

UNIVERSAL ASSEMBLER VERSION 1.2 JANUARY 4, 1978 (IN-HOUSE)

CONFIDENTIAL PROPRIETARY INFORMATION

THIS ITEM IS THE PROPERTY OF DATAPoint CORPORATION, SAN ANTONIO, TEXAS, AND CONTAINS CONFIDENTIAL AND TRADE SECRET INFORMATION. THIS ITEM MAY NOT BE TRANSFERRED FROM THE CUSTODY OR CONTROL OF DATAPoint EXCEPT AS AUTHORIZED BY DATAPoint AND THEN ONLY BY WAY OF LOAN FOR LIMITED PURPOSES. IT MUST NOT BE REPRODUCED IN WHOLE OR IN PART AND MUST BE RETURNED TO DATAPoint UPON REQUEST AND IN ALL EVENTS UPON COMPLETION OF THE PURPOSE OF THE LOAN.

NEITHER THIS ITEM NOR THE INFORMATION IT CONTAINS MAY BE USED OR DISCLOSED TO PERSONS NOT HAVING A NEED FOR SUCH USE OR DISCLOSURE CONSISTENT WITH THE PURPOSE OF THE LOAN, WITHOUT THE PRIOR WRITTEN CONSENT OF DATAPoint.

COMMAND LINE WAS: SNAP3 COMM,,,PROC;GBQLX

INCLUSION A: PROCPARM/TXT
 INCLUSION B: PMACMIC/TXT
 INCLUSION C: GMACROZ/TXT
 INCLUSION D: PORTASGN/TXT
 INCLUSION E: PROCEQUS/TXT
 INCLUSION F: MDEF1800/TXT
 INCLUSION G: BDEF1800/TXT
 INCLUSION H: PORTEQUS/TXT
 INCLUSION I: DDEF1800/TXT
 INCLUSION J: HDEF1800/TXT

PROGRAM NAME: COMM

PROGRAM ADDRESS BLOCKS:	010000	/ABSOLUTE/	SIZE=000000	(ABS)
	167400	/SYSIVR/	SIZE=000400	(ABS)
	170000	/SYSROM/	SIZE=000047	(ABS)
	006000	/CDOXL/	SIZE=001000	(ABS)
	000000	/CDOXP/	SIZE=002000	(REL)
	007000	/CDORL/	SIZE=001000	(ABS)
	000000	/CDORP/	SIZE=002000	(REL)

EXTERNAL DEFINITIONS:

SIOUT	006555	SISTART	006660	SICIO	006707	SIIN	007674
-------	--------	---------	--------	-------	--------	------	--------

EXTERNAL REFERENCES (UNDEFINED SYMBOLS):

FETCHW	SRVRPT	FETCHI	MEMPF\$	SRVNXT	IVIOLS	FETCH
--------	--------	--------	---------	--------	--------	-------

UNUSED LABELS:

CVER	CREV	CPRE	INSZ	SDLCX	SIMODOT	SDLCR	SI11
------	------	------	------	-------	---------	-------	------

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.
11.
12.
13.
14.
15.
16.
17.
18.
19.
20.
21.
22.
23.
24.
25.
26.

000002
000011
000112

. BISYNC ,GENSYNC ,AND ASYNC XMTR AND RCVR
. 3-MAR-78 - 5:00 PM

*
CVER EQU 2
CREV EQU 9
CPRE EQU 'J'

COMMUNICATIONS VERSION NUMBER
COMPATIBLE RELEASE NUMBER
PRE-RELEASE CHANGE LETTER

- * . 2,9.K HJS 17 APR 78
- . 2,9.J MURF 3 MAR 78
- . 2,9.I HJS 23 FEB 78
- . 2,9.H M,HS 7/13 FEB 78
- . 2,9.G HJS 30 JAN 78
- . 2,9.C HJS 13 DEC 77
- . 2,9.B HJS 20 NOV 77
- * . 2,8.A HJS 19 SEPT 77
- . 2,7 MURF 7 SEPT 77
- . 2,6.A MURF 29 AUG 77
- * . 2,5.C MURF 15 JULY 77
- . 2,5.B HJS 14 JULY 77
- . 2,5.A HJS 13 JULY 77
- .
.*

- SET UP FOR RELOCATABILITY (NO CODE CHANGES)
- DELETE 'BREAK' MODE
- CLEAN MEMORY FAULT ACTIONS
- ADD 'BREAK' MODE TO ASYNC XMIT (MEMPF BACK)
- CHANGE INTERRUPT ROUTINE MEMPF TO NO-OPS
- NO CHANGE
- JUST TO KEEP IN LINE WITH OTHERS
- JUST TO STAY IN LINE WITH NEWER VERSION
- CLR USER FLAG AFTER MWAIT
- FIX LA;SDLC 8 BITS ONLY;SDLC GLOBAL ADDR
- FIX SIACUIN & SIMODIN FOR NEW BOARDS
- FIX COPYING MISTAKE
- GENERATE STANDARD VRP FORMAT UPDATING
- CORRECT REGISTER EQU'S TO WORK FROM PROC/EPT

INC PROCPARM

29.
30.
31.
32. 157000
33. 157400
34.

.
. BUFFER AREAS
. XBUF EQU SYSCOM
RBUF EQU SYSCOM+256

37,
38,
39,
40,
41,
42,
43,
44,
45,
46,
47,
48,
49,
50,
51,
52,
53,
54,
55,
56,
57,
58,
59,
60,
61,
62,
63,
64,
65,
66,
67,
68,
69,
70,
71,
72,
73,
74,
75,
76,
77,
78,
79,
80,
81,
82,
83,
84,
85,
86,
87,
88.

```

.COMMODE EQU    F5+F6+011          COM CONTROL
.COMMODE FORMAT
.BISYNC        00 00S 001
.GENSYNC       LL L0S 010
.ASYNC(1)      LL L00 100
.ASYNC(1.5)    LL L00 101
.ASYNC(2)      LL L00 110
.SDLC          00 00A 011

.S=SYNDEL, 1 TO STRIP OFF RECEIVED SYN CHARACTERS
.L=WLENGTH, 8=N WHERE N IS NUMBER OF BITS PER CHARACTER
.(1),(1.5),(2)=> NUMBER OF STOP ELEMENTS
.A=ADDET, 1 TO RECEIVE ONLY IF ADDRESS.FIELD=SNID

.BISYNC REGISTERS

.XSTAT EQU      F5+F6+013          XMIT STATUS
.XPNTR EQU      F5+F6+014          XMIT BUFFER POINTER
.XDATA EQU      F5+F6+015
.XCRCH EQU      F5+F6+016          (SYNCODE IN GENSYNC,AISTOR IN ASYNC)
.XCRCL EQU      F5+F6+017          (XICOUNT IN ASYNC)

.RSTAT EQU      F5+F6+003          RCV STATUS
.RPNTR EQU      F5+F6+004          RCV BUFFER POINTER
.RDATA EQU      F5+F6+005
.RCRCH EQU      F5+F6+006          (RICOUNT IN ASYNC)
.RCRCL EQU      F5+F6+007          (SPCOUNT IN ASYNC)

.LINK EQU       F5+F6+000          TEMP!!!
.TEMP1 EQU      F5+F6+001          TEMP!!!
.TEMP2 EQU      F5+F6+002          TEMP!!!
.G EQU         F5+002             DEFAULT (DON'T WRITE!!!)

.BSTAT EQU      TEMP1             RVC BUFFER STATUS
.RDATA2 EQU     RCRCL             FOR SYN DETECTION IN BISYNC AND GENSYNC

.GENSYNC REGISTERS
.SYNCODE EQU    XCRCH

.ASYNC REGISTERS
.AISTOR EQU     SYNCODE           NUMBER OF INTERUPTS PER BIT
.XICOUNT EQU    XCRCL            XMT INTERUPT COUNTER
.RICOUNT EQU    RCRCH            RCV INTERUPT COUNTER
.SPCOUNT EQU    RCRCL            RCV SPACE COUNTER

.USER BUFFER POINTERS (FOR EXXMIT AND EXRCV)
    
```

030001
030007

030016

030016
030017
030006
030007

89.
90.
91.
92.
93.

.UXPNTR EQU F5+F6+010
.URPNTR EQU F5+F6+012
.

96,		.			
97,		.	COMMODE		
98,	000010	.	SYNDEL EQU	010	BISYNC AND GENSYNC ONLY
99,	000010	.	ADDEET EQU	SYNDEL	SDLC ONLY
100,	000340	.	WLNTH EQU	0340	WORD LENGTH; GENSYNC AND ASYNC ONLY
101,		.			
102,		.	XSTAT		
103,		.			
104,	000001	.	EOB EQU	B0	END OF BLOCK (BISYNC)
105,	000001	.	CIP EQU	EOB	CHARACTER IN PROGRESS (ASYNC)
106,	000002	.	LDLE EQU	B1	LEADING DLE (BISYNC)
107,	000002	.	SIP EQU	LDLE	START ELEMENT IN PROGRESS (ASYNC)
108,	000004	.	FILIP EQU	B2	FILL IN PROGRESS IF NOT EOB, ELSE CRC2 SENT (BISYNC)
109,	000010	.	XPAR EQU	B3	TRANSPARENT (BISYNC)
110,	000010	.	CRC EQU	XPAR	CRC BEING SENT (SDLC)
111,	000020	.	BIP EQU	B4	BLOCK IN PROGRESS (BISYNC)
112,	000020	.	FIP EQU	BIP	FRAME IN PROGRESS (SDLC)
113,	000040	.	BCOUNT EQU	B5	BIT COUNT (3 BITS)
114,		.			
115,		.	RSTAT		
116,		.			
117,		.	EOB EQU	B0	END OF BLOCK (BISYNC)
118,	000001	.	LA EQU	B0	LINE ACTIVE (SDLC)
119,		.	CIP EQU	EOB	CHARACTER IN PROGRESS (ASYNC)
120,		.	LDLE EQU	B1	LEADING DLE (BISYNC)
121,		.	SIP EQU	LDLE	START ELEMENT IN PROGRESS (ASYNC)
122,	000004	.	SYNWAIT EQU	B2	WAITING FOR SYN SYN (BISYNC & GENSYNC)
123,	000004	.	IDLE EQU	SYNWAIT	WAITING FOR FLAG (SDLC)
124,		.	CRC EQU	XPAR	CRC IF EOB, XPAR IF NOT (BISYNC)
125,	000010	.	CHRCOM EQU	XPAR	CHARACTER COMPLETE (ASYNC)
126,		.	BIP EQU	B4	BLOCK IN PROGRESS
127,		.	FIP EQU	BIP	FRAME IN PROGRESS (SDLC)
128,		.	BCOUNT EQU	B5	BIT COUNT (3 BITS)
129,		.			
130,		.			

133.
134.
135.
136. 000062
137. 000001
138. 000002
139. 000003
140. 000046
141. 000037
142. 000055
143. 000020
144.
145.
146.
147.
148. 000176
149. 000177
150. 000377
151. 000174
152. 000020
153. 000010
154. 000205
155. 000117
156.

