FOUR XXDP MASS STORAGE DEVICES, PLUS PAPER TAPE.

UPD1A IS A 4K PROGRAM THAT RELOCATES IT SELF TO THE TOP OF MEMORY, AND LEAVES LOWER STORAGE FREE FOR OTHER PROGRAMS. UPD1A CAN PERFORM ON XXDP MASS STORAGE DEVICES, PLUS PAPER TAPE.

UPD2 IS A 8K PROGRAM WHICH RELOCATES ITSELF TO THE TOP OF MEMORY, LEAVING LOWER STORAGE FREE FOR OTHER PROGRAMS. IT CAN PERFORM OPERATIONS ON ADDITIONAL XXDP MASS STORAGE DEVICES.

UPDATE PROGRAM #2 IS AN EXPANSION OF UPDATE PROGRAM #1. IT INCLUDES ALL THE FEATURES OF UPD1, WITH ADDED FACILITIES FOR HANDLING AND CHECKING GROUPS OF FILES. THE ABILITY TO EXECUTE A COMMAND FILE, AND THE USE OF THE "ASTERISK" AND "WILD CHARACTER" CONSTRUCTIONS HAVE BEEN ADDED TO FACILITATE USER FILE STORAGE MANIPULATIONS.

UPDATE PROGRAM #3 IS A 12K PROGRAM THAT CONTAINS ALL FEATURES PROVIDED BY UPD2, BUT IT CAN PERFORM OPERATIONS ON ALL XXDP MASS STORAGE DEVICES.

A FOURTH UPDATE PROGRAM CALLED UPD3R IS AVAILABLE FOR DEC INTERNAL USE. ITS PURPOSE IS TO PROVIDE A FACILITY FOR RELIABLE GENERATION OF XXDP PACKAGES BY THE RELEASE ENGINEERING GROUP OF DIAGNOSTIC ENGINEERING.

2. REQUIREMENTS

2.1 THE FOLLOWING MINIMUM CONFIGURATION IS REQUIRED TO RUN UPD1:

CONSOLE TERMINAL
XXDP INPUT MEDIUM FOR UPD1: (RK11, TC11, RX01, TA11)
AT LEAST 8K MEMORY

2.1.1 THE FOLLOWING MINIMUM CONFIGURATION IS REQUIRED TO RUN UPD1A:

CONSOLE TERMINAL
XXDP INPUT MEDIA FOR UPD1A: (RK11, RX11, RX02)
AT LEAST 8K OF MEMORY.

2.2 THE FOLLOWING MINIMUM CONFIGURATION IS REQUIRED TO RUN UPD2:

CONSOLE TERMINAL
1 OR MORE XXDP MEDIA (TC11,RK11,TA11,TM11,TM02,RX11,RXV11,RS11,
RP11,TS03,RP04,RK06)
AT LEAST 16K MEMORY

2.3 THE FOLLOWING MINIMUM CONFIGURATION IS REQUIRED TO RUN UPD3:

CONSOLE TERMINAL
1 OR MORE XXDP MEDIA (TC11,RK11,TA11,TU11,TM02,RX11,RXV11,
RS11,RP11,TS03,RP04,RK06,RL01,RX02)
AT LEAST 20K MEMORY.

2.4 IN ORDER TO SUCCESSFULLY LOAD A PROGRAM USING THE UPDATE #1 PROGRAM, ONE MUST HAVE AT LEAST 4K MORE STORAGE THAN THE LARGEST PROGRAM TO BE LOADED REQUIRES. UPD2 NEEDS 8K MORE. UPD3 NEEDS 12K MORE.

2.5 WHEN THE USER ARE TYPING A COMMAND OR DATA UNDER UPD1/UPD1A/UPD2/UPD3, THEY SHOULD BE AWARE OF THE FOLLOWING SPECIAL CHARACTERS:

C (CONTROL C) EXITS TO COMMAND MODE.

Z (CONTROL Z) EXITS TEXT MODE, RETURNING TO COMMAND MODE

RUBOUT - DELETES THE LAST CHARACTER TYPED.

THE ONLY OUTPUT AND INPUT FILE SPECIFICATION SEPARATOR CHARACTERS ARE:

< (LEFT ANGLE BRACKET), (EQUAL SIGN) = AND _ (UNDERSCORE).

LEADING SPACES ARE IGNORED.

CARRIAGE RETURN IS THE ONLY LEGAL COMMAND TERMINATOR, EXCEPT IN THE CASE OF THE "MOD" AND "TEXT" COMMANDS.

FILENAMES ARE CONSIDERED TO BE ALWAYS 6 CHARACTERS LONG, PLUS A 3 CHARACTER EXTENSION. THE NAME AND EXTENSION ARE LEFT-JUSTIFIED WITH TRAILING BLANKS.

2.6 DEVICES SUPPORTED

RK11, TA11, PT11, TC11, RP11, RP04, RS11, TM11, TM02, RX11,
RXV11, TS03, RK06, RL01, RX02

3. LOADING AND STARTING PROCEDURE

UPD1/UPD1A/UPD2/UPD3 IS LOADED VIA THE XXDP MONITOR BY TYPING R UPD1<CR>/R UPD1A<CR>/R UPD2<CR>
R UPD3<CR>. ONCE LOADED, IT OUTPUTS THE FOLLOWING MESSAGE:

```
DZQUA-I - XXDP UPDATE PROGRAM #1 21-JUL-76
DATE (DD-MMM-YY):

OR

CZQU4-A - XXDP UPDATE PROGRAM #1A 15-AUG-78
DATE (DD-MMM-YY):

OR

DZQUB-J - XXDP UPDATE PROGRAM #2 22-OCT-77
DATE (DD-MMM-YY):

OR

CZQU1-B - XXDP UPDATE PROGRAM #3 15-AUG-78
DATE (DD-MMM-YY):
```

TYPE THE DATE ACCORDING TO FOLLOWING FORMAT:

```
DATE (DD-MMM-YY): DD-MMM-YY<CR>
```

DD IS THE DAY OF THE MONTH, MMM IS JAN, FEB, MAR, APR, MAY,
JUN, JUL, AUG, SEP, OCT, NOV, DEC, AND YY IS BETWEEN 70 AND 99.

A TEST IS MADE TO MAKE SURE NO MONTH HAS MORE THAN 31 DAYS.
DATES LIKE FEB 30, APR 31, ETC., WILL NOT BE DETECTED AS ERRORS
BUT WILL BE STORED AS FEB 30, APR 1, ETC.

THE PROGRAM WILL TYPE BACK THE DATE FOLLOWED BY:

```
PROGRAM RELOCATED TO: YYYYYY      ;INITIAL ADDR WHERE PROGRAM RELOCATED TO.
RESTART: XXXXXX                   ;RESTART ADDRESS.
@                                 ;@ INDICATES READY FOR KEYBOARD COMMANDS.
```

4. COMMAND DESCRIPTIONS

4.1 IN THE COMMAND DESCRIPTIONS THAT FOLLOW, AN INDICATION IS PROVIDED AS TO THE AVAILABILITY OF THE COMMAND UNDER UPD1, UPD2, OR UPD3. ALL COMMANDS ARE AVAILABLE FOR UPD2 AND UPD3. ONLY A SMALL SUBSET IS AVAILABLE FOR UPD1 . ALL COMMANDS DESCRIBED ARE PART OF THE UPD3R PROGRAM. COMMANDS THAT ARE UNIQUE TO THE UPD3R PROGRAM ARE INDICATED.

4.2 THE FILL COMMAND (UPD1, UPD1A, UPD2, UPD3)

THE CONSOLE TERMINAL OUTPUT ROUTINE OF THE UPDATE PROGRAM NORMALLY OUTPUTS 14(8) FILLER CHARACTERS AFTER A CARRIAGE RETURN, IN ORDER TO INSURE THAT THE LA30 TERMINAL PRINTS CORRECTLY. HOWEVER, ON TERMINALS OTHER THAN THE LA30 THE FILLER CHARACTERS ARE NOT REQUIRED.
THE NUMBER OF FILLER CHARACTERS OUTPUT CAN BE CHANGED BY MEANS OF THE "FILL" COMMAND.
TYPE:

```
FILL<CR>

000014  1               ;THE 000014 IS TYPED BY THE PROGRAM AND
                        ;INDICATES THE CURRENT FILLER COUNT. THE 1
                        ;INDICATES THE USER TYPED A FILLER COUNT OF 1.
```

THE FILLER COUNT SHOULD BE SET TO A 1 FOR ASR33 AND ASR35 TERMINALS. FOR OTHER TERMINALS, SET THE NUMBER TO WHATEVER PRODUCES CORRECT PRINTING AFTER A CARRIAGE RETURN, WITHOUT UNDUE DELAY.

4.3 THE "CLR" COMMAND (UPD1, UPD1A, UPD2, UPD3)

THE "CLR" COMMAND IS USED TO CLEAR TO ZEROES ALL CORE STORAGE BELOW THE UPDATE PROGRAM. IT IS PROVIDED IN CASE THE USER WISHES CORE STORAGE TO BE "ZEROED" PRIOR TO LOADING A PROGRAM. TYPE:

```
CLR<CR>
```

THE PROGRAM RESPONDS WITH *

4.4 LOAD COMMAND (UPD1, UPD1A, UPD2, UPD3)

THE LOAD COMMAND IS USED TO LOAD FILES STORED IN ABS FORMAT. (FILES WITH EXTENSIONS OF .BIN, .BIC, OR OTHER EXTENSIONS KNOWN TO INDICATE ABS FORMAT).

```
LOAD DEV: FILNAM.EXT    ;COMMAND FORMAT
```

IF THE DEVICE HAS NO DIRECTORY, THEN THE FILE NAME AND EXTENSION SHOULD BE OMITTED.

```
LOAD PR:                ;USER COMMAND TO LOAD FROM PAPER TAPE.
XFRADR: 000050  CORE: 000000,017670
*
```

XFRADR: INDICATES THE STARTING ADDRESS OF THE PROGRAM LOADED. IF IT IS 000001 OR ODD, THE PROGRAM IS NOT SELF-STARTING.

CORE: LEFT NUMBER INDICATES THE LOWEST LOCATION LOADED INTO DURING THE LOAD. THE RIGHT NUMBER INDICATES THE HIGHEST LOCATION LOADED INTO DURING THE LOAD. THE LEFT AND RIGHT NUMBERS IN EFFECT INDICATE THE CORE LIMITS OF THE PROGRAM.

4.5 DUMP COMMAND (UPD1, UPD1A, UPD2, UPD3)

THE MEMORY CONTENTS CAN BE WRITTEN TO A XXDP MEDIUM IN ABS FORMAT BY THE DUMP COMMAND.

```
DUMP DEV: FILNAM.EXT    ;COMMAND FORMAT
```

PROCESSING STARTS FROM PROGRAM'S LOW CORE LIMIT AND PROCEEDS TO BUT DOES NOT INCLUDE THE PROGRAM'S HIGH CORE LIMIT.

```
*DUMP DK0:XXXX.BIN      ;DUMP PROGRAM ONTO DK0:.  CALL IT XXXX.BIN
*DIR DK0:

12-JAN-76
ENTRY#          FILNAM  .EXT    DATE            LENGTH          START
000001          XXXX    .BIN    26-AUG-72       17              000105
000002          2                2-AUG-72       12C             000172
000003          3                2-AUG-72       12C             000206
FREE FILES: 445

*
```

4.6 THE "XFR" COMMAND (UPD1, UPD1A, UPD2, UPD3)

ONCE A PROGRAM HAS BEEN LOADED INTO CORE VIA THE "LOAD" COMMAND, IT CAN BE MADE SELF-STARTING OR NOT SELF-STARTING AT THE USER'S DISCRETION. AS DESCRIBED UNDER "LOAD COMMAND", THE LOAD ROUTINE TYPES: XFRADR : XXXXXX INDICATING WHETHER A PROGRAM IS OR IS NOT SELF-STARTING. THE USE OF "XFR" IS:

```
XFR<CR>          ;REQUEST CURRENT TRANSFER ADDRESS.
000001  000050   ;000001 IS THE CURRENT XFR ADDRESS. 000050 IS THE
                 ;NEW XFR ADDRESS ENTERED BY THE USER.
```

NOTE: DIAGNOSTIC PROGRAMS ARE PURPOSELY MADE NOT SELF-STARTING.

4.7 THE "START" COMMAND (UPD1, UPD1A, UPD2, UPD3)

THE "START" COMMAND IS USED TO BEGIN EXECUTION OF A PROGRAM IN CORE

```
START<CR>        ;USED TO START A SELF-STARTING PROGRAM.

START ADR<CR>    ;USED TO A START A PROGRAM AT A SPECIFIC LOCATION.
```

NOTE: IF THE COMMAND START<CR> IS GIVEN FOR A NON-SELF-START PROGRAM, THE PROCESSOR WILL TRAP OUT WITHOUT AN ERROR MESSAGE.

4.8 THE SAVE COMMAND (UPD1, UPD1A, UPD2, UPD3)

THE CONTENTS OF CORE ARE WRITTEN ONTO THE OUTPUT DEVICE AS A SINGLE BLOCK OF DATA, STARTING AT LOC 000000 AND PROCEEDING TO THE HIGH LIMIT OF THE PROGRAM IN CORE. THE SAVE COMMAND IN EFFECT, SAVES A "CORE IMAGE" OF THE CONTENTS OF CORE. FOR XXDP PURPOSES THE ONLY VALID EXTENSION FOR SAVED PROGRAMS IS .SAV.

THE ONLY CURRENT USE OF THE SAVE COMMAND IS TO PLACE A CORE IMAGE OF THE XXDP MONITOR ON CASSETTE AND MAGTAPE. XXDP PACKAGES DO NOT CONTAIN ANY OTHER CORE IMAGE FILES

NOTE: .SAV IS A CORE IMAGE FILE.

```
SAVE DEV: FILNAM.EXT    ;COMMAND FORMAT.

*SAVE DK0: UPDATE.SAV
*DIR DK0:

12-JAN-76
ENTRY#          FILNAM  .EXT    DATE            LENGTH          START
000001          UPDATE  .BIN    26-AUG-72       17              000105
000002          2                2-AUG-72       12C             000!/2
000003          UPDATE  .SAV    26-AUG-72       12C             000247
FREE FILES. 445
*
```

4.9 THE GET COMMAND (UPD2 AND UPD3)

THE GET COMMAND PLACES THE "SAVED" PROGRAM INTO CORE STARTING AT LOC 000000

```
GET DEV:FILNAM.EXT

 C
*GET DK0:UPDATE.SAV
*
```

NOTE: SAVE CORE IMAGE FILES (.SAV FILES) ARE NO LONGER IN USE, THE "GET" COMMAND IS NO LONGER VERY USEFUL. IT HAS BEEN LEFT AS THE COMPLEMENTARY COMMAND FOR THE SAVE COMMAND

4 10 THE MOD COMMAND (UPD1, UPD1A, UPD2, UPD3)

ONCE A PROGRAM IS LOADED IT CAN BE PATCHED BY THE MOD COMMAND.

MOD ADR CAUSES UPDATE TO PRINT THE FOLLOWING:

ADR CONTENTS OF ADR,

AND WAITS FOR USER RESPONSE.

THE USER MAY TYPE IN AN OCTAL NUMBER AND A TERMINATOR, OR JUST A TERMINATOR.

IF A NUMBER IS TYPED, IT IS USED AS THE NEW CONTENT OF ADR.

THE TERMINATOR CAN BE EITHER A CARRIAGE RETURN OR A LINE FEED. CARRIAGE RETURN TAKES THE PROGRAM BACK TO COMMAND MODE, WHEREAS THE LINE FEED CAUSES THE NEXT WORD (ADR+2) TO BE OPENED FOR MODIFICATION

```
*MOD 50
000050  000005  3 <LF>
000052  012737  4 <LF>
000054  000340  5 <CR>
*MOD 50
000050  000003    <LF>
000052  000004    <CR>
*
```

THE MOD COMMAND WILL NOT ALLOW THE USER TO GO BEYOND THE PROGRAM'S PROTECTION LIMIT, AN "INVCOR" ERROR WILL OCCUR (SEE SECTION 4 13)

4.10.1 THE MODALL COMMAND (UPD2 AND UPD3)

THE MODALL COMMANDS FUNCTIONS EXACTLY AS THE MOD COMMAND, BUT ALLOWS MODIFICATION OF ANY LOCATION, EVEN THOSE OUTSIDE THE LOW AND HIGH CORE LIMITS.

