

```

LOC  OBJ          LINE      SOURCE STATEMENT
      1 $TITLE('MIP-ISIS MIPSEND')
      2 ;ISIS$mipsend:
      3 ;do;
      4 ;
      5 ; THIS ROUTINE IS CALLED BY USERS IN ORDER TO SENT A MESSAGE.
      6 ;
      7 ; INPUT:  TOP OF STACK = THE DESTINATION SOCKET
      8 ;         B/C = THE POINTER TO THE MESSAGE TO BE SENT
      9 ;         D/E = THE LENGTH OF THE MESSAGE TO BE SENT
     10 ;
     11 ; OUTPUT: THE RESULTS OF THE ATTEMPT TO SEND ARE IN THE A REG
     12 ;         UPON RETURN. THIS ROUTINE WILL ALWAYS RETURN SINCE
     13 ;         IT HAS A TIMEOUT BUILTIN TO IT.
     14 ;
     15 ;         A REG VALUES UPON RETURN ARE:
     16 ;
     17 ;         80H - DELIVERED TO DESTINATION
     18 ;         82H - DELIVERED TO DESTINATION
     19 ;         81H - UNKNOWN PORT ON DESTINATION
     20 ;         85H - THE DESTINATION HAD INSUFFICIENT RESOURCES
     21 ;         TO RECEIVE THE MESSAGE
     22 ;         89H - THE DESTINATION DEVICE DID NOT RESPOND
     23 ;
     24 ;
     25 ;
     26 ;declare RQEntry1 literally '
     27 ;
     28 ;         Request$id byte,
     29 ;         Src$request$id byte,
     30 ;         Dest$dev$id byte,
     31 ;         Dest$port$id byte,
     32 ;         Src$dev$id byte,
     33 ;         Buf$base$adr (2) word,
     34 ;         Length word,
     35 ;         IDS$id byte,
     36 ;         Owner$dev byte;
     37 ;
     38 ;declare Out$RQD word external;
     39 ;declare Wreply byte public;
     40 ;
     41 ;         NAME      MIPSEND
     42 ;         PUBLIC    MIPSEN,WREPLY
     43 ;         EXTRN     TRQGPT
     44 ;         EXTRN     INTASK,RQGPTR,RLGPTR,OUTRQD,INITT,BUMPT
     45 ;         CSEG
     46 ;
     47 $INCLUDE(:F1:MIP.EQU)
= 48 ;
= 49 ; DEFINE RQD RESULTS
= 50 ;
0001 = 51 GERROR EQU      1H
0004 = 52 GBUSY EQU      4H
0008 = 53 FIRSTG EQU      8H

```

LOC	OBJ	LINE	SOURCE STATEMENT
0020		= 55	GFULL EQU 20H
0040		= 56	DISABT EQU 40H
0080		= 57	FULLF EQU 80H
		= 58	
00C1		= 59	TERROR EQU 1H
00C4		= 60	TBUSY EQU 4H
00C8		= 61	FIRSTT EQU 8H
0010		= 62	TDISAB EQU 10H
0020		= 63	EMPTY EQU 20H
0040		= 64	DISABG EQU 40H
0080		= 65	EMPTYF EQU 80H
		= 66	;
		= 67	; DEFINE MIP CMDS AND RESPONSES
		= 68	;
0070		= 69	CSEND EQU 70H
0080		= 70	SENTOK EQU 80H
0081		= 71	UNKNP EQU 81H
0083		= 72	ACTIVP EQU 83H
0085		= 73	INSUFM EQU 85H
0087		= 74	INACTP EQU 87H
0089		= 75	DEADP EQU 89H
		= 76	;
		= 77	; DEFINE MIP-ISIS PARAMETERS
		= 78	;
0000		= 79	MYIDS EQU 0
00C3		= 80	THIDEV EQU 3
		81	
		82	;/*****/
		83	
		84	;MIP\$sen:
		85	; procedure(Dsocket,Msgptr,Length) byte public;
		86	
		87	; declare Dsocket word,
		88	; Msgptr word,
		89	; Length word;
		90	;/*
		91	; declare local variables
		92	;/*
		93	; declare Msg based Msgptr structure(Mip\$msg\$format),
		94	; RQEntry based Out\$rqd structure(Rqentry\$format),
		95	; Give\$state byte,
		96	; I byte;
		97	MIPSEN:
0000	EB	98	XCHG ; PUT LENGTH IN H/L
00C1	220000 D	99	SHLD LENGTH
0004	E1	100	POP H ; RET ADR
0005	D1	101	POP C ; DSOCKET IN D/E
0006	E5	102	PUSH H
		103	;
		104	; /* First get the RQD for the Request queue.
		105	; */
		106	
0007	C5	107	a5: PUSH B ; SAVE MSGPTR
0008	D5	108	PUSH D ; SAVE DSOCKET

