SOFTWARE COVER SHEET

APPLICATION				REVISION	
TOP ASSY	PROD LINE	APPROVED	DATE	DESCRIPTION	REV
	PDS	[illegible]	[illegible]	INITIAL RELEASE - 777	A
		[illegible]	1-19-79	DCN # IMP- 3094	A'
		[illegible]	2-9-79	DCN # IMP- 3119	B
		B. Eldridge	4/4/80	DCN # 3497	C

CLASSIFICATION

FIRMWARE ☐ SOFTWARE ☒ PROGRAM FORMAT TEST PROCEDURE ☐

		DATE
ORIGINATOR	[illegible]	[illegible]
CHK	[illegible]	9/7/79
APPD	[illegible]	[illegible]
APPD	[illegible]	[illegible]

TITLE COPS SOFTWARE PACKAGE

PROG NO 440305834

REV C

National Semiconductor Corporation

SHEET 1 OF 37

1. OVERVIEW:

THIS IS A DETAILED DESCRIPTION OF PDS SOFTWARE, INCLUDING A DESCRIPTION OF THE FUNCTION OF EACH PROGRAM.

SEE THE INDIVIDUAL 43030XXXX DOCUMENTATION FOR FURTHER DETAILS OF THE PROGRAMS.

NAME	DESCRIPTION AND ERRORS
FM	FILE MANAGER PROGRAM FOR LISTING DISK DIRECTORIES, COPYING FILES, ETC.
EDIT	TEXT EDITOR FOR EDITING TEXT FILES
ASM	COP CROSS ASSEMBLER
DSKIT	DISK INITIALIZATION AND TEST PROGRAM
COPMON	COP MONITOR PROGRAM FOR LOADING AND ALTERING SHARED MEMORY, MAKING PROMS, SINGLE STEPPING, BREAKPOINTING, AND TRACING COP CHIPS, ETC.
LIST	LISTS FILES ON CONSOLE OR PRINTER
XREF	GENERATES CROSS REFERENCE LISTING OF COP LABELS FROM SOURCE FILE.
MASKTR	GENERATES A DISK FOR TRANSFER OF CUSTOMER DEVELOPED PROGRAMS TO NSC FOR MASK GENERATION.
MDIAG	MEMORY DIAGNOSTIC FOR PDS SYSTEM.

THERE ARE SIX (6) INCLUDE FILES WHICH ARE INCLUDED IN ALL PROGRAMS FOR THE PDS SYSTEM. THEY ARE:

1) .INCLD LST
2) .INCLD MACLIB
3) .INCLD GENERAL
4) .INCLD SYSRAM
5) .INCLD HWADDR
6) .INCLD ENTRY

THE INCLUDES DEFINE ALL THE GENERALLY USED DEFINITIONS, ALL THE LIST DIRECTIVES, THE HARDWARE ADDRESSES, THE ENTRY POINTS OF ALL UTILITY PROGRAMS, THE RESERVED MEMORY LOCATIONS AND THE GENERALLY USED MACROS.

ALL PDS PROGRAMS ARE SET UP IN THE FORMAT

SHOWN BELOW. THE MACROS DEFINED IN MACLIB AND ENTRY USE INFORMATION FROM THE FORMAT TO ASSIGN THE PROPER REV LEVEL, PROGRAM NAME, DESCRIPTION, CHANNGES SINCE LAST REVISION, AND ENTRY POINTS TO THE PROGRAMS.

OTHER MACROS ASSIGN BASE PAGE VARIABLES START OF BASE PAGE, END OF BASE PAGE, BEGINNING OF PROGRAM, END OF PROGRAM, ASSIGNMENT OF SYSRAM LOCATIONS.

```
        .LIST   OFFFF
        .INCLD  LST
        .INCLD  MACLIB
        .NAME   NEWNAME,SOFT,POXXXX,'TODAYS DATE',A
; (NONE)
; EDIT LINE 4, AND DELETE THIS LINE

; DESCRIPTION OF PROGRAM:
; ADD YOUR OWN DESCRIPTION
;
        .INCLD  GENERAL
        .PAGE   'ADDITIONAL CONSTANT DEFINITIONS'
; DEFINE YOUR OWN CONSTANTS
;
        .INCLD  SYSRAM
        .INCLD  HWADDR
        .INCLD  ENTRY

        ENTRY POINTS TO UTILITY ROUTINES


        START OF BASE PAGE
;
; BASE PAGE VARIABLES GO HERE
;
        END OF BASE PAGE

        START OF PROGRAM

; FIRST LINE OF PROGRAM, MUST BE ENTRY LINE. THE NAME IN LINE
; 4 IS AUTOMATICALLY MADE A LABEL HERE, LOCATION 100
; THE REST OF YOUR PROGRAM GOES HERE


        END OF PROGRAM

        .END    NEWNAME

; EDIT THE .END LINE, USE YOUR PROGRAM NAME
```

2. CONVENTIONS AND STANDARDS:

```
THE FOLLOWING VARIABLES CONTROL THE LISTING OF INCLUDE
FILES, ETC. THESE LIST PARAMETERS ARE USED AS
.INCLUDES IN ALL PDS PROGRAMS. (.INCLD LST)

MACLIB   =      001              MACLIB LIST
GENERAL  =      001            ; GENERAL LIST
SYSRAM   =      001            ; SYSRAM LIST
HWADDR   =      001            ; HWADDR LIST
ENTRY    =      001            ; ENTRY LIST
COPCOM   =      001            ; COPCOM LIST
PROM     =      021            ; PROM LIST
PUTCMD   =      021            ; PUTCMD LIST
FRMEP    =      001            ; FIRMWARE ENTRY POINT LIST
FRMEJ    =      001            ; FIRMWARE ENTRY JUMP LIST
XBTD     =      021            ;BREAKPOINT/TRACE LIST
ELAPTI   =      021            ; ELAPSED TIME LIST
XFPIN    =      021            ; COPMON FP,I/O LIST
MODCOP   =      021            ; MODIFY LIST
UNASM    =      021            ; UNASM LIST
XLIST    =      001            ; OTHER LIST

        .MSTAT  0               ; NO MICRO EXPANSION

        .PAGE   'ASSIGNMENT OF CONSTANTS'
        THE FOLLOWING .INCLUDE FILE DEFINES MOST OF
        THE GENERALLY USED EQUATES IN ALL PDS PROGRAMS
        THE FILE IS INCLUDED IN ALL PDS PROGRAMS.
        (.INCLD GENERAL)

GENERAL REGISTERS

AC0      =      0
AC1      =      1
AC2      =      2
AC3      =      3

R0       =      AC0
R1       =      AC1
R2       =      AC2
R3       =      AC3

CONDITIONAL CODES FOR THE BOC INSTRUCTION

INT      =      0               ; INTERRUPT LINE EQUAL TO ONE
ZRO      =      1               ; AC0 EQUAL TO ZERO
POS      =      2               ; AC0 IS POSITIVE OR ZERO
```

```
BITO     =     3          ; BIT 0 OF ACO IS ONE
BIT1     =     4          ; BIT 1 OF ACO IS ONE
NZRO     =     5          ; ACO NOT EQUAL TO ZERO
CPINT    =     6          ; CONTROL PANEL INTERRUPT
STFL     =     8          ; STACK FULL LINE EQUAL ONE
INEN     =     9          ; INTERRUPT ENABLE EQUAL ONE
CRY      =     10         ; CARRY EQUAL TO ONE
OVFL     =     10         ; OVERFLOW EQUAL TO ONE
LEZ      =     11         ; ACO LESS OR EQUAL TO ZERO
SERBIT   =     12         ; CONSOLE INPUT FOR BAUDET
SELT     =     13         ; SEL FLAG SET
NRDY     =     14         ; DISK DATA NOT READY

  CONTROL FLAGS

INTEN    =     1          ; INTERRUPT ENABLE
SEL      =     2          ; SELECT FLAG FOR SHIFTS

;
  STATUS FLAGS

CY       =     X'2000     ; CARRY BIT
OV       =     X'4000     ; OVERFLOW BIT
L        =     X'8000     ; LINK BIT FOR SHIFTS

  ASCII CHARACTER DEFINITIONS

CR       =     X'D        ; CARRIAGE RETURN
LF       =     X'A        ; LINE FEED
CNTLA    =     X'1        ; CONTROL/A
SHFTO    =     X'5F       ; SHIFT/O (BACK ARROW ON TTY)
NULL     =     X'0
ESCAPE   =     X'1B
RUBOUT   =     X'7F
B        =     X'20       ; BLANK OR SPACE
                          ;  FOR SOURCE FILES)

  PROMPT CHARACTERS USED BY EACH MAIN PROGRAM IN CDS

ZASM     =     'A>'       ; ASM
ZCOPMON  =     'C>'       ; COPMON
ZDSKIT   =     'D>'       ; DSKIT
ZEDIT    =     'E>'       ; EDIT
ZFM      =     'F>'       ; FM
ZPATCH   =     'H>'       ; PATCH
ZCDSASM  =     'I>'       ; CDSASM
ZLINK    =     'K>'       ; LINK
ZLIST    =     'L>'       ; LIST
ZMDIAG   =     'M>'       ; MEMORY DIAGNOSTICS
```

```
ZPROMPT =        'P>'          ; PROMPT
ZXREF   =        'R>'          ; XREF
ZMASKTR =        'T'>          ; MASK TRANSFER
ZEXEC   =        'X>'          ; EXEC
```