EXAMPLE:

```
MODALL 177740
177749  000010  4               ;MODIFIES LOCATION IN I/O PAGE
```

4.11 THE CORE COMMAND (UPD1, UPD1A, UPD2, UPD3)

THE CORE COMMAND CAUSES THE LOWER AND UPPER LIMITS OF THE PROGRAM IN CORE TO BE TYPED:

```
*CORE<CR>
000000,014776           ;LEFT NUMBER IS THE LOWER CORE LIMIT,
                        ;RIGHT NUMBER IS THE UPPER CORE LIMIT
```

4.12 THE "LOCORE" COMMAND (UPD1, UPD1A, UPD2, UPD3)

THE "LOCORE" COMMAND IS USED TO CHANGE THE LOWER LIMIT OF THE PROGRAM IN CORE.

```
*LOCORE ADR<CR>         ;WHERE ADR IS THE NEW LOW CORE LIMIT.  IT IS RECOMMENDED
                        ;THAT ADDRESS BE EVEN.
```

4.13 THE "HICORE" COMMAND (UPD1, UPD1A, UPD2, UPD3)

THE "HICORE" COMMAND IS USED TO CHANGE THE UPPER LIMIT OF THE PROGRAM IN CORE:

```
*HICORE ADR<CR>         ;WHERE ADR IS THE NEW HIGH CORE LIMIT.  RECOMMEND THAT
                        ;ADDRESS BE EVEN, BUT MUST BE HIGHER THAN THE LOWER
                        ;LIMIT, AND MUST BE LOWER THAN START OF UPDATE PROGRAM.
```

TYPICALLY, THE HICORE COMMAND IS USED TO RESERVE AN AREA FOR PATCHING A PROGRAM. THE UPDATE PROGRAM WILL NOT ALLOW MODIFICATION OF CORE BELOW THE LOW CORE LIMIT, AND WILL NOT ALLOW MODIFICATION OF LOCATIONS WHOSE ADDRESS IS EQUAL OR HIGHER THAN THE HIGH CORE LIMIT. THEREFORE, WHEN ADDING A PATCH, THE HIGH CORE LIMIT MUST BE SET SUFFICIENTLY HIGH SO AS TO INCLUDE THE COMPLETE PATCH.

4.14 THE DIRLP AND DIR COMMANDS

```
DIR     (UPD1,UPD1A, UPD2, UPD3)
DIRLP   (UPD2 AND UPD3)
*DIRLP DEV:     ;COMMAND FORMAT


COMMAND EXAMPLES;
UPD1,UPD1A, UPD2, AND UPD3
*DIR DEV:                       THIS GIVES AN ENTIRE DIRECTORY OF THE DEVICE.


UPD2 AND UPD3
*DIR DEV:*.BIN                  ;GIVES A DIRECTORY OF ALL FILES
                                WITH A ".BIN" EXTENSION.

*DIR DEV:*.BI?                  ;GIVES A DIRECTORY OF ALL FILES
                                WITH AN EXTENSION BEGINING WITH
                                "BI" AND ANY OTHER CHARACTER
                                SUCH AS BIN OR BIC.

DIR DEV ZTC??? BI?              ;GIVES A DIRECTORY OF ALL FILES
                                WITH THE FIRST THREE CHARACTERS
                                OF THE FILENAME BEING "ZTC"
                                AND HAVING AN EXTENSION BEGINING
                                WITH "BI". EXAMPLES; ZTCA.BIN,
                                ZTCB.BIN, ZTCC.BIC.
```

NOTE: AT THE END OF THE DIRECTORY THE FREE FILES AND FREE BLOCKS WILL BE INDICATED ONLY ON RANDOM ACCESS DEVICES AND ONLY FOR UPD2 AND UPD3.

NOTE: DIR IN UPDATE #1 GIVES ONLY THE SHORT DIRECTORY (NO LENGTH, NO START).

DIRLP CAUSES THE DIRECTORY OF DEV: TO PRINTED ON LINE PRINTER IF DIR IS USED, THE DIRECTORY IS TYPED ON CONSOLE DEVICE DO NOT USE DIRLP UNLESS A LINE PRINTER EXISTS, AS NO CHECK IS MADE FOR ITS EXISTENCE.

```
*DIR DK0:
12-JAN-76
ENTRY#          FILNAM  .EXT    DATE            LENGTH          START
000001          1               2-AUG-72        14              000105
000002          2               2-AUG-72        12C             000172
000003          3               2-AUG-72        12C             000206
000004          5               2-AUG-72        12C             000222
FREE FILES: 444
*
```

LENGTH IS THE NUMBER OF BLOCKS (10) THE FILE OCCUPIES. A "C" AFTER
THE FILE LENGTH INDICATES THE FILE IS CONTIGUOUS.

START IS THE ADDR OF FIRST BLOCK OF FILE. OCTAL NUMBER.
DATE IS THE FILE CREATION DATE.

4.15 THE DELETE COMMAND (UPD1, UPD1A, UPD2, UPD3)

```
DEL DEV:FILNAM.EXT

CAUSES THE FILE NAMED TO BE DELETED FROM THE DIRECTORY.

*DEL DK0:1
*DIR DK0:

12-JAN-76
ENTRY#          FILNAM  .EXT    DATE            LENGTH          START
000002          2               2-AUG-72        12C             000172
000003          3               2-AUG-72        12C             000206
000004          5               2-AUG-72        12C             000222
FREE FILES: 444
*
```

4.16 THE ZERO COMMAND (UPD1, UPD1A, UPD2, UPD3)

```
ZERO DEV:

DESTROYS THE DIRECTORY. AS FAR AS UPDATE IS CONCERNED, THERE IS
NOTHING ON THE DEVICE. THIS SHOULD BE DONE ON A BRAND NEW TAPE
OR CARTRIDGE SINCE UPDATE USES THE ZERO COMMAND TO RESERVE SOME
ROOM FOR USE BY THE XXDP MONITOR. VALID FOR ALL MASS STORAGE DEVICES.

.ZERO DK0:
.DIR DK0:

26-AUG-72

FILNAM.EXT      LENGTH  START   DATE

FREE FILES: 442
.

NOTE!!! WHEN THE DEVICE BEING ZEROED CONTAINS A "BAD-SECTOR" TRACK, THE UPD2, UPD3, AND
        UPD3R PROGRAMS WILL OUTPUT ONE OF THE FOLLOWING MESSAGES:

        NBBKS           ;DISK ZEROED CONTAINS NO DETECTED
                        ;BAD BLOCKS AS PER BAD-SECTOR TRACK.

        BBKS            ;DISK ZEROED CONTAINS DETECTED
                        ;BAD BLOCKS AS PER BAD-SECTOR TRACK.
```

4.17 THE BOOT AND SAVM. (UPD1, UPD1A, UPD2, UPD3)

4.17.1 BOOT DEV:

CAUSES BLOCK 0 OF DEV TO BE LOADED INTO MEMORY, STARTING AT LOC 000000.
BLOCK 0 IS ASSUMED TO HAVE A BOOT LOADER. THE PROGRAM THEN JUMPS TO
LOC 000000 TO START THE BOOT LOADER.

EXAMPLE:

```
BOOT DT0: <CR>          ;BOOTS IN THE TCDP MONITOR.

BOOT DK0: <CR>          ;BOOTS IN THE RKDP MONITOR.

BOOT MT0: <CR>          ;BOOTS IN THE TMDP MONITOR.

BOOT MM0: <CR>          ;BOOTS IN THE THDP MONITOR.

BOOT CT0: <CR>          ;BOOTS IN THE TADP MONITOR.

BOOT DX0: <CR>          ;BOOTS IN THE RXDP MONITOR.

BOOT DP0: <CR>          ;BOOTS IN THE RPDP MONITOR.

BOOT DB0: <CR>          ;BOOTS IN THE RBDP MONITOR.

BOOT DS0: <CR>          ;BOOTS IN THE RSDP MONITOR.

BOOT DM0: <CR>          ;BOOTS IN THE RMDP MONITOR.

BOOT DL0: <CR>          ;BOOTS IN THE RLDP MONITOR.

BOOT DY0: <CR>          ;BOOTS IN THE RYDP MONITOR.
```

4.17.2 SAVM DEV:

CAUSES THE FIRST 4K TO BE WRITTEN IN .SAV FORMAT (CORE IMAGE)
STARTING AT THE MONITOR CORE IMAGE BLOCK OF THE DEVICE.
THIS COMMAND IS USED TO WRITE THE XXDP MONITOR ON THE
DEVICE AS A CORE IMAGE THAT IS BOOTABLE.

```
*LOAD DK1:RKDP.BIN      ;LOAD RKDP MONITOR.
*SAVM DK0:              ;SAVE IT AS CORE IMAGE ON DK0:
```

THE SAVM COMMAND IS VALID ONLY ON RANDOM ACCESS DEVICES.

NOTE: SAVM IS NOT A DIRECTORY ENTRY IT WILL NOT SHOW
ON DIRECTORY.

4.18 THE RENAME COMMAND (UPD1, UPD2, UPD3)

```
*REN DEV:NEWNAM.EXT_DEV:OLDNAM.EXT

RENAMES THE OLD FILE. THE DEVICES MUST BE THE SAME. NOT ALLOWED
ON MAGTAPE OR CASSETTE.

*DIR DK0:

12-JAN-76
ENTRY#          FILNAM  .EXT    DATE            LENGTH          START
000001          ASD     .123    26-AUG-76       16C             000105
FREE FILES: 447


*REN DK0:123.ASD_DK0:ASD.123
*DIR    DK0:

12-JAN-76
ENTRY#          FILNAM  .EXT    DATE            LENGTH          START
000001          123     .ASD    26-AUG          16C             000105
FREE FILES: 447
*
```

4.19 PIP COMMAND (UPD1, UPD2, UPD3)

PIP IS USED TO COPY A LINKED FILE FROM ANY DEVICE THAT CAN INPUT
TO ANY DEVICE THAT CAN PERFORM OUTPUT OPERATIONS. FILE DATA IS
NOT CHECKED FOR FORMAT OR CHECKSUMS. THE OUTPUT FILE IS GIVEN TODAY'S DATE,
AND NOT THE DATE OF THE INPUT FILE.

```
PIP DEV1:FILNAM.EXT_DEV2:FILNAM.EXT

PIP PP:_PR:       (COPIES PAPER TAPE)
$PIP DK0:123.456<PR:     ;PAPER TAPE TO DISK
$PIP PP:<DK0:123.456             ;DISK TO PAPER TAPE PUNCH.
$DIR DK0:

12-JAN-76
ENTRY#          FILNAM  .EXT    DATE            LENGTH          START
000001          123     .ASD    26-AUG-72       16C             000105
000002          123     .456    26-AUG-72         3             000125
FREE FILES: 446
$
```

THE USER SHOULD MAKE SURE THAT THE OUTPUT FILE NAME DOESN'T EXIST
ALREADY ON THE OUTPUT DEVICE DIRECTORY.

```
$PIP    DK0:A_DK0:A     ;IS A NO NO.
DELOLD                   ;CAUSES THIS ERROR.  DELETE OLD FILE 1ST.
```

PIP HAS OTHER USEFUL FEATURES:

```
PIP PP:_PR:       COPIES A PAPER TAPE.
```

IMPORTANT!!!

A PROGRAM THAT HAS BEEN "PIPPED" TO A XXDP DEVICE
SHOULD BE LOADED IMMEDIATELY VIA THE "LOAD" COMMAND
TO INSURE THAT NO ERRORS HAVE OCCURRED DURING THE
"PIP" COMMAND AS THE PIP COMMAND DOES NOT CHECKSUM
INPUT DATA!

4.20 THE "FILE" COMMANDS (UPD2 AND UPD3)

UPD2 AND UPD3 INCLUDE A GROUP OF COMMANDS WHICH CAN EXECUTE ON MULTIPLE FILES WITHOUT REQUIRING THE NAME OF EACH FILE TO BE INDIVIDUALLY LISTED IN THE COMMAND STRINGS. THESE ARE THE "FILE" COMMANDS, INCLUDING FILE, FILEF, FILEL, FILEG, FILED, AND FILET. FOLLOWING THIS GENERAL DESCRIPTION, THEIR DIFFERENCES WILL BE INDIVIDUALLY EXPLAINED. NOTE THAT THE "FILE" COMMANDS IN GENERAL, CAN NOT BE USED WITH NON-DIRECTORY DEVICES (SUCH AS PR, PP, LP).

THE "FILE" COMMANDS RECOGNIZE TWO SPECIAL CHARACTERS IN THE FILE NAME AND EXTENSION. THESE CHARACTERS, THE ASTERISK (*) AND THE QUESTION-MARK (?) ALLOW A SINGLE NAME TO REFERENCE SEVERAL FILES.

NOTE THAT FILE NAMES ARE ALWAYS RECORDED AS HAVING 6 CHARACTERS, AND EXTENSIONS ALWAYS HAVE 3 CHARACTERS. THEY ARE LEFT-JUSTIFIED WITH TRAILING BLANKS ADDED, AND THE BLANKS ARE PART OF THE NAME.

BECAUSE THE "FILE" COMMANDS CAN HANDLE SEVERAL FILES PER COMMAND ISSUED, THEIR TREATMENT OF ERROR CONDITIONS SHOULD BE NOTED. IF A DEVICE ERROR OCCURS IN THE PROCESS OF FINDING A FILE (I.E. WHEN THE DIRECTORY IS REFERENCED IN THE CASE OF DISK OR DECTAPE, OR THE BLOCKS ARE SCANNED IN THE CASE OF CASSETTE OR MAGTAPE), THE "FILE" COMMAND IS ABORTED AND THE ERROR IS PRINTED. IF A DEVICE ERROR, CHECKSUM ERROR, OR END OF MEDIUM ERROR OCCURS WHILE READING A FILE (FILEL, FILEG, AND FILET ONLY) THE ERROR IS REPORTED AND THEN PROCESSING OF THE COMMAND IS CONTINUED.

THE "FILE" COMMANDS LIST THE DESCRIPTIVE INFORMATION ABOUT EACH FILE AS IT IS PROCESSED, INCLUDING FILE NAME, TRANSFER ADDRESS, AND LOCORE AND HICORE VALUES. THE /N AND /LP SWITCHES ARE INCLUDED TO ALTER THIS IF DESIRED.

THE "FILE" COMMAND DO NOT DO A FORMAT OR CHECKSUM VERIFY OF THE FILE DATA.

4.21 THE "ASTERISK" CONSTRUCTION

THE "ASTERISK" CONSTRUCTION PERMITS REFERENCE TO ALL FILES HAVING A DESIRED EXTENSION (ANY FILENAME), TO ALL FILES HAVING A DESIRED FILENAME (ANY EXTENSION), OR TO ALL FILES ON A DEVICE. ITS USE IN THE FILENAME POSITION MEANS "ANY FILENAME" AND IN THE FILE EXTENSION POSITION MEANS "ANY EXTENSION".

TO REFER TO ALL FILES HAVING A DESIRED EXTENSION (ANY FILENAME), AN ASTERISK IS TYPED FOR THE FILENAME:

DK0:*.OBJ	MEANS ALL FILES ON DISK 0 WITH A .OBJ EXTENSION
DT3:*.P11	MEANS ALL FILES ON DECTAPE 3 WITH THE EXTENSION .P11

TO REFER TO ALL FILES WITH A DESIRED FILENAME (ANY EXTENSION), AN ASTERISK IS TYPED FOR THE EXTENSION:

DK0:UPD2.*	MEANS ALL FILES ON DISK 0 WITH THE FILENAME UPD2, SUCH AS UPD2.P11, UPD2.LST, AND UPD2.DOC
DT1:SYSTST.*	MEANS ALL FILES ON DECTAPE 1 WITH THE FILENAME SYSTST, SUCH AS SYSTST.V1, SYSTST.LST, AND SYSTST.OBJ

TO REFER TO ALL FILES ON A DEVICE (ANY FILENAME, ANY EXTENSION), ASTERISKS ARE TYPED FOR BOTH THE FILENAME AND THE EXTENSION:

MT3:*.*	MEANS ALL FILES ON MAGTAPE 3
CT0:*.*	MEANS ALL FILES ON CASSETTE 0

4.22 THE "WILD CHARACTER" CONSTRUCTION

THE "WILD CHARACTER" CONSTRUCTION PERMITS REFERENCE TO ALL FILES WHOSE FILE NAMES DIFFER IN SPECIFIC CHARACTER POSITIONS. WHEN SEARCHING FOR FILES CORRESPONDING TO THE NAME IN THE COMMAND STRING, ANY CHARACTER IS ACCEPTED AS MATCHING A QUESTION MARK. FOR EXAMPLE:

DK0:UPD?.DOC	MEANS ANY FILE WHOSE NAME BEGINS WITH "UPD", HAS ANY CHARACTER NEXT (INCLUDING A BLANK) AND THEN TWO BLANKS, WITH A .DOC EXTENSION. UPD1.DOC AND UPD2.DOC WOULD BOTH QUALIFY.
DT1:TEST??.P11	WOULD INCLUDE ANY FILES ON DT1 WHOSE FILENAMES BEGIN WITH "TEST" AND WHOSE EXTENSIONS ARE .P11, SUCH AS TEST2.P11, TEST34.P11, AND TEST.P11.
CT1.SYSTST.V?	INCLUDES ANY FILE ON CASSETTE 1 WHOSE FILENAME IS "SYSTST" AND WHOSE EXTENSION BEGINS WITH "V" AND ENDS WITH A BLANK. THUS, SYSTST.V1 AND SYSTST.VA WOULD QUALIFY, BUT SYSTST.V14 AND SYSTST.LST WOULD NOT.