. EBCDIC CONTROL CHARACTERS

.
SYN EQU 062
SOH EQU 001
STX EQU 002
ETX EQU 003
ETB EQU 046
ITB EQU 037
ENQ EQU 055
DLE EQU 020

. INTERESTING SDLC CHARACTERS AND STUFF

.
FLAG EQU 0176
ABORT EQU 0177
GLOBAL EQU 0377
INSZ EQU 0174
POLYHI EQU 020
POLYLO EQU 010
GDCRCH EQU 0205
GDCRCL EQU 0117

29AUG77
29AUG77
INSERTED ZERO (LSB=X)
HI BYTE OF CRC POLY
LO " " "
HI BYTE OF GOOD CRC (INCLUDING SIX BITS OF FLAG)
LO " " " " " " "


```

179,
180,
181,
182,
183,
184,
185,
186,
187,
188,
189,
190,
191, 006007L 01110001 10111011
      006010L 01000101 00011000
192, 006011L 11000011 11101011
193, 006012L 01110001 11111101
      006013L 01010101 00000001
194, 006014L 01110010 11111011
195, 006015L 01110000 11001011
196, 006016L 11000010 11101111
197, 006017L 01010101 11111000
198, 006020L 01101111 11111011
199, 006021L 01010101 00000111
200, 006022L 01000000 00000101
201, 006023L 11000011 11000000
202, 006024L 01110001 11111101
203, 006025L 00010111 10110010
204, 006026L 00010111 10010010
205, 006027L 01101111 11111101
206, 006030L 01110001 10111011
      006031L 01010010 00100000
      006032L 01101111 11111011
207, 006033L 11000000 11011000
208, 006034L 01110001 11111011
      006035L 01000101 00001000
209, 006036L 11000010 10101010
210, 006037L 01010001 11011110
211, 006040L 11001111 01001010
212,
213, 006041L 11000011 10111001
214, 006042L 01000101 00000010
215, 006043L 11000010 10110110
216, 006044L 01110001 11111011
      006045L 01010011 00010000
      006046L 01101111 11111011
217,
218, 006047L 01110001 11111011
      006050L 01000101 00010000
219, 006051L 11000011 11000011
220, 006052L 01110001 11111101
      006053L 01010101 00000001
221, 006054L 01110000 11111110
    
```

```

*
. XSTAT USAGE
.   0,1,2 --- ONE'S COUNT (FOR ZERO INSERTION)
.   3 --- CRC (BEING SENT)
.   4 --- FIP (FRAME IN PROGRESS)
.   5,6,7 --- BCOUNT (BITS PER CHARACTER)
.
. XBUF STAT
.   0 --- RDY (PREVIOUS BUFFER BYTE IS DATA TO XMIT)
.   1 --- SPX (SPECIAL: END FRAME WITH CRC/FLAG OR ABORT)
.   2,3,4,5,6,7 --- 0
.
SDLCX  TSTIR  ,FIP+CRC,XSTAT,CC  SEE IF WE ARE DOING ZERO INSERTION
      BRA    INCBIT,TZ          NOT IF NO FRAME OR CRC IN PROGRESS
      DORIR  ,ND,1,XDATA       GET LAST BIT SENT
      DORIR  ,AC,XSTAT         ADD LAST BIT TO ONES COUNT; LSB => LINK
      TSTRT  XR,XSTAT          IF IT WAS A ZERO WE DIDNT CHANGE ANYTHING
      BRA    $+2,FZ
      DORIR  ,ND,0370         AND WE THEREFORE CLEAR THE COUNTER
      LDRT   XSTAT
      DORIR  ,ND,7           GET ONES COUNT
      TSTIT  XR,5            FIVE ONES?
      BRA    STUFF0,TZ        THEN STUFF IT
      INCBIT LDTR   XDATA      DO A CIRCULAR SHIFT
      CCLR   SR
      SHIFT  SR
      LDRT   XDATA
      DORIR  XSTAT,AC,BCOUNT,XSTAT,CC  BUMP THE BIT COUNTER
      BRA    DOSXCRC,FC       JUST DO THE CRC IF NOT END OF BYTE
      TSTIR  ,CRC,XSTAT      SENDING CRC?
      BRA    XSDLCRC,FZ
      LDRT   SDLCX1          SET RETURN ADDRESS
      BRA    CHKBUF         CHECK FOR MORE DATA
      SDLCX1 BRA    MTBUF,TZ
      TSTIT  ,2             CHECK BUFF STAT FOR CONTROL
      BRA    SPX,FZ         AND GO HANDLE SPECAIL XMIT IF SO
      DORIR  XSTAT,OR,FIP,XSTAT SET FRAME IN PROGRESS - 29AUG77
      DOSXCRC TSTIR  ,FIP,XSTAT  FRAME IN PROGRESS?
      SXCRC1 BRA    SDXBIT,TZ    JUST SEND A BIT IF NOT
      DORIR  ,ND,1,XDATA       GET LSB
      DORIR  ,XR,XCRCH
    
```

222.	006055L	01101111	11111110	LDRT	XCRCH	
223.	006056L	00010111	10110010	CCLR		PUT THE FEEDBACK IN THE LINK
224.	006057L	01000101	00000001	TSTIT	,1	FEEDBACK ONE OR ZERO?
225.	006060L	11000011	11001001	BRA	XFB0,TZ	
226.	006061L	01010000	00010000	DORI	XCRCH,XR,POLYHI	
	006062L	01101111	11111110			
227.	006063L	01110001	11111111	DORIR	XCRCL,XR,POLYLO,XCRCL	
	006064L	01010000	00001000			
	006065L	01101111	11111111			
228.	006066L	01110001	11111111	XFB0 LDTR	XCRCL	
229.	006067L	00010111	10010010	SHIFT	SR	FEEDBACK IS STILL IN LINK
230.	006070L	01101111	11111111	LDRT	XCRCL	
231.	006071L	01110001	11111110	LDTR	XCRCH	
232.	006072L	00010111	10010010	SHIFT	SR	
233.	006073L	01101111	11111110	LDRT	XCRCH	
234.						
235.	006074L	01110001	11111101	SDXBIT LDTR	XDATA	GET THE DATA
236.	006075L	00110111	00100011	SDXBIT1 LDPT	SDLCOT	SEND THE LSB
237.	006076L	11011111	00000011	BRA	CRETURN	
238.						
239.	006077L	01110001	11111101	STUFF0 DORIR	XDATA,ND,0376,XDATA	MAKE ME THINK I SENT A ZERO
	006100L	01010101	11111110			
	006101L	01101111	11111101			
240.	006102L	01110001	11111011	DORIR	XSTAT,ND,0370,XSTAT	CLEAR THE ONES COUNTER
	006103L	01010101	11111000			
	006104L	01101111	11111011			
241.	006105L	11001111	11000010	BRA	SDXBIT1	AND SEND A ZERO
242.						
243.	006106L	01110001	11111011	MTBUF TSTIR	,FIP,XSTAT	FRAME IN PROGRESS?
	006107L	01000101	00010000			
244.	006110L	11000010	10011011	BRA	XABORT,FZ	SDLC UNDERRUNS ARE A NO=NO!
245.	006111L	01110001	11111101	SPX TSTIR	XR,FLAG,XDATA	WAS LAST BYTE A FLAG?
	006112L	01000000	01111110			
246.	006113L	11000010	10011011	BRA	XABORT,FZ	SEND MARKS IF NOT
247.	006114L	01110001	11111011	TSTIR	,FIP,XSTAT	FRAME IN PROGRESS?
	006115L	01000101	00010000			
248.	006116L	11000011	10100000	BRA	INTXCRC,TZ	
249.	006117L	01010011	00001000	DORI	XSTAT,OR,CRC	YES, END IT WITH CRC
	006120L	01101111	11111011			
250.	006121L	01110001	11111110	DORIR	XDATA,XR,0377,XCRCH	INVERT THE CRC BEFORE SENDING IT
	006122L	01010000	11111111			
	006123L	01101111	11111101			
251.	006124L	11001111	11010101	BRA	SXCRC1	
252.						
253.	006125L	01000101	00010000	XSDLCRC TSTIT	,FIP	SEND SECOND BYTE OF CRC?
254.	006126L	11000011	10100010	BRA	XEOF,TZ	YES
255.	006127L	01010101	11101111	DORI	XSTAT,ND,-1-FIP	NO, CLEAR FRAME IN PROGRESS
	006130L	01101111	11111011			
256.	006131L	01110001	11111110	DOTIR	,XR,0207,XCRCH	THIS GETS INVERTED CRCL
	006132L	01010000	10000111			
257.	006133L	01101111	11111101	SDXDATA LDRT	XDATA	
258.	006134L	11001111	11000010	BRA	SDXBIT1	

259,						
260,	006135L	01010001	01111110	XEOF	LDRI	XDATA,FLAG
	006136L	01101111	11111101			FOLLOW CRC WITH FLAG
261,						
262,	006137L	01010001	11111111	INTXCRC	LDRI	XCRCH,0377
	006140L	01101111	11111110			
263,	006141L	01101111	11111111		LDRT	XCRCL
264,	006142L	01101110	01111011		INCR	XSTAT
265,	006143L	11001111	11000011		BRA	SDXBIT
266,						
267,	006144L	01010001	00000000	XABORT	LDRI	XSTAT,0
	006145L	01101111	11111011			
268,	006146L	01010001	11111111		LDTI	0377
269,	006147L	11001111	10100100		BRA	SDXDATA
270,						SEND MARKS

```

273,
274,
275,
276,
277,
278,
279,
280,
281,
282,
283,
284,
285,
286,
287,
006150L 01110001 10111011
006151L 01010010 00100000
006152L 01101111 11111011
288,
006153L 11000000 00100101
289,
006154L 01000101 00000001
290,
006155L 11000010 00011111
291,
006156L 01000101 00000100
292,
006157L 11000010 00010010
293,
006160L 01110001 11111101
006161L 01000000 00010000
294,
006162L 11010011 11110010
295,
006163L 01110001 11111011
296,
006164L 01010101 00001010
297,
006165L 01000000 00001000
298,
006166L 11000011 01110101
299,
006167L 01110001 11111101
006170L 01000000 00000011
300,
006171L 11000011 00001111
301,
006172L 01000000 00100110
302,
006173L 11000011 00001111
303,
006174L 01000000 00011111
304,
006175L 11000011 00001100
305,
006176L 01000000 00101101
306,
006177L 11000011 00000101
307,
006200L 01000000 00000010
308,
006201L 11000011 00000001
309,
006202L 01000000 00000001
310,
006203L 11010011 11111100
311,
006204L 01110001 11111011
006205L 01010101 11111101
006206L 01101111 11111011
312,
006207L 01110001 11111101
006210L 01000000 00110010
313,
006211L 11000011 01001011
314,
006212L 01110001 11111101
006213L 01110000 11111111
006214L 01101111 11110001
315,
006215L 01110010 00110001
    
```

```

*
. XSTAT USAGE
.      0 --- EOB (END OF BLOCK)
.      1 --- LDLE (LEADING DLE)
.      2 --- FILIP (FILL IN PROGRESS IF NOT EOB, ELSE CRC2 SENT)
.      3 --- XPAR (TRANSPARENT MODE)
.      4 --- BIP (BLOCK IN PROGRESS)
.      5,6,7 --- BCOUNT (BITS PER CHARACTER)
.
. XBUF STATUS
.      0 --- RDY (PRECEDING BYTE IS DATA TO XMIT)
.      1 --- COM2 (LIKE COM2 IN MFCA)
.      2,3,4,5,6,7 --- 0
.
BSCX  DORIR  XSTAT,AC,BCOUNT,XSTAT,CC  INC THE BIT COUNT

      BRA    XBIT,FC                JUST SEND A BIT IF NOT 8 BITS
      TSTIT  ,EOB                    TIME TO SEND SECOND CRC?
      BRA    XCRC2,FZ                DO IT IF SO
      TSTIT  ,FILIP                  XPAR FILL IN PROGRESS?
      BRA    XFILLX,FZ                SEND SYN IF SO
      TSTIR  XR,DLE,XDATA            SENT DLE?

      BRA    XDLE,TZ
      LDTR   XSTAT                    NO, GET XSTAT AGAIN
      DOTI   ,ND,XPAR+LDLE,TW        MASK OFF ALL BUT XPAR,LDLE
      TSTIT  XR,XPAR                  XPAR,LDLE' ?
      BRA    DOXCRC,TZ                JUST SEND THE BYTE IF SO
      TSTIR  XR,ETX,XDATA            SENT ETX?

      BRA    XEOB,TZ
      TSTIT  XR,ETB
      BRA    XEOB,TZ
      TSTIT  XR,ITB
      BRA    XEOB1,TZ
      TSTIT  XR,ENQ
      BRA    XENQ,TZ
      TSTIT  XR,STX
      BRA    XSTX,TZ
      TSTIT  XR,SOH
      BRA    XSOH,TZ
XNCNTL DORIR  XSTAT,ND,-1=LDLE,XSTAT  RESET LDLE

      TSTIR  XR,SYN,XDATA

DOXCRC DORRR  TEMP1,XR,XCRCCL,XDATA  DATA,XR.CRCL=>TEMP1

      DOTR   ,AC,TEMP1,,C0          SL (LSB=0)
    
```