ALL DISK FILE OPERATIONS MAKE USE OF A 'FILE STATUS TABLE' (FST) WHICH DEFINES THE VOLUME NAME, FILE NAME, FILE MODIFIER, ETC. OF A FILE. FOLLOWING IS A DEFINITION OF THE CONTENTS OF THE FST:

```
FSV1    =        0             ; VOLUME NAME, SUPPLIED BY
FSV2    =        1             ;  ROUTINE 'PGFR'
FSV3    =        2
FSV4    =        3
FSN1    =        4             ; FILE NAME, SUPPLIED BY
FSN2    =        5             ;  ROUTINE 'PGFR'
FSN3    =        6
FSN4    =        7
FSM1    =        8             ; FILE MODIFIER, SUPPLIED BY
FSM2    =        9             ;  ROUTINE 'PGFR', DEFAULT
                               ;  SUPPLIED BY USER BEFORE
                               ;  CALLING PGFR
FSIT    =        9             ; INTERNAL TYPE (BITS 0-7)
                               ;  SUPPLIED BY USER. IF = 0,
                               ;  ROUTINE 'DISER' RETURNS
                               ;  ACTUAL INTERNAL TYPE IF
                               ;  FILE IS FOUND
FSPL    =        10            ; PROTECT LEVEL (BITS 12-15)
                               ;  SUPPLIED BY USER WHEN
                               ;  FILE IS CREATED.
                               ;  OTHERWISE IT IS NOT USED.
FSCT    =        11            ; CHANNEL TABLE ADDRESS
                               ;  SUPPLIED BY 'OPEN'
                               ;  ROUTINES
FSBA    =        12            ; BUFFER ADDRESS, SUPPLIED
                               ;  BY USER (256 WORDS)
FSST    =        13            ; DISK I/O STATUS, RETURNED
                               ;  WHEN ERROR OCCURS, SEE
                               ;  DISKIO STATUS RETURN CODES
                               ;  BELOW FOR DETAILS ON LEFT
                               ;  BYTE. SEE FIOSYS ERROR
                               ;  CODES BELOW FOR DETAILS
                               ;  ON RIGHT BYTE.
FSVN    =        14            ; FILE VERSION NUMBER,
                               ;  SUPPLIED BY 'DISER',
                               ;  INCREMENTED BY 'OPENW' OR
                               ;  'OPENM'
```

FILE STATUS TABLE SIZE

```
FSSIZE  =       15              ; WORDS/FILE STATUS TABLE
```

WHEN A FILE IS OPENED BY ANY OF THE ROUTINES 'OPENR', 'OPENW' OR 'OPENM', AN INTERNAL TABLE IS ASSIGNED TO THE FST. THIS INTERNAL TABLE, CALLED A 'CHANNEL TABLE' (CT), IS USED FOR ALL DISK READ AND WRITE OPERATIONS. A MAXIMUM OF 5 CHANNEL TABLES ARE ALLOWED AT ONE TIME. WHEN A FILE IS CLOSED BY THE 'CLOSE' ROUTINE, ITS CHANNEL TABLE(S) IS RELEASED, MAKING IT AVAILABLE FOR THE NEXT 'OPEN' OPERATION. FOLLOWING ANY DISK READ OR WRITE OPERATION, THE DISK I/O STATUS IS INDICATED IN BOTH THE FST AND THE CHANNEL TABLE. FOLLOWING IS A DEFINITION OF THE CONTENTS OF THE CT:

```
CTCT    =       0               ; CHANNEL TABLE ADDRESS
CTV1    =       1               ; VOLUME NAME
CTV2    =       2
CTV3    =       3
CTV4    =       4
CTN1    =       5               ; FILE NAME
CTN2    =       6
CTN3    =       7
CTN4    =       8
CTM1    =       9               ; FILE MODIFIER
CTM2    =       10
CTFG    =       10              ; FLAGS (BITS 0-7), SEE
                                ;  CHANNEL TABLE FLAGS BELOW
                                ;  (INDICATE OPEN FOR READ,
                                ;  WRITE, OR MODIFY)
CTBS    =       11              ; BEGINNING SECTOR
CTLS    =       12              ; LAST SECTOR OPERATED ON
CTBP    =       13              ; BUFFER POINTER
CTOP    =       14              ; DISK I/O COMMAND
CTNS    =       15              ; NEXT SECTOR
CTBA    =       16              ; BUFFER ADDRESS
CTST    =       17              ; DISK I/O STATUS
```

CHANNEL TABLE SIZE

```
CTSIZE  =       18              ; WORDS/CHANNEL TABLE
```

CHANNEL TABLE FLAGS

```
RD      =       1               ; READ STATUS FLAG
WR      =       2               ; WRITE STATUS FLAG
MD      =       4               ; MODIFY STATUS FLAG
```

MAXIMUM NUMBER OF CHANNEL TABLES

```
MAXCHN  =       5
```

EACH DISKETTE HAS A TABLE, LOCATED IN SECTOR 0, CALLED THE 'NEXT SECTOR TABLE', OR 'NST'. THE FIRST 154 WORDS OF THIS TABLE ARE LINKS TO THE NEXT LOGICAL EXTENT OF THE DISKETTE. AN EXTENT IS A GROUP OF 4 CONSECUTIVE SECTORS WHOSE SECTOR ADDRESS IS THE SAME EXCEPT FOR THE LAST TWO BITS. THERE ARE 154 EXTENTS ON EACH DISKETTE. THESE EXTENTS ARE NOT USED IN CONSECUTIVE ORDER, BUT RATHER ARE ASSIGNED TO INDIVIDUAL FILES AS THEY ARE NEEDED. THE NST DEFINES THE ORDER THAT EXTENTS ARE TO BE ACCESSED IN WHEN ACCESSING A FILE. THE NST ALSO CONTAINS THE DIRECTORY ADDRESS AND DISKETTE VOLUME NAME AND HEADER STRING. FOLLOWING ARE DEFINITIONS OF THESE TERMS:

```
NSTDF   =       154             ; 155TH WORD OF NST IS FIRST
                                ;  DIRECTORY SECTOR NUMBER
NSTVOL  =       155             ; START OF DISK VOLUME NAME
NSTV1   =       0               ; VOLUME NAME OFFSETS
NSTV2   =       1
NSTV3   =       2
NSTV4   =       3
NSTHDR  =       159             ; START OF DISK HEADER STRING
```

DISK HEADER STRING LENGTH

```
HDRLEN  =       40              ; HEADER IS 40 CHARS MAXIMUM
```

NST SIZE

```
NSTSIZ  =       179             ; WORDS USED BY NST
```

EACH DISKETTE HAS A DIRECTORY CONTAINING INFORMATION ABOUT BAD AND AVAILABLE SECTORS, AND FILE DATA FOR EACH FILE ON THE DISKETTE. FOLLOWING IS A DESCRIPTION OF THE DIRECTORY FORMAT:

```
DRV1    =       1               ; 2ND WORD OF DIRECTORY IS
DRV2    =       2               ;  START OF DISK VOLUME NAME
DRV3    =       3
DRV4    =       4
BSF     =       6               ; FIRST BAD SECTOR
BSL     =       7               ; LAST BAD SECTOR
BSCNT   =       8               ; BAD SECTOR COUNT
NAS     =       16              ; NEXT AVAILABLE SECTOR
LAS     =       17              ; LAST AVAILABLE SECTOR
ASCNT   =       18              ; AVAILABLE SECTOR COUNT
```

```
DRF     =       26              ; DIRECTORY STARTING SECTOR
DRL     =       27              ; DIRECTORY ENDING SECTOR
DRCNT   =       28              ; DIRECTORY SIZE (SECTORS)
DRFE    =       30              ; FIRST FILE ENTRY IN
                                ;  DIRECTORY STARTS AT THE
                                ;  31ST WORD AND HAS A FORMAT
                                ;  AS SHOWN BELOW.
```

EACH FILE HAS AN ENTRY IN THE DISKETTE DIRECTORY. THE ENTRY IS CREATED WHEN THE FILE IS 'CLOSED' FOR THE FIRST TIME. THE FORMAT OF EACH ENTRY IS AS FOLLOWS:

```
DRN1    =       0               ; FILE NAME
DRN2    =       1
DRN3    =       2
DRN4    =       3
DRM1    =       4               ; FILE MODIFIER
DRM2    =       5
DRIT    =       5               ; INTERNAL TYPE (BITS 0-7)
DRBS    =       6               ; STARTING SECTOR NUMBER
DRES    =       7               ; ENDING SECTOR NUMBER
DRDF    =       8               ; DELETE FLAG CODE (BIT 15)
DRPL    =       8               ; PROTECT LEVEL (BITS 11-14)
DRTS    =       8               ; TOTAL # SECTORS USED (BITS
                                ;  0-9)
DRVN    =       9               ; FILE VERSION NUMBER
```

THE END OF THE DIRECTORY IS INDICATED BY A VALUE FOR DRN1 OF X'FFFF. THE NEXT SECTOR TABLE MARKS THE END OF THE BAD SECTOR LIST, THE END OF THE FREE SECTOR LIST, AND THE END OF EACH FILE WITH A SECTOR NUMBER OF X'FFFF.