4.23 THE FILE COMMAND (UPD2 AND UPD3)

THE FILE COMMAND IS USED TO DO BULK TRANSFERS FROM ONE DEVICE TO ANOTHER. IT IS SIMILAR TO A PIP COMMAND EXCEPT THAT IT CAN UTILIZE THE "ASTERISK" AND "WILD CHARACTER" CONSTRUCTIONS. IF A FILE OF THE SAME NAME ALREADY EXISTS ON THE OUTPUT DEVICE, THE FILE COMMAND (UNLIKE THE PIP COMMAND) WILL DELETE THE OLD FILE. NOTE ALSO THAT THE FILE COMMAND CAN TRANSFER BOTH LINKED AND CONTIGUOUS (CORE-IMAGE) FILES. THE OUTPUT FILE(S) IS GIVEN THE SAME DATE AS THE INPUT FILE(S).

```
FILE DEV:<DEV:FILNAM.EXT                    ;COMMAND FORMAT
```

WHERE THE DEVICE NAME ON THE LEFT IS THE OUTPUT DEVICE AND THAT ON THE RIGHT IS THE INPUT DEVICE.

4.24 THE FILEF COMMAND (UPD2 AND UPD3)

THE FILEF COMMAND IS USED TO DO FAST TRANSFERS ONTO ALL DIRECTORY DEVICES. FOR MAG TAPE LOGICAL END OF TAPE IS FOUND AND ALL THE REQUESTED FILES ARE TRANSFERRED SEQUENTIALLY ONTO THE TAPE STARTING AT THAT POINT. THIS FAST TRANSFER COMMAND ELIMINATES THE CHECK OF THE TAPE DIRECTORY WHICH IS MADE BEFORE EACH FILE TRANSFER IF THE FILE COMMAND IS USED.

FOR RANDOM ACCESS DEVICES THE FILE IS TRANSFERED TO THE FIRST AVAILABLE SPACE ON THE DEVICE.

```
FILEF   DEV:<DEV:FILNAM.EXT                 ,COMMAND FORMAT
```

4.25 THE FILED COMMAND (UPD2 AND UPD3)

THE FILED COMMAND DELETES THE FILES NAMED FROM THE DEVICE'S DIRECTORY.

```
FILED   DEV:FILNAM.EXT                      ;COMMAND FORMAT
```

UPD2 AND UPD3 NOW PERMIT THE USE OF THE DEL(ETE) COMMAND WITH * AND WILD CHARACTER FILENAME CONSTRUCTION. EXAMPLE:

```
DEL DK0:*.BIN                       ;DELETES ALL FILES IN DK0:  WITH .BIN
                                    ;EXTENSION.
```

CAUTION!!! THE UPD2 AND UPD3 PROGRAMS DO NOT REQUIRE VERIFICATION OF A MASS DELETION COMMAND. THE USER MUST BE CAREFUL NOT TO SPECIFY A DELETE THAT HE DOES NOT REALLY MEAN TO OCCUR. IF IT SHOULD, TYPING CONTROL C WILL ABORT THE COMMAND AT THE EARLIEST OPPORTUNITY.

4.26 THE FILEL COMMAND (UPD2 AND UPD3)

THE FILEL COMMAND SEQUENTIALLY LOADS INTO CORE EACH FILE REFERENCED. IT ASSUMES THAT ALL REFERENCED FILES ARE ABS FORMAT (IF NOT A CKSMER OR EOM ERROR WILL OCCUR). ITS PURPOSE IS TO SHOW THAT ALL ABS FORMATTED FILES CAN BE CORRECTLY LOADED (CHECKS FOR DEVICE AND CHECKSUM ERRORS). IF AN ERROR OCCURS, IT WILL IDENTIFY THE TYPE OF ERROR AND THE DEVICE.

```
FILEL   DEV:FILNAM.EXT                  ;COMMAND FORMAT
```

THE LOAD COMMAND MAY ALSO BE USED IN UPD2 AND UPD3 TO PERFORM THE SAME FUNCTIONS AS THE FILEL COMMAND.

4.27 THE FILEG COMMAND (UPD2 AND UPD3)

THE FILEG (FILE GET) COMMAND IS SIMILAR TO THE FILEL COMMAND EXCEPT THAT IT LOADS AND CHECKS CONTIGUOUS (CORE-IMAGE) FILES INSTEAD OF ABS FORMAT FILES. DEVICE ERRORS AND SIZE ERRORS WILL BE REPORTED.

```
FILEG   DEV:FILNAM.EXT                  ;COMMAND FORMAT
```

THE GET COMMAND MAY ALSO BE USED IN UPD2 AND UPD3 TO PERFORM THE SAME FUNCTIONS AS THE FILEG COMMAND.

4.28 THE FILET COMMAND (UPD2 AND UPD3)

THE FILET COMMAND TESTS ALL FILES NAMED BY READING THEM INTO A BUFFER TO MAKE CERTAIN THAT NO DEVICE ERRORS OCCUR. ANY DEVICE ERRORS ARE LISTED AS THEY OCCUR.

```
FILET   DEV:FILNAM.EXT                  ;COMMAND FORMAT
```

WHEN USED IN THE UPD3R PROGRAM, THE FILET OUTPUTS A MESSAGE INDICATING THE TOTAL NUMBER OF FILE BLOCKS PROCESSED.

```
#OF BLOCKS : XXXXX
```

THIS FEATURE IS USEFUL TO RELEASE ENGINEERING IN DETERMINING THE TOTAL NUMBER OF BLOCKS WRITTEN IN A CASSETTE OR MAGTAPE, AS THEY MUST NOT USE UP MORE THAN 75 PERCENT OF THE MEDIUM.

4.29 THE /LP AND /N SWITCHES (UPD2 AND UPD3 ONLY)

THE "FILE" COMMANDS NORMALLY CAUSE PRINTING OF THE NAMES OF THE FILES CHECKED, THEIR TRANSFER ADDRESSES, AND LOCORE AND HICORE VALUES, ON THE CONSOLE TERMINAL. THE /LP SWITCH CAUSES THIS INFORMATION TO BE OUTPUT ON THE LINE PRINTER INSTEAD. THE /N SWITCH INHIBITS PRINTING OF THIS INFORMATION, SO THAT ONLY ERROR PRINTOUTS ARE OUTPUT. SWITCHES MUST NOW BE SPECIFIED AT END OF THE COMMAND STRING.

```
FILET DK0:*.*/LP          ;TEST ALL FILES ON DK0 AND PRINT
                          ;THE FILE INFORMATION AND ERROR
                          ;INFORMATION ON THE LINE PRINTER

FILEG DT1:*.SA?/N         ;DO A CORE-IMAGE LOAD OF ALL THE
                          ;.SAV FILES ON DECTAPE 1,
                          ;REPORTING ONLY ERROR INFORMATION

FILEL /N MT2:*.BIN/LP     ;LOAD ALL .BIN FILES FROM MAGTAPE 2,
                          ;REPORTING ONLY ERROR INFORMATION
                          ;ON THE LINE PRINTER

DEL DK0:*.TXT/LP          ;DELETE ALL .TXT FILES FROM DK0: AND
                          ;PRINT DELETED FILES ON LINE PRINTER.
```

4.30 THE "EOT" COMMAND (UPD2 AND UPD3)

THE "EOT" COMMAND IS PROVIDED AS A MEANS OF PLACING AN "END-OF-TAPE" MARK OR SENTINEL FILE AT A SELECTED SPOT ON MAGTAPE OR CASSETTE. APPLICATIONS OF THIS COMMAND INCLUDE REPLACING AN "EOT" MARK WHEN IT HAS BEEN ACCIDENTALLY DESTROYED, OR WHEN THE USER WISHES TO DELETE FILES AT THE END OF THE MEDIUM, AND STILL BE ABLE TO USE THE SPACE TAKEN UP BY THOSE DELETED FILES.

THE PROCEDURE TO BE USED IS AS FOLLOWS:

A. POSITION THE MAGTAPE BY PERFORMING A FILET COMMAND ON THE FILE PRECEDING THE SPOT WHERE THE "EOT" IS TO BE PLACED. IN PRACTICE, IF AN "EOT" HAS BEEN LOST, THE USER SHOULD FILET THE NEXT TO THE LAST FILE, SINCE THE LAST FILE MAY BE UNRECOVERABLE.

B. PERFORM AN "EOT" COMMAND.

EXAMPLE:

```
*FILET MT0:ZQRADO.BIN<CR>      ,READS FILE ZQRADO BIN AND STOPS.
*EOT<CR>                       ,WRITES EOT
```

4.31 THE TEXT COMMAND (UPD2 AND UPD3)

UPD2 AND UPD3 INCLUDE THE FACILITY TO EXECUTE A SEQUENCE OF COMMANDS CONTAINED IN AN ASCII TEXT FILE. THIS ASCII TEXT FILE IS CREATED VIA THE TEXT COMMAND. ALSO SEE CHAPTER 4. XTECO TEXT EDITOR.

```
TEXT DEV:FILNAM.TXT                    ;COMMAND FORMAT
```

WHEN THE TEXT COMMAND IS ISSUED, UPD2 AND UPD3 OPEN THE NAMED FILE FOR OUTPUT AND RESPONDS WITH A QUOTATION MARK (") TO INDICATE ITS READINESS TO ACCEPT TEXT. ANY ASCII CHARACTER (EXCEPT CONTROL C AND RUBOUT) WILL BE ACCEPTED AS INPUT TO THE TEXT FILE. EACH LINE OF INPUT IS TERMINATED BY A CARRIAGE-RETURN. AFTER THE LAST LINE HAS BEEN ENTERED, THE FILE IS CLOSED WITH A CONTROL Z (^Z). CONTROL C (^C) WILL ABORT TEXT MODE, RETURNING TO COMMAND MODE AND CLOSING THE OUTPUT FILE. RUBOUT CAN BE USED TO DELETE CHARACTERS

ON THE CURRENT LINE (BUT NOT ON PRECEDING LINES).

THREE CHARACTERS, THE POUND SIGN (#), THE SEMICOLON (;), AND THE DOLLAR SIGN ($), HAVE SPECIAL SIGNIFICANCE IN THE TEXT FILE. THE # SIGN AND ; ARE USED TO START A COMMENT WHICH IS TO BE PRINTED DURING COMMAND FILE EXECUTION. THE $ SIGN IS USED TO START A COMMENT WHICH IS TO BE PRINTED AND FOLLOWED BY A HALT DURING COMMAND FILE EXECUTION (SUCH AS "$PRESS CONT WHEN READY").

4.32 THE PRINT COMMAND (UPD2 AND UPD3)

THE PRINT COMMAND OUTPUTS A FILE ON THE LINE PRINTER. IT IS USED TO PRINT TEXT FILES, AND WILL OUTPUT TO THE LINE PRINTER. AFTER THE TEXT FILE IS PRINTED THE PROGRAM OUTPUTS 10 CARRIAGE RETURNS AND LINE FEEDS TO SIMULATE A FORM FEED. NOTE THAT BOTH PRINT AND TYPE COMMANDS ACCEPT * AND WILD CHARACTER CONSTRUCTION IN FILENAMES, SO THAT MULTIPLE TEXT FILES MAY BE PRINTED WITH ONE COMMAND.

```
PRINT DEV:FILNAM.EXT                   ;COMMAND FORMAT

PRINT DEV:*.TXT
```

WHERE DEV IS THE SOURCE DEVICE ON WHICH THE FILE RESIDES.

NOTE THAT NO CHECK IS MADE OF FILE PRINTABILITY

4.33 THE TYPE COMMAND (UPD2 AND UPD3)

SAME AS THE PRINT COMMAND EXCEPT THAT IT OUTPUTS TO THE CONSOLE TERMINAL INSTEAD OF TO THE LINE PRINTER.

```
TYPE DEV:FILNAM.EXT                    ;COMMAND FORMAT
```

4.34 THE DO COMMAND (UPD2 AND UPD3)

THE DO COMMAND IS USED TO CAUSE THE EXECUTION OF A COMMAND FILE.
THE FILE MUST BE ON ONE OF THE XXDP STORAGE MEDIA (DECTAPE, MAGTAPE,
CASSETTE, OR DISK). THE FILE IS EXECUTED LINE BY LINE, AND MUST
BE TERMINATED BY A Z (CONTROL Z). EXECUTABLE FILES ARE CREATED
VIA THE TEXT COMMAND, OR VIA THE XTECO TEXT EDITOR PROGRAM (SEE CHAPTER 4.)
FOR NOTES ON THE FILE'S FORMAT AND THE USE OF SPECIAL CHARACTERS,
SEE THE PRECEDING TEXT COMMAND DESCRIPTION.

```
DO DEV:FILNAM.EXT                        ;COMMAND FORMAT
```

4.35 THE ASG (ASSIGN) COMMAND (UPD2 AND UPD3)

THE ASG (ASSIGN) COMMAND ALLOWS THE USE OF LOGICAL DEVICE NAMES IN
COMMAND FILES. ALLOWED LOGICAL DEVICE NAMES ARE 1,2,3,4, AND SYS.
A COMMAND FILE MAY USE A LOGICAL NAME SUCH AS "1" INSTEAD OF
SPECIFYING, FOR EXAMPLE, DK0 OR DK1. THEN, BEFORE EXECUTING
THE COMMAND FILE, THE USER CAN ASSIGN THE DESIRED PHYSICAL DEVICE
TO THE LOGICAL NAME, PERMITTING USE OF ANY AVAILABLE UNIT.

```
ASG PHYSICAL DEV = LOGICAL DEV           ;COMMAND FORMAT
```

REVERSAL OF PHYSICAL AND LOGICAL DEVICE NAMES IN THE COMMAND STRING
RESULTS IN "INVDEV" ERROR MESSAGE. THE COMMAND IS NOT PERFORMED.

```
ASG DK1: = 2:    ;ASSIGNS DISK 1 TO LOGICAL DEVICE "2"

ASG DT3: = SYS:  ;ASSIGNS DECTAPE 3 TO LOGICAL DEVICE "SYS"
```

4.36 THE FILCMP COMMAND (UPD3R)

THE FILCMP COMMAND IS USED TO COMPARE TWO FILES WHICH ARE THE SAME
BUT ON DIFFERENT XXDP MEDIUMS. IT CAN UTILIZE
THE "ASTERISK" AND "WILD CHARACTER" CONSTRUCTIONS.

```
FILCMP DEV:<DEV:FILNAM.EXT      ;COMMAND FORMAT.
```

WHERE THE FILE ON THE DEVICE ON THE RIGHT IS COMPARED TO THE FILE
OF THE SAME NAME ON THE DEVICE ON THE LEFT.

```
EXAMPLE;
FILCMP DK1:<DK0:*.*     ;COMPARES ALL FILES ON DISK 0 TO
                        ;ALL FILES ON DISK 1.
```

FOR ERRORS UNIQUE TO THE FILCMP COMMAND SEE CHAPTER 5.1.

4.37 THE PATCH COMMAND (UPD2 AND UPD3)

THE PATCH COMMAND ENABLES THE USER TO PATCH A PROGRAM ON ANY DIRECTORY-
ORIENTED (RANDOM ACCESS) XXDP SUPPORTED DEVICE. NO OUTPUT
DEV: FILE SPECIFICATION IS REQUIRED OR PERMITTED. THE INPUT DEVICE
IS ASSUMED TO BE THE DESIRED OUTPUT DEVICE.

THE FILE(S) TO BE PATCHED MUST BE IN ABS FORMAT
BINARY FILE. THE PATCH ROUTINE DOES NOT CHECK IN ADVANCE
FOR CORRECT FILE FORMAT. THE FOLLOWING EXTENSION
ARE FOR XXDP ABS FORMAT FILES: .BIN, .BIC, .MPG.

CARRIAGE-RETURN OR LINE-FEED ARE THE ONLY CHARACTERS WHICH MAY BE USED
FOR TERMINATING A TYPED ENTRY. THE LINE-FEED MAY BE THOUGHT OF AS AN
"ADVANCE" KEY, WHICH WILL GO TO THE NEXT ADDRESS. THE RUBOUT KEY MAY
BE USED TO CORRECT TYPING MISTAKES MADE ON INPUT. ALL ADDRESSES ENTERED MUST
BE EVEN. IF AN ADDRESS IS TYPED (IN RESPONSE TO A PROMPT) WHICH IS ODD, THE
PROMPT WILL BE RE-ASKED.