LOC	OBJ	LINE	SOURCE STATEMENT
		110 ;	now loop until we can put the item into the RQ, find out the
		111 ;	device is dead, or timeout.
		112 ;	*/
		113 ;	do forever;
		114 a9:	
		115 ;	if TRQGPT and GERROR then return DEADP;
0009	CD0000	E 116	CALL TRQGPT
000C	C1	117	POP B ; get back dsocket
000D	D1	118	POP D ; get back msgptr
000E	1F	119	RAR
000F	3E89	120	MVI A,DEADP
0011	D8	121	RC
		122 ;	do;
		123 ;	/*
		124 ;	fill in the entry first
		125 ;	*/
		126 ;	RQEntry.Request\$Id = CEND;
0012	3670	127	MVI M,CSEND ; PTR TO RQE IS IN H/L UPON RETURN
		128 ;	RQEntry.Dest\$dev\$Id = Ddevice;
0014	23	129	INX H ; PAST SRC REQ ID SINCE NOT USED
0015	23	130	INX H
0016	70	131	MOV M,B ; PUT DEST DEVICE IN
0017	23	132	INX H
		133 ;	RQEntry.Dest\$port\$Id = Dport;
0018	71	134	MOV M,C
0019	23	135	INX H ; PUT DEST PORT IN
		136 ;	RQEntry.Src\$dev\$Id = Thi\$dev;
001A	3603	137	MVI M,THIDEV ; SRC IS THIS DEVICE
001C	23	138	INX H
		139 ;	RQEntry.Buf\$baseadr(0) = Msgptr;
		140 ;	/*
		141 ;	base is always 0
		142 ;	*/
001D	73	143	MOV M,E
001E	23	144	INX H
001F	72	145	MOV M,D
0020	23	146	INX H
		147 ;	RQEntry.Buf\$baseadr(1) = 0;
0021	AF	148	XRA A
0022	77	149	MOV M,A
0023	23	150	INX H
0024	77	151	MOV M,A
0025	23	152	INX H
		153 ;	RQEntry.Length = Length
0026	EB	154	XCHG
0027	2A000C	D 155	LHLD LENGTH
002A	EB	156	XCHG
002B	73	157	MOV M,E
002C	23	158	INX H
002D	72	159	MOV M,D
002E	23	160	INX H
		161 ;	RQEntry.ID\$Id = MY\$ids;
002F	3600	162	MVI M,MYIDS
0031	23	163	INX H

```

LOC  OBJ          LINE      SOURCE STATEMENT
0032 3603          165      MVI      M,THIDEV
          166 ;      /*
          167 ;      now release the entry and always signal
          168 ;      */
0034 CD0000  E      169 ;      Give$state = Rlgptr(.Cut$rqd);
          170      CALL      RLGPTR
          171 ;      /*
          172 ;      have placed the request into the queue, now wait for ack/nak
          173 ;      */
          174 ;      /*
          175 ;      SET PLACE TO WAIT
          176 ;      */
          177 ;      Wreply = OFFH
0037 21020C  D      178      LXI      H,WREPLY
003A 36FF          179      MVI      M,OFFH
          180 ;      call INITT;
003C CD0000  E      181      CALL      INITT
          182 ;      do while (W$reply = OFFH);
          183 ;      call Intask
          184 ;      if Bumppt then return DEADP
          185 ;      end;
003F CD0000  E      186 @11A:  CALL      INTASK          ; SEE IF ANYTHING THERE
0042 CD0000  E      187      CALL      BUMPT
0045 3E89          188      MVI      A,DEADP
0047 C8           189      RZ
0048 21020C  D      190      LXI      H,WREPLY
004B 7E           191      MOV      A,M
004C FEFF          192      CPI      OFFH
004E CA3F0C  C      193      JZ      @11A
          194 ;      /*
          195 ;      return to calling task
          196 ;      */
0051 C9          197 @6:    RET
          198 ;      return W$reply;
          199 ;      end;
          200 ;      end;
          201 ;end MIP$send;
          202
          203
          204 ;end ISIS$Mip$send;
          205
          206      DSEG
0000 0000          207 LENGTH: DW      0
0002 00          208 WREPLY: DB      0
          209
          210      END

```

PUBLIC SYMBOLS

MIPSEN C 000C WREPLY D 0002

EXTERNAL SYMBOLS

BUMPT E 0000 INITT E 0000 INTASK E 0000 OUTRQD E 0000 RLGPTR E 0000 RQGPT E 0000 TRQGPT E 0000

@11A	C	003F	@5	C	0007	@6	C	0051	@9	C	0009	ACTIVP	A	0083	BUMPT	E	0000	CSEND	A	0070
DEADP	A	0089	DISABG	A	0040	DISABT	A	0040	EMPTYF	A	0080	FIRSTG	A	0008	FIRSTT	A	0008	FULLF	A	0080
GBUSY	A	0004	GDISAB	A	0010	GERROR	A	0001	GFULL	A	0020	INACTP	A	0087	INITT	E	0000	INSUFM	A	0085
INTASK	E	0000	LENGTH	D	0000	MIPSEN	C	0000	MYIDS	A	0000	OUTRQD	E	0000	RLGPTR	E	0000	RQGPT	E	0000
SENTOK	A	008C	TBUSY	A	0004	TDISAB	A	0010	EMPTY	A	0020	TERROR	A	0001	THIDEV	A	0003	TRQGPT	E	0000
UNKNF	A	0081	WREPLY	D	0002															

ASSEMBLY COMPLETE, NO ERRORS