DIRECTORY ENTRY SIZE

```
DESIZ   =       10              ; WORDS/DIRECTORY ENTRY
DEPS    =       25              ; DIRECTORY ENTRIES PER SECTOR
```

SPECIAL DISK FILE DEFINITIONS

```
FS      =       X'1C            ; FILE SEPARATOR (EOF MARK)
```

INTERNAL TYPE CODES FOR FILES

```
ITUNIV  =       0               ; UNIVERSAL
ITLM    =       1               ; LOAD MODULE
ITMP    =       2               ; MAIN PROGRAM
ITOV    =       3               ; OVERLAY
```

```
ITBL     =      4              ; BLOCK
ITSYM    =      5              ; SYMBOLIC
ITSYT    =      6              ; SYSTEM
ITDATA   =      7              ; DATA
```

FLOPPY DISK SPECIFICATIONS

```
TRACKS   =      77             ; 77 TRACKS/DISKETTE
SECTORS  =      8              ; 8 SECTORS/TRACK
EXTENT   =      4              ; 4 SECTORS/EXTENT
SECTSZ   =      256            ; 256 WORDS/SECTOR

SHIFT    =      2              ; EXTENT = 2**SHIFT

SYNC     =      X'AAAA         ; DISK SYNC WORD

DSKADR   =      X'7D06         ; DISK HARDWARE ADDRESS
DRESET   =      X'7D02         ; ADDRESS FOR MASTER RESET

DSKSTAT  =      0              ; STATUS OFFSET FROM DISK ADR
DSKDATA  =      -1             ; DATA OFFSET FROM DISK ADR
```

DISK I/O PARAMETER LIST DEFINITIONS

```
OPCODE   =      0              ; READ/WRITE COMMAND (BELOW)
PSECT    =      1              ; PHYSICAL SECTOR
BUFADR   =      2              ; BUFFER ADDRESS
STATUS   =      3              ; DISK STATUS RETURN
```

DISK I/O PARAMETER LIST SIZE

```
PLSIZE   =      4              ; WORDS/PARAMETER LIST
```

DISK I/O COMMANDS

```
WCODE    =      1              ; OPCODE FOR DISK WRITE
RCODE    =      2              ; OPCODE FOR DISK READ
```

MAXIMUM ALLOWED NUMBER OF BAD SECTORS WHEN INITIALIZING
A DISKETTE

```
BSMAX    =      150
```

DISK INTERFACE STATUS SIGNAL DEFINITIONS

```
DUROPN   =      X'8            ; DRIVE DOOR WAS OPENED
```

ROUTINES 'DISK1' AND 'DISK2' IN DISKIO RETURN A STATUS CODE WORD IN THE PARAMETER LIST ('STATUS'), WHICH DESCRIBES THE RESULTS OF THE DISK I/O OPERATION. THIS STATUS WORD HAS THE FOLLOWING FORMAT:

UPPER BYTE = TOTAL NUMBER OF ERRORS ENCOUNTERED
LOWER BYTE = DISKIO STATUS RETURN CODE

DISKIO STATUS RETURN CODE BIT DEFINITIONS:

```
NOERR   =   1     ; NO ERRORS
DNRDY   =   2     ; DRIVE NOT READY
ADDRER  =   4     ; ADDRESSING ERROR
MISYNC  =   8     ; MISSING SYNC/WRITE PROTECT
WRERR   =   16    ; WRITE ERROR (NO CRC VERIFY)
RDERR   =   32    ; READ ERROR (NO CRC VERIFY)
CMDER   =   64    ; ILLEGAL DISK COMMAND
```