IF AN ADDRESS IS TYPED WHICH IS NOT WITHIN THE CORE LOAD LIMITS OF THE FILE
BEING OPERATED UPON, THE UNKNOWN CONTENTS OF THE SPECIFIED ADDRESS WILL
BE INDICATED BY "XXXXXX". THE PROGRAM WILL THEN GIVE THE USUAL "?"
PROMPT, ASKING IF MODIFICATION IS DESIRED.

IN RESPONSE TO THE "ADDR?" PROMPT, IF A CARRIAGE-RETURN OR A
LINE-FEED IS TYPED AS THE ONLY THING ON THE INPUT LINE, THE EXIT
SEQUENCE WILL BE ENTERED, AT SUCH TIME, THE USER IS ASKED TO WRITE-ENABLE
THE OUTPUT DEVICE AND CONFIRM THE FACT THAT THE PATCHES SHOULD BE ENTERED
INTO THE SPECIFIED FILE.

IF A FILE IS MODIFIED BY THE USE OF THE "PATCH" COMMAND, THE DATE AND
LENGTH OF THE FILE OPERATED UPON ARE UPDATED IN THE DEVICE DIRECTORY AS
REQUIRED.

THE PATCH COMMAND CREATES A BLOCK CONTAINING OVERLAY ADDRESSES/CORRECTED
INFORMATION. THIS BLOCK IS LINKED TO THE FILE CONTAINING THE PROGRAM
BEING PATCHED.

IF THE FILE BEING PATCHED CONTAINS REPRESENTATIONS OF ISOLATED SINGLE-BYTE
DATA, FOR EXAMPLE THOSE GENERATED BY THE FOLLOWING ASSEMBLY
CODE SEQUENCES;

```
A.      .=24
        .BYTE 120
        .EVEN           ;GENERATES ONLY 1 BYTE OF DATA

B.      .=413
        .BYTE-1
        .EVEN           ;GENERATES ONLY 1 BYTE OF DATA

C.      .ODD
        .BYTE 6
        .=.+1           ;GENERATES ONLY 1 BYTE OF DATA
```

THE CONTENTS OF THE DATA BYTE REPRESENTED IN THE FILE WILL BE PROPERLY
REPORTED IF EXAMINED USING THE "PATCH" COMMAND, BUT THE CONTENTS
OF THE ADJACENT DATA BYTE WHICH OCCUPIES THE SAME WORD ADDRESS WILL BE
REPORTED TO BE 0'S, SINCE IT IS NOT REPRESENTED IN THE FILE.
FOR EXAMPLE, IN THE CASE OF A ABOVE,

```
        ADDR? 24 <CR>
        000024 000120
                ---- NOTE THAT THE CONTENTS OF THE
                     UPPER BYTE ARE ACTUALLY UNKNOWN.
```

AND B

```
        ADDR? 412 <CR>
        000   177400
                ----NOTE UNKNOWN DATA IN LOW BYTE
                     REPRESENTED BY 0'S.
```

5. ERRORS

INVCMD	INVALID COMMAND. CHECK COMMAND JUST GIVEN.
INVDEV	INVALID DEVICE SPECIFIED FOR COMMAND GIVEN.
INVADR	INVALID ADDRESS. MUST BE EVEN, WITHIN EXISTING LOCORE AND HICORE LIMITS, AND MUST NOT BE WITHIN UPDATE PROGRAM.
INVNAM	INVALID FILE NAME. NO SPECIAL CHARACTERS ALLOWED. A THROUGH Z, AND 0 THROUGH 9 ARE ONLY VALID CHARACTERS. ALSO OCCURS IF * OR WILD CHARACTER CONSTRUCTION FILENAMES ARE SPECIFIED TO A COMMAND THAT DOES NOT ALLOW IT.
NEXFIL	NON-EXISTENT FILE. FILE DOES NOT EXIST IN DEVICE DIRECTORY.
DELOLD	DELETE OLD FILE BEFORE GIVING COMMAND THAT WOULD CREATE FILE WITH SAME NAME.
DEVERR	DEVICE ERROR ON EITHER INPUT OR OUTPUT DEVICE. CHECK THAT OUTPUT DEVICE IS WRITE-ENABLED.
NOTRDY	PAPER TAPE DEVICE IS NOT READY. MAKE IT READY.
CKSMER	CHECKSUM ERROR DURING "LOAD" COMMAND.
EOM	END OF MEDIUM. OCCURS DURING INPUT OPERATIONS WHEN THE PROGRAM ATTEMPTS TO INPUT AND THE FILE IS AT AN END. SERIOUS PROBLEM. FILE IN STORAGE IS PROBABLY WIPED OUT. REFER TO CHAPTER 4 FOR MEDIUM TESTING COMMANDS.
DEVFUL	DEVICE FULL. APPLIES TO DECTAPE AND DISK. NO MORE FILE STORAGE AVAILABLE. DELETE UNNECESSARY FILES AND TRY AGAIN, OR USE ANOTHER MEDIUM.
INVCOR	HIGH CORE LIMIT LOWER THAN LOWER CORE LIMIT. CORRECT CORE LIMITS. OCCURS DURING DUMP COMMAND.
DIRERR	INVALID NAME IN DEVICE DIRECTORY.
DELERR	BIT MAP ERROR DURING DELETE OPERATION ON DECTAPE OR DISK. NOT USUAL UNLESS MEDIUM HAS BEEN WIPED OUT. TRANSFER FILES TO OTHER MEDIUM. (SEE CHAPTER 4.).
POFLOW	PROGRAM TOO LARGE TO LOAD WITHIN EXISTING CORE SPACE.
INVSW	INVALID SWITCH SPECIFIED IN COMMAND STRING.
DUMP ERROR	ACT MODE ONLY (SEE CHAPTER 7). OCCURS DURING DUMP COMMAND WHEN DATA DUMPED ON OUTPUT DEVICE DOES NOT MATCH DATA IN CORE.

5.1 ERRORS UNIQUE TO THE FILCMP COMMAND

UNEQUAL FILE TYPES	INDICATES THE TWO FILES BEING COMPARED ARE NOT OF SIMILAR STRUCTURE.
UNEQUAL FILE SIZES	INDICATES THE TWO FILES BEING COMPARED ARE NOT THE SAME SIZE.
SCRATCH FILE SHORTER THAN MASTER FILE	THE SCRATCH FILE IS THE FILE ON THE DEVICE WHICH IS ON THE LEFT OF THE BACK ARROW IN THE COMMAND STRING.
SCRATCH FILE LONGER THAN MASTER FILE	THE SCRATCH FILE WHICH IS ON THE LEFT OF THE BACK ARROW IS LONGER THAN THE FILE ON THE RIGHT.
BLOCK COMPARE ERROR XTH BLOCK, YTH BYTE	THIS INDICATES THERE WAS AN ERROR IN THE COMPARE, X AND Y INDICATE THE BLOCK NUMBER AND BYTE NUMBER WHERE THE ERROR OCCURRED.

6. UPDATING XXDP MEDIA

UPDATING XXDP MEDIA CONSISTS OF:

A. PATCHING EXISTING PROGRAMS (DEPO), OR
B. REPLACING PROGRAMS WITH NEWER VERSIONS, OR
C. ADDING NEW PROGRAMS.

WHEN FIRST BECOMING ACQUAINTED WITH THE USE OF THE UPDATE PROGRAMS THE USER SHOULD MAKE EXTRA SURE THAT A BACKUP FOR THE MEDIUM TO BE MODIFIED EXISTS, IN ORDER TO BE ABLE TO RECOVER FROM FATAL ERRORS. (ZEROING THE MEDIUM, DELETING THE WRONG FILE, ETC.).

6.1 PATCHING EXISTING PROGRAMS

THERE ARE TWO METHODS TO PATCH A PROGRAM IN AN XXDP MEDIUM:

A. USE OF THE "PATCH" COMMAND TO CREATE AN OVERLAY BLOCK CONTAINING THE CORRECTIONS. THIS METHOD CAN BE USED WITH ANY SIZE MEMORY LARGE ENOUGH TO HOLD THE UPD2 OR UPD3 PROGRAM. THE DIAGNOSTIC PROGRAM BEING PATCHED IS NOT BROUGHT INTO MEMORY.

B. USE OF THE "LOAD/MOD/DUMP" COMMANDS TO BRING THE DIAGNOSTIC PROGRAM INTO MEMORY, INSERT THE CHANGES, AND WRITE THE NEW VERSION BACK TO THE XXDP MEDIUM.

IT IS IMPORTANT WHEN IMPLEMENTING DEPO'S THAT THE NAME OF THE PROGRAM REFLECT THE DEPO LEVEL OF THE PROGRAM. SEE APPENDIX D PROGRAM NAMING CONVENTIONS.

6.2 REPLACING PROGRAMS WITH NEWER VERSIONS, OR ADDING NEW PROGRAMS

TO REPLACE A PROGRAM, OR TO ADD A NEW ONE.

A. DELETE OLD PROGRAM IF REPLACING IT,
B. LOAD NEW PROGRAM INTO MEMORY,
C. DUMP PROGRAM ONTO DEVICE.

EXAMPLE 1:

```
.DEL DTO:DOSA1.BIN              (DELETE OLD PROGRAM)
.LOAD PR:                       (LOAD NEW PROGRAM )
.DUMP DTO:DOSB0.BIN             (STORE NEW PROGRAM)
.LOAD DTO:DOSB0.BIN             (LOAD NEW PROGRAM)
.START 200                      (TRY NEW PROGRAM)
```

EXAMPLE 2:

```
DEL CTO:DOSA1.BIN       ;DELETES OLD PROGRAM
LOAD PR:                ;LOADS NEW PROGRAM FROM PAPER TAPE.
DUMP CTO:DOSB0.BIN      ;ADDS NEW PROGRAM
LOAD CTO:DOSB0.BIN      ;CHECKS THAT PROGRAM LOADS CORRECTLY.
```

NOTE: DELETING A PROGRAM FROM CASSETTE OR MAGTAPE DOES NOT PHYSICALLY REMOVE THE PROGRAM FROM THE MEDIUM. IT STILL TAKES UP SPACE. TO CLEAN UP THE CASSETTE OR MAGTAPE, IT MUST BE COPIED VIA ITS XXDP MONITOR'S COPY ROUTINE, WHICH COPIES ONLY "GOOD" FILES.

```
.PIP DTO:OVLY.BIN_PR:           (PIP TO DTO: FROM PR: )
.LOAD DTO:OVLY.BIN              (LOAD OVERLAY)
```

RELOADING OF A PROGRAM THAT HAS BEEN "PIPPED" DIRECTLY TO A DEVICE IS IMPORTANT, TO INSURE THAT NO READING ERRORS HAVE OCCURRED. THE PIP AND FILE COMMANDS DO NOT CHECKSUM INPUT DATA.

6.3 GENERATING A XXDP MEDIUM

IT MAY BE DESIRABLE TO CREATE A CUSTOM MADE MEDIUM CONTAINING ONLY THOSE PROGRAMS REQUIRED TO TEST A PARTICULAR SYSTEM. AS AN EXAMPLE, SUCH A MEDIUM COULD CONTAIN:

A. PROCESSOR TESTS
B. MEMORY TESTS
C. I/O PROGRAMS FOR THAT SYSTEM

WITH SUCH A MEDIUM, THE ENTIRE SYSTEM COULD BE TESTED USING THE CHAIN MODE OF OPERATION, WITHOUT HAVING TO SWITCH DECTAPES, OR CASSETTES.

THE PROCEDURES FOR GENERATING A NEW MEDIUM FOLLOW.

6.3.1 CREATING A NEW XXDP DECTAPE

```
*ZERO DT1:                      (ZERO OUT NEW DECTAPE)
*LOAD DT0:TCDP.BIN              (GET DECTAPE MONITOR)
*SAVM DT1:                      ;SAVE TCDP AS BOOTABLE CORE IMAGE.
*DUMP DT1:TCDP.BIN              (SAVE MONITOR AS A FILE)
*LOAD DT0:UPD1.BIN              (LOAD UPD1 PROGRAM)
*DUMP DT1:UPD1.BIN              (COPY OF UPD1 GOES ON NEW TAPE)
LOAD DT0:UPD2.BIN               (GET UPD2 PROGRAM)
DUMP DT1:UPD2.BIN               (COPY OF UPD2 GOES ON NEW TAPE)
```

FROM THIS POINT ON, THE DESIRED PROGRAMS ARE TRANSFERRED FROM THE OTHER TCDP DECTAPES TO THE NEW DECTAPE, USING THE PIP, AND THE LOAD AND DUMP COMMANDS AS REQUIRED BY TYPE OF FILE

IMPORTANT ABS FORMAT FILES (.BIN,.BIC) CAN BE TRANSFERRED BY MEANS OF THE PIP COMMAND. CORE IMAGE FILES (.SAV) MUST NOT. TO TRANSFER A CORE IMAGE FILE, THE GET AND SAVE COMMANDS MUST BE USED AS FOLLOWS:

```
 C
*GET DT0:A.SAV
*SAVE DT1:A.SAV
```

CORE IMAGE FILES MUST BE TRANSFERRED TO THE NEW DECTAPE FIRST, SINCE THEY REQUIRE CONTIGUOUS BLOCK ALLOCATION. WAITING UNTIL OTHER LINKED FILES HAVE BEEN TRANSFERRED MAY RESULT IN THE LACK OF SUFFICIENT CONTIGUOUS BLOCKS TO STORE A CORE IMAGE FILE.

AFTER THE NEW DECTAPE IS COMPLETED, ALL PROGRAMS SHOULD BE LOADED FROM IT, TO INSURE THEY HAVE BEEN STORED CORRECTLY. ADDITIONALLY, THE DECTAPE SHOULD BE DUPLICATED, TO PROVIDE A BACKUP.

6.3.2 CREATING A NEW XXDP DECPACK

A. MOUNT THE "NEW" DISK ON DRIVE 1
B. MOUNT THE "OLD" DISK ON DRIVE 0.
C. PERFORM THE FOLLOWING COMMANDS:

```
ZERO DK1:                    ;ZERO NEW DISK.
LOAD DK0:RKDP.BIN
SAVM DK1:
DUMP DK1:RKDP.BIN
LOAD DK0:UPD1.BIN
DUMP DK1:UPD1.BIN
LOAD DK0:UPD2.BIN
DUMP DK1 UPD2.BIN
```

6.3.3 CREATING A NEW XXDP MAGTAPE

A. MOUNT "NEW" MAGTAPE ON DRIVE 1
B. MOUNT "OLD" MAGTAPE ON DRIVE 0
C. PERFORM THE FOLLOWING COMMANDS·

FOR A TM11

```
ZERO MT1:
LOAD MT0:THDP.BIN
SAVE MT1:THDP.SAV
LOAD MT0:TMDP.BIN
SAVE MT1:TMDP.SAV
LOAD MT0:THDP.BIN
DUMP MT1:THDP.BIN
LOAD MT0:TMDP.BIN
DUMP MT1:TMDP.BIN
LOAD MT0:UPD1.BIN
DUMP MT1:UPD1.BIN
LOAD MT0:UPD2.BIN
DUMP MT1:UPD2.BIN
```

FOR A TM02

```
ZERO MM1:
LOAD MM0:THDP.BIN
SAVE MM1:THDP.SAV
LOAD MM0:TMDP.BIN
SAVE MM1:TMDP.SAV
LOAD MM0:THDP.BIN
DUMP MM1:THDP.BIN
LOAD MM0:TMDP.BIN
DUMP MM1:TMDP.BIN
LOAD MM0:UPD1.BIN
DUMP MM1:UPD1.BIN
LOAD MM0:UPD2.BIN
DUMP MM1:UPD2.BIN
```

6.3.4 CREATING NEW XXDP CASSETTE

THE TADP CASSETTE ITSELF DOES NOT CONTAIN DIAGNOSTIC PROGRAMS. HOWEVER, A NEW CASSETTE COULD BE BUILT CONTAINING THE REQUIRED PROGRAMS AND TO BE RUN UNDER TADP.