316.	006216L	01110000	10110001	DORR	TEMP1,XR,TEMP1,,CC	TEMP1,XR,T->TEMP1 (LINK=0)
	006217L	01101111	11110001			
317.	006220L	00010111	11010010	DORR	TEMP2,FT,Q,,CF+SR	SR->TEMP2
	006221L	01101111	11110010			
318.	006222L	00010111	11010010	SHIFT	SR,CF	SR
319.	006223L	00010111	11010010	SHIFT	SR,CF	SR
320.	006224L	01110000	11110010	DORR	TEMP2,XR,TEMP2	TEMP2,XR,T->TEMP2
	006225L	01101111	11110010			
321.				RPT	4	
322.	006226L	00010111	11010010	SHIFT	SR,CF	SR BY 4
322.	006227L	00010111	11010010	SHIFT	SR,CF	SR BY 4
322.	006230L	00010111	11010010	SHIFT	SR,CF	SR BY 4
322.	006231L	00010111	11010010	SHIFT	SR,CF	SR BY 4
323.	006232L	01110000	11110010	DORR	TEMP2,XR,TEMP2	TEMP2,XR,T->TEMP2
	006233L	01101111	11110010			
324.	006234L	01010101	00000001	DOTI	,ND,1	
325.	006235L	01110000	11111110	DORR	XCRCL,XR,XCRCH	CRCH,XR,T->CRCL
	006236L	01101111	11111111			
326.	006237L	01110001	11110001	LDTR	TEMP1	
327.	006240L	00010111	10110010	CCLR		LSB->LINK
328.	006241L	00010111	10010010	SHIFT	SR	SRC
329.	006242L	00010111	10110010	CCLR		LSB->LINK
330.	006243L	00010111	10010010	SHIFT	SR	SRC
331.	006244L	01101111	11110001	LDRT	TEMP1	T->TEMP1
332.	006245L	01010101	11000000	DOTI	,ND,0300	
333.	006246L	01110000	11111111	DORR	XCRCL,XR,XCRCL	CRCL,XR,T->CRCL
	006247L	01101111	11111111			
334.	006250L	01110001	11110010	LDTR	TEMP2	TEMP2->T
335.				RPT	5	
336.	006251L	00010111	10100010	SHIFT	SL	SL BY 5
336.	006252L	00010111	10100010	SHIFT	SL	SL BY 5
336.	006253L	00010111	10100010	SHIFT	SL	SL BY 5
336.	006254L	00010111	10100010	SHIFT	SL	SL BY 5
336.	006255L	00010111	10100010	SHIFT	SL	SL BY 5
337.	006256L	01010101	11000000	DOTI	,ND,0300	
338.	006257L	01110000	11110001	DORR	XCRCH,XR,TEMP1	TEMP1,XR,T->CRCH
	006260L	01101111	11111110			
339.						
340.	006261L	01110001	11111011	TSTIR	,EOB,XSTAT	TIME TO SEND FIRST CRC?
	006262L	01000101	00000001			
341.	006263L	11000010	00001000	BRA	XCRC,FZ	CHECK BUFFER IF NOT
342.						
343.	006264L	01010001	00101010	XBUMP	LDTA XBUMP1	
344.						
345.						
346.						
347.						
348.						
349.	006265L	01101111	11110000	CHKBUF	LDRT LINK	PUT RETURN ADDRESS IN LINK
350.	006266L	11000100	01001001	MWAIT	,CMEMPK	
	006267L	11010111	00000111			
351.	006270L	00110001	11011100	DOPIP	MODW,ND,-1-SWUSER,PSWI	MAKE SURE WE CAN USE THE MEMORY!

	006271L	01010101	11111011			
	006272L	00110111	00000100			
352.	006273L	01010001	11011110	LDPI	MAR0H,XBUF>8	POINT TO CURRENT STATUS
	006274L	00110111	11100000			
353.	006275L	01110001	11111100	DOPIR	MAR0L,OR,1,XPNTR	
	006276L	01010011	00000001			
	006277L	00110111	11000000			
354.	006300L	00110111	01000111	STB	SMR	AND READ IT
355.	006301L	01101111	11111100	LDRT	XPNTR	MAKE SURE XPNTR POINTS TO STATUS
356.	006302L	11000100	00111101	MWAIT	,CMEMPK	
	006303L	11010111	00000111			
357.	006304L	00110001	00110110	LDRP	TEMP1,MDR	SAVE CURRENT STATUS IN TEMP1
	006305L	01101111	11110001			
358.	006306L	11100011	00000000	BRR	LINK,TZ	RETURN IF BUFFER MT (STATUS=0)
359.	006307L	01010001	00000000	LDPI	MDW,0	MAKE OLD BUF LOC RDY
	006310L	00110111	00100001			
360.	006311L	01110001	11111100	LDTR	XPNTR	GET THE BUFFER POINTER
361.	006312L	11000100	00110101	MWAIT	,CMEMPK	
	006313L	11010111	00000111			
362.	006314L	00110111	00001101	STB	DMAR	POINT TO DATA
363.	006315L	00110111	01000111	STB	SMR	AND READ IT
364.	006316L	01101110	01111100	INCR	XPNTR	BUMP THE POINTER TO NEXT DATA
365.	006317L	11000100	00110000	MWAIT	,CMEMPK	
	006320L	11010111	00000101			
366.	006321L	00110001	00110110	LDRP	XDATA,MDR	PUT NEW DATA IN XDATA
	006322L	01101111	11111101			
367.	006323L	01110001	11110001	LDTR	TEMP1	GET THE STATUS
368.	006324L	11101111	00000000	BRR	LINK	AND RETURN
369.						
370.	006325L	11010011	11100100	XBUMP1	BRA	XFILL,TZ
371.	006326L	01000101	00000010	TSTIT	,2	FILL IF BUFFER MT (STATUS=0)
372.	006327L	11000011	00100101	BRA	XBIT,TZ	CHECK FOR "COM2" BIT
373.	006330L	01110001	11111100	INCR	XPNTR,XPNTR	LEAVE XPNTR AT NEXT DATA IF NORMAL
	006331L	01101110	01111100			BUMP IT TO NEXT STATUS IF "COM2"
374.	006332L	01110001	11111101	XBIT	LDRP	SDLCOT,XDATA
	006333L	00110111	00100011			
375.	006334L	00010111	10110010	CCLR		MAKE THE SHIFT CIRCULAR
376.	006335L	00010111	10010010	SHIFT	SR	SRC
377.	006336L	01101111	11111101	LDRT	XDATA	AND STORE IT AWAY AGAIN
378.	006337L	11011111	00000011	BRA	CRETURN	
379.						
380.	006340L	01000101	00000100	XCRC2	TSTIT	,FILIP
381.	006341L	11000010	00011000	BRA	XCRCXIT,FZ	HAVE WE SENT THE SECOND CRC?
382.	006342L	01010011	00000100	DORI	XSTAT,OR,FILIP	CLEAN UP AND GO ON IF SO
	006343L	01101111	11111011			ELSE SEND IT
383.	006344L	01110001	11111110	LDRR	XDATA,XCRCH	CRCH->DATA
	006345L	01101111	11111101			
384.	006346L	11001111	00100101	BRA	XBIT	
385.						
386.	006347L	01010101	00010000	XCRCXIT	DORI	XSTAT,ND,BIP
	006350L	01101111	11111011			
387.	006351L	01010001	00000000	LDRI	XCRCL,0	CLR CRC

388.	006352L	01101111	11111111			
389.	006353L	01101111	11111110	LDRT	XCRCH	
390.	006354L	11001111	01001011	BRA	XBUMP	
391.	006355L	01010101	11111001	XFILLX	DORI	XSTAT,ND,-1-FILIP-LDLE RESET FILIP,LDLE
392.	006356L	01101111	11111011			
393.	006357L	11011111	11011100	BRA	XSYN	
394.	006360L	01110001	11111011	XEOB	DORIR	XSTAT,ND,-1-BIP,XSTAT RESET BIP
	006361L	01010101	11101111			
	006362L	01101111	11111011			
395.	006363L	01110001	11111011	XEOB1	DORIR	XSTAT,OR,EOB,XSTAT SET EOB
	006364L	01010011	00000001			
	006365L	01101111	11111011			
396.	006366L	11001111	01110101	BRA	DOXCRC	INCLUDE THE END CHAR IN THE CRC
397.						
398.	006367L	01110001	11111111	XCRC	LDRR	XDATA,XCRCL CRCL->DATA
	006370L	01101111	11111101			
399.	006371L	11001111	00100101	BRA	XBIT	
400.						
401.	006372L	01110001	11111011	XENQ	DORIR	XSTAT,ND,-1-BIP-XPARG-LDLE,XSTAT RESET BIP,XPARG,LDLE
	006373L	01010101	11100101			
	006374L	01101111	11111011			
402.	006375L	11001111	01001011	BRA	XBUMP	
403.						
404.	006376L	01110001	11111011	XSTX	TSTIR	,LDLE,XSTAT DLE-STX ?
	006377L	01000101	00000010			
405.	006400L	11010011	11111100	BRA	XSOH,TZ	
406.	006401L	01010011	00001000	DORI	XSTAT,OR,XPARG	SET XPARG IF SO
	006402L	01101111	11111011			
407.	006403L	01110001	11111011	XSOH	TSTIR	,BIP,XSTAT BLOCK ALREADY IN PROGRESS?
	006404L	01000101	00010000			
408.	006405L	11000010	01111011	BRA	XNCNTL,FZ	
409.	006406L	01010011	00010000	DOTI	,OR,BIP	THERE IS NOW!
410.	006407L	01010101	11111101	DORI	XSTAT,ND,-1-LDLE	CLR LDLE
	006410L	01101111	11111011			
411.	006411L	01010001	00000000	LDRI	XCRCL,0	CLR CRC
	006412L	01101111	11111111			
412.	006413L	01101111	11111110	LDRT	XCRCH	
413.	006414L	11001111	01001011	BRA	XBUMP	
414.						
415.	006415L	01110001	11111011	XDLE	TSTIR	,XPARG,XSTAT TRANSPARENT?
	006416L	01000101	00001000			
416.	006417L	11010010	11101100	BRA	XXDLE,FZ	
417.	006420L	01010011	00000010	DORI	XSTAT,OR,LDLE	NO, SET LDLE
	006421L	01101111	11111011			
418.	006422L	11001111	01110101	BRA	DOXCRC	
419.						
420.	006423L	01000101	00000010	XXDLE	TSTIT	,LDLE LEADING DLE?
421.	006424L	11000010	01111011	BRA	XNCNTL,FZ	
422.	006425L	01010011	00000010	DORI	XSTAT,OR,LDLE	NO, SET LDLE
	006426L	01101111	11111011			

423.	006427L	01110001	11111100	TSTIR	,1,XPNTR	DATA (NORMAL) OR STATUS (COM2)
	006430L	01000101	00000001			
424.	006431L	11000010	01001011	BRA	XBUMP,FZ	COM2
425.	006432L	11001111	00100101	BRA	XBIT	NORMAL, SEND IT AGAIN
426.						
427.	006433L	01110001	11111011	XFILL	TSTIR ,XPAR,XSTAT	TRANSPARENT MODE?
	006434L	01000101	00001000			
428.	006435L	11010011	11011100	BRA	XSYN,TZ	JUST SEND SYN IF NOT
429.	006436L	01010011	00000100	LDRI	XSTAT,OR,FILIP	ELSE SET FILIP
	006437L	01101111	11111011			
430.	006440L	01010001	00010000	LDRI	XDATA,DLE	AND SEND DLE
	006441L	01101111	11111101			
431.	006442L	11001111	00100101	BRA	XBIT	
432.						
433.	006443L	01010001	00110010	XSYN	LDRI XDATA,SYN	
	006444L	01101111	11111101			
434.	006445L	11001111	00100101	BRA	XBIT	
435.						

```

438.
439.
440.
441.
442.
443.
444.
445.
446.
447. 006446L 01110001 10111011
      006447L 01010010 00100000
      006450L 01101111 11111011
448. 006451L 11000000 00100101
449. 006452L 01110001 11111001
      006453L 01010101 11100000
450. 006454L 01110011 11111011
      006455L 01101111 11111011
451. 006456L 01110001 11111110
      006457L 01101111 11111101
452. 006460L 01010001 00100101
453. 006461L 11001111 01001010
454.