FIOSYS ERROR CODE DEFINITIONS

```
ER01    =   X'01  ; WRONG DISKETTE VOLUME
ER02    =   X'02  ; DISK FULL, DIRECTORY FULL
ER03    =   X'03  ; UNABLE READ NST
ER04    =   X'04  ; UNABLE TO ACCESS DIRECTORY
ER05    =   X'05  ; UNABLE TO WRITE DIRECTORY
ER06    =   X'06  ; DIR DOES NOT MATCH NST
ER07    =   X'07  ; DRIVE NOT ACTIVE
ER08    =   X'08  ; ATTEMPT TO READ PAST THE
                  ;  END OF THE DIRECTORY
ER09    =   X'09  ; INVALID CHANNEL TABLE
ER10    =   X'10  ; FILE NOT OPEN FOR READ
ER11    =   X'11  ; DISK READ ERROR ON FILE
ER12    =   X'12  ; ATTEMPT TO READ PASS EOF
ER13    =   X'13  ; FILE ALREADY OPEN
ER14    =   X'14  ; FILE NOT FOUND
ER15    =   X'15  ; NO CHANNEL TABLE AVAILABLE
ER16    =   X'16  ; FILE NOT OPEN FOR WRITE
ER17    =   X'17  ; ATTEMPT TO READ/WRITE PAST
                  ;  THE END OF THE DISK
ER18    =   X'18  ; DISK WRITE ERROR ON FILE
ER19    =   X'19  ; FILE NOT OPEN FOR MODIFY
ER20    =   X'20  ; NO PERMISSION TO DEL FILE
ER21    =   X'21  ; NO PERMISSION TO MOD FILE
ER22    =   X'22  ; UNABLE TO UPDATE NST
ER23    =   X'23  ; UNABLE TO FIND END OF DIR
ER24    =   X'24  ; SYNTAX ERROR IN FILE NAME
ER25    =   X'25  ; TOO MANY VOLUMES
ER26    =   X'26  ; ILLEGAL DEVICE
```

```
NUMERRS =       X'26            ; NUMBER OF ERROR CODES

  DEFAULT PARAMETERS FOR COP DEVELOPMENT SYSTEM:

  DEFAULT TAB STOPS

DTAB1   =       9
DTAB2   =       17
DTAB3   =       33

  CARRIAGE WIDTHS

DWIDTH  =       72              ; INITIAL WIDTH
MNWDTH  =       10              ; MINIMUM WIDTH
MXWDTH  =       80              ; MAXIMUM WIDTH

  SERIAL I/O PARAMETER AND CONTROL WORDS

PMODE   =       0FA             ; 150/9600 PARITY MODE WORD
NPMODE  =       0EE             ; 150/9600 NO PARITY MODE
PTMODE  =       0FE             ; 110 PARITY MODE WORD
NPTMOD  =       0EE             ; 110 NO PARITY MODE WORD
UCMND1  =       037             ; UART CMND WORD
RS32    =       RSBAUD/32       ; RS232 TYPE/32
TY32    =       TYBAUD/32       ; CUR LOOP TYPE/32
CRDL30  =       200             ; 150/300 BAUD CR/LF DEFAULT
LFDL30  =       40              ;  DELAYS IN MILLISECONDS,
                                ;  START UP ONLY

  INITIAL CONSOLE SET UP PARAMETERS

DCNBD   =       9600            ; BAUD RATE
DCNTYP  =       RS32            ; TYPE
DCNMOD  =       NPMODE          ; MODE

  INITIAL PRINTER SETUP PARAMETERS

DPRBD   =       1200            ; BAUD
DPRTYP  =       RS32            ; TYPE
DPRMOD  =       NPMODE          ; MODE

  PERIPHERAL DEVICE MNEMONICS

MNEMPT  =       'PT'            ; PAPER TAPE
```

```
MNEMCN   =        'CN'              ; CONSOLE
MNEMPR   =        'PR'              ; PRINTER

  PERIPHERAL DEVICE CODES

PTDEV    =        X'2               ; PAPER TAPE
CNDEV    =        X'4               ; CONSOLE
PRTDEV   =        X'8000            ; PRINTER

  DEFAULT COP CHIP NUMBER

DCHIP    =        X'420

  INITIAL VALUES FOR TARGET CARD FLAGS

DTCTAC   =        0                 ; TCTAC
DOCNTR   =        0                 ; OCNTR
DBPFLG   =        0                 ; BPFLG
DFLAGS   =        2                 ; FLAGS
DBPDEX   =        0                 ; BPDEX

  MAXIMUM NUMBER OF BREAKPOINTS ALLOWED IN COPMON

NBRCON   =        10

        THE FOLLOWING INCLUDE FILE CONTAINS MOST OF
 THE MACROS USED BY THE PDS PROGRAMS. (.INCLD MACLIB)

        .PAGE    'MACRO DEFINITIONS'
;
; MACRO FOR GENERATING COMMENTS AND ASSIGNING PROGRAM NAME
;
        .MACRO   .NAME,MNAME,MWARE,PARTNUM,MDATE,MREV
        .MICRO   PGMNM,1,8,*MNAME*
        .MICRO   PGMREV,1,1,*MREV*
        .LIST    XLIST!OD
;
        .TITLE   MNAME,' PARTNUM^MREV MDATE'
;
; PROGRAM NAME:                          MNAME
;
; SOURCE LANGUAGE OF THIS PROGRAM:  IMP MACRO-CROSS-ASSEMBLER
;                                   RUNNING ON PACE.
;
; OBJECT LANGUAGE OF THIS PROGRAM:  IMP16C/200A MACHINE CODE
;
```

```
;     MNAME IS A MWARE^WARE PROGRAM WHICH IS INTENDED FOR USE
; WITH THE NATIONAL SEMICONDUCTOR 'COP DEVELOPMENT SYSTEM'
; (CDS), WHICH CONTAINS AN IMP16C/200A MICROCOMPUTER BOARD.
; FOR ADDITIONAL INFORMATION ON THIS DEVELOPMENT SYSTEM, SEE
; 'COP DEVELOPMENT SYSTEM USER'S MANUAL'.
;
; REVISION LEVEL (A=NO REVISIONS, B=FIRST REVISION, ETC.): MREV
;
; DATE OF THIS REVISION:          MDATE
;
; FOLLOWING IS A SUMMARY OF THE REVISIONS MADE IN EACH
; REVISION LEVEL OF THIS PROGRAM FOLLOWING REVISION A:
;
        .LIST   XLIST
        .ENDM
;
; MACRO FOR GENERATING PROGRAM PROMPT AND SIGNON LINE
;
        .MACRO  PRINT,STRING
        .MSTAT  1
        .IFC    'STRING' EQ 'SIGNON'
        MESSAGE 1,'"PGMNM",REV:"PGMREV","PGMDAT"'
        OUTPUT  'NEW LINE'
        .ELSE
        .IFC    'STRING' EQ 'PROMPT'
        OUTPUT  'NEW LINE'
        JSR     CNSTUP
        LD      AC0,.+3
        JSR     PUT2C
        JMP     .+2
        .WORD   Z^"PGMNM"
        .ELSE
        .ERROR  'PRINT'
        .ENDIF
        .ENDIF
        .MSTAT  0
        .ENDM
;
; MACRO FOR SETTING START OF BASE PAGE AND PROGRAM
;
        .MACRO  START,DUMMY,MBP
        .MSTAT  1
        .LIST   XLIST!OD
        .IFC    'MBP' EQ 'BASE'
        .PAGE   'BASE PAGE DEFINITIONS'
        .ASECT
;
        .=010                   ; BASE PAGE STARTS AT 010
        .ELSE
        .IFC    'MBP' EQ 'PROGRAM'
        .PAGE   'START OF PROGRAM'
        .LOCAL
```

```
        .ASECT
;
        .=0100                          ; PROGRAM STARTS AT 0100
;
"PGMNM":
        .ELSE
        .ERROR   'START'
        .ENDIF
        .ENDIF
        .LIST    XLIST
        .MSTAT   0
        .ENDM
;
; MACRO FOR SETTING END OF BASE PAGE AND PROGRAM
;
        .MACRO   END,DUMMY,MBP
        .IFC     'MBP' EQ 'BASE'
        EASECT=.                        ;SAVE LOC CTR FOR USE BELOW
        .ELSE
        .IFC     'MBP' EQ 'PROGRAM'
        .BSECT                          ; SET UP BASE PAGE LOC CTR
        .=.+EASECT
        .MSTAT   1
        .ELSE
        .ERROR   'END'
        .ENDIF
        .ENDIF
        .ENDM
;
;  MACRO FOR OUTPUT CHARACTER ROUTINE
;
        .MACRO   OUTPUT,STRING
        .IFC     'STRING' EQ 'NEW LINE'
        JSR      CRLF
        .ELSE

        .IFC     'STRING' EQ '       '
        LI       AC0,6
        JSR      PUTB
        .ELSE
        .IFC     'STRING' EQ '      '
        LI       AC0,5
        JSR      PUTB
        .ELSE
        .IFC     'STRING' EQ '     '
        LI       AC0,4
        JSR      PUTB
        .ELSE
        .IFC     'STRING' EQ '    '
        LI       AC0,3
        JSR      PUTB
        .ELSE
```

```
        .IFC    'STRING' EQ '  '
        LI      AC0,2
        JSR     PUTB
        .ELSE

        .IFC    'STRING' EQ 'FORM FEED'
        LI      AC0,X'0C
        JSR     PUTC
        .ELSE
        LI      AC0,'STRING'/256
        JSR     PUTC
        .ENDIF
        .ENDIF
        .ENDIF
        .ENDIF
        .ENDIF
        .ENDIF
        .ENDIF
        .ENDIF
        .ENDM
;
;   MACRO FOR OUTPUT MESSAGE ROUTINE
;
        .MACRO  MESSAGE,NEWLIN,STRING
        .MLOC   MESS,OUTPUT
        JMP     OUTPUT
MESS:
        .IF     NEWLIN > 0      ; CR-LF PRIOR TO MESSAGE
        .DO     NEWLIN
        .WORD   0D0A
        .ENDDO
        .ENDIF
        .ASCII  'STRING'
        .WORD   0
OUTPUT:
        JSR     MESG2
        .WORD   MESS
        .ENDM
;
; MACRO FOR OUTPUT ERROR MESSAGE
;
        .MACRO  ERROR,STRING
        JSR     CNSTUP          ; SET UP FOR CONSOLE OUTPUT
        MESSAGE 1,'STRING'
        JMP     PROMPT
        .ENDM
;
;   MACRO FOR FRONT PANEL MESSAGE ROUTINE
;
        .MACRO  PNLMSG,STRING
        .MLOC   MESS,OUTPUT
        .MSTAT  1
        .MICRO  SHORT,8,1,*STRING*
```

```
        .MICRO   LONG,9,1,*STRING*
        .IFC     '"SHORT"' EQ ''
        .ERROR   'STRING TOO SHORT'
        .ELSE
        .IFC     '"LONG"' NE ''
        .ERROR   'STRING TOO LONG'
        .ELSE
        JMP      OUTPUT
MESS:
        .ASCII   'STRING'
OUTPUT:
        JSR      FPOUT
        .WORD    MESS
        .ENDIF
        .ENDIF
        .MDEL    SHORT,LONG
        .MSTAT   0
        .ENDM
;
; MACRO FOR FRONT PANEL MESSAGE WITH DECIMAL POINT
;
        .MACRO   COPPNL,PSTR
        .MLOC    MESS,OUTPUT
        JMP      OUTPUT
MESS:   .PNLASC  'PSTR'
OUTPUT: JSR      MSGOUT
        .WORD    MESS
        .ENDM
;
; MACRO FOR ERROR MESSAGE TO CONSOLE AND FRONT PANEL
;
        .MACRO   CPMSG,ST1,ST2
        .MLOC    MESS
        JSR      MSGOUT
        .WORD    MESS
        ERROR    'ST2'
MESS:   .PNLASC  'ST1'
        .ENDM
;
; MACRO FOR DISK I/O
;
        .MACRO   WAIT,STRING
        .IFC     'STRING' EQ 'UNTIL DATA IS READY'

        BOC      NRDY,.

        .ELSE
        .IFC     'STRING' EQ 'UNTIL DATA HAS CLEARED'

        BOC      NRDY,.+2
        JMP      .-1
```

```
        .ELSE
        .ERROR   'WAIT'
        .ENDIF
        .ENDIF
        .ENDM
;
; MACROS FOR THE REFERENCED FIOSYS UTILITY ROUTINES
;
        .MACRO   CLEAR
        JSR      CLRHS              ; CLEAR HARDWARE STACK
        .ENDM
;
        .MACRO   RESET
        JSR      RESETD             ; RESET DISK STATUS
        .ENDM
;
        .MACRO   REPORT
        LD       AC2,=#4
        JSR      DERROR             ; REPORT DISK ERRORS
        .ENDM
;
        .MACRO   SEARCH
        JSR      DISER              ; DIRECTORY FILE SEARCH
        JMP      DSKERR             ; (?? DISK ERRORS ??)
        JMP      DSKERR             ; (FILE NOT FOUND)
        .ENDM
;
        .MACRO   OPEN
        LD       AC2,=#2
        .IFC     '#4' EQ 'READ'
        JSR      OPENR              ; OPEN FILE FOR READ
        .ELSE
        .IFC     '#4' EQ 'WRITE'
        JSR      OPENW              ; OPEN FILE FOR WRITE
        .ELSE
        .IFC     '#4' EQ 'MODIFY'
        JSR      OPENM              ; OPEN FILE FOR MODIFY
        .ELSE
        .ERROR   'OPEN'
        .ENDIF
        .ENDIF
        .ENDIF
        JMP      DSKERR             ; DISK ERRORS
        .ENDM
;
        .MACRO   CLOSE
        LD       AC2,=#2
        JSR      CLOSE              ; CLOSE FILE
        JMP      DSKERR             ; DISK ERRORS
        .ENDM
;
        .MACRO   READ
```

```
LD      AC2,=#4
.IFC    '#1' EQ 'SECTOR'
JSR     RCSECT          ; READ NEXT SECTOR OF FILE
.ELSE
.IFC    '#1' EQ 'WORD'