A. MOUNT THE TADP CASSETTE ON CT0:
B. MOUNT A "SCRATCH" CASSETTE ON CT1:
C. PERFORM THE FOLLOWING COMMANDS:

```
ZERO CT1:
LOAD CT0:TALDRB.BIN
SAVE CT1:TALDRB.SYS     ;PLACES CASSETTE LOADER IN IMAGE FORM
DUMP CT1:TALDRB.BIN     ;PLACES CASSETTE LOADER IN FILE FORM.
```

LOAD AND DUMP THE REQUIRED PROGRAMS. BEFORE EACH PROGRAM IS LOADED THE CASSETTE CONTAINING THE PROGRAM MUST BE LOADED IN CT0· EXAMPLE

```
LOAD CT0:GTP.BIN
DUMP CT1:GTP.BIN
```

6.3 5 CREATING A NEW XXDP DISKETTE

A. MOUNT THE "NEW" DISKETTE ON DRIVE 1
B. MOUNT THE "OLD" DISKETTE ON DRIVE 0.
C. PERFORM THE FOLLOWING FUNCTIONS

```
ZERO DX1:
LOAD DX0:  RXDP.BIN
SAVM DX1:
DUMP DX1:  RXDP.BIN
LOAD DX0:  UPD1.BIN
DUMP DX1:  UPD1.BIN
LOAD DX0:  UPD2.BIN
DUMP DX0:  UPD2.BIN
```

6.3.6 CREATING A NEW XXDP RP03 DISK

A. MOUNT THE "NEW" DISK PACK ON DRIVE 1.
B. MOUNT THE "OLD" DISK PACK ON DRIVE 0.
C. PERFORM THE FOLLOWING FUNCTIONS:

```
ZERO DP1:
LOAD DP0: RPDP.BIN
SAVM DP1:
DUMP DP1: RPDP.BIN
LOAD DP0: UPD1.BIN
DUMP DP1: UPD1.BIN
LOAD DP0: UPD2.BIN
DUMP DP1: UPD2.BIN
```

6.3.7 CREATING A NEW XXDP RP04 DISK

A. MOUNT THE "NEW" DISK PACK ON DRIVE 1.
B. MOUNT THE "OLD" DISK PACK ON DRIVE 0.
C. PERFORM THE FOLLOWING FUNCTIONS:

```
ZERO DB1:
LOAD DB0: RBDP BIN
SAVM DB1:
DUMP DB1: RBDP.BIN
LOAD DB0: UPD1.BIN
DUMP DB1: UPD1.BIN
LOAD DB0: UPD2.BIN
DUMP DB1: UPD2.BIN
```

6.3 8 CREATING THE NEW XXDP RS03 DISK

A. SELECT THE "NEW" DISK AS DRIVE 1.
B. SELECT THE "OLD" DISK AS DRIVE 0.
C. PERFORM THE FOLLOWING FUNCTIONS:

```
ZERO DS1:
LOAD DS0: RSDP.BIN
SAVM DS1:
DUMP DS1: RSDP.BIN
LOAD DS0: UPD1.BIN
DUMP DS1: UPD1.BIN
LOAD DS0: UPD2.BIN
DUMP DS1: UPD2.BIN
```

6.3.9 CREATING THE NEW XXDP RK06 DISK

A. SELECT THE "NEW" DISK AS DRIVE 1.
B. SELECT THE "OLD" DISK AS DRIVE 0.
C. PERFORM THE FOLLOWING FUNCTIONS.

```
ZERO DM1:
LOAD DM0: RMDP.BIN
SAVM DM1:
DUMP DM1: RMDP.BIN
LOAD DM0: UPD1.BIN
DUMP DM1: UPD1.BIN
LOAD DM0: UPD2.BIN
DUMP DM1: UPD2.BIN
```

6.3.10 CREATING THE NEW XXDP RL01 DISK

A. SELECT THE "NEW" DISK AS DRIVE 1.
B. SELECT THE "OLD" DISK AS DRIVE 0.
C. PERFORM THE FOLLOWING FUNCTIONS

```
ZERO DL1:
LOAD DL0: RLDP.BIN
SAVM DL1:
DUMP DL1: RLDP.BIN
LOAD DL0: UPD1.BIN
DUMP DL1: UPD1.BIN
LOAD DL0: UPD2.BIN
DUMP DL1: UPD2.BIN
```

6.3.11 CREATING THE NEW XXDP RX02 DISKETTE

A. MOUNT THE "NEW" DISKETTE ON DRIVE 1
B. MOUNT THE "OLD" DISKETTE ON DRIVE 0.
C. PERFORM THE FOLLOWING FUNCTIONS:

```
ZERO DY1:
LOAD DY0: RYDP.BIN
SAVM DY1:
DUMP DY1: RYDP.BIN
LOAD DY0: UPD1A.BIN
DUMP DY1: UPD1A.BIN
LOAD DY0: UPD3.BIN
DUMP DY1: UPD3.BIN
```

6.3.12 CREATING A XXDP MEDIUM - COMMON PROCEDURE

ONCE THE MONITOR HAS BEEN SAVED ON THE MEDIUM, UPD1.BIN, UPD1A.BIN, UPD2.BIN, AND UPD3.BIN SHOULD BE SAVED:

```
FILEF DEV1:<DEV0:UPD?.BIN       ;TRANSFERS UPD1.BIN, UPD1A BIN, UPD2.BIN, AND UPD3.BIN
```

CONTIGUOUS (CORE-IMAGE) FILES SHOULD BE TRANSFERRED NEXT (TO GUARANTEE ROOM ON THE MEDIUM). THIS CAN BE DONE VIA THE FILEF COMMAND:

```
FILEF DEV1:<DEV0:A.SAV          ;TRANSFER A.SAV
```

FROM THIS POINT ON, THE DESIRED PROGRAMS ARE TRANSFERRED FROM THE INPUT MEDIA TO THE OUTPUT MEDIUM VIA THE FILEF COMMAND. USE OF THE SPECIAL FEATURES CAN CONSIDERABLY DECREASE THE NUMBER OF COMMANDS REQUIRED. FOR EXAMPLE, TO TRANSFER ALL DECTAPE DIAGNOSTICS TO THE NEW MEDIUM A SINGLE FILEF COMMAND WILL SUFFICE:

```
FILEF DEV1:<DEV0:XTC???.*       ;TRANSFERS ALL PROGRAMS WHOSE
                                ;NAMES START WITH "XTC"
```

AFTER ALL THE DESIRED FILES HAVE BEEN STORED ON THE NEW MEDIUM, IT SHOULD BE TESTED VIA THE FILET, FILEL, AND FILEG COMMANDS:

```
FILET  DEV1:*.*/LP      ;READ EVERY FILE ON THE NEW MEDIUM,
                        ;LISTING ALL INFORMATION ON THE
                        ;LINE PRINTER
FILEL  DEV1:*.BI?/N     ;LOAD ALL ABS FORMAT FILES
                        ;TO VERIFY THAT NO ERRORS
                        ;OCCUR. LIST ERRORS ONLY.
FILEG  DEV1:*.SA?/N     ;LOAD ALL CONTIGUOUS FILES TO
                        ;VERIFY THAT NO ERRORS OCCUR.
                        ;LIST ERRORS ONLY.
```

IT IS ALSO A GOOD IDEA TO DUPLICATE THE NEW MEDIUM TO PROVIDE A BACKUP.

7. ACT MODE OPERATION

THE XXDP UPDATE PROGRAMS UPD2 AND UPD3 HAS A SPECIAL MODE OF OPERATION REFERRED TO AS THE "ACT MODE". THE USE OF THE UPD3 PROGRAM IN "ACT MODE" IS RESTRICTED TO MANUFACTURING USES AND HAS NO APPLICATION IN THE FIELD.

THIS CHAPTER DESCRIBES THE COMMANDS PECULIAR TO "ACT MODE", AND DIFFERENCES IN OPERATION.

THE COMMANDS AND SWITCHES AFFECTED BY "ACT MODE" ARE:

ACT COMMAND

NOTACT COMMAND

LOAD AND FILEL COMMANDS

DUMP COMMAND

7.1 THE "ACT" COMMAND

THE UPD2 AND UPD3 PROGRAMS AS LOADED ARE IN "NOTACT" MODE. TO PUT THE PROGRAM IN "ACT" MODE, TYPE:

```
ACT<CR>                    ;PUTS PROGRAM IN ACT MODE.
```

THE PROGRAM ENTERS ACT MODE AND THEN TYPES:

```
*
```

THE UPD3R PROGRAM WHEN LOADED IS ALREADY IN ACT MODE.

7.2 THE "NOTACT" COMMAND

TO TAKE THE PROGRAM OUT OF ACT MODE TYPE:

```
NOTACT<CR>
```

THE PROGRAM EXITS ACT MODE AND THEN TYPES:

```
*
```

THE UPD3R PROGRAM TYPES FOLLOWING MESSAGE BEFORE TYPING
THE * BEFORE EACH COMMAND;
NOTE IN ACT MODE!!!

THE MESSAGE IS A WARNING TO THE USER THAT THE PROGRAM
SHOULD BE IN ACT MODE WHEN GENERATING XXDP MEDIA.

7.3 LOAD AND FILEL COMMANDS

THE LOAD AND FILEL COMMANDS FUNCTION EXACTLY AS IN "NOTACT" MODE, WITH THE EXCEPTION THAT THE CONTENTS OF CORE LOCATIONS 46 AND 52 ARE TYPED IN ADDITION TO THE USUAL DATA THAT IS TYPED AFTER A PROGRAM HAS BEEN LOADED. LOCATIONS 46 AND 52 CONTAIN INFORMATION THAT IS USED BY THE ACT11 MONITOR IN ACT11 TEST LINES IN MANUFACTURING FACILITIES. THE LOCATION 46 AND 52 INFORMATION IS TYPED ONLY IF THE PROGRAM'S LOWER CORE LIMIT IS EQUAL OR LOWER THAN 46.

EXAMPLE:

```
LOAD DK0:CKBNA0.BIN<CR> ;LOADS PROGRAM FROM DISK 0. THEN TYPES:

XFR: 000001  CORE: 000000,005711  LOG46: 000000  LOG52: 000000
```

IN THIS CASE LOC 46 AND 52 ARE 0, INDICATING THAT THE REQUIRED CORE INFORMATION IS MISSING.

```
LOAD DK0:CKBRD0.BIC     ;LOADS PROGRAM AND TYPES:

XFR: 000001  CORE: 000000,015151  LOG46: 012042 LOG52: 040000
```

IN THIS CASE, LOC 46 AND 52 CONTAIN THE REQUIRED INFORMATION

7.4 THE "DUMP" COMMAND

THE "DUMP" COMMAND FUNCTIONS EXACTLY AS IN NOTACT MODE, BUT IN ADDITION PERFORMS THE FOLLOWING FUNCTIONS:

- PERFORMS AN AUTOMATIC SIMULATED "LOAD" OF THE PROGRAM JUST STORED ON THE OUTPUT DEVICE, AND COMPARES IT AGAINST THE CONTENTS IN CORE. IF THE DATA DOES NOT MATCH, A "DUMP ERROR" MESSAGE OCCURS, INDICATING THAT THE "DUMP" OPERATION DID NOT SUCCEED IN STORING THE PROGRAM CORRECTLY. AT THIS POINT A RETRY OF THE COMMAND SHOULD BE DONE AND IF UNSUCCESSFUL, A BAD OUTPUT DEVICE IS INDICATED.

EXAMPLE:

```
DUMP DK0:CKBRD0.BIC<CR> ;STORES PROGRAM ON DISK 0.

XFR: 000001  CORE: 000000,015151  LOG46: 012402  LOG52: 040000
```

THE LAST PRINTOUT LINE IS THE RESULT OF SUCCESSFULLY COMPLETING THE AUTOMATIC SIMULATED "LOAD" OF THE PROGRAM FROM THE DISK.

CHAPTER 4. XTECO - XXDP TEXT EDITOR

TABLE OF CONTENTS

1. ABSTRACT

THE XTECO - XXDP TEXT EDITOR PROGRAM ENABLES THE USER OF XXDP TO CREATE AND EDIT ASCII TEXT FILES. ALL EDITING CAN BE DONE BY USING A FEW SIMPLE COMMANDS.

XTECO IS A CHARACTER ORIENTED EDITOR. ONE OR MORE CHARACTERS IN A LINE CAN BE MODIFIED WITHOUT RETYPING THE REST OF THE LINE. XTECO DOES NOT REQUIRE THAT LINE NUMBERS OR OTHER EXTRANEOUS INFORMATION BE ASSOCIATED WITH THE ASCII TEXT.

XTECO OPERATES ON ASCII DATA FILES. A FILE IS AN ORDERED SET OF DATA ON SOME PERIPHERAL DEVICE. IN THE CASE OF XTECO, A DATA FILE IS SOME TYPE OF DOCUMENT. AN INPUT FILE MAY BE A NAMED FILE ON ANY DIRECTORY DEVICE (DISK, MAGTAPE, DECTAPE, CASSETTE). AN OUTPUT FILE CAN BE WRITTEN ONTO ANY OF THE SAME DEVICES.

THE INPUT FILE FOR A GIVEN EDITING OPERATION IS THE FILE TO WHICH THE USER WISHES TO MAKE CHANGES. IF THE USER IS USING XTECO TO CREATE A NEW FILE, THERE IS NO INPUT FILE. THE OUTPUT FILE IS EITHER THE NEWLY CREATED FILE, OR THE EDITED VERSION OF THE INPUT FILE.

IN GENERAL, THE EDITING PROCESS PROCEEDS AS FOLLOWS. THE USER SPECIFIES THE FILE HE WISHES TO EDIT, AND THEN A BLOCK OF TEST IS READ INTO CORE. THE USER MODIFIES THE TEXT BY USING THE VARIOUS EDITING COMMANDS. HE THEN APPENDS ADDITIONAL BLOCKS OF TEXT AND EDITS THEM UNTIL THE ENTIRE FILE HAS BEEN EDITED, AT WHICH POINT HE OUTPUTS THE EDITED FILE AND CLOSES IT.

XTECO IS CAPABLE OF PERFORMING EDITING OPERATIONS FROM AND TO DEVICES CURRENTLY SUPPORTED BY THE XXDP UPDATE PROGRAMS # 1 AND # 2. REFER TO CHAPTER 3 FOR DETAILS.

2. REQUIREMENTS

THE MINIMUM CONFIGURATION FOR USING XTECO IS AS FOLLOWS:

A. PDP-11 PROCESSOR WITH 12K MEMORY

B. CONSOLE TERMINAL

C. XXDP SUPPORTED INPUT/OUTPUT DEVICE AS FOLLOWS:

1. SINGLE RANDOM ACCESS DEVICE (RK11/RK05, DECTAPE, ETC).
2. SEQUENTIAL ACCESS DEVICE WITH 2 DRIVES (MAGTAPE, CASSETTE).

3. LOADING AND STARTING PROCEDURE

XTECO IS LOADED BY TYPING R XTECO<CR> WHILE UNDER CONTROL OF THE XXDP MONITOR. ONCE LOADED THE PROGRAM AUTOMATICALLY STARTS AND TYPES THE FOLLOWING MESSAGE:

```
CZQUG-G XTECO - XXDP TEXT EDITOR
DATE (DD-MMM-YY):
```

TYPE THE DATE ACCORDING TO THE FOLLOWING FORMAT, FOLLOWED BY <CR>

DD-MMM-YY

WHERE:

```
DD      IS THE DAY OF THE MONTH
MMM     IS THE MONTH OF THE YEAR (1ST THREE LETTERS)
YY      IS THE YEAR (LAST 2 NUMBERS)
```

THE DASHES MUST ALSO BE TYPED.

EXAMPLE: 22-OCT-77

THE PROGRAM ECHOES BACK THE DATE AND THEN TYPES.

```
RESTART: 005730          ;PROGRAM'S RESTART ADDRESS.
```

NOW GO TO STEP 4. HOW TO USE XTECO.

4. HOW TO USE XTECO

AS PACKAGED, THE XTECO PROGRAM PROVIDES SEVERAL OF THE COMMANDS AVAILABLE UNDER THE UPD1/UPD1A/UPD2/UPD3 PROGRAMS, IN ADDITION TO THOSE COMMANDS PROVIDED FOR EDITING PURPOSES. IT IS DONE SO AS TO MINIMIZE THE NEED FOR SWAPPING BACK AND FORTH BETWEEN THE XTECO AND UPD1/UPD1A/UPD2/UPD3 PROGRAMS.

THE COMMANDS THAT ARE COMMON BETWEEN UPD1/UPD2 AND XTECO ARE LISTED HERE, BUT NOT DESCRIBED. REFER TO CHAPTER 3. FOR DETAILED DESCRIPTIONS OF THOSE COMMANDS.

XTECO COMMANDS ARE OF TWO TYPES: NON-EDIT TYPE COMMANDS, AND EDIT TYPE COMMANDS.