```

```

*
. XSTAT USAGE
.   0,1,2,3,4 --- 0
.   5,6,7 --- BCOUNT (BITS PER CHARACTER)
.
. XBUF STATUS
.   0 --- RDY (PRECEDING BYTE IS DATA TO XMIT)
.   1,2,3,4,5,6,7 --- 0
.
GSX  DORR  XSTAT,AC,BCOUNT,XSTAT,CC  BUMP THE BIT COUNT

BRA  XBIT,FC  JUST SEND A BIT IF NOT DONE WITH BYTE
DOTIR ,ND,WLNGTH,COMMODE  GET THE WORD LENGTH

DORR  XSTAT,OR,XSTAT  AND PUT IT INTO XSTAT

LDRR  XDATA,SYNCODE  SET UP FOR FILL IN CASE BUF NOT RDY

LDTA  XBIT  SET UP RETURN ADDRESS
BRA  CHKBUF  GO CHECK FOR MORE DATA

```

```

457,
458,
459,
460,
461,
462,
463,
464,
465,
466,
467,
468, 006462L 01110001 10111111
469, 006463L 11010010 10111111
470, 006464L 01110001 11111011
    006465L 01000101 00000011
471, 006466L 11010011 10110101
472, 006467L 01000101 00000010
473, 006470L 11010011 10101011
474, 006471L 01110001 11111001
    006472L 01010101 11100000
475, 006473L 01010011 00000001
    006474L 01101111 11111011
476, 006475L 01110001 11111110
    006476L 01101111 11111111
477, 006477L 01110001 10111111
478, 006500L 01010100 00000001
479, 006501L 01101111 11111111
480, 006502L 01110001 11111011
    006503L 01000101 00000010
481, 006504L 11010010 10110111
482, 006505L 01110001 11111101
483, 006506L 00110111 00100011
484, 006507L 11011111 00000011
485,
486, 006510L 01010001 00000000
487, 006511L 11011111 10111001
488,
489, 006512L 01011001 11110010
490, 006513L 01010001 10110010
491, 006514L 11001111 01001010
492, 006515L 11010010 10101110
493, 006516L 01010001 00000001
    006517L 00110111 00100011
494, 006520L 11011111 00000011
495,
496, 006521L 01010001 00000010
    006522L 01101111 11111011
497, 006523L 11011111 11000010
498,
499, 006524L 01010010 00100000
    006525L 01101111 11111011
500, 006526L 01110001 11111101
    
```

```

*
. XSTAT USAGE
.      0 --- CIP (CHARACTER IN PROGRESS)
.      1 --- SIP (START ELEMENT IN PROGRESS)
.      2,3,4 --- 0
.      5,6,7 --- BCOUNT (BITS PER CHARACTER)
.
. XBUF STATUS
.      0 --- RDY (PRECEDING BYTE IS DATA TO XMIT)
.      1,2,3,4,5,6,7 --- 0
.
ASX   LDTR   XICOUNT,CC      GET INTRP CNT
      BRA   DECCIC1,FZ    DECREMENT IT IF NOT DONE WITH BIT
      TSTIR ,SIP+CIP,XSTAT START OR CHARACTER IN PROGRESS?
      BRA   AXIDLE,TZ     IDLE IF NOT
      TSTIT ,SIP         START IN PROGRESS?
      BRA   AXCIP,TZ     CHARACTER IN PROGRESS IF NOT
      DOTIR ,ND,WLNTH,COMMODE GET WORD LENGTH
      DORI  XSTAT,OR,CIP  SET CHARACTER IN PROGRESS, CLEAR SIP
.
ASETIC LDRR   XICOUNT,AISTOR  INIT INTERRUPTS/BIT
.
DECCIC LDTR   XICOUNT,CC      GET ICOUNT
DECCIC1 DECT  DEC IT
      LDRT  XICOUNT         REPLACE IT
      YSTIR ,SIP,XSTAT     SENDING START ELEMENT?
.
AXBIT  LDTR   XDATA         SEND SPACE IF SO
      LDPT  SDLCOT         ELSE OUTPUT THE LSB
      BRA   CRETURN
.
AXSP   LDTI   0             SEND SPACE
      BRA   AXBIT
.
AXIDLE BPGX   AXIDLE1
      LDTA  AXIDLE1        SET RETURN ADDRESS
      BRA  CHKBUF         CHECK BUFFER FOR DATA
AXIDLE1 BRA   AXSTRT,FZ    SEND A START ELEMENT IF RDY
      LDPI  SDLCOT,1      ELSE SEND SOME MARK
.
      BRA   CRETURN
.
AXSTRT LDRI   XSTAT,SIP     SET START IN PROGRESS
.
      BRA   ASETIC
.
AXCIP  DORI  XSTAT,AC,BCOUNT BUMP THE BIT COUNT
.
      LDTR  XDATA         SHIFT THE DATA
    
```

501.	006527L	00010111	11010010	SHIFT	SR,CF	
502.	006530L	01101111	11111101	LDRT	XDATA	
503.	006531L	11010000	11000010	BRA	ASETIC,FC	INIT ICOUNT IF NOT DONE WITH CHARACTER
504.	006532L	01010001	00000000	LDRI	XSTAT,0	CLEAR CIP
	006533L	01101111	11111011			
505.	006534L	01010001	11111111	LDRI	XDATA,0377	SET UP STOP ELEMENT
	006535L	01101111	11111101			
506.	006536L	01110001	11111110	DORRR	XICOUNT,AC,AISTOR,AISTOR,C0	ASSUME 2 STOP ELEMENTS
	006537L	01110010	00111110			
	006540L	01101111	11111111			
507.	006541L	01110001	11111001	TSTIR	,2,COMMODE	IS IT 2 STOP ELEMENTS?
	006542L	01000101	00000010			
508.	006543L	11010010	11000000	BRA	DECXIC,FZ	JUMP IF SO
509.	006544L	01110001	10111110	DOTIR	,AC,1,AISTOR,CC	ELSE ASSUME 1.5 STOP ELEMENTS
	006545L	01010010	00000001			
510.	006546L	00010111	10010010	SHIFT	SR	"
511.	006547L	01110010	00111110	DORR	XICOUNT,AC,AISTOR,,C0	"
	006550L	01101111	11111111			
512.	006551L	01110001	11111001	TSTIR	,1,COMMODE	IS IT 1.5 STOP ELEMENTS?
	006552L	01000101	00000001			
513.	006553L	11010010	11000000	BRA	DECXIC,FZ	JUMP IF SO
514.	006554L	11011111	11000010	BRA	ASETIC	ELSE MUST BE 1 STOP ELEMENT
515.						

518,
 519, 006555L
 520,
 521,
 522,
 523,
 524,
 525,
 526,
 527,
 528,
 529,
 530,
 531,
 532,
 533,
 534,
 535,
 536,
 537,
 538,
 539,
 540,
 541,
 542,
 543,
 544,
 545,
 546,
 547,
 548,
 549,
 550,
 551,
 552,
 553,
 554,
 555, 006555L 00110001 11011100
 006556L 01000101 00000100
 556, 006557L 11010010 00000001
 557,
 558, 006560L 01010001 01110100
 559, 006561L 11011010 10001011
 560,
 561, 006562L 01010001 01101111
 562, 006563L 00110111 00001001
 563, 006564L 01101111 11110000
 564, 006565L 01110001 11111000
 006566L 01010011 00000001
 006567L 00110111 11000000
 565, 006570L 01101110 01110001
 566, 006571L 01110001 11111100

```

*
SIOUT:
.   (   167)  SICOUT          MOVE (A) TO XMIT BUFF COM2 7.4/5.2
.   (062 167)  SIOUT          MOVE (A) TO XMIT BUFF WRITE 9.4/7.2
.   (111 167)  SIMOUT        MOVE C BYTES FROM (HL) TO XBUF COM2 *
.   (113 167)  SIMOUT        MOVE C BYTES FROM (HL) TO XBUF WRITE *
.                                     * 4.85/7.35 +7N
.                                     SETS TZ IF BUFFER FULL
.
.   IF USER THEN IVIOL
.   ELSE LUF <= XPNTR,AND,0376-UXPNTR,OR,1+1
.   IF TZ THEN SIOEXIT
.   ELSE MAR <= UXPNTR,OR,1
.   IF (   167) OR (111 167) THEN (MAR) <= 3
.   ELSE (MAR) <= 1
.   TEMP1 <= UXPNTR
.   UXPNTR <= UXPNTR+2
.   IF (111 167) OF (113 167) THEN SIMOT
.   ELSE MAR <= MAR-1
.   (MAR) <= (URA)
.   END
.   END
.   END
.
SIMOT MAR <= HL
.   HL <= HL+1
.   MDR <= (MAR)
.   MAR <= XBUF,TEMP1
.   (MAR) <= MDR
.   URC <= URC-1
.   IF TZ THEN SIOEXIT
.   ELSE IF SRVREQ THEN SRVRPT
.   ELSE SIO
.
SIOEXIT: PSW <= PSW,AND,-1-SWRPT
.   END
.
TSTIP ,SWUSER,PSWI          USER MODE?
.
BRA CIVIOL,FZ              USERS CAN'T DO THIS!
.
BAL ,SIOT                  ASSUME SINGLE MOVE (EVEN)
BRA SIOCB,F#,ID           JMP IF (   167) OR (062 167)
.
BAL ,SIMOT                 SET UP FOR (111 167) OR (113 167)
STB DIMP                   SET UP FOR ZERO TEST
SIOCB BAS LINK
SIORPT DOPIR MAROL,OR,1,UXPNTR SET TZ IF FULL
.
INCR TEMP1                 29I- 23FEB78 - HJS - SAVE A WORD
DOTIR ,ND,0376,XPNTN
    
```

567.	006572L 01010101 11111110			
	006573L 01110100 00110001	DOPR	LUF,SB,TEMP1,,C0	
	006574L 00110111 00000110			
568.	006575L 11010011 01010100	BRA	SIOEXIT,TZ	DO NO MORE IF FULL
569.	006576L 01010001 11011110	LDPI	MAROH,XBUF>8	
	006577L 00110111 11100000			
570.	006600L 01010001 00000011	LDTI	3	ELSE MARK LOCATION USED
571.	006601L 11011001 01111100	BRA	SIOWB,T0,IZ	LIKE EX=COM2 IF (167) OR (111 167)
572.				29AUG77
573.	006602L 01010001 00000001	LDTI	1	LIKE EX=WRITE IF (062 167) OR (113 167)
574.	006603L 00110111 00100001	LDPT	MDW	"
575.	006604L 01110001 10111000	LDRR	TEMP1,UXPNTR,CC	SAVE USER POINTER
	006605L 01101111 10110001			
576.	006606L 01010010 00000010	DORI	UXPNTR,AC,2	INC USER POINTER BY 2
	006607L 01101111 11111000			
577.	006610L 11010100 01110111	MWAIT	,CMEMPF	" (400)
	006611L 11010111 00000101			
578.	006612L 11101111 00000000	BRR	LINK	
579.				
580.	006613L 00110111 00001101	SIOT	STB	POINT AT BUFFER DATA LOCATION
581.	006614L 00110001 11010000	LDPP	MDW,URI+URA	MOVE A TO BUFFER
	006615L 00110111 00100001			
582.	>006616L 01011001 11111111	BRAX	FETCHW	AND CALL IT QUILTS
	>006617L 11001111 11111111			
583.				
584.	006620L 00110001 11000110	SIMOT	DLDX	HL2MR,,SMR
	006621L 00110001 11100101			
	006622L 00110111 01000111			
585.	006623L 00110001 10010000	DADDP	URO+UR,MARI	INC HL
	006624L 00010110 01110010			
	006625L 00110111 10000110			
	006626L 00110001 10110000			
	006627L 00110110 10000101			
586.	006630L 11010100 01100111	MWAIT	,CMEMPF	(200)
	006631L 11010111 00000101			
587.	006632L 01010001 11011110	LDPI	MAROH,XBUF>8	POINT AT BUFFER
	006633L 00110111 11100000			
588.	006634L 01110001 10110001	LDPR	MAROL,TEMP1,,CC	USE UN=INCREMENTED UXPNTN
	006635L 00110111 11000000			
589.	006636L 00110001 00110110	LDPP	MDW,MDR	MOVE DATA FROM (HL) TO BUFFER
	006637L 00110111 00100001			
590.	006640L 00110001 11010010	DOPIP	URO+URC,SB,1,URI+URC	DEC C
	006641L 01010100 00000001			
	006642L 00110111 10000010			
591.	006643L 11010011 01010100	BRA	SIOEXIT,TZ	QUIT IF IT'S NOW 0
592.				
593.	006644L 00110001 00110000	LDTP	SRVREQ	CHECK FOR SERVICE REQUEST
594.	006645L 11010100 01011010	MWAIT	,CMEMPF	(200)
	006646L 11010111 00000101			
595.	006647L 11010011 10001010	BRA	SIORPT,TZ	REPEAT IF NO SERVICE
596.				29I - HJS - 23FEB78 - RE-ORGANIZATION
597.	006650L 00110111 00001000	STB	IIMP	RESTORE IMP IF IT WAS 111 OR 113

598. >006651L 01011001 11111111
>006652L 11001111 11111111

BRAX SRVRPT HANDLE SERVICE IF IT'S THERE

599.
600. 006653L 00110001 11011100
006654L 01010101 11011111
006655L 00110111 10001100

SIOEXIT DOPIP PSWD,ND,-1-SWRPT,PSWI CLEAR THE REPEAT FLAG

601. >006656L 01011001 11111111
>006657L 11001111 11111111

BRAX FETCHW AND IT'S ALL OVER!