JSR     RWORD           ; READ NEXT WORD FROM FILE
.ELSE
.ERROR  'READ'
.ENDIF
.ENDIF
JMP     DSKERR          ; DISK ERRORS
.IFC    '#1' EQ 'WORD'
.IFC    '#5' EQ '8'
AND     AC0,=OFF        ; LM WORDS ARE 8 BITS ONLY
.ENDIF
.ENDIF
.ENDM
;
.MACRO  WRITE
LD      AC2,=#4
.IFC    '#1' EQ 'SECTOR'
JSR     WCSECT          ; WRITE NEXT SECTOR FILE
.ELSE
.IFC    '#1' EQ 'WORD'
JSR     WWORD           ; WRITE NEXT WORD TO FILE
.ELSE
.ERROR  'WRITE'
.ENDIF
.ENDIF
JMP     DSKERR          ; DISK ERRORS
.ENDM
;
.MACRO  DELETE,AFST
LD      AC2,=#2
JSR     DELF            ; DELETE FILE
JMP     DSKERR          ; DISK ERRORS
.ENDM
;
.MACRO  DIRECTORY
.IFC    '#1' EQ 'READ'
.IFC    '#2' EQ 'FIRST'
JSR     DIRF            ; READ FIRST DIRECTORY SECTOR
.ELSE
.IFC    '#2' EQ 'NEXT'
JSR     DIRN            ; READ NEXT DIRECTORY SECTOR
.ELSE
.ERROR  'DIRECTORY'
.ENDIF
.ENDIF
.ELSE
.IFC    '#1' EQ 'WRITE'
JSR     DIRW            ; WRITE DIRECTORY SECTOR
```

```
        .ELSE
        .ERROR   'DIRECTORY'
        .ENDIF
        .ENDIF
        JSR      DSKERR              ; DISK ERRORS
        .ENDM
;
;   MACRO FOR DOUBLE LOAD AND DOUBLE STORE INSTRUCTIONS
;
        .MACRO   DLD,REGISTER,DISPLACEMENT
        LD       REGISTER,DISPLACEMENT
        LD       REGISTER+1,1+DISPLACEMENT
        .ENDM
;
        .MACRO   DST,REGISTER,DISPLACEMENT
        ST       REGISTER,DISPLACEMENT
        ST       REGISTER+1,1+DISPLACEMENT
        .ENDM
;
;   MACROS FOR LOAD AND STORE WITH AUTOINCREMENT ON INDEX
;   REGISTERS.
;
        .MACRO   LDAI,REGISTER,INDEX
        LD       REGISTER,INDEX
        INC      INDEX
        .ENDM
;
        .MACRO   STAI,REGISTER,INDEX
        ST       REGISTER,INDEX
        INC      INDEX
        .ENDM
;
        .MACRO   INC,INDEX
        .IFC     'INDEX' EQ '(AC3)'
        AISZ     AC3,1
        .ELSE
        .IFC     'INDEX' EQ '(AC2)'
        AISZ     AC2,1
        .ELSE
        .ERROR   'INDEXING'
        .ENDIF
        .ENDIF
        .ENDM
;
;   MACROS FOR BYTE HANDLING
;
        .MACRO   LLB,REGISTER,ADDRESS
        LD       REGISTER,ADDRESS; LOAD LEFT BYTE
        SHR      REGISTER,8
        .ENDM
;
        .MACRO   LRB,REGISTER,ADDRESS
```

```
        LD      REGISTER,ADDRESS; LOAD RIGHT BYTE
        AND     REGISTER,=X'00FF
        .ENDM
;
;   MACRO FOR SKIP IF EQUAL INSTRUCTION
;
        .MACRO  SKE,REGISTER,DISPLACEMENT
        SKNE    REGISTER,DISPLACEMENT
        JMP     .+2
        .ENDM
;
;   MACRO FOR SKIP IF GREATER THAN OR EQUAL TO
;
        .MACRO  SKGE,REGISTER,DISPLACEMENT
        SKG     REGISTER,DISPLACEMENT
        SKNE    REGISTER,DISPLACEMENT
        JMP     .+2
        .ENDM
;
;   MACRO FOR COPYING THE TOP OF THE STACK CONTENT INTO
;   REGISTER
;
        .MACRO  CSR,REGISTER
        PULL    REGISTER
        PUSH    REGISTER
        .ENDM
;
;   MACRO FOR DROPPING THRU TO NEXT ROUTINE
;
        .MACRO  DROP,ADDRESS
        .MLOC   ADDR
        .SET    ADDR,ADDRESS
        .IF     ADDR = .
        .ELSE
        .ERROR  'ADDRESSING'
        .ENDIF
        .ENDM
;
;  MACRO FOR PUTTING ASCII CHARACTERS ONE PER WORD
;
        .MACRO  .ASCI1,STRING
        .NSTAT  1
        .SET    POSITION,1
        .DO     100
        .MICRO  P1,POSITION,1,*STRING*
        .IFC    '"P1"' NE ''
        .WORD   '"P1"'/256
        .SET    POSITION,POSITION+1
        .ELSE
        .EXIT
        .ENDIF
        .ENDDO
```

```
        .NSTAT  0
        .ENDM
;
;  MACRO FOR PUTTING UNPACKED ASCII WITH DECIMAL POINT
;
        .MACRO  .PNLASC,STRING
        .NSTAT  1
        .SET    POSITION,1
        .DO     100
        .MICRO  C1,POSITION,1,*STRING*
        .SET    PSNP1,POSITION+1
        .MICRO  C2,PSNP1,1,*STRING*
        .IFC    '"C1"' EQ ''
        .EXIT
        .ENDIF
        .IFC    '"C2"' EQ '.'
        .WORD   '"C1"'/256+080
        .SET    POSITION,POSITION+1
        .ELSE
        .WORD   '"C1"'/256
        .ENDIF
        .SET    POSITION,POSITION+1
        .ENDDO
        .NSTAT  0
        .ENDM
;
;    MACRO FOR GENERATING COMMAND TABLE
;
        .MACRO  COMMAND,COMMAND,STRING
        .NSTAT  1
        .MICRO  CMND,1,2,*COMMAND*
        .MICRO  CMND1,1,1,*COMMAND*
        .MICRO  STRNG,1,9,*STRING*
;
        .WORD   '"CMND"'
        .WORD   COMMAND
;
        .IFC    '"STRNG"' EQ 'MOST USED'
        .WORD   '"CMND1"'-B
        .WORD   COMMAND
        .ENDIF
;
        .NSTAT  0
        .ENDM
        .LIST   GENERAL
;
        THE FOLLOWING MACRO DEFINES THE FIRMWARE PROGRAM
 STARTING ADDRESSES AND THE ENTRY POINTS FOR THE ROUTINES
 IN THOSE PROGRAMS.  (.INCLD ENTRY)


        .LIST   ENTRY!OD
```

```
        .PAGE   'FIRMWARE ENTRY POINTS'
;
; ROM = 0 -> FIRMWARE IN RAM
; ROM = 1 -> FIRMWARE IN ROM/PROM
;
ROM     =       1
;
;   THESE ARE THE MEMORY ADDRESSES OF FIRMWARE WITHIN THE
; COP DEVELOPMENT SYSTEM.
;
        .IF     ROM
        .ERROR  'FIRMWARE IN ROM'   ; WARNING
CONIO   =       0E800
FIOSYS  =       0EAD0
DISKIO  =       0F720
MEMDI   =       0F900
EXEC    =       0FA60
;
TOPUSE  =       03FFF               ; TOP OF USER RAM
        .ELSE
CONIO   =       02800
FIOSYS  =       02AD0
DISKIO  =       03730
MEMDI   =       03900
EXEC    =       03A60
;
TOPUSE  =       027FF               ; TOP OF USER RAM
        .ENDIF
        .LIST   ENTRY
;
; MACRO FOR ASSIGNING ENTRY POINTS FOR FIRMWARE PROGRAMS
;
        .MACRO  ENTRY,DUM1,DUM2,PGMNAME
        .MSTAT  1
        .IFC    'PGMNAME' NE 'UTILITY'
        .LIST   FRMEJ
        .PAGE   '"PGMNM" ENTRY JUMPS'
        .ASECT
        .NOBAS
        .="PGMNM"
        .LIST   ENTRY
        .ENDIF
        .MACRO  UTILITY,UTILNAME
        .IFC    'PGMNAME' EQ 'UTILITY'
        .LIST   FRMEP
        .PAGE   'UTILNAME ENTRY POINTS'
;
        .SET    ADDRESS,UTILNAME
;
        .ELSE
        .IFC    '"PGMNM"' EQ 'CONIO'
        JSR     SYSERR              ; SAFETY JUMPS IN CONIO
```

```
        JSR     SYSERR
        .PTR    SYSERR
        .ENDIF
        .ENDIF
;
        .LIST   ENTRY
        .MACRO  ENTPT,ENTNAME
        .IFC    'PGMNAME' EQ 'UTILITY'
        .LIST   FRMEP
ENTNAME =       ADDRESS
        .LIST   ENTRY
        .SET    ADDRESS,ADDRESS+2
        .ELSE
        .IFC    '"PGMNM"' EQ 'UTILNAME'
        .LIST   FRMEJ
        JMP     @.+1
        .WORD   E^ENTNAME
        .LIST   ENTRY
        .ENDIF
        .ENDIF
        .ENDM
        .ENDM
;
        UTILITY CONIO
        .SET    ADDRESS,ADDRESS+3 ; CONIO SAFETY JUMPS
;
        ENTPT   AUTOBD          ; AUTO BAUD DETECT
        ENTPT   PRSTUP          ; SET UP FOR PRINTER OUTPUT
        ENTPT   CNSTUP          ; SET UP FOR CONSOLE I/O
        ENTPT   GETC            ; GET CHAR FROM CONSOLE
        ENTPT   GECO            ; GET CHAR WITH ECHO
        ENTPT   PUTC            ; SEND CHAR TO CONSOLE
        ENTPT   PUT2C           ; SEND TWO CHARS TO CONSOLE
        ENTPT   OHEX4           ; PRINT AC1 AS 4 HEX CHARS
        ENTPT   OHEX3           ;              3
        ENTPT   OHEX2           ;              2
        ENTPT   OHEX1           ;              1
        ENTPT   CRLF            ; PRINT NEW LINE
        ENTPT   PUTB            ; PRINT BLANKS
        ENTPT   MESG            ; MESSAGE TO CONSOLE
        ENTPT   MESG2           ; MESSAGE, NO NEW LINE
        ENTPT   CRESET          ; CONSOLE RESET
        ENTPT   INTEST          ; CONSOLE INTERRUPT TEST
        ENTPT   ABSTTY          ; LOAD 16 BIT LM TAPE
        ENTPT   FPOUT           ; SEND MESSAGE TO DISPLAY
        ENTPT   FPO4            ; MESSAGE TO HALF OF DISPLAY
        ENTPT   READKB          ; READ FRONT PNL KYBD
        ENTPT   READSW          ; READ FRONT PNL SWITCHES
;
        UTILITY FIOSYS
;
        ENTPT   CLRHS           ; CLEAR HARDWARE STACK
```

```
        ENTPT   RDLIN           ; READ ONE INPUT LINE
        ENTPT   GNCHAR          ; GET NEXT CHAR
        ENTPT   GNSP            ; GET NON-SPACE CHAR
        ENTPT   GNVC            ; GET NEXT CHAR
        ENTPT   DECLPT          ; DECREMENT LINE PTR
        ENTPT   GCOMMA          ; TEST FOR COMMA
        ENTPT   GETHEX          ; GET HEX NUMBER
        ENTPT   GETDEC          ; GET DECIMAL NUMBER
        ENTPT   GETYN           ; GET YES OR NO RESPONSE
        ENTPT   GETCMD          ; GET COMMAND
        ENTPT   OUTNAM          ; PRINT FILE NAME
        ENTPT   RESETD          ; RESET DISK DRIVE PTRS
        ENTPT   DERROR          ; REPORT DISK ERRORS
        ENTPT   PGFR            ; PARSE FILE NAME
        ENTPT   DISER           ; DIRECTORY SEARCH
        ENTPT   FILCOM          ; COMPARE FILE NAMES
        ENTPT   EXEC1           ; GET AND EXECUTE DISK FILE
        ENTPT   EXEC2           ; EXECUTE DISK FILE,AC3=INPTR
        ENTPT   EXEC3           ; EXECUTE DISK FILE,AC2=FST
        ENTPT   OVERLAY         ; PROGRAM OVERLAY
        ENTPT   OPENR           ; OPEN FILE FOR READ
        ENTPT   OPENW           ; OPEN FILE FOR WRITE
        ENTPT   OPENM           ; OPEN FILE FOR MODIFY
        ENTPT   CLOSE           ; CLOSE FILE
        ENTPT   RCSECT          ; READ CONSECUTIVE SECTOR
        ENTPT   WCSECT          ; WRITE CONSECUTIVE SECTOR
        ENTPT   RRSECT          ; READ RELATIVE SECTOR
        ENTPT   WRSECT          ; WRITE RELATIVE SECTOR
        ENTPT   RWORD           ; READ WORD FROM FILE
        ENTPT   WWORD           ; WRITE WORD TO FILE
        ENTPT   REWIND          ; REWIND FILE
        ENTPT   DELF            ; DELETE FILE
        ENTPT   DIRF            ; READ FIRST DIRECTORY SECTOR
        ENTPT   DIRN            ; READ NEXT DIRECTORY SECTOR
        ENTPT   DIRW            ; WRITE CURRENT DIRECT SECTOR
;
        UTILITY DISKIO
;
        ENTPT   DISK1           ; DISK I/O, PLIST ADR AT CALL
        ENTPT   DISK2           ; DISK I/O, PLIST ADR IN AC2
;
        UTILITY EXEC
;
        ENTPT   PWRON           ; POWER ON INIT
        ENTPT   EXECA           ; NORMAL EXEC ENTRY
        ENTPT   SYSERR          ; ERROR ENTRY POINT
        ENTPT   STKOVR          ; STACK OVFL PROCESSING
        ENTPT   BRKPT           ; BRKPT ENTRY
        ENTPT   MPASS           ; CHECK FOR MEMORY PASSWORD
        ENTPT   DPASS           ; CHECK FOR DISK PASSWORD
;
        .MSTAT  0
```

```
        .ENDM
        .LIST   XLIST
        .PAGE
```