THE NON-EDIT TYPE COMMANDS ARE:

```
FILL            ;UPD1/UPD1A/UPD2/UPD3 EQUIVALENT

BOOT            ;UPD1/UPD1A/UPD2/UPD3 EQUIVALENT

DIRLP           ;UPD2/UPD3 EQUIVALENT

DIR             ;UPD1/UPD1A/UPD2/UPD3 EQUIVALENT

DELETE          ;UPD1/UPD1A/UPD2/UPD3 EQUIVALENT

RENAME          ;UPD1/UPD2/UPD3 EQUIVALENT

TYPE            ;UPD2/UPD3 EQUIVALENT

PRINT           ;UPD2/UPD3 EQUIVALENT

TEXT            ;XTECO UNIQUE
EDIT            ;XTECO UNIQUE
TECO            ;XTECO UNIQUE

<CR>            ;CARRIAGE RETURN IS THE NON-EDIT TYPE COMMAND
                ;STRING TERMINATOR.
```

```
                    THE EDIT TYPE COMMANDS ARE:

                    L               ;USED TO MOVE POINTER ONE OR MORE LINES

                    C               ;USED TO MOVE POINTER ONE OR MORE CHARACTERS.

                    J               ;USED TO MOVE POINTER TO BEGINNING OF TEXT IN CORE.

                    ZJ              ;USED TO MOVE POINTER TO END OF TEXT IN CORE.

                    S               ;USED TO SEARCH FOR A CHARACTER SEQUENCE IN TEXT IN CORE.

                    N               ;USED TO SEARCH CORE AND REMAINDER OF INPUT FILE FOR
                                    ;A SPECIFIED CHARACTER SEQUENCE.

                    T               ;USED TO TYPE ONE OR MORE TEXT LINES.

                    D               ;USED TO DELETE ONE OR MORE CHARACTERS.

                    K               ;USED TO DELETE (KILL) ONE OR MORE TEXT LINES.

                    I               ;USED TO INSERT ASCII TEXT INTO THE TEXT BUFFER.

                    A               ;USED TO APPEND ONE OR MORE TEXT BLOCKS TO TEXT BUFFER.

                    EX              ;OUTPUTS EDITED FILE TO OUTPUT DEVICE AND CLOSES OUTPUT.

                    <ALT>           ;ECHOES A "$" .  USED TO TERMINATE AN EDIT COMMAND.

                    <ALT><ALT>      ;ECHOES 2 "$".  USED TO TERMINATE LAST EDIT COMMAND,
                                    ;AND TO CAUSE EXECUTION OF ENTIRE COMMAND STRING.

                    NOTE:   ALT MAY BE ESC ON SOME TERMINALS.

                    THE USER SHOULD BE AWARE OF THE USE OF THE FOLLOWING SPECIAL CHARACTERS:

                     C (CTRL C)     ;USED TO EXIT OUT OF ANY COMMAND AND RETURN TO COMMAND
                                    ;MODE.  WILL CAUSE AN OPEN OUTPUT FILE TO BE CLOSED.
                                    ;THE USER MUST BE CAREFUL NOT TO TYPE CTRL C, UNLESS
                                    ;HE WISHES TO ABORT HIS OPERATION.  IT IS SPECIALLY
                                    ;TRUE WHEN EDITING A FILE, AS ALL WORK WILL BE WASTED.

                     O (CTRL O)      ;USED TO STOP PRINTING ON THE CONSOLE TERMINAL,
                                    ;AS WHEN TYPING MULTIPLE LINES OF TEXT WHEN EDITING
                                    ;A FILE.

                     U (CTRL U)     ;USED TO EMPTY OUT CONTENTS OF KEYBOARD BUFFER,
                                    ;AS WHEN THE USER WISHES TO START TYPING HIS COMMAND
                                    ;SEQUENCE ALL OVER AGAIN.

                    RUBOUT          ;USED TO REMOVE ONE OR MORE CHARACTERS TYPED FROM
                      OR            ;COMMAND OR TEXT STRING.  ONE DEPRESSION OF THE
                    DELETE          ;RUBOUT KEY REMOVES ONE CHARACTER.
```

4.1 THE "TEXT", "EDIT", AND "TECO" COMMANDS

TEXT, EDIT, AND TECO COMMANDS ARE THE BASIC COMMANDS PROVIDED TO CREATE OR EDIT AN ASCII TEXT FILE.

THE TEXT COMMAND IS USED WHEN THE USER WISHES TO CREATE A NEW TEXT FILE. THE TEXT COMMAND DOES NOT REQUIRE AN INPUT FILE, ONLY AN OUTPUT FILE. ALL EDITING COMMANDS ARE AVAILABLE WITH THE EXCEPTION OF THE "A" (APPEND) COMMAND WHICH BECOMES A NO-OP COMMAND WHEN NO INPUT FILE EXISTS.

THE EDIT COMMAND IS THE GENERAL PURPOSE COMMAND FOR EDITING AN EXISTING TEXT FILE. IT PERMITS THE USER TO EDIT AN INPUT FILE IN ONE TYPE OF DEVICE AND TO OUTPUT THE EDITED FILE TO A DIFFERENT TYPE DEVICE. ALL EDITING COMMANDS ARE AVAILABLE WHEN UNDER THE EDIT COMMAND.

THE TECO COMMAND IS A SPECIALIZED VERSION OF THE EDIT COMMAND. UNDER THE TECO COMMAND THE INPUT AND OUTPUT DEVICE/DRIVE MUST BE THE SAME (IT ASSUMES THEY ARE THE SAME), AND MUST BE RANDOM ACCESS TYPE DEVICES (DISK, DECTAPE). THE EDITED OUTPUT FILE TAKES ITS NAME FROM THE NAME AND EXTENSION OF THE INPUT FILE, AND THE INPUT FILE IS RENAMED TO A ".BAK" EXTENSION (FOR BACKUP). ALL EDITING COMMANDS ARE AVAILABLE.

INDIVIDUAL COMMAND DESCRIPTIONS FOLLOW.

4.2 THE "TEXT" COMMAND

THE TEXT COMMAND IS USED TO CREATE A NEW TEXT FILE. THE FORMAT IS·

TEXT OUTDEV:FILNAM.EXT<CR>

WHERE OUTDEV: IS ANY DIRECTORY DEVICE.

THE PROGRAM WILL TYPE:

"MAKE OUTPUT READY. TYPE <CR> WHEN READY"

INSURE THAT THE OUTPUT DEVICE IS READY AND WRITE ENABLED. PRESS THE "CR" KEY ON THE CONSOLE TERMINAL WHEN READY TO PROCEED. THE PROGRAM IS NOW IN EDIT MODE, AND ONLY EDITING TYPE COMMANDS ARE VALID. THE PROGRAM PROMPTS THE USER BY TYPING AN ASTERISK (*).

THE USER CAN AT THIS POINT TYPE AND EDIT HIS TEXT. REFER TO CHAPTER 4.5 INTRODUCTION TO EDIT TYPE COMMANDS.

EXAMPLE:

TEXT DK0:1.TXT<CR>

4.3 THE "EDIT" COMMAND

THE "EDIT" COMMAND PERMITS THE USER TO EDIT A TEXT FILE FROM A SPECIFIED INPUT DEVICE, AND TO OUTPUT THE EDITED TEXT FILE ON TO A SPECIFIED OUTPUT DEVICE. THE COMMAND FORMAT IS:

EDIT OUTDEV:FILNAM.EXT_INDEV:FILNAM.EXT<CR>

BOTH OUTDEV: AND INDEV: MUST BE DIRECTORY DEVICES. FOR MAGTAPE OR CASSETTE, THE INPUT DEVICE DRIVE MUST BE DIFFERENT FROM THE DRIVE ASSIGNED TO THE OUTPUT. IF THE USER WISHES TO EDIT A FILE RESIDING ON PAPER TAPE, THE TAPE MUST FIRST BE TRANSFERRED TO A DIRECTORY DEVICE BY MEANS OF THE "PIP" COMMAND OF THE UPD2 OR UPD3 PROGRAM, AND THEN EDITED AS DESCRIBED IN THIS DOCUMENT. ONCE EDITED, THE FILE MAY AGAIN BE TRANSFERRED TO PAPER TAPE BY MEANS OF THE "PIP" COMMAND OF THE UPD2 OR UPD3 PROGRAM.

THE PROGRAM WILL TYPE:

"MAKE OUTPUT READY. TYPE <CR> WHEN READY"

INSURE THAT THE OUTPUT DEVICE IS READY AND WRITE ENABLED. PRESS THE "CR" KEY ON THE CONSOLE TERMINAL WHEN READY TO PROCEED. THE PROGRAM IS NOW IN EDIT MODE, AND ONLY EDITING TYPE COMMANDS ARE VALID. THE PROGRAM PROMPTS THE USER BY TYPING AN ASTERISK (*).

THE USER CAN AT THIS POINT TYPE AND EDIT HIS TEXT REFER TO CHAPTER 4.5 INTRODUCTION TO EDIT TYPE COMMANDS.

EXAMPLES:

EDIT DK0:2.TXT_DK0:1.TXT<CR>

EDIT MT1:1.TXT_MT0:1.TXT<CR>

4.4 THE "TECO" COMMAND

THE "TECO" COMMAND IS A SPECIALIZED, SHORT HAND VERSION OF THE "EDIT" COMMAND. WHEN USING THE "TECO" COMMAND THE INPUT DEVICE/DRIVE MUST BE THE SAME AS FOR OUTPUT, AND THE TECO COMMAND SO ASSUMES. IN ADDITION, THE COMMAND IS RESERVED FOR USE WITH RANDOM ACCESS DEVICES ONLY (DISKS, DECTAPE). THE COMMAND FORMAT IS:

TECO DEV:FILNAM.EXT<CR>

WHERE DEV: IS ANY RANDOM ACCESS DEVICE.

IT IS IMPORTANT THAT THE USER BE AWARE OF THE MECHANICS INVOLVED IN THE OPERATION OF THE TECO COMMAND. THE SEQUENCE IS AS FOLLOWS:

1. OPEN INPUT FILE
2. OPEN OUTPUT FILE, AND ASSIGN IT A .TMP EXTENSION
3. EDIT OPERATIONS ARE PERFORMED
4. EDITING DONE. OUTPUT EDITED FILE TO .TMP FILE.
5. CLOSE THE .TMP FILE.
6. RENAME THE .TMP FILE TO SAME NAME AND EXTENSION AS THE INPUT FILE.
7. RENAME THE INPUT FILE TO A .BAK EXTENSION

WHEN USING THE TECO COMMAND THE INPUT DEVICE MUST NOT CONTAIN A FILE WITH THE SAME NAME AS THE INPUT FILE AND .BAK EXTENSION, IF THE USER WISHES TO PRESERVE THAT FILE, AS IT WILL BE DELETED IN THE PROCESS OF RENAMING THE INPUT FILE TO A .BAK EXTENSION.

THERE IS NO CONCERN IF THE EXISTING .BAK FILE IS MERELY A BACKUP FROM A PREVIOUS EDITING OPERATION.

ALSO NOTE THAT THE TECO COMMAND MUST NOT BE USED TO EDIT A FILE WHICH HAS THE FILENAME EXTENSION .BAK. THE FILE MUST FIRST BE RENAMED TO ANOTHER EXTENSION.

THE PROGRAM WILL TYPE:

"MAKE OUTPUT READY. TYPE <CR> WHEN READY"

INSURE THAT THE OUTPUT DEVICE IS READY AND WRITE ENABLED. PRESS THE "CR" KEY ON THE CONSOLE TERMINAL WHEN READY TO PROCEED. THE PROGRAM IS NOW IN EDIT MODE, AND ONLY EDITING TYPE COMMANDS ARE VALID. THE PROGRAM PROMPTS THE USER BY TYPING AN ASTERISK (*).

THE USER CAN AT THIS POINT TYPE AND EDIT HIS TEXT. REFER TO CHAPTER 4.5 INTRODUCTION TO EDIT TYPE COMMANDS.

EXAMPLES: TECO DK0:1.TXT<CR> TECO DT1:ABC.TXT<CR>

4.5 INTRODUCTION TO EDIT TYPE COMMANDS

4.5.1 GENERAL EDITING COMMAND STRING SYNTAX

XTECO COMMANDS MAY BE GIVEN ONE AT A TIME. HOWEVER, IT IS USUALLY MORE CONVENIENT TO TYPE IN A SINGLE COMMAND STRING, SEVERAL COMMANDS THAT FORM A LOGICAL GROUP. AN EXAMPLE OF A COMMAND STRING IS SHOWN BELOW

```
*IHEADINGSNTAG:$2LT$$   ;INSERTS WORD "HEADING", SEARCHES FOR STRING "TAG:", MOVES
                        ;POINTER FORWARD 2 LINES AND TYPES LINE POINTED TO.
```

A COMMAND STRING IS TYPED AFTER XTECO INDICATES ITS READINESS BY PRINTING AN ASTERISK. COMMAND STRINGS ARE FORMED BY MERELY TYPING ONE COMMAND AFTER ANOTHER. COMMAND STRINGS ARE TERMINATED BY TYPING TWO CONSECUTIVE ALTMODES.

EXECUTION OF THE COMMAND STRING BEGINS ONLY AFTER THE DOUBLE ALTMODE HAS BEEN TYPED. AT THAT POINT, EACH COMMAND IN THE STRING IS EXECUTED IN TURN, STARTING AT THE LEFT. WHEN ALL COMMANDS HAVE BEEN EXECUTED, XTECO PRINTS ANOTHER ASTERISK, INDICATING READINESS TO ACCEPT ANOTHER COMMAND.

IF SOME COMMAND IN THE STRING CAN NOT BE EXECUTED BECAUSE OF A COMMAND ERROR, EXECUTION OF THE COMMAND STRING STOPS AT THAT POINT, AND AN ERROR MESSAGE IS PRINTED. COMMANDS PRECEDING THE BAD COMMAND ARE EXECUTED. THE BAD COMMAND AND THOSE FOLLOWING IT ARE NOT EXECUTED.

4.5.2 COMMAND ARGUMENTS

THERE ARE TWO TYPES OF ARGUMENTS FOR XTECO EDITING COMMANDS. SOME COMMANDS REQUIRE NUMERIC ARGUMENTS AND SOME OTHER COMMANDS REQUIRE ALPHANUMERIC (TEXT) ARGUMENTS. NUMERIC ARGUMENTS ARE DECIMAL INTEGERS. NUMERIC ARGUMENTS ALWAYS PRECEDE THE COMMAND TO WHICH THEY APPLY. A TYPICAL EXAMPLE OF A COMMAND TAKING A NUMERIC ARGUMENT IS THE COMMAND TO DELETE THREE CHARACTERS: "3D".

ALPHANUMERIC ARGUMENTS ARE TEXTUAL ARGUMENTS MEANT TO BE INTERPRETED AS ASCII CODE BY XTECO. ALPHANUMERIC ARGUMENTS ALWAYS FOLLOW THE COMMAND TO WHICH THEY APPLY, AND THEY MUST ALWAYS BE TERMINATED BY AN ALTMODE. EXAMPLES OF ALPHANUMERIC ARGUMENTS ARE (1) TEXT TO BE INSERTED, AND (2) CHARACTER STRINGS TO BE SEARCHED FOR.

EXAMPLE:

```
*ISOMETHING$$   ;THE ARGUMENT IS "SOMETHING"
```

AS SHOWN IN THE ABOVE EXAMPLE, THE ALTMODE USED TO TERMINATE AN ALPHANUMERIC ARGUMENT MAY ALSO SERVE AS ONE OF THE TWO ALTMODES NECESSARY TO TERMINATE A COMMAND STRING.

4.6 XTECO EDIT COMMANDS

4.6.1 INPUT COMMANDS

THE "A" (APPEND) COMMAND

THE "A" COMMAND READS IN THE NEXT BLOCK OF TEXT FROM THE INPUT DEVICE AND ADDS IT TO THE CONTENTS OF THE TEXT BUFFER IN CORE.

THE "A" COMMAND ACCEPTS NUMERIC ARGUMENTS. EXAMPLE: 3A$$. HOWEVER, IT DOES NOT EXECUTE ANY OTHER COMMANDS FOLLOWING IT IN THE COMMAND STRING. IT IS MEANT TO BE USED SINGLY IN A COMMAND STRING. WHEN NOT ENOUGH CORE IS AVAILABLE TO SATISFY AN "A" COMMAND, XTECO OUTPUTS PART OF THE TEXT BUFFER ONTO THE OUTPUT DEVICE UNTIL THE REQUIREMENTS OF THE "A" COMMAND ARE SATISFIED.

4.6.2 BUFFER POINTER POSITIONING COMMANDS

SINCE XTECO IS A CHARACTER-ORIENTED EDITOR, IT IS VERY IMPORTANT THAT THE USER UNDERSTAND THE CONCEPT OF THE "BUFFER POINTER". THE POSITION OF THE BUFFER POINTER DETERMINES THE EFFECT OF MANY OF THE EDITING COMMANDS. FOR EXAMPLE, INSERTION AND DELETION ALWAYS TAKES PLACE AT THE CURRENT POSITION OF THE BUFFER POINTER.