602.

```

605.
606. 006660L
607.
608.
609.
610.
611.
612.
613.
614.
615.
616.
617.
618.
619.
620.
621.
622.
623.
624.
625. 006660L 00110001 11011100
      006661L 01000101 00000100
626. 006662L 11010010 00000001
627.
628. 006663L 11011000 00111100
629.
630. 006664L 00110001 11010000
      006665L 01101111 11111001
631. 006666L 00110001 11010001
      006667L 01101111 11111110
632. 006670L 01010001 00000001
      006671L 00110111 00100011
633. 006672L 01010001 00000000
      006673L 01101111 11111100
634. 006674L 01101111 11111000
635. 006675L 01101111 11111111
636. 006676L 01101111 11110100
637. 006677L 01101111 11111010
638. 006700L 01101111 11110110
639. 006701L 01010001 11100000
      006702L 01101111 11111011
640. 006703L 01010001 00000100
      006704L 01101111 11110011
641. >006705L 01011001 11111111
      >006706L 11001111 11111111
642.

```

```

*
SISTART:
. ( 165) SISTART          COMMENSE COMMUNICATING COMMAND 4.65
. (111 165) SISYNC       SEEK SYNC (LOOK FOR SYN=SYN) 5.10
.
.   IF USER THEN IVIOL
.   ELSE IF FIZ THEN SISYN      *** ',.' MISSING
.   ELSE COMMODE <= A-REG
.   SYNCODE <= B-REG
.   SDLCOT <= 1
.   XPNTR <= 0
.   UXPNTR <= 0
.   XICOUNT <= 0
.   RPNTR <= 0          *** ILLEGAL CHARACTERS (000)
.   RPNTR <= 0          *** SEEN INSTEAD OF ',.'
.   RICOUNT <= 0       *** SEEN ON ONE OF THESE LINES
.   XSTAT <= BCOUNT#7  *** WARNING FOR BAD FILE
.   RSTAT <= SYNWAIT
.   END
.
.   SISYN
.
.   TSTIP ,SWUSER,PSWI      USER MODE?
.
.   BRA CIVIOL,FZ          USERS CAN'T DO THIS!
.
.   BRA SISYN,F#,IZ       JUST SET SYNWAIT IF IMP#0
.
.   LDRP COMMODE,URI+URA  A-REG => COMMODE
.
.   LDRP SYNCODE,URI+URB  B-REG => SYNCODE
.
.   LDPI SDLCOT,1         SEND MARKS
.
.   LDRI XPNTR,0          INIT XMT BUFFER POINTER
.
.   LDRT UXPNTR           AND USER XMIT BUFFER POINTER
.   LDRT XICOUNT          AND XMIT INTERRUPT COUNTER
.   LDRT RPNTR            AND RCV BUFFER POINTER
.   LDRT URPNTR           AND USER RCV BUFFER POINTER
.   LDRT RICOUNT          AND RCV INTERRUPT COUNTER
.   LDRI XSTAT,0340      BCOUNT#7, ALL ELSE RESET
.
.   SISYN LDRI RSTAT,SYNWAIT  ALL ELSE RESET
.
.   BRAX FETCHI

```



```

645.
646. 006707L
647.
648.
649.
650.
651.
652.
653.
654.
655.
656.
657.
658.
659.
660.
661.
662.
663.
664.
665. 006707L 00110001 11011100
      006710L 01000101 00000100
666. 006711L 11010010 00000001
667. 006712L 11011010 00101001
668. 006713L 00110111 00001001
669. 006714L 00110001 00010111
670. 006715L 11011001 00101101
671. 006716L 00110001 11010000
      006717L 00110111 00100100
672. >006720L 01011001 11111111
      >006721L 11001111 11111111
673.
674. 006722L 01010101 00111111
      006723L 00110111 10000000
675. >006724L 01011001 11111111
      >006725L 11001111 11111111
676.
677. 006726L 11011001 00100100
678. 006727L 00110001 11010000
      006730L 00110111 00100101
679. >006731L 01011001 11111111
      >006732L 11001111 11111111
680.
681. 006733L 00110001 00010000
      006734L 01010101 01111000
      006735L 00110111 10000000
682. 006736L 01110001 11111100
      006737L 01010101 11111110
683. 006740L 01110100 00111000
      006741L 00110111 00000110
684. 006742L 01110001 11111001
      006743L 01010101 00000111

```

```

*
SICID:
. ( 161) SIMODIN GET THE MODEM AND XMIT BUFFER STATUS 4.45
. 5.1 IF SDLC MODE
. SET TZ IF XMIT BUFFER EMPTY
. SET TC IF LINE ACTIVE (SDLC)
. (062 161) SIMODOUT SET THE MODEM CONTROLS 5.3
. (111 161) SIACUIN GET THE ACU STATUS 5.55
. (113 161) SIACUOUT SET THE ACU CONTROLS 5.5
.
. IF USER THEN INVIOL
. ELSE CASE OF IMP
. 000: A-REG <= MODIN.AND.0170
. LUF <= UXPNTR-XPNTR
. LUCF <= LINE ACTIVE
. 062: SDLCMD <= A-REG
. 111: A-REG <= ACUIN.AND.077
. 113: ACUOT <= A-REG
.
. END
.
. TSTIP ,SWUSER,PSWI USER MODE?
.
. BRA CIVIOL,FZ USERS CAN'T DO THIS!
. BRA SIMOD,F#,IO MODEM CONTROL IF ( 161) OR (062 161)
. STB DIMP (111 161)=>TIZ, (113 161)=>FIZ
. LDTP ACUIN GET THE ACU INPUT AND ALLOW TIZ SETUP
. BRA SIACUIN,T#,IZ (111 161)=>ACUIN
. LDPP ACUOT,URI+URA ELSE DO ACUOUT
.
. BRAX FETCHI
.
. SIACUIN DOPI URO+URA,ND,077 ONLY 6 BITS - 15JUL77
.
. BRAX FETCHI
.
. SIMOD BRA SIMODIN,T#,IZ ( 161)=>MODIN
. SIMODOT LDPP SDLCMD,URI+URA ELSE DO MODOUT
.
. BRAX FETCHI
.
. SIMODIN DOPIP URO+URA,ND,0170,MODIN ONLY 4 BITS - 15JUL77
.
. DOTIR ,ND,0376,XPNTR SET TZ IF XMIT BUFFER MT
.
. DOPR LUF,SB,UXPNTR,,C0
.
. DOTIR ,ND,7,COMMODOE CHECK MODE - 29AUG77

```

685. 006744L 01000000 00000011
686. 006745L 11010010 00010010
687. 006746L 01110001 11110011
688. 006747L 00010111 10110010
689. 006750L 01010101 11111110
690. 006751L 01101111 11110011
691. 006752L 00010111 10010010
692. 006753L 01010010 10000000
006754L 00110111 00000111
693. >006755L 01011001 11111111
>006756L 11001111 11111111
694.

TSTIT	XR,3	IS IT SDLC? = 29AUG77
BRA	SICIOX,FZ	ALL DONE IF NOT = 29AUG77
LDTR	RSTAT	
CCLR		LA=LSB => LINK
DDTI	,ND,-1=LA	RESET LA
LDRT	RSTAT	AND PUT RSTAT BACK
SHIFT	SR	LA => MSB
DOPI	LUCF,AC,0200	LA => CARRY => UCF
SICIOX	BRAX	FETCHI

697.
698. 006757L 11111111 11111111
 006760L 11111111 11111111
 006761L 11111111 11111111
 006762L 11111111 11111111
 006763L 11111111 11111111
 006764L 11111111 11111111
 006765L 11111111 11111111
 006766L 11111111 11111111
 006767L 11111111 11111111

PAD -S-010,AND,0377

699.
700. 006770L 01010001 00000000
 006771L 01101111 11111001
 701. >006772L 01011001 11111111
 >006773L 11001111 11111111
 702. >006774L 01011001 11111111
 >006775L 11001111 11111111
 703. >006776L 01011001 11111111
 >006777L 11001111 11111111

CMEMPK LDRI COMMODE,0

TURN COMM OFF IF MEMORY FAULT

CMEMPF BRAX MEMPFS

CRETURN BRAX SRVNXT

CTVIOL BRAX IVIOLS

704.
705. 001000
706. 006000
707. 006000
708.

CDOXLEN EQU S=CDOXP
 USE CDOXL
 SKIP CDOXLEN

```

711.
712. 007000
713. 000000
714. 007000
715. 000000
716. 007000L
717.
718.
719.
720.
721. 007000L 01110001 11111001
      007001L 01000101 00000100
722. 007002L 11010010 01111001
723. 007003L 01000101 00000010
724. 007004L 11000011 10001101
725. 007005L 01000101 00000001
726. 007006L 11010011 11000110
727.
728.

```

```

CDORL  ORG  CDOR          LOGICAL SPACE DEFINED IN PLACE
CDORP  ORG  0            PHYSICAL SPACE RELOCATABLE
CDORL  USE  CDORL        USE THEM BOTH
        USE  CDORP        PUT THE CODE IN PHYSICAL SPACE
CDORP  LOC  CDORL,2      WITH ADDRESSES IN LOGICAL SPACE

```

```

*
. SDOR COMES HERE TO SEE IF BISYNC, GENSYNC, ASYNC, OR SDLC
.

```

```

TSTIR  ,4,COMMODE

```

```

BRA    ASR,FZ

```

```

IT'S ASYNC (XX XXX 1XX)

```

```

TSTIT  ,2

```

```

BRA    BSCR,TZ

```

```

IT'S BISYNC (XX XXX 00X)

```

```

TSTIT  ,1

```

```

BRA    GSR,TZ

```

```

IT'S GENSYNC (XX XXX 010)
ELSE IT'S SDLC (XX XXX 011)

```


777.	007045L	11000011	11010100	BRA	FB0,TZ	
778.	007046L	01010000	00010000	DORI	RCRCH,XR,POLYHI	
	007047L	01101111	11110110			
779.	007050L	01110001	11110111	DORIR	RCRCL,XR,POLYLO,RCRCL	
	007051L	01010000	00001000			
	007052L	01101111	11110111			
780.	007053L	01110001	11110111	FB0	LDTR	RCRCL
781.	007054L	00010111	10010010	SHIFT	SR	LINK IS STILL FEEDBACK
782.	007055L	01101111	11110111	LDRT	RCRCL	
783.	007056L	01110001	11110110	LDTR	RCRCH	
784.	007057L	00010111	10010010	SHIFT	SR	
785.	007060L	01101111	11110110	LDRT	RCRCH	
786.						
787.	007061L	01110001	10110011	DORIR	RSTAT,AC,BCOUNT,RSTAT,CC	BUMP THE BIT COUNT
	007062L	01010010	00100000			
	007063L	01101111	11110011			
788.	007064L	11010000	00000011	BRA	CRETURN,FC	DONE IF NOT 8 BITS
789.						
790.	007065L	01000101	00010000	TSTIT	,FIP	FRAME IN PROGRESS?
791.	007066L	11000010	01000101	BRA	RSETRDY,FZ	STORE THE BYTE IF SO
792.	007067L	01110001	11111001	TSTIR	,ADDEDET,COMMODE	ADDRESS MODE?
	007070L	01000101	00001000			
793.	007071L	11000011	11000000	BRA	SETFIP,TZ	
794.	007072L	01110001	11110101	TSTIR	XR,GLOBAL,RDATA	GOT GLOBAL ADDRESS? - 29AUG77
	007073L	01000000	11111111			
795.	007074L	11000011	11000000	BRA	SETFIP,TZ	YES, ACCEPT IT -29AUG77
796.	007075L	00110000	01010001	TSTTP	XR,SNID	GOT MY ADDRESS? - 29AUG77
797.	007076L	11000010	10111100	BRA	GOIDLE,FZ	IGNORE IF NOT -29AUG77
798.	007077L	01110001	11110011	SETFIP	DORIR	RSTAT,OR,FIP,RSTAT
	007100L	01010011	00010000			SET FRAME IN PROGRESS
	007101L	01101111	11110011			
799.	007102L	11001111	01000101	BRA	RSETRDY	
800.						
801.	007103L	01110001	11110011	GOIDLE	DOTIR	,ND,LA,RSTAT
	007104L	01010101	00000001			0 BITS, NO FIP
802.	007105L	01010011	00000100	DORI	RSTAT,OR,IDLE	SET IDLE
	007106L	01101111	11110011			
803.	007107L	01110001	11111001	DORIR	COMMODE,ND,037,COMMODE	SET ONES COUNTER TO 7 (PHYSICAL=0)
	007110L	01010101	00011111			
	007111L	01101111	11111001			
804.	007112L	11011111	00000011	BRA	CRETURN	
805.						
806.	007113L	01110001	11110011	RFLAG	TSTIR	,FIP,RSTAT
	007114L	01000101	00010000			FRAME IN PROGRESS?
807.	007115L	11000010	10011001	BRA	EOF,FZ	MUST BE END OF FRAME IF SO
808.	007116L	01010001	11111111	LDRI	RCRCH,0377	
	007117L	01101111	11110110			
809.	007120L	01101111	11110111	LDRT	RCRCL	
810.	007121L	01110001	11110011	DORIR	RSTAT,ND,LA,RSTAT	0 BITS, NO FIP, NOT IDLE
	007122L	01010101	00000001			
	007123L	01101111	11110011			
811.	007124L	11011111	00000011	BRA	CRETURN	