3. OPERATING ENVIRONMENT:

STANDARD COPS PDS SYSTEM WHICH CONSISTS OF:

A) 16K WORDS OF RAM
B) 6K WORDS OF ROM (440305549)
C) SINGLE DRIVE FLOPPY DISK (PACE COMPATIBLE)
D) STANDARD RS-232 OR CURRENT LOOP TERMINAL
E) IMP-16/200 (980304736-007)
F) RS-232 PRINTER (OPTIONAL)
G) 1K BUFFER FOR ALL SYSTEM STORAGE

THE FOLLOWING INCLUDE FILE DEFINES THE HARDWARE AND THE SHARED MEMORY ADDRESSES FOR THE PDS SYSTEM. (.INCLD HWADDR)

```
        .PAGE   'HARDWARE ADDRESSES'

; HARDWARE BASE ADDRESS FOR THE FRONT PANEL, SERIALIO AND
; PROM PROGRAMMER:
;
PNLIO   =       X'8380          ; BASE ADDRESS OF PNL HW
;
;
RDSW1   =       PNLIO   + 010   ; ADR OF FRONT PANEL SWITCHES
RDSTA1  =       PNLIO   + 020   ; ADR OF KYBD UART STATUS
;
; OFFSETS FROM THE BASE ADDRESS FOR ALL PNL RELATED HARDWARE:
;
PNLOFF  =       0               ; HARDWARE OFFSET
RDPR    =       PNLOFF + 000    ; ADR FOR READING PROM
LDD     =       PNLOFF + 000    ; LOAD DATA TO BE PROGRAMMED
LDA     =       PNLOFF + 008    ; LOAD ADR TO BE PROGRAMMED
KYBRD   =       PNLOFF + 008    ; READ KEYBOARD
RDSW    =       PNLOFF + 010    ; READ FRONT PANEL SWITCHES
DWRT0   =       PNLOFF + 010    ; WRT PULSE TO DIGIT 0
DWRT4   =       PNLOFF + 014    ; WRT PULSE TO DIGIT 4
DWRT7   =       PNLOFF + 017    ; WRT PULSE TO LEFT DGT (7)
RDC     =       PNLOFF + 018    ; READ MODE FOR PROMS
CSC     =       PNLOFF + 019    ; PROM CHIP SELECT CONTROL
VSS     =       PNLOFF + 01A    ; VSS,VDD,VBB CONTROL
VPC     =       PNLOFF + 01B    ; EPROM PGM PULSE CONTROL
GRDC    =       PNLOFF + 01C    ; GROUND CONTROL FOR PROMS
BPPC    =       PNLOFF + 01D    ; BP PROM PROG DATA ENABLE
BVC     =       PNLOFF + 01E    ; BP PROG VOLTAGE CONTROL
WRDM    =       PNLOFF + 01F    ; WRT CONTROL FOR DSPLY MEM
RDSTA   =       PNLOFF + 020    ; READ KYBD AND UART STATUS
```

```
TYBAUD   =        PNLOFF + 020     ; TTY BAUD RATE DETECT CNTRL
RDSTB    =        PNLOFF + 040     ; READ KYBD AND UART STATUS
RSBAUD   =        PNLOFF + 040     ; 232 BAUD RATE DETECT CNTRL
UARTWD   =        PNLOFF + 078     ; WRITE UART DATA
UARTRD   =        PNLOFF + 078     ; READ UART DATA
UARTRC   =        PNLOFF + 079     ; READ UART STATUS REGISTER
UARTWC   =        PNLOFF + 079     ; WRT UART MODE/CNTRL WORDS
BAUDET   =        PNLOFF - 080     ; BAUD RATE DETECTOR
URESET   =        PNLOFF - 080     ; UART/BAUD RATE GEN RESET
TTYEN    =        PNLOFF - 07F     ; TTY ENABLE
CRTEN    =        PNLOFF - 07E     ; CRT ENABLE
PRTTTY   =        PNLOFF - 07D     ; PRINTER ENABLE (TTY)
PRT232   =        PNLOFF - 07C     ; PRINTER ENABLE (232)
KBEN     =        PNLOFF - 07B     ; KEYBOARD ENABLE
WBDGEN   =        PNLOFF - 060     ; SET UP BAUD RATE GENERATOR
PNLRST   =        PNLOFF - 040     ; RESET ALL IFACE CONTROLS
RDRLY    =        PNLOFF - 020     ; READER RELAY CLOCK
;
; HARDWARE BASE ADDRESS FOR THE TARGET CARD
;
TGTIO    =        X'B000
;
; OFFSETS FROM THE BASE ADDRESS FOR ALL TARGET CARD RELATED
; HARDWARE.
;
TGTOFF   =        0                ; HARDWARE OFFSET
TCTAC    =        TGTOFF + 000     ; TRACE/TRACE AFTER COUNTER
OCNTR    =        TGTOFF + 001     ; OCCURENCE COUNTER
BPFLG    =        TGTOFF + 002     ; BRKPT FLAGS
FLAGS    =        TGTOFF + 003     ; CONTROL FLAGS
BPDEX    =        TGTOFF + 004     ; BRKPT DUMP EXIT
;
; SHARED MEMORY DEFINITIONS
;
SHRMEM   =        X'A000           ; SHARED MEMORY BASE ADDRESS
SMSIZ    =        X'1000           ; SHARED MEMORY SIZE
;

        THE FOLLOWING INCLUDE FILE DEFINES THE RESERVED
 MEMORY LOCATIONS USED BY THE PDS SOFTWARE AND FIRMWARE.
 (.INCLD SYSRAM)

        .PAGE   'SYSTEM BUFFER AND BASE PAGE DEFINITIONS'
;
;
;     THIS IS A COMPLETE DEFINITION OF RESERVED MEMORY
; LOCATIONS IN THE COP DEVELOPMENT SYSTEM.  'SYSBUF
; VARIABLES' ARE MEMORY LOCATIONS IN THE SYSTEM BUFFER REGION
; AT ADDRESSES X'DC00 TO X'DFFF, WHICH ARE RESERVED FOR USE
; BY FIRMWARE PROGRAMS.  'BASE PAGE VARIABLES' ARE MEMORY
; LOCATIONS IN THE IMP BASE PAGE AT ADDRESSES X'0 TO X'F,
; WHICH ARE PERMANENTLY RESERVED FOR USE BY ONE OR MORE
```

```
; PROGRAMS.
;
;
; BASE PAGE VARIABLES AT ADDRESSES 0 TO X'F:
;
ZERJMP   =        0                    ;SYSTEM ERROR IF PC HERE,
                                       ; INDIRECT JUMP THROUGH X'2
FULSTK   =        1                    ;STACK FULL INTERRUPT,
                                       ; INDIRECT JUMP THROUGH X'3
SYSER    =        2                    ;ADDRESS OF ROUTINE 'SYSERR'
STKOV    =        3                    ;ADDRESS OF ROUTINE 'STKOVR'
CHIPN    =        4                    ;CHIP NUMBER FOR 'COPMON'
TCTACB   =        5                    ;TRACE CONTROL FOR 'COPMON'
OCNTRB   =        6                    ;OCCURENCE COUNT FOR 'COPMON'
BPFLGB   =        7                    ;BRKPT CONTROL FOR 'COPMON'
FLAGSB   =        8                    ;FLAG BUFFER FOR 'COPMON'
BPDEXB   =        9                    ;DUMP CONTROL FOR 'COPMON'
SRCPTR   =        0A                   ;USED BY MEMORY DIAGNOSTICS
DSTPTR   =        0B                   ;USED BY MEMORY DIAGNOSTICS
ENDBLK   =        0C                   ;USED BY MEMORY DIAGNOSTICS
BASENU   =        0D                   ;NOT USED
BRKPTR   =        0E                   ;ADDRESS OF EXEC 'BRKPT'
MAXRAM   =        0F                   ;BASE PAGE PTR TO SYSBUF
;
;
         .SET     OFFSET,X'80          ;SYSBUF OFFSET
;
; MACRO FOR ASSIGNING SYSTEM BUFFER SPACE
;
         .MACRO   RESERV,NAME,NUMWDS
         .IFC     'NUMWDS' NE ''
         .SET     OFFSET,OFFSET-NUMWDS+1
         .ENDIF
         .SET     OFFSET,OFFSET-1
NAME     =        OFFSET
         .ENDM
;
; SYSBUF DEFINITIONS:
;
TOPSYS   =        0DFFF                ;TOP OF SYSBUF
SSIZ     =        0400                 ;SYSBUF SIZE
SYSBUF   =        TOPSYS-OFFSET+1      ;SYSBUF ADR LOCATED AT MAXRAM
;
; SYSBUF VARIABLES:
;
         RESERV   TAB3                 ;TAB COLUMN 3(FIOSYS,EDIT)
         RESERV   TAB2                 ;TAB COLUMN 2(FIOSYS,EDIT)
         RESERV   TAB1                 ;TAB COLUMN 1(FIOSYS,EDIT)
         RESERV   TRKDIR               ;TRACK DIRECTION (DISKIO)
         RESERV   ERRCNT               ;ERROR COUNT(DISKIO)
         RESERV   VECTOR               ;PARAM VECTOR ADR(DISKIO)
         RESERV   TRKCNT               ;TRK CTR(DISKIO)
```

```
RESERV  DSTAT       ;DISK STATUS ADR(DISKIO)
RESERV  STACK,16    ;STACK(EX,DISKIO)
RESERV  FLGS        ;STATUS FLAGS(EX,DISKIO)
RESERV  DR3         ;AC3(EX,DISKIO)
RESERV  DR2         ;AC2(EX,DISKIO)
RESERV  DR1         ;AC1(EX,DISKIO)
RESERV  DRO         ;ACO(EX,DISKIO)
RESERV  ET1         ;TEMPORARY(EX)
RESERV  ET2         ;TEMPORARY(EX)
RESERV  MEMRNG      ;NON-SYSBUF TOPRAM ADR(EX)
RESERV  BDATA2      ;DATA OF 2ND BREAK(EX)
RESERV  BLOC2       ;ADR OF 2ND BRK(EX)
RESERV  BDATA1      ;DATA OF 1ST BRK(EX)
RESERV  BLOC1       ;ADR OF 1ST BRK(EX)
RESERV  BRKNUM      ;CTR FOR # OF BRKS SET(EX)
RESERV  CWRD        ;LAST ALTERED MEMORY ADR(EX)
RESERV  DATA        ;DATA REG(EX)
RESERV  RNGE        ;END OF RANGE INDEX(EX)
RESERV  MODE        ;MODE(EX)
RESERV  ADDR        ;ADDR(EX)
RESERV  ACCESS      ;SECURITY ACCESS CODE(FIOSYS)
RESERV  HIGH        ;TEMP RANGE FOR MP LOADING
RESERV  LOW         ; (FIOSYS)
RESERV  ASHI        ;ABSOLUTE SECTOR RANGE OF
RESERV  ASLO        ; MP'S VECTOR(FIOSYS)
RESERV  TSHI        ;TOP SECT RNGE OF MP'S VECTOR
RESERV  TSLO        ;(FIOSYS)
RESERV  PTRHI       ;PTR RNG OF MP'S VECTOR
RESERV  PTRLO       ;(FIOSYS)
RESERV  BSHI        ;BASE SECT RNG OF MP'S VECTOR
RESERV  BSLO        ;(FIOSYS)
RESERV  ENTPT       ;ENTPT OF MP'S VECTOR(FIOSYS)
RESERV  CHCNT       ;ACTIVE CHANNEL COUNT(FIOSYS)
RESERV  DIR         ;DIRECTORY BUF PTR(FIOSYS)
RESERV  DIREND      ;END OF DIRECTORY BUF(FIOSYS)
RESERV  DIRP        ;TEMP PTR TO DIR(FIOSYS)
RESERV  DIR1        ;DIR SECTOR FOR NST1(FIOSYS)
RESERV  DIR2        ;DIR SECTOR FOR NST2(FIOSYS)
RESERV  DIRS        ;DIRECTORY SECTOR(FIOSYS)
RESERV  DIRSECT     ;CURRENT DIR SECTOR #(FIOSYS)
RESERV  FREE1       ;FREE SPACE PTR, NST1(FIOSYS)
RESERV  FREE2       ;FREE SPACE PTR, NST2(FIOSYS)
RESERV  NST1CT      ;CHNL TBL CNT, NST1(FIOSYS)
RESERV  NST2CT      ;CHNL TBL CNT, NST2(FIOSYS)
RESERV  CURVOL      ;CURRENT VOL NST ADR(FIOSYS)
RESERV  FSADR       ;FST ADDRESS(FIOSYS)
RESERV  CTADR       ;CHANNEL TBL ADR(FIOSYS)
RESERV  FR3         ;AC3(FIOSYS)
RESERV  FR2         ;AC2(FIOSYS)
RESERV  FR1         ;AC1(FIOSYS)
RESERV  FRO         ;ACO(FIOSYS)
RESERV  FTO         ;TEMPORARY(FIOSYS)
```

```
        RESERV  FT1                 ;TEMPORARY(FIOSYS)
        RESERV  FT2                 ;TEMPORARY(FIOSYS)
        RESERV  FT3                 ;TEMPORARY(FIOSYS)
        RESERV  FT4                 ;TEMPORARY(FIOSYS)