THE "BUFFER" IS THE CURRENT TEXT CONTENTS IN CORE, FROM THE FIRST CHARACTER, UP TO AND INCLUDING THE LAST CHARACTER.

THE BUFFER POINTER IS SIMPLY A MOVABLE POSITION INDICATOR. IT IS ALWAYS POSITIONED BETWEEN TWO CHARACTERS IN THE BUFFER, OR BEFORE THE FIRST CHARACTER IN THE BUFFER, OR AFTER THE LAST CHARACTER IN THE BUFFER. THE POINTER MAY BE MOVED FORWARD OR BACKWARD OVER ANY NUMBER OF CHARACTERS.

THE "J" COMMAND

THE "J" COMMAND MOVES THE POINTER TO THE BEGINNING OF THE BUFFER
I.E., IMMEDIATELY BEFORE THE FIRST CHARACTER IN THE BUFFER.

THE "ZJ" COMMAND

THE "ZJ" COMMAND MOVES THE POINTER TO THE END OF THE BUFFER. I.E.,
TO POSITION FOLLOWING LAST CHARACTER IN THE BUFFER.

THE "C" COMMAND

THE "C" COMMAND MOVES THE POINTER ONE CHARACTER IN THE BUFFER. THE
"C" COMMAND MAY BE PRECEDED BY A (DECIMAL) NUMERIC ARGUMENT. THE
COMMAND "NC" MOVES THE POINTER FORWARD OVER "N" CHARACTERS. THE
COMMAND "-NC" MOVES THE POINTER BACKWARD OVER "N" CHARACTERS. (THE
POINTER CANNOT BE ADVANCED BEYOND THE ENDS OF THE BUFFER).

THE "L" COMMAND

THE "L" COMMAND IS USED TO ADVANCE THE BUFFER POINTER OR MOVE IT
BACKWARD, ON A LINE-BY-LINE BASIS. THE "L" COMMAND TAKES A NUMERIC
ARGUMENT, WHICH MAY BE POSITIVE, NEGATIVE, OR ZERO, AND IS UNDERSTOOD
TO BE ONE (1) IF OMITTED.

SUPPOSE THE BUFFER POINTER IS POSITIONED AT THE BEGINNING OF LINE "B"
OR AT SOME POSITION WITHIN LINE "B".

THE COMMAND L OR 1L, ADVANCES THE POINTER TO THE BEGINNING OF LINE B+1.

THE COMMAND NL, WHERE N>0, ADVANCES THE POINTER TO THE BEGINNING
OF LINE B+N.

THE COMMAND -0L MOVES THE POINTER TO THE BEGINNING OF LINE B. IF THE
POINTER IS ALREADY AT THE BEGINNING, NOTHING HAPPENS.

THE COMMAND -L OR -1L MOVES THE POINTER BACK TO THE BEGINNING OF
LINE B-1.

THE COMMAND -NL MOVES THE POINTER BACK TO THE BEGINNING OF LINE B-N

NOTE: EXECUTION OF THE "A" (APPEND) COMMAND DOES NOT CHANGE THE
POSITION OF THE BUFFER POINTER.

4.6.3 TEXT TYPE-OUT COMMANDS

THE "T" COMMAND

VARIOUS PARTS OF THE TEXT IN THE BUFFER CAN BE TYPED OUT FOR EXAMINATION BY USE OF THE "T" COMMAND. JUST WHAT IS TYPED OUT DEPENDS ON THE POSITION OF THE BUFFER POINTER AND THE ARGUMENT GIVEN. THE "T" COMMAND NEVER MOVES THE BUFFER POINTER.

THE "T" COMMAND TYPES OUT EVERYTHING FROM THE BUFFER POINTER THROUGH THE NEXT LINE FEED. THUS, IF THE POINTER IS AT THE BEGINNING OF A LINE, THE T COMMAND CAUSES THAT LINE TO BE TYPED OUT. IF THE POINTER IS IN THE MIDDLE OF A LINE, T CAUSES THE PORTION OF THE LINE FOLLOWING THE POINTER TO BE TYPED.

THE COMMAND NT (N>0) IS USED TO TYPE OUT N LINES I.E., EVERYTHING FROM THE BUFFER POINTER THROUGH THE NTH LINE FEED FOLLOWING IT.

THE USER, ESPECIALLY ONE NEW TO XTECO, SHOULD USE THE T COMMAND OFTEN, TO MAKE SURE THE BUFFER POINTER IS WHERE HE THINKS IT IS.

DURING EXECUTION OF ANY T COMMAND, THE USER MAY STOP THE TERMINAL OUTPUT BY TYPING THE ^O (CTRL O) CHARACTER. THE TYPEOUT STOPS AND EXECUTION OF THE REMAINDER OF THE COMMAND STRING IS ABORTED. THEREFORE, LONG TYPEOUTS SHOULD BE RESTRICTED TO SINGLE COMMAND, COMMAND STRINGS.

4.6.4 DELETION COMMANDS

THE "D" COMMAND

INDIVIDUAL CHARACTERS ARE DELETED BY USING THE "D" COMMAND. THE COMMAND "D" DELETES THE CHARACTER IMMEDIATELY FOLLOWING THE BUFFER POINTER. THE COMMAND "ND", WHERE N > 0 DELETES THE N CHARACTERS IMMEDIATELY FOLLOWING THE POINTER. EXAMPLES: D$$, 3D$$, 5D$$

THE "K" COMMAND

LINES ARE DELETED BY USING THE "K" COMMAND. THE "K" COMMAND MAY BE PRECEDED BY A NUMERIC ARGUMENT, WHICH IS UNDERSTOOD TO BE A 1 IF OMITTED. THE COMMAND "NK" (N >0) DELETES EVERYTHING FROM THE CURRENT POSITION OF THE BUFFER POINTER THROUGH THE NTH LINE FEED CHARACTER FOLLOWING THE POINTER.

AT THE CONCLUSION OF A D OR K COMMAND THE BUFFER POINTER IS POSITIONED BETWEEN THE CHARACTERS WHICH PRECEDE AND FOLLOW THE DELETION.

4.6.5 INSERTION COMMAND

THE ONLY INSERTION COMMAND IS THE "I" COMMAND. THE ASCII TEXT THAT IS TO BE INSERTED INTO THE BUFFER IS TYPED IMMEDIATELY AFTER THE LETTER I. THE TEXT TO BE INSERTED IS TERMINATED BY AN ALTMODE.

ANY ASCII CHARACTER EXCEPT NULL, ALTMODE, RUBOUT, CTRL C, CTRL O, AND CTRL U MAY BE INCLUDED IN THE TEXT TO BE INSERTED.

IF A CARRIAGE RETURN IS TYPED IN AN INSERTION, IT IS AUTOMATICALLY FOLLOWED BY A LINE FEED. THE TEXT TO BE INSERTED IS PLACED IN THE BUFFER AT THE POSITION OF THE BUFFER POINTER, I.E., BETWEEN THE CHARACTERS. AT THE CONCLUSION OF THE INSERTION COMMAND THE BUFFER POINTER IS POSITIONED AT THE END OF THE INSERTION.

ANY NUMBER OF LINES MAY BE INSERTED WITH A SINGLE "I" COMMAND. HOWEVER, IT IS RECOMMENDED THAT NO MORE THAN 10 TO 20 LINES SHOULD BE INSERTED WITH EACH I COMMAND.

4.6.6 OUTPUT COMMANDS

THE ONLY OUTPUT COMMAND AVAILABLE WITH XTECO IS THE "EX" (EXIT) COMMAND. THE "EX" COMMAND IS USED TO CONCLUDE AN EDITING JOB WITH A MINIMUM OF EFFORT. ITS USE IS BEST SHOWN BY AN EXAMPLE:

SUPPOSE THE USER IS EDITING A 30 PAGE FILE AND SUPPOSE THE LAST ACTUAL CHANGE TO THE FILE IS MADE ON PAGE 10. AT THIS POINT THE USER GIVES THE COMMAND:

```
EX$$
*
```

THE ACTION OF XTECO IS (1) TO RAPIDLY MOVE ALL OF THE REST OF THE INPUT FILE TO THE OUTPUT FILE, (2) CLOSE THE FILE, AND (3) TO RETURN TO COMMAND MODE SO THAT THE USER MAY GIVE OTHER NON-EDIT MODE COMMANDS.

4.6.7 SEARCH COMMANDS

IN MANY CASES THE SIMPLEST WAY TO POSITION THE BUFFER POINTER IS BY USING A CH ACTER STRING SEARCH. A SEARCH COMMAND CAUSES XTECO TO SEARCH THROUGH THE TEXT UNTIL A SPECIFIED STRING OF CHARACTERS IS FOUND, AND THEN TO POSITION THE POINTER AT THE END OF THIS STRING. THERE ARE TWO SEARCH COMMANDS.

THE "S" COMMAND

THE "S" COMMAND IS USED TO SEARCH FOR A CHARACTER STRING WITHIN THE BUFFER. THE STRING TO BE SEARCHED FOR IS SPECIFIED AS AN ALPHANUMERIC ARGUMENT FOLLOWING THE S COMMAND. THIS ARGUMENT MUST BE TERMINATED BY AN ALTMODE.

EXECUTION OF THE S COMMAND BEGINS AT THE POSITION OF THE BUFFER POINTER AND CONTINUES TO THE END OF THE BUFFER. IF THE SPECIFIED STRING IS NOT FOUND AN ERROR MESSAGE IS PRINTED AND THE BUFFER POINTER IS SET TO THE POINT WHERE THE SEARCH BEGAN.

THE "N" COMMAND

THE "N" COMMAND WORKS JUST LIKE THE "S" COMMAND. THE DIFFERENCE IS THAT AN S COMMAND ENDS AT THE END OF THE BUFFER, WHEREAS THE N COMMAND DOES NOT. AN N SEARCH BEGINS LIKE AN S SEARCH BUT IF THE CHARACTER STRING IS NOT FOUND IN THE CURRENT BUFFER AN AUTOMATIC "A" (APPEND) COMMAND IS EXECUTED AND THE SEARCH CONTINUED UNTIL THE SEARCH IS SUCCESSFUL OR THE INPUT FILE EXHAUSTED.

IF THE N COMMAND FINDS THE SPECIFIED STRING THE POINTER IS POSITIONED AT THE END OF THE STRING FOUND. IF THE STRING IS NOT FOUND, AN ERROR MESSAGE IS PRINTED AND THE POINTER IS SET AT THE BEGINNING OF THE BUFFER. SINCE A GOOD PART OF THE FILE MAY ALREADY HAVE BEEN OUTPUT TO THE OUTPUT DEVICE, THE USER MAY HAVE NO OTHER CHOICE THAN TO EXIT VIA THE "EX" COMMAND, AND TO REOPEN THE FILE AND TRY THE N SEARCH AGAIN WITH A CHARACTER STRING THAT CAN BE FOUND.

WARNING: WHEN ATTEMPTING TO SEARCH IT IS VERY EASY TO OVERLOOK AN OCCURRENCE OF THE SEARCH STRING PRECEDING THE ONE THE USER DESIRES. FOR EXAMPLE, HE MAY WANT TO MOVE THE POINTER AFTER THE WORD "AND" BUT ERRONEOUSLY POSITION THE POINTER AFTER A PRECEDING OCCURRENCE OF A WORD LIKE "THOUSAND".

FOR THIS REASON, THE USER IS STRONGLY URGED TO EXECUTE A "T" COMMAND TO ASCERTAIN THE POSITION OF THE POINTER AFTER EACH SEARCH COMMAND.

5. ERRORS

ERROR MESSAGES GENERATED BY XTECO ARE THE SAME AS THOSE GENERATED BY XXDP UPDATE PROGRAMS # 1 AND # 2 (UPD1, UPD2), AND HAVE THE SAME MEANINGS.

IN ADDITION, ONE ERROR MESSAGE IS GENERATED BY XTECO WHEN A SEARCH FOR A CHARACTER STRING BY EITHER THE "S" OR "N" COMMANDS FAILS. IN THAT CASE XTECO TYPES:

"NOT FOUND: ASCII STRING"

CHAPTER 5. COPY - XXDP COPY PROGRAMS

1. ABSTRACT
2. REQUIREMENTS
3. LOADING AND STARTING PROCEDURE
4. HOW TO USE COPY1 AND COPY2
5. ERRORS

1. ABSTRACT

THE COPY1 AND COPY2 - XXDP COPY PROGRAMS ENABLE THE USER OF XXDP TO CREATE A NEW MEDIUM EXACTLY THE SAME AS THE ORIGINAL XXDP MEDIUM.

THE COPY PROGRAMS ALLOW ONLY COPYING ON THE SAME MEDIUMS. THE PROGRAM WILL NOT COPY ANYTHING OTHER THAN XXDP MATERIAL THEY ARE NOT GENERAL PURPOSE COPY PROGRAMS.

THE COPY PROGRAMS CONSIST OF TWO CHAPTERS, THE COPY CHAPTER AND THE VERIFY CHAPTER. THE COMPLETION OF EACH CHAPTER WITHOUT ERRORS IS INDICATED ON THE TTY.

2. REQUIREMENTS

THE MINIMUM CONFIGURATION FOR USING COPY1 AND COPY2 IS AS FOLLOWS

A. PDP-11 PROCESSOR WITH 8K MEMORY FOR COPY1 AND 16K MEMORY FOR COPY2.

B. CONSOLE TERMINAL

C. XXDP SUPPORTED INPUT/OUTPUT DEVICE AS FOLLOWS.

1. RANDOM ACCESS DEVICE WITH 2 DRIVES (RK11/RK05, DECTAPE, ETC)
2. SEQUENTIAL ACCESS DEVICE WITH 2 DRIVES (MAGTAPE, CASSETTE)

3. LOADING AND STARTING PROCEDURE

COPY IS LOADED BY TYPING R COPY1<CR> OR R COPY2<CR> WHILE UNDER CONTROL OF THE XXDP MONITOR. ONCE LOADED THE PROGRAM AUTOMATICALLY BEGINS OPERATION.

4. HOW TO USE COPY1 AND COPY2

THE COPY PROGRAMS PROVIDE SEVERAL OF THE COMMANDS AVAILABLE UNDER THE UPD1/UPD1A/UPD2/UPD3 PROGRAMS. IN ADDITION TO THOSE COMMANDS PROVIDED FOR COPYING PURPOSES. IT IS DONE SO AS TO MINIMIZE THE NEED FOR SWAPPING BACK AND FORTH BETWEEN COPY1/2 AND UPD1/UPD1A/UPD2/UPD3 PROGRAMS.

THE COMMANDS THAT ARE COMMON BETWEEN UPD1/UPD1A/UPD2/UPD3 AND COPY1/2 ARE LISTED HERE, BUT NOT DESCRIBED. REFER TO CHAPTER 3 FOR A DETAILED DESCRIPTION OF THOSE COMMANDS.

THE COPY COMMANDS ARE:

```
FILL                    ;UPD1/UPD1A/UPD2/UPD3 EQUIVALENT.

BOOT                    ;UPD1/UPD1A/UPD2/UPD3 EQUIVALENT.

DIRLP                   ;UPD1/UPD1A/UPD2/UPD3 EQUIVALENT.

DIR                     ;UPD1/UPD1A/UPD2/UPD3 EQUIVALENT.

COPY                    ;COPY UNIQUE

VERIFY                  ;COPY UNIQUE.
```

4.1 THE "COPY" COMMAND

COPY IS THE BASIC COMMAND TO COPY XXDP SOFTWARE. THE SOURCE AND DESTINATION MUST BE ON THE SAME MEDIUM. THE COMMAND STRING IS AS FOLLOWS:

```
*COPY DEVN:=DEVNN:<CR>.
```

OR

```
*COPY DEVN:=DEVNN:/NEW<CR>.
```

WHERE N AND NN ARE THE DEVICE LOGICAL NUMBERS

EXAMPLE: COPY RK0:=RK1

THIS COPIES RK1 ONTO RK0.

THE PROGRAM THEN TYPES:

```
*MAKE OUTPUT READY, TYPE <CR> WHEN READY.
```

THIS IS TO INFORM THE USER THAT THE OUTPUT DEVICE MUST BE POWERED UP, READY AND WRITE ENABLED. WHEN ALL THESE REQUIREMENTS ARE MET TYPE <CR> TO START THE COPY PROCESS.

WHEN THE COPY IS COMPLETED A VERIFICATION PASS IS MADE. THIS PASS IS STARTED WHEN THE PROGRAM TYPES:

```
*STARTING VERIFICATION.
```

WHEN THE VERIFY PASS IS COMPLETE THE PROGRAM TYPES·

```
VERIFY COMPLETE, COPY COMPLETE.
```

THE COPY HAS NOW BEEN COMPLETED.