812.					
813.	007125L	01110001	10111001	GOTAONE TSTIR	,0340,COMMODE,CC SEVEN ONES?
	007126L	01000101	11100000		
814.	007127L	11010011	00000011	BRA	CRETURN,TZ WAIT FOR A 0 IF SEVEN ONES (PHYSICAL=0)
815.	007130L	01010010	00100000	DDRI	COMMODE,AC,040 BUMP THE COUNTER
	007131L	01101111	11111001		
816.	007132L	01010101	11100000	DOTI	,ND,0340 LOOK AT THE COUNTER
817.	007133L	11000010	11100110	BRA	SDLCR1,FZ GO ON IF NOT SEVEN ONES (PHYSICAL=0)
818.	007134L	01110001	11110011	TSTIR	,FIP,RSTAT FRAME IN PROGRESS?
	007135L	01000101	00010000		
819.	007136L	11000011	10111100	BRA	GOIDLE,TZ
820.	007137L	01010001	01111111	LDRI	RDATA,ABORT TELL HIM ABOUT ABORT - 29AUG77
	007140L	01101111	11101010		
821.	007141L	01110001	11110011	BADCRC DDRI	RSTAT,ND,LA,RSTAT 0 BITS, NO FIP, NOT IDLE
	007142L	01010101	00000001		
	007143L	01101111	11110011		
822.	007144L	01010001	10000001	LDTI	0201 BAD END OF FRAME
823.	007145L	11001111	01000100	BRA	RPUTRBS TELL HIM ABOUT IT
824.					
825.	007146L	01010001	01111110	EOF LDRI	RDATA,FLAG TELL HIM ABOUT END FLAG - 29AUG77
	007147L	01101111	11110101		
826.	007150L	01110001	11110110	TSTIR	XR,GDCRCH,RCRCH CHECK CRC
	007151L	01000000	10000101		
827.	007152L	11000010	10011110	BRA	BADCRC,FZ
828.	007153L	01110001	11110111	TSTIR	XR,GDCRCL,RCRCL
	007154L	01000000	01001111		
829.	007155L	11000010	10011110	BRA	BADCRC,FZ
830.	007156L	01010001	00000001	LDRI	RSTAT,LA 0 BITS, NO FIP, NOT IDLE
	007157L	01101111	11110011		
831.	007160L	01010001	11000001	LDTI	0301 GOOD END OF FRAME
832.	007161L	11001111	01000100	BRA	RPUTRBS TELL HIM ABOUT IT
833.					

836,
837,
838,
839,
840,
841,
842,
843,
844,
845,
846,
847,
848,
849,
850,
851,
852, 007162L 00110001 00010110
853, 007163L 01110001 10110101
854, 007164L 00010111 10010010
855, 007165L 01101111 11110101
856, 007166L 01110001 11110011
007167L 01000101 00000100
857, 007170L 11000010 00110000
858, 007171L 01010010 00100000
007172L 01101111 11110011
859, 007173L 11010000 00000011
860, 007174L 01000101 00000001
861, 007175L 11000010 00100110
862, 007176L 01000101 00000010
863, 007177L 11000010 00010000
864, 007200L 01000101 00001000
865, 007201L 11010010 11001101
866, 007202L 01110001 11110101
007203L 01000000 00010000
867, 007204L 11010011 11101010
868, 007205L 01000000 00110010
869, 007206L 11010011 11100110
870, 007207L 01000000 00000010
871, 007210L 11010011 11100010
872, 007211L 01000000 00000001
873, 007212L 11010011 11100010
874, 007213L 01000000 00000011
875, 007214L 11010011 11011001
876, 007215L 01000000 00100110
877, 007216L 11010011 11011001
878, 007217L 01000000 00011111
879, 007220L 11010011 11010110
880, 007221L 01000000 00101101
881, 007222L 11010011 11010001
882, 007223L 01110001 11110101
007224L 01110000 11110111
007225L 01101111 11110001

```

*
. RSTAT USAGE
.      0 --- EOB (END OF BLOCK)
.      1 --- LDLE (LEADING DLE)
.      2 --- SYNWAIT
.      3 --- CRC (CRC IF EOB, XPAR IF NOT)
.      4 --- BIP (BLOCK IN PROGRESS)
.      5,6,7 --- BCOUNT (BITS PER CHARACTER)

. RBUF STATUS
.      0 --- RDY (PRECEDING BYTE IS RECEIVED DATA      =7 )
.      1,2,3,4,5 --- 0
.      6 --- GOOD EOB (0 UNLESS BIT=7=1)
.      7 --- EOB (END OF BLOCK; IF BIT=6 =1, PRECEDING BYTE AND
.              LAST RECEIVED BYTE ARE CRC)

BSCR  LDTP      SDLCIN      GET NEW BIT (BIT 0)
      LDTR      RDATA,CC    LINK=NEW BIT
      SHIFT     SR          SHIFT IN NEW BIT
      LDRT      RDATA
      TSTIR     ,SYNWAIT,RSTAT  WAITING FOR SYN=SYN?

      BRA      RSYNLOP,FZ
      DORI     RSTAT,AC,BCOUNT  INC THE BIT COUNT

      BRA      CRETURN,FC      RETURN IF NOT 8 BITS
      TSTIT    ,EOB           TIME TO CHECK CRC?
      BRA      CHKCRC,FZ
      TSTIT    ,LDLE          LEADING DLE?
      BRA      RLDLE,FZ
      TSTIT    ,XPAR          TRANSPARENT?
      BRA      CHKXDLE,FZ     THEN CHECK FOR DLE
RCHKDLE TSTIR   XR,DLE,RDATA  RECEIVED DLE?

RCHKSYN BRA      RDLE,TZ
      TSTIT   XR,SYN
      BRA      RSYN,TZ
      TSTIT   XR,STX
      BRA      RSOB,TZ
      TSTIT   XR,SOH
      BRA      RSOB,TZ
      TSTIT   XR,ETX
      BRA      REOB,TZ
      TSTIT   XR,ETB
      BRA      REOB,TZ
      TSTIT   XR,ITB
      BRA      REOB1,TZ
      TSTIT   XR,ENQ
      BRA      RENQ,TZ
DORCRC DORRR   TEMP1,XR,RCRCL,RDATA  DATA,XR.T->TEMP1
    
```


883.	007226L	01110010	00110001	DOTR	,AC,TEMP1,,C0	SL (LSB=0)
884.	007227L	01110000	10110001	DORR	TEMP1,XR,TEMP1,,CC	TEMP1,XR,T=>TEMP1 (LINK=0)
	007230L	01101111	11110001			
885.	007231L	00010111	11010010	DORR	TEMP2,FT,Q,,CF+SR	SR->TEMP2
	007232L	01101111	11110010			
886.	007233L	00010111	11010010	SHIFT	SR,CF	SR
887.	007234L	00010111	11010010	SHIFT	SR,CF	SR
888.	007235L	01110000	11110010	DORR	TEMP2,XR,TEMP2	TEMP2,XR,T=>TEMP2
	007236L	01101111	11110010			
889.				RPT	4	
890.	007237L	00010111	11010010	SHIFT	SR,CF	SR BY 4
890.	007240L	00010111	11010010	SHIFT	SR,CF	SR BY 4
890.	007241L	00010111	11010010	SHIFT	SR,CF	SR BY 4
890.	007242L	00010111	11010010	SHIFT	SR,CF	SR BY 4
891.	007243L	01110000	11110010	DORR	TEMP2,XR,TEMP2	TEMP2,XR,T=>TEMP2
	007244L	01101111	11110010			
892.	007245L	01010101	00000001	DOTI	,ND,1	
893.	007246L	01110000	11110110	DORR	RCRCL,XR,RCRCH	CRCH,XR,T=>CRCL
	007247L	01101111	11110111			
894.	007250L	01110001	11110001	LDTR	TEMP1	
895.	007251L	00010111	10110010	CCLR		LSB->LINK
896.	007252L	00010111	10010010	SHIFT	SR	SRC
897.	007253L	00010111	10110010	CCLR		LSB->LINK
898.	007254L	00010111	10010010	SHIFT	SR	SRC
899.	007255L	01101111	11110001	LDRT	TEMP1	T->TEMP1
900.	007256L	01010101	11000000	DOTI	,ND,0300	
901.	007257L	01110000	11110111	DORR	RCRCL,XR,RCRCL	CRCL,XR,T=>CRCL
	007260L	01101111	11110111			
902.	007261L	01110001	11110010	LDTR	TEMP2	TEMP2->T
903.				RPT	5	
904.	007262L	00010111	10100010	SHIFT	SL	SL BY 5
904.	007263L	00010111	10100010	SHIFT	SL	SL BY 5
904.	007264L	00010111	10100010	SHIFT	SL	SL BY 5
904.	007265L	00010111	10100010	SHIFT	SL	SL BY 5
904.	007266L	00010111	10100010	SHIFT	SL	SL BY 5
905.	007267L	01010101	11000000	DOTI	,ND,0300	
906.	007270L	01110000	11110001	DORR	RCRCH,XR,TEMP1	TEMP1,XR,T=>CRCH
	007271L	01101111	11110110			
907.	007272L	01010001	00000001	RSETRDY	LDTR	1
908.	007273L	01101111	11110001	RPUTRBS	LDRT	BSTAT
909.	007274L	11000100	01000011	RPUTDAT	MWAIT	,CHEMPK
	007275L	11010111	00000111			
910.	007276L	00110001	11011100	DOPIP	MODW,ND,-1-SWUSER,PSWI	MAKE SURE WE CAN USE THE MEMORY!
	007277L	01010101	11111011			
	007300L	00110111	00000100			
911.	007301L	01010001	11011111	LOPI	MAR0H,RBUF>8	POINT TO NEXT DATA
	007302L	00110111	11100000			
912.	007303L	01110001	11110100	LDPR	MAR0L,RPNTR	
	007304L	00110111	11000000			
913.	007305L	01110001	11110101	LDPR	MDW,RDATA	WRITE THE DATA
	007306L	00110111	00100001			
914.	007307L	01110001	11110001	LDTR	BSTAT	GET THE STATUS