        RESERV  FT5                 ;TEMPORARY(FIOSYS)
        RESERV  FT6                 ;TEMPORARY(FIOSYS)
        RESERV  FT7                 ;TEMPORARY(FIOSYS)
        RESERV  FT8                 ;TEMPORARY(FIOSYS)
        RESERV  FT9                 ;TEMPORARY(FIOSYS)
        RESERV  CNBAUD              ;CON BAUD RATE(CONIO,FIOSYS)
        RESERV  PRBAUD              ;PRTR BAUD RATE(CONIO,FIOSYS)
        RESERV  CNCRDL              ;CON CR DELAY(CONIO,FIOSYS)
        RESERV  PRCRDL              ;PRTR CR DELAY(CONIO,FIOSYS)
        RESERV  CNLFDL              ;CON LF DELAY(CONIO,FIOSYS)
        RESERV  PRLFDL              ;PRTR LF DELAY(CONIO,FIOSYS)
        RESERV  PRFFDL              ;PRTR FF DELAY(CONIO,FIOSYS)
        RESERV  PRVTDL              ;PRTR VT DELAY(CONIO,FIOSYS)
        RESERV  CNMODE              ;CONSOLE MODE(CONIO)
        RESERV  PRMODE              ;PRINTER MODE(CONIO)
        RESERV  CNTYPE              ;CONSOLE TYPE(CONIO)
        RESERV  PRTYPE              ;PRINTER TYPE(CONIO)
        RESERV  CONPRT              ;SERIALI/O SET UP 0=CON/3=PRT
        RESERV  FST,FSSIZE          ;FILE STATUS TBL(FIOSYS)
        RESERV  ANY,16              ;TEMP FOR ANYBODY
        RESERV  ANYTMP              ;TEMP FOR ANYBODY @MAXRAM
        .IF     OFFSET = 0
;
; ABOVE VARIABLES USE UP ALL 128 TOPSYS +RELATIVE WORDS
;
        .ELSE
        .ERROR  'SYSBUF'
        .ENDIF
        RESERV  NU1,34              ;UNUSED TOPSYS -RELATIVE WRDS
        RESERV  LINPTR              ;INPUT LINE BUF PTR(FIOSYS)
        RESERV  CWIDTH              ;CARRIAGE WIDTH(FIOSYS)
        RESERV  PLIST,PLSIZE        ;DISKIO PARAM LIST(FIOSYS)
        RESERV  FIOSTK,16           ;STACK SAVE(FIOSYS)
        RESERV  TRPLST,51           ;PLIST FOR MEMDI
        RESERV  TRDLST,4            ;DLIST FOR FPOUT4
        RESERV  DIRBUF,SECTSZ       ;DIRECTORY BUF(FIOSYS)
MWP1    =       MXWDTH+1
        RESERV  LINBUF,MWP1         ;INPUT LINE BUF(FIOSYS)
CHNSIZ  =       CTSIZE*MAXCHN       ;CHANNEL TBL BUFFER SIZE
        RESERV  CHANNEL,CHNSIZ      ;CHANNEL TABLES(FIOSYS)
        RESERV  NSTBF1,NSTSIZ       ;NST1(FIOSYS)
        RESERV  NSTBF2,NSTSIZ       ;NST2(FIOSYS)
        .IF     OFFSET+SSIZ-128 = 0
;
; ABOVE VARIABLES USE UP ALL WORDS IN SYSBUF
;
        .ELSE
        .ERROR  'SYSBUF'
```

```
        .ENDIF
;
;
4. DETAILED COMPONENT DESIGN:

                NOT APPLICABLE


5. SYSTEM GENERATION ENVIRONMENT:

                IMP MACRO ASSEMBLER RUNNING
        ON 28K PACE. (MODIFIED PIMPAS)

          ***********************************

35  SIZE12  =       1-SIZE8
36  ;
37  ;       COPS CHANGES
38  ;
39  FIRE    =       1
40  ;
*************************************************************
63          .ENDIF
64          .ENDIF
65          .IF     IMP16
66          .TITLE  PIMPAS, ' P05493A   08/16/78 COPS'
67          .ENDIF
68          .ELSE
*************************************************************
1402            LD      RO,=$MBUF       ; TEMP BUFF POINTER
1403            ST      RO,$MPNTR
1404  ;
1405  ;         COPS CHANGE
1406  ;
1407  $1:
1408   ;
1409            JSR     NXCHAR          ; GET A CHARACTER
1410            SKNE    RO,MMARK        ; CHECK FOR MICRO-MARK
*************************************************************
1425            SKNE    RO,MMARK
1426            JMP     $5              ; PROCESS MICRO
1427  ;
1428  ;         COPS CHANGE
1429  ;
1430            DSZ     INPTR
1431  $3:       ISZ     ITEMLN
1432            LD      RO,ITEMLN
1433            BOC     ZRO,$1
1434  $4:       DSZ     INPTR
1435            AISZ    RO,-1
1436            JMP     $4
1437            JMP     $1
```

```
1438  ;
1439  ;
1440  ;
1441  $5:      LI      RO,0              ; SEARCH FOR DEFINED MICRO
1442           JSR     MCSRCH
1443           JMP     $8                ; NOT FOUND - ERROR
*************************************************************
1576  $3:      JSR     NXCHAR            ; PROCESS STRING
1577           SKNE    RO,$DELIM
1578  ;
1579  ;        COPS CHANGE
1580  ;
1581           JMP     $6                ; TERMINATING DELIMITER
1582  ;
1583            LD      RO,CHARTP         ; TEST FOR EOL
1584           SKNE    RO,=NCEOL
*************************************************************
1743           LI      RO,DFPRM          ; PARAMETER FLAG
1744           JMP     $4
1745  ;
1746  ;        COPS CHANGES
1747  ;
1748  $LOC:
1749           JSR     $CKLVL            ; IGNORE LOCAL IF MACRO NEST
1750           JMP     $STORE            ; LEVEL IS NOT 1
1751  ;
1752           SFLG    CRYFLG            ; ADJUST LOCAL NUMBER TO 3*NUM
1753           SUBB    RO,$PLIM
*************************************************************
1796  $LOCS:   .=.+1                     ; LIMIT OF LOCAL SYMBOL STACK
1797  $PLIM:   .=.+1                     ; LIMIT OF PARAMETER STACK
1798  ;
1799  ;        COPS CHANGES
1800  ;
1801  $CKLVL:
1802           PUSH    RO
1803           LD      RO,$MCNT
1804           SKNE    RO,=1
1805           JMP     $LEV1
1806           PULL    RO
1807           RTS
1808  $LEV1:
1809           PULL    RO
1810           RTS     1
1811  ;
1812           .POOL   16
1813  ;
1814  ; CHECK FOR MACRO DIRECTIVE
1815  ;
*************************************************************
1818           LD      R3,$DTAB          ; R3 POINTS AT MACRO DIRECTIVE TA
1819           JSR     TBSRCH            ; SEARCH TABLE
```

```
1820  ;
1821  ;          COPS CHANGE
1822  ;
1823  $32:       JMP      $STORE            ; NOT FOUND, SIMPLY PUT IN DEFINI
1824             SKNE     R3,=$MLOC         ; .MLOC DIRECTIVE
1825             JMP      $5
1826             SKNE     R3,=$ENDM         ; .ENDM DIRECTIVE
1827             JMP      $7
1828             SKNE     R3,=$MACRO        ; .MACRO DIRECITVE
1829             JMP      $9
1830             SKNE     R3,=$END          ; .END DIRECTIVE
1831             JMP      $32
1832  ;
1833  ;
1834             HALT                       ; SYSTEM ERROR
*************************************************************
1844  ; .MLOC PROCESSING
1845  ;
1846  ;
1847  ;          COPS CHANGES
1848  ;
1849  $5:
1850             JSR      $CKLVL            ; IGNORE MLOC IF MACRO NEST
1851             JMP      $STORE            ;  LEVEL IS NOT 1
1852  ;
1853             JSR      $CHKV             ; CHECK WHETHER TO RECOGNIZE DIRE
1854             LI       R0,DFSKIP         ; PUT SKIP FLAG IN DEF SO THIS LI
*************************************************************
1926             JMP      $STORE            ; NO SPECIAL SIGNIFICANCE IN ITEM
1927                                        ;    SIMPLY SAVE IN DEFINITION
1928  ;
1929  ;          COPS CHANGES
1930  ;
1931  $PREF:
1932             JSR      $CKLVL            ; IGNORE PARAM DEF IF MACRO
1933             JMP      $STORE            ;  NEST LEVEL IS NOT 1
1934  ;
1935             JSR      NXITEM            ; NEXT ITEM A CONSTANT?
1936             LI       R0,0              ; CLEAR BACKUP FLAGS AND BACK
*************************************************************
1956             JMP      $NXT
1957  $20:       LI       R0,DFARG          ; SET ARGUMENT REFERENCE FLAG
1958  ;
1959  ;          COPS CHANGES
1960  ;
1961             JMP      .+2
1962  $CONC:
1963             LI       R0,DFCON          ; SET CONCATENATION FLAG.
1964             JSR      $CKLVL            ; IGNORE CONCATENATION IF
1965             JMP      $STORE            ;  MACRO NEST LEVEL IS NOT 1
1966  ;
*************************************************************
```

```
$LOOP:   JSR     GETNM1          ; GET MACRO NAME
         JMP     XERROR          ; NO NAME, SYNTAX ERROR
;
;        COPS CHANGE
;
         LD      R0,XE000
;
;
         JSR     MCSRCH          ; SEARCH FOR DEFINED MACRO
         JMP     $UND            ; MACRO UNDEFINED
*************************************************************
                                 ;     BEEN DELETED SINCE LAST COMPR
         BOC     ZR0,$RTS        ; NO
;
;        COPS CHANGES
;
         LD      R0,MACPTR       ; SEE IF MACRO BEING EXPANDED
         BOC     NZR,$RTS        ; IF SO, DON'T COMPRESS NOW
         LI      R0,0            ; CLEAR COMPRESS FLAG
         ST      R0,CMPFLG
;
;
         LD      R3,BASE         ; R3 POINTS AT NEXT MACRO DEF
$2:      SKNE    R3,NXTMAC       ; AT END OF STACK?
*************************************************************
PRMSTK:  .=.+299                 ; PARAMETER STACK BUILDS UP
CALSTK:  .=.+1                   ; MACRO CALL STACK BUILDS DOWN
;
;        COPS CHANGE
;
PTRTAB:  .=.+2300
PTREND:
         .END    START
*************************************************************
```