WHEN THE OPTIONAL /NEW SWITCH IS USED THE COPY2 PROGRAM WILL COPY INFORMATION ON A FILE BY FILE BASIS RATHER THAN ON A BLOCK BY BLOCK BASIS. HOWEVER, IF THE DISK CONTAINS A "BAD-SECTOR" TRACK, THE COPY PROGRAM WILL ONLY DO THIS FILE BY FILE COPY (EVEN IF THE SWITCH IS NOT USED). THIS SWITCH IS NOT AVAILABLE IN THE COPY1 PROGRAM.

4.2 THE VERIFY COMMAND

THE VERIFY COMMAND WILL ONLY DO A VERIFICATION OF A XXDP MEDIUM. THE COMMAND STRING IS AS FOLLOWS:

```
*VERIFY DEVN: DEVNN: <CR>
```

WHERE THE N AND NN ARE THE DEVICE LOGICAL NUMBERS.

THE PROGRAM THEN TYPES:

```
*STARTING VERIFICATION.
```

THE VERIFICATION HAS NOW BEGUN.

WHEN THE VERIFICATION IS COMPLETE THE PROGRAM TYPES:

```
*VERIFY COMPLETE.
```

5. ERRORS

```
NEXFIL          ;REPORTS FILE NOT FOUND.
DEVERR          ;REPORTS A DEVICE ERROR, CHECK FOR READY, ON LINE ETC
DEVFUL          ;REPORTS A DEVICE FULL, NO MORE ROOM FOR FILES.
INVCMD          ;INVALID COMMAND, CHECK LAST COMMAND STRING.
INVNAM          ,INVALID NAME, FOR FILE OR COMMAND.
INVDEV          ;INVALID DEVICE, CHECK DEVICE TABLE.
INVADR          ;INVALID ADDRESS, ADDRESS SHOULD BE EVEN.
CKSMER          ;LOAD (CHECKSUM) ERROR.
EOM             ;END OF MEDIUM ERROR, REACHED END OF MEDIUM BEFORE END OF FILE
DELOLD          ;TELL HIM TO DELETE OLD FILE FIRST.
DELERR          ;DELETE ERROR.
INVCOR          ;CORE ERROR.
INVSW           ;INVALID SWITCH.
POFLOW          ;PROGRAM OVERFLOW ERROR, NOT ENOUGH CORE
```

APPENDIX A XXDP RESIDENT MONITOR COMMANDS

F<CR>	SET CONSOLE FILL COUNT.
D<CR>	DIRECTORY ON THE TTY CONSOLE.
D/F<CR>	SHORT DIRECTORY ON THE TTY CONSOLE.
D/L	DIRECTORY ON THE LINE PRINTER.
D/L/F	SHORT DIRECTORY ON LINE PRINTER.
R COPY1	STARTS THE COPY1 PROGRAM
R FILENAME	STARTS INDICATED PROGRAM
L FILENAME	LOADS DESIRED PROGRAM.
S FILENAME	STARTS DESIRED PROGRAM WHICH WAS LOADED UNDER "L" COMMAND
S ADDR	STARTS PROGRAM AS SPECIFIED ADDRESS.
C FILENAME	RUNS DESIRED CHAIN TABLE
C FILENAME/QV	RUNS DESIRED CHAIN IN QUICK VERIFY
E 0<CR>	ENABLE DRIVE 0(TADP ONLY)
E 1<CR>	ENABLE DRIVE 1(TADP ONLY)

APPENDIX B. XXDP RESIDENT MONITOR ERRORS

INVCMD/SW	INVALID COMMAND AND/OR SWITCH. CHECK COMMAND JUST GIVEN.
DEVERR	DEVICE ERROR ON INPUT DEVICE.
EOM	END OF MEDIUM. OCCURS DURING INPUT OPERATIONS WHEN THE PROGRAM ATTEMPTS TO INPUT AND THE FILE IS AT AN END. SERIOUS PROBLEM. FILE IN STORAGE IS PROBABLY WIPED OUT.
INVADR	INVALID ADDRESS. MUST BE EVEN WITHIN EXISTING LOCORE AND HICORE LIMITS, AND MUST NOT BE WITHIN UPDATE PROGRAM.
CKSMER	CHECKSUM ERROR DURING "LOAD" COMMAND.
POFLO	PROGRAM TOO LARGE TO LOAD WITHIN EXISTING CORE SPACE.
INVNAM	INVALID CHARACTER TYPED FOR FILE NAME.
NEXFIL	NON-EXISTENT FILE. IF IN CHAIN MODE THE PROGRAM TO BE RUN DOES NOT HAVE .BIC EXTENSION.

APPENDIX C. UPD1/UPD1A PROGRAM COMMANDS

FILL<CR>	SETS UP TERMINAL FOR CORRECT PRINT AFTER CRLF.
CLR<CR>	CLEARS CORE BELOW UPDATE PROGRAM
XFR<CR>	PERMITS MAKING PROGRAM SELF-STARTING, OR NON SELF-STARTING.
DUMP DEV:FILNAM.EXT	WRITES MEMORY CONTENTS IN ABS FORMAT
LOAD DEV:FILNAM.EXT	LOADS ABS FORMAT PROGRAM (.BIN, .BIC)
PIP DEV1:FILNAM.EXT_DEV2:FILNAM.EXT	COPIES FILE FROM ONE DEVICE TO ANOTHER.
SAVE DEV:FILNAM.EXT	WRITES MEMORY CONTENTS ONTO CONTIGUOUS BLOCKS
MOD ADR	MODIFIES CORE CONTENTS
CORE	TYPES PROTECTION LIMITS
LOCORE ADR	ENTERS LOW PROTECTION LIMIT
HICORE ADR	ENTERS HIGH PROTECTION LIMIT
DIR DEV:	TYPES DEV DIRECTORY ON TTY
ZERO DEV:	ZEROES DEVICE DIRECTORY
BOOT DEV:	LOADS BLOCK 0 OF DEV STARTING AT LOC 000000
SAVM DEV:	WRITES 4K ONTO DEV STARTING AT BLOCK 30
START	STARTS PROGRAM AT LOC 000000
START ADR	STARTS PROGRAM AT ADR
C (CONTROL C)	RETURN TO COMMAND MODE (OPEN OUTPUT FILE IS CLOSED).
DEL DEV:FILNAM.EXT	DELETES FILE FROM DEVICE DIRECTORY.
REN DEV:NEWNAM.EXT=DEV:OLDNAM.EXT	RENAMES OLD FILE

APPENDIX D. UPD2 AND UPD3 PROGRAM COMMANDS

FILL<CR>	SETS UP TERMINAL FOR CORRECT PRINT AFTER CRLF.
CLR<CR>	CLEARS CORE BELOW UPDATE PROGRAM
XFR<CR>	PERMITS MAKING PROGRAM SELF-STARTING, OR NON SELF-STARTING.
DUMP DEV:FILNAM.EXT	WRITES MEMORY CONTENTS IN ABS FORMAT
LOAD DEV:FILNAM.EXT	LOADS ABS FORMAT PROGRAM (.BIN, .BIC)
PIP DEV1:FILNAM.EXT_DEV2:FILNAM.EXT	COPIES FILE FROM ONE DEVICE TO ANOTHER.
SAVE DEV:FILNAM.EXT	WRITES MEMORY CONTENTS ONTO CONTIGUOUS BLOCKS
GET DEV:FILNAM.EXT	LOADS CORE IMAGE PROGRAM
MOD ADR	MODIFIES CORE CONTENTS, WITHIN LIMITS
MODALL ADR	MODIFIES ANY CORE CONTENTS
CORE	TYPES PROTECTION LIMITS
LOCORE ADR	ENTERS LOW PROTECTION LIMIT
HICORE ADR	ENTERS HIGH PROTECTION LIMIT
DIR DEV:	TYPES DEV DIRECTORY ON TTY
DIRLP DEV:	TYPES DEV DIRECTORY ON LINE PRINTER.
DEL DEV:FILNAM.EXT	DELETES FILE FROM DEV DIRECTORY
REN DEV:NEWNAM.EXT_DEV:OLDNAM.EXT	RENAMES OLD FILE
ZERO DEV:	ZEROES DEVICE DIRECTORY
BOOT DEV:	LOADS BLOCK 0 OF DEV STARTING AT LOC 000000
SAVM DEV:	WRITES 4K ONTO DEV STARTING AT BLOCK 30
START	STARTS PROGRAM AT LOC 000000
START ADR	STARTS PROGRAM AT ADR
ACT	PUTS UPD2/UPD3 PROGRAM IN "ACT MODE"
NOTACT	TAKES UPD2/UPD3 PROGRAM OUT OF "ACT MODE"
FILE DEV:<DEV:FILNAM.EXT	COPIES FILE(S) FROM ONE DEVICE TO ANOTHER, DELETING FILE OF SAME NAME

	BEFORE DOING THE TRANSFER
FILEF DEV:<DEV:FILNAM.EXT	SAME AS FILE EXCEPT THAT WITH CASSETTE OR MAGTAPE FAST TRANSFERS ARE PERFORMED (NO DIR CHECKING)
FILET DEV:FILNAM.EXT	READS FILE AND CHECKS FOR DEVICE ERRORS (FILE "TEST")
FILEL DEV:FILNAM.EXT	LOADS FILES (ASSUMES ABS FORMAT) CHECKING FOR DEVICE AND CHECKSUM ERRORS
FILEG DEV:FILNAM.EXT	LOADS FILES (ASSUMES CONTIGUOUS FORMAT) CHECKING FOR DEVICE AND FILE SIZE ERRORS
FILED DEV:FILNAM.EXT	DELETES NAMED FILES
FILCMP DEV:<DEV:FILNAM.EXT	COMPARES TWO FILES AGAINST EACH OTHER ON TWO XXDP MEDIUMS.
PATCH	ENABLE THE USER TO PATCH A PROGRAM
TEXT DEV:FILNAM.EXT	CREATES TEXT FILE FOR PRINTING OR FOR COMMAND EXECUTION
PRINT DEV:FILNAM.EXT	OUTPUTS A FILE TO THE LINE PRINTER (ASSUMES IT ENDS WITH A ^Z)
TYPE DEV:FILNAM.EXT	OUTPUTS A FILE TO THE CONSOLE TERMINAL
DO DEV:FILNAM.EXT	EXECUTES A COMMAND FILE.
ASG PHYSICAL = LOGICAL	ASSIGNS A PHYSICAL DEVICE TO A LOGICAL DEVICE NAME
EOT	WRITES END OF TAPE MARK (FILE) ON MAGTAPE OR CASSETTE AFTER TAPE HAS BEEN POSITIONED.
PATCH DEV:FILNAM.EXT	ENABLES PATCHING CAPABILITIES TO A FILE ON THE XXDP MEDIA.
^C (CONTROL C)	RETURNS TO COMMAND MODE (OPEN OUTPUT FILE IS CLOSED).
^Z (CONTROL Z)	ENDS INPUT TO A TEXT FILE
*	USED FOR FILE NAMING TO MEAN "ANY" (ANY FILE NAME OR ANY FILE EXTENSION)
?	USED FOR FILE NAMING TO INDICATE A WILD CHARACTER (ANY CHARACTER WILL MATCH IT)
# OR ;	USED IN A FILE OF EXECUTABLE COMMANDS TO START A COMMENT LINE WHICH IS TO BE TYPED DURING EXECUTION
$	SAME AS # BUT CAUSES A HALT AFTER

THE COMMENT IS PRINTED

T

APPENDIX E. PERIPHERALS SUPPORTED BY UPDATE PROGRAMS

XXDP SUPPORTS THE FOLLOWING DEVICES:

PR:	PC11 HIGH SPEED PAPER TAPE READER	(UPD1, UPD1A, UPD2, UPD3)
PP.	PC11 HIGH SPEED PAPER TAPE PUNCH	(UPD1, UPD1A, UPD2, UPD3)
KB:	TTY KEYBOARD, OR LOW SPEED READER	(UPD1, UPD1A, UPD2, UPD3)
PT:	TTY PRINTER AND PUNCH	(UPD1, UPD1A, UPD2, UPD3)
DTN:	TC11 DECTAPE	(UPD1 N=0 OR 1), (UPD2, UPD3, N=0-3)
DKN:	RK11/RK05 DISK	(UPD1, UPD1A N=0 OR 1) (UPD2, UPD3, N=0-3)
MTN:	TM11/TU10 MAGTAPE 7/9 TRACK	(UPD2, UPD3, N=0-3)
CTN:	TA11 CASSETTE	(UPD1, N=0 OR 1), (UPD2, UPD3, N=0 OR 1).
DXN:	RX11/RX01 FLOPPY DISK	(UPD1, UPD1A N=0 OR 1), (UPD2, UPD3, N=0 OR 1)
MMN:	TM02/TU16 MAGTAPE	(UPD2, UPD3 ONLY, N=0-3)
DPN:	RP11C/RP02/RP03	(UPD2, UPD3, ONLY, N=0 OR 1)
DBN:	RP04 DISK	(UPD2, UPD3 ONLY, N=0 OR 1)
DSN:	RS04/RH11 DISK	(UPD2, UPD3 ONLY, N=0 OR 1)
DMN:	RK611/RK06 DISK	(UPD2, UPD3 ONLY, N=0-3)
DLN:	RL11/RLV11/RL01 DISK	(UPD3 ONLY, N=0-3)
DYN:	RX211/RX02 DISK	(UPD1A, UPD3, N=0-1)

APPENDIX F. PERIPHERALS SUPPORTED BY COPY PROGRAMS

XXDP SUPPORTS THE FOLLOWING DEVICES:

PR:	PC11 HIGH SPEED PAPER TAPE READER	(COPY1, COPY2)
PP:	PC11 HIGH SPEED PAPER TAPE PUNCH	(COPY1, COPY2)
KB:	TTY KEYBOARD, OR LOW SPEED READER	(COPY1, COPY2)
PT:	TTY PRINTER AND PUNCH	(COPY1, COPY2)
DTN:	TC11 DECTAPE	(COPY1 ONLY, N=0 OR 1)

DKN: RK11/RK05 DISK (COPY1 ONLY, N=0-3)

MTN: TM11/TU10 MAGTAPE 7/9 TRACK (COPY1 ONLY, N=0-3)

CTN: TA11 CASSETTE (COPY1 ONLY, N=0 OR 1)

DXN: RX11/RX01 FLOPPY DISK (COPY1 ONLY, N=0 OR 1)

MMN: TM02/TU16 MAGTAPE (COPY1 ONLY, N=0-3)

DPN: RP11C/RP02/RP03 (COPY2, ONLY, N=0 OR 1)

DBN: RP04 DISK (COPY2, ONLY, N=0 OR 1)

DSN: RS04/RH11 DISK (COPY2, ONLY, N=0 OR 1)

DMN: RK611/RK06 DISK (COPY2, ONLY, N=0-3)

DLN: RL11/RLV11/RL01 DISK (COPY2 ONLY, N=0-3)

DYN: RX211/RX02 DISK (COPY1 ONLY, N=0-1)

APPENDIX X. PROGRAM NAMING CONVENTIONS

THE FOLLOWING PROGRAM NAMING CONVENTION HAS BEEN USED FOR XXDP FILES STORED ON A LOAD MEDIA. THE USE OF THIS CONVENTION WILL PERMIT USERS TO DETERMINE BOTH THE VERSION AND THE DEPO LEVEL OF THE FILES. USE OF THIS CONVENTION WHEN PROGRAMS ARE UPDATED IN THE FIELD IS HIGHLY RECOMMENDED.

```
C       ZFPKA#

|       || |||
|       || |||-----# = INDICATES DEPO LEVEL
|       || ||       0 = INDICATES NO DEPO ISSUED
|       || ||------A THRU Z = REVISION DESIGNATION
|       || |-------A THRU Z = PROGRAM DESIGNATION
|       ||          0 THRU 9 = OVERLAY DESIGNATION
|       ||---------2 DIGITS = OPTION DESIGNATION
|       |----------A = 11/05,15,20 PROCESSORS
|                   B = 11/40 PROCESSOR
|                   C = 11/45 PROCESSOR
|                   Z = ALL PROCESSORS
|
|-----------------D INDICATES A DIAGNOSTIC PROGRAM, AND IS NOT USED
                    IN NAMING A PROGRAM.
```

.BIN EXTENSION USED TO STORE PROGRAM IN ABS FORMAT
.SAV EXTENSION USED TO STORE PROGRAM IN CORE IMAGE FORMAT
.BIC EXTENSION INDICATES ABS FORMAT CHAINABLE PROGRAM.
%