915.	007310L	11000100	00110111	MWAIT	,CHEMPK	
	007311L	11010111	00000111			
916.	007312L	00110111	00001100	STB	IMAR	POINT TO STATUS
917.	007313L	00110111	00100001	LDPT	MDW	WRITE THE STATUS
918.	007314L	00110001	10010000	LDTP	MARIL	GET THE POINTER
919.	007315L	01101110	01110100	INCR	RPNTR	AND BUMP IT
920.	007316L	11011111	00000011	BRA	CRETURN	
921.						
922.	007317L	01110001	11110111	RSYNLOP	LDTR	GET LAST BYTE
923.	007320L	00010111	10010010	SHIFT	SR	SHIFT IN LSB JUST SHIFTED OUT OF RDATA
924.	007321L	01101111	11110111	LDRT	RDATA2	AND PUT IT BACK
925.	007322L	01000000	00110010	TSTIT	XR,SYN	WAS LAST BYTE A SYN?
926.	007323L	11010010	00000011	BRA	CRETURN,FZ	NO, KEEP WAITING
927.	007324L	01110001	11110101	TSTIR	XR,SYN,RDATA,,TW	IS CURRENT BYTE A SYN?
	007325L	01010000	00110010			
928.	007326L	11010010	00000011	BRA	CRETURN,FZ	NO, KEEP WAITING
929.	007327L	01101111	11110011	LDRT	RSTAT	YES! ZERO RSTAT
930.	007330L	11011111	11100110	BRA	RSYN	
931.						
932.	007331L	01000101	00001000	CHKCRC	TSTIT	,CRC
933.	007332L	11000010	00100001	BRA	CHKCRC1,FZ	ALREADY HAVE FIRST CRC?
934.	007333L	01010011	00001000	DORI	RSTAT,OR,CRC	NO,BUT WE DO NOW!
	007334L	01101111	11110011			
935.	007335L	11001111	01101100	BRA	DORCRC	
936.						
937.	007336L	01010001	10000001	CHKCRC1	LDRI	BSTAT,0201
	007337L	01101111	11110001			RDY AND EOB
938.	007340L	01110001	11110101	DOTRR	,XR,RCRCL,RDATA	(DATA,XR,CRCL)+CRCH=0?
	007341L	01110000	11110111			
939.	007342L	01110011	11110110	DOTR	,OR,RCRCH	
940.	007343L	11000010	00011001	BRA	CHKCRC2,FZ	
941.	007344L	01010001	11000001	LDRI	BSTAT,0301	RDY,EOB,OK IF ALL IS COOL
	007345L	01101111	11110001			
942.	007346L	01110001	11110011	CHKCRC2	DOTIR	,ND,BIP,RSTAT
	007347L	01010101	00010000			CLR ALL BUT BIP
943.	007350L	11000010	00010101	BRA	CHKCRC3,FZ	IF BIP IS SET LEAVE IT ALONE
944.	007351L	01010001	00000100	LDTI	SYNWAIT	ELSE CLR ALL, SET SYNWAIT
945.	007352L	01101111	11110011	CHKCRC3	LDRT	RSTAT
946.	007353L	01010001	00000000	LDRI	RCRCL,0	CLR CRC
	007354L	01101111	11110111			
947.	007355L	01101111	11110110	LDRT	RCRCH	
948.	007356L	11001111	01000011	BRA	RPUTDAT	
949.						
950.	007357L	01010101	11111101	RLDLE	DORI	RSTAT,ND,-1-LDLE
	007360L	01101111	11110011			RESET LDLE
951.	007361L	01000101	00001000	TSTIT	,XPAR	
952.	007362L	11000010	00000101	BRA	RXLBLE,FZ	
953.	007363L	01110001	11110101	TSTIR	XR,STX,RDATA	DLE=STX?
	007364L	01000000	00000010			
954.	007365L	11000010	01111101	BRA	RCHKDLE,FZ	
955.	007366L	01110001	11110011	DORIR	RSTAT,OR,XPAR,RSTAT	SET XPAR IF SQ
	007367L	01010011	00001000			

956.	007370L	01101111	11110011		
957.	007371L	11011111	11100010	BRA	RSOB
958.	007372L	01110001	11110101	RXLBLE	TSTIR XR,SYN,RDATA DLE=SYN?
959.	007373L	01000000	00110010		
960.	007374L	11010011	11101110	BRA	RDLESYN,TZ
961.	007375L	11000100	00000010	RPUTDLE	MWAIT ,CMEMPK
962.	007376L	11010111	00000111		
963.	007377L	00110001	11011100	DOPIP	MODW,ND,-1=SWUSER,PSWI MAKE SURE WE CAN USE THE MEMORY!
964.	007400L	01010101	11111011		
965.	007401L	00110111	00000100		
966.	007402L	01010001	11011111	LDPI	MAR0H,RBUF>8 ELSE STUFF DLE IN BUFFER
967.	007403L	00110111	11100000		
968.	007404L	01110001	11110100	LDPR	MAR0L,RPNTR
969.	007405L	00110111	11000000		
970.	007406L	01010001	00010000	LDPI	MDW,DLE
971.	007407L	00110111	00100001		
972.	007410L	01010001	00000001	LDTI	1 SET UP RDY STATUS
973.	007411L	11010100	11110110	MWAIT	,CMEMPK
974.	007412L	11010111	00000111		
975.	007413L	00110111	00001100	STB	IMAR POINT TO STATUS
976.	007414L	00110111	00100001	LDPT	MDW WRITE THE STATUS
977.	007415L	00110001	10010000	LDTP	MARIL GET THE POINTER
978.	007416L	01101110	01110100	INCR	RPNTR AND BUMP IT
979.	007417L	01110001	11110101	LDTR	RDATA
980.	007420L	11001111	01111010	BRA	RCHKSYN
981.	007421L	01110001	11111001	RDLESYN	TSTIR ,SYNDEL,COMMODE
982.	007422L	01000101	00001000		
983.	007423L	11010010	00000011	BRA	CRETURN,FZ
984.	007424L	11001111	00000010	BRA	RPUTDLE
985.	007425L	01110001	11110011	RDLE	DORIR RSTAT,OR,LDLE,RSTAT SET LDLE
986.	007426L	01010011	00000010		
987.	007427L	01101111	11110011		
988.	007430L	11001111	01101100	BRA	DORCRC
989.	007431L	01110001	11111001	RSYN	TSTIR ,SYNDEL,COMMODE STRIP SYN?
990.	007432L	01000101	00001000		
991.	007433L	11010010	00000011	BRA	CRETURN,FZ
992.	007434L	11001111	01000101	BRA	RSETRDY
993.	007435L	01110001	11110011	RSOB	TSTIR ,BIP,RSTAT BLOCK ALREADY IN PROGRESS?
994.	007436L	01000101	00010000		
995.	007437L	11000010	01101100	BRA	DORCRC,FZ
996.	007440L	01010011	00010000	DORI	RSTAT,OR,BIP IT IS NOW!
997.	007441L	01101111	11110011		
998.	007442L	01010001	00000000	LDRI	RCRCL,0 CLR CRC
999.	007443L	01101111	11110111		
1000.	007444L	01101111	11110110	LDRT	RCRCH
1001.	007445L	11001111	01000101	BRA	RSETRDY

992.	007446L	01110001	11110011	REOB	DORIR	RSTAT,ND,-1-BIP,RSTAT CLR BIP	
	007447L	01010101	11101111				
	007450L	01101111	11110011				
993.	007451L	01110001	11110011	REOB1	DOTIR	,ND,-1-XPAR,RSTAT CLR XPAR	
	007452L	01010101	11110111				
994.	007453L	01010011	00000001		DORI	RSTAT,OR,EOB SET EOB	
	007454L	01101111	11110011				
995.	007455L	11001111	01101100		BRA	DORCRC	
996.							
997.	007456L	01010001	00000100	RENG	LDRI	RSTAT,SYNWAIT	
	007457L	01101111	11110011				
998.	007460L	01010001	10000001		LDTI	0201 RDY AND EOB FOR BSTAT	
999.	007461L	11001111	01000100		BRA	RPUTRBS	
1000.							
1001.	007462L	01110001	11110101	CHKXDLE	TSTIR	XR,DLE,RDATA IS THIS A DLE?	
	007463L	01000000	00010000				
1002.	007464L	11000010	01101100		BRA	DORCRC,FZ NO	
1003.	007465L	01110001	11110011		DORIR	RSTAT,OR,LDLE,RSTAT YES, REMEMBER IT!	
	007466L	01010011	00000010				
	007467L	01101111	11110011				
1004.	007470L	11011111	00000011		BRA	CRETURN	
1005.							

```

1008.
1009.
1010.
1011.
1012.
1013.
1014.
1015.
1016.
1017.
1018.
1019. 007471L 01110001 11110011
      007472L 01010101 00000100
      007473L 11010011 10101011
1020.
1021.
1022. 007474L 01110001 10110111
1023. 007475L 00010111 10010010
1024. 007476L 01101111 11110111
1025. 007477L 01110001 11111001
      007500L 01010101 11100000
1026. 007501L 00010111 10100010
1027. 007502L 00010111 10100010
1028. 007503L 00010111 10100010
1029. 007504L 01010100 01010011
      007505L 01101111 10110000
1030. 007506L 01110001 11110101
      007507L 01110100 11110101
1031. 007510L 00010111 10010010
1032. 007511L 01011001 11110000
1033. 007512L 11101111 00000000
1034. 007513L 00010111 10010010
1035. 007514L 00010111 10010010
1036. 007515L 00010111 10010010
1037. 007516L 00010111 10010010
1038. 007517L 00010111 10010010
1039. 007520L 00010111 10010010
1040. 007521L 00010111 10010010
1041. 007522L 01110011 11110111
      007523L 01101111 11110111
      007524L 01010001 00000000
1042.
1043. 007525L 01110001 10110101
1044. 007526L 00010111 10010010
1045. 007527L 01101111 11110101
1046. 007530L 01110001 11111001
      007531L 01010101 11100000
1047. 007532L 00010111 10100010
1048. 007533L 00010111 10100010
1049. 007534L 00010111 10100010
1050. 007535L 01010100 01101101
      007536L 01101111 11110000
1051. 007537L 00110001 00010110
1052. 007540L 00010111 10110010
    
```

```

*
. RSTAT USAGE
.   0,1 --- 0
.   2 --- SYNWAIT
.   3,4 --- 0
.   5,6,7 --- BCOUNT (BITS PER CHARACTER)
.
. RBUF STAT
.   0 --- RDY (PRECEDING BYTE IN RECEIVED DATA)
.   1,2,3,4,5,6,7 --- 0
.
GSR   TSTIR   ,SYNWAIT,RSTAT,,TW   WAITING FOR SYN=SYN?
      BRA     GSR1,TZ              THEN DON'T BOTHER WITH EXTENDED DATA
      LDTR    RDATA2,CC            ELSE GET LAST BYTE (0->LINK)
      SHIFT   SR                   SHIFT RIGHT WITH ZERO FILL
      LDRT    RDATA2              AND PUT IT BACK
      DOTIR   ,ND,WLNGTH,COMMODE  GET WLNGTH IN 3 MSB
      SHIFT   SL                   AND MOVE IT...
      SHIFT   SL                   ...TO 3...
      SHIFT   SL                   ...LSB
      DORI    LINK,SB,GSRTAP1+1,,CC SET UP BASED JUMP IN LINK
      DOTRR   ,SB,RDATA,RDATA     RDATA LSB->LINK 0->T
      SHIFT   SR                   RDATA LSB->MSB OF ZERO BYTE
      BPGX    $
      BRR     LINK                 DO 0-7 SR'S
      SHIFT   SR                   (1 BIT WORDS)
      SHIFT   SR                   (2 " " )
      SHIFT   SR                   (3 " " )
      SHIFT   SR                   (4 " " )
      SHIFT   SR                   (5 " " )
      SHIFT   SR                   (6 " " )
      SHIFT   SR                   (7 " " )
GSRTAP1 DORR   RDATA2,OR,RDATA2   SHIFT RDATA LSB INTO RDATA2 MSB
GSR1   LDTI    0                   GET READY TO ZERO LINK
      LDTR    RDATA,CC            GET CURRENT BYTE (0->LINK)
      SHIFT   SR                   SHIFT RIGHT WITH ZERO FILL
      LDRT    RDATA              AND PUT IT BACK
      DOTIR   ,ND,WLNGTH,COMMODE  GET WLNGTH IN 3 MSB...
      SHIFT   SL                   ...AND MOVE...
      SHIFT   SL                   ...TO 3...
      SHIFT   SL                   ...LSB
      DORI    LINK,SB,GSRTAP+1    SET UP BASED JUMP IN LINK
      LDTP    SDLCIN
      CCLR
      NEW BIT => LINK
    
```