6. MANUFACTURING REPRODUCTION

LOAD DSKIT INTO THE PDS SYSTEM.

INSERT DISK TO BE INITIALIZED.

DSKIT WILL PROMPT FOR A COMMAND
ENTER THE FOLLOWING:

D> IN "PDS","PDS MASTER DISK"

WHEN THE DISK IS INITIALIZED IT MAY
BE DUPLICATED ON A PACE MACHINE IN THE
NORMAL MANNER OR IT MAY BE DUPLICATED
ON PDS.

TO DUPLICATE ON PDS :

LOAD FM
FM WILL PROMPT FOR A COMMAND
ENTER THE FOLLOWING:

F> DU CDS TO PDS
FOLLOW THE INSTRUCTIONS ON THE
CONSOLE.

7. LIST OF RELATED DOCUMENTS:

430305767-000	COPMON
430305768-000	EDIT
430305769-000	DSKIT
430305770-000	FM
430305807-000	ASM
430305771-000	LIST
430305772-000	XREF
430305938-000	MASKTR
430305937-000	MDIAG
420305548-001	COP SYSTEM USERS MANUAL
091305548-000	CUSTOMER SUPPORT PACKAGE

8. CHANGES:

CHANGES FROM REV "A" TO "B"

MODIFIED COPMON, EDIT, FM, ASM AND DSKIT.
ADDED LIST, XREF ,MASKTR AND MDIAG.

CHANGES FROM REV "B" TO "C"

MODIFIED COPMON AND MASKTR

SEE APPROPRIATE 43030XXXX-000 PACKAGE FOR DETAILS.

9. CONTENTS OF MASTER DISKETTE

COPMON	430305767	REV. C
EDIT	430305768	REV. B
DSKIT	430305769	REV. B
FM	430305770	REV. B
ASM	430305807	REV. B
LIST	430305771	REV. A
XREF	430305772	REV. A
MASKTR	430305938	REV. B
MDIAG	430305937	REV. A

LABEL DISKETTE USING A BLANK INDEX LABEL MARKED WITH:

"PDS MASTER 440305834-200C"

NOTE: THE 440305834-200 IS CREATED FROM A BLANK, UNINITIALIZED, SINGLE DENSITY DISKETTE P/N 205004514-001