1053,	007541L	01010001	00000000	LDTI	0	
1054,	007542L	00010111	10010010	SHIFT	SR	NEW BIT=>MSB OF ZERO BYTE
1055,	007543L	01011001	11110000	BPGX	\$	
1056,	007544L	11101111	00000000	BRR	LINK	DO 0-7 SR'S
1057,	007545L	00010111	10010010	SHIFT	SR	(1 BIT WORDS)
1058,	007546L	00010111	10010010	SHIFT	SR	(2 BIT WORDS)
1059,	007547L	00010111	10010010	SHIFT	SR	(3 BIT WORDS)
1060,	007550L	00010111	10010010	SHIFT	SR	(4 BIT WORDS)
1061,	007551L	00010111	10010010	SHIFT	SR	(5 BIT WORDS)
1062,	007552L	00010111	10010010	SHIFT	SR	(6 BIT WORDS)
1063,	007553L	00010111	10010010	SHIFT	SR	(7 BIT WORDS)
1064,	007554L	01110011	11110101	GSRTAP	DOOR	RDATA,OR,RDATA SHIFT NEW BIT INTO RDATA MSB
	007555L	01101111	11110101			
1065,	007556L	01110001	10110011	DOORIR	RSTAT,AC,BCOUNT,RSTAT,CC	BUMP THE BIT COUNT
	007557L	01010010	00100000			
	007560L	01101111	11110011			
1066,	007561L	01000101	00000001	TSTIT	,CIP	ALWAYS 0 IF GENSYNC, 1 IF ASYNC
1067,	007562L	11010010	01101110	BRA	ARBCCHK,FZ	THIS WILL GO TO ASYNC CODE
1068,	007563L	01000101	00000100	TSTIT	,SYNWAIT	
1069,	007564L	11010010	01111111	BRA	GSYNLOP,FZ	GO LOOK FOR SYN=SYN
1070,	007565L	11010000	00000011	BRA	CRETURN,FC	RETURN IF NOT DONE WITH BYTE
1071,						
1072,	007566L	01110001	11111001	GSR2	DOTIR	,ND,WLNGTH,COMMODE GET THE WORD LENGTH
	007567L	01010101	11100000			
1073,	007570L	01101111	11110011	LDRT	RSTAT	AND PUT IT IN RSTAT (RESETS SYNWAIT)
1074,	007571L	01110001	11110101	TSTRR	XR,SYNCODE,RDATA	DID WE GET A SYNCODE?
	007572L	01110000	11001110			
1075,	007573L	11000010	01000101	BRA	RSETRDY,FZ	GO STUFF BUFF IF NOT
1076,						
1077,	007574L	01110001	11111001	TSTIR	,SYNDEL,COMMODE	ARE WE STRIPING?
	007575L	01000101	00001000			
1078,	007576L	11010010	00000011	BRA	CRETURN,FZ	YES, JUST RETURN
1079,	007577L	11001111	01000101	BRA	RSETRDY	NO, STUFF THE BUFF
1080,						
1081,	007600L	01110001	11111110	GSYNLOP	TSTRR	XR,RDATA2,SYNCODE WAS LAST BYTE A SYN?
	007601L	01110000	11000111			
1082,	007602L	11010010	00000011	BRA	CRETURN,FZ	NO, KEEP WAITING
1083,	007603L	01110000	11000101	TSTRR	XR,RDATA	IS CURRENT BYTE A SYN?
1084,	007604L	11010010	00000011	BRA	CRETURN,FZ	NO, KEEP WAITING
1085,	007605L	11011111	10001001	BRA	GSR2	YES!
1086,						

```

1089,
1090,
1091,
1092,
1093,
1094,
1095,
1096,
1097,
1098,
1099,
1100,
1101,
1102,
1103, 007606L 01110001 10110110
1104, 007607L 11010011 01110100
1105, 007610L 01010100 00000001
      007611L 01101111 11110110
1106, 007612L 11011111 00000011
1107,
1108, 007613L 01110001 11110011
      007614L 01000101 00000001
1109, 007615L 11010011 01101001
1110, 007616L 01000101 00001000
1111, 007617L 11010010 01011101
1112, 007620L 11011111 10101011
1113,
1114,
1115,
1116, 007621L 11010000 01101011
1117, 007622L 01010001 00001001
      007623L 01101111 11110011
1118, 007624L 01110001 10111110
1119, 007625L 11011111 01110111
1120,
1121, 007626L 01000101 00000010
1122, 007627L 11010011 01010110
1123, 007630L 01110001 10110111
      007631L 01010100 00000001
      007632L 01101111 11110111
1124, 007633L 11010011 01001011
1125, 007634L 00110001 00010110
1126, 007635L 01000101 00000001
1127, 007636L 11010011 00000011
1128, 007637L 01010001 00000000
      007640L 01101111 11110011
1129, 007641L 11011111 00000011
1130,
1131, 007642L 01010001 00000000
      007643L 01101111 11110011
1132, 007644L 00110001 00010110
1133, 007645L 01000101 00000001
    
```

```

*
. RSTAT USAGE
.      0 --- CIP (CHARACTER IN PROGRESS)
.      1 --- SIP (START ELEMENT IN PROGRESS)
.      2 --- 0
.      3 --- CHRCOM (CHARACTER COMPLETE)
.      4 --- 0
.      5,6,7 --- BCOUNT (BITS PER CHARACTER)
.
. RBUF STATUS
.      0 --- RDY (PRECEDING CHARACTER IS RECEIVED DATA)
.      1,2,3,4,5,6 --- 0
.      7 --- FRAMING ERROR
.
ASR      LDTR      RICOUNT,CC      GET INTRP CNT
ACHKIC   BRA      ACHKCIP,TZ      CHECK FOR CHARACTER IN PROGRESS IF IC=0
DECRIE   DORI     RICOUNT,SB,1    DECREMENT THE ICOUNT
.
      BRA      CRETURN
.
ACHKCIP  TSTIR    ,CIP,RSTAT      CHARACTER IN PROGRESS?
.
      BRA      ACHKSR,TZ          CHECK FOR START ELEMENT IF NOT
      TSTIT    ,CHRCOM           CHARACTER COMPLETE?
      BRA      ACHKSTP,FZ        CHECK FOR STOP ELEMENT IF SD
      BRA      GSR1              GO DO COMMON STUFF IF NOT
.
. GSR WILL RETURN HERE AFTER BUMPING THE BIT COUNT
.
ARBCCCHK BRA      AREXIT,FC        JUMP IF CHARACTER NOT COMPLETE
      LDRI     RSTAT,CHRCOM+CIP    SET CHRCOM
.
AREXIT   LDTR     AISTOR,CC        INIT NUMBER OF INTERRUPTS PER BIT
      BRA      DECRIE
.
ACHKSRT  TSTIT    ,SIP            START ELEMENT IN PROGRESS?
      BRA      ACHKSP,TZ          CHECK FOR SPACE IF NOT
      DORIR    SPCOUNT,SB,1,SPCOUNT,CC DECREMENT THE SPACE COUNT
.
      BRA      GOTSRT,TZ          GOT A START IF SPACE COUNT=0
      LDTP     SDLCIN            GET THE NEW BIT
      TSTIT    ,1                LOOK AT LSB
      BRA      CRETURN,TZ        RETURN IF SPACE
      LDRI     RSTAT,0           CLEAR START IN PROGRESS IF NOT
.
      BRA      CRETURN
.
ACHKSTP  LDRI     RSTAT,0
.
      LDTP     SDLCIN            GET THE NEW BIT
      TSTIT    ,1                LOOK AT LSB
    
```

1134.	007646L	11000010	01000101	BRA	RSETROY,FZ	ALL IS COOL IF MARK
1135.	007647L	01010001	10000001	LDTI	0201	RDY & FRAMING ERROR
1136.	007650L	11001111	01000100	BRA	RPUTRBS	
1137.						
1138.	007651L	00110001	00010110	ACHKSP	LDTP	SDLCIN
1139.	007652L	01000101	00000001		TSTIT	,1
1140.	007653L	11010010	00000011		BRA	CRETURN,FZ
1141.	007654L	01010001	00000010		LDRI	RSTAT,SIP
	007655L	01101111	11110011			SET START IN PROGRESS IF SPACE
1142.	007656L	01110001	10111110		DOTIR	,AC,1,AISTOR,CC
	007657L	01010010	00000001			INIT SPCOUNT FOR 1/2 BIT
1143.	007660L	00010111	10010010		SHIFT	SR
1144.	007661L	01010101	01111111		DORI	SPCOUNT,ND,0177
	007662L	01101111	11110111			" (FIX UP LINK SCREW UP)
1145.	007663L	11011111	00000011		BRA	CRETURN
1146.						
1147.	007664L	01110001	11111001	GOTSRT	DOTIR	,ND,WLNGTH,COMMODE
	007665L	01010101	11100000			GET WORD LENGTH
1148.	007666L	01010011	00000001		DORI	RSTAT,OR,CIP
	007667L	01101111	11110011			SET CHARACTER IN PROGRESS
1149.	007670L	01110001	10111110		DORIR	RICOUNT,SB,1,AISTOR,CC
	007671L	01010100	00000001			INIT INTERUPTS PER BIT
	007672L	01101111	11110110			
1150.	007673L	11011111	01111000		BRA	ACHKIC
1151.						CHECK ICOUNT FOR 0


```

1154.
1155. 007674L
1156.
1157.
1158.
1159.
1160.
1161.
1162.
1163.
1164.
1165.
1166.
1167.
1168.
1169.
1170.
1171.
1172.
1173.
1174.
1175.
1176.
1177.
1178.
1179.
1180.
1181.
1182.
1183.
1184.
1185.
1186.
1187.
1188.
1189. 007674L 00110001 11011100
      007675L 01000101 00000100
1190. 007676L 11010010 00000001
1191. 007677L 01010001 11011111
      007700L 00110111 11100000
1192. 007701L 01110001 11111010
      007702L 01010011 00000001
      007703L 00110111 11000000
1193. 007704L 00110111 01000111
1194. 007705L 11001110 11111111
1195. 007706L 11010100 00111001
      007707L 11010111 00000101
1196. 007710L 00110001 00110110
      007711L 00110111 00000110
1197. 007712L 11010011 00001100
1198.
1199. 007713L 01010001 00000000
    
```

```

*
SIIN:
.   (   163) SIIN           MOVE ONE BYTE FROM RCV BUFFER TO (A) *
.   (062 163) SIMIN        MOVE C BYTES FROM RCV BUFFER TO (HL) **
.                               * 7,8/5,75
.                               ** 4,7/7,75/5,05 +7,4N FOR NRML/MT/EOB
.                               SETS TZ IF BUFFER EMPTY
.                               TS IF EOB (FRAMING ERROR IN ASYNC)
.                               TP IF EOB AND GOOD CRC
.
.   IF USER THEN IVIOL
.   ELSE MAR <= RBUF,URPNTR+1
.   LUF <= (MAR)
.   IF TZ THEN SIIEXIT
.   ELSE (MAR) <= 0
.   URPNTR <=URPNTR+2
.   MAR <= MAR-1
.   IF (062 163) THEN SIMIN
.   ELSE URA <= (MAR)
.   END
.   END
.
SIMIN MRD <= (MAR)
      MAR <= HL
      HL <= HL+1
      (HL) <= MDR
      URC <= URC-1
      IF TZ THEN SIIEXIT
      ELSE IF TS IN UCFLG THEN SIIEXIT
      ELSE IF SRVREQ THEN SRVRPT
      ELSE SII
.
SIIEXIT: PSW <= PSW.AND.-1-SWRPT
        END
.
.   TSTIP ,SWUSER,PSWI      USER MODE?
.
.   BRA CIVIOL,FZ           USERS CAN'T DO THIS!
SII LDPI MAROH,RBUF>8      CHECK RBUF FOR DATA
.
.   DOPIR MAROL,OR,1,URPNTR "
.
.   STB SMR                 "
.   NOOP                    "
.   MWAIT ,CMEMPF           " (800)
.
.   LDPP LUF,MDR            SET FLAGS FROM RBUF STAT
.   BRA SIIEXIT,TZ         DO NO MORE IF BUFF MT
.   LDPI MDW,0             ELSE MARK LOCATION MT
    
```

1200.	007714L 00110111 00100001			
	007715L 01110001 10111010	DORIR	URPNTR,AC,2,URPNTR,CC	INC USER BUFF POINTER BY 2
	007716L 01010010 00000010			
	007717L 01101111 11111010			
1201.	007720L 11010100 00101111	MWAIT	,CMEMPF	(600)
	007721L 11010111 00000101			
1202.	007722L 00110111 00001101	STB	DMAR	GET DATA FROM BUFFER
1203.	007723L 00110111 01000111	STB	SMR	"
1204.	007724L 11001110 11111111	NOOP		
1205.	007725L 11010100 00101010	MWAIT	,CMEMPF	" (600)
	007726L 11010111 00000101			
1206.	007727L 11011000 00100011	BRA	SIMIN,F0,IZ	
1207.				
1208.	007730L 00110001 00110110	SI1I	LDPP	URO+URA,MDR AND PUT IT IN A
	007731L 00110111 10000000			
1209.	>007732L 01011001 11111111	BRAX	FETCH	AND THAT'S IT!
	>007733L 11001111 11111111			
1210.				
1211.	007734L 00110001 11000110	SIMIN	DLDX	HL2MR AND POINT TO (HL)
	007735L 00110001 11100101			
1212.	007736L 00110001 10010000	DADDP	URO+UR,MARI	INC HL
	007737L 00010110 01110010			
	007740L 00110111 10000110			
	007741L 00110001 10110000			
	007742L 00110110 10000101			
1213.	007743L 00110001 00110110	LDPP	MDW,MDR	MOVE DATA FROM BUFFER TO (HL)
	007744L 00110111 00100001			
1214.	007745L 00110001 11010010	DOPIP	URO+URC,SB,1,URI+URC	DEC C
	007746L 01010100 00000001			
	007747L 00110111 10000010			
1215.	007750L 11010011 00001100	BRA	SIIEXIT,TZ	QUIT IF IT'S NOW 0
1216.				
1217.	007751L 00110001 00110101	TSTIP	,0100,UCFLG	CHECK FLAGS FOR EOB
	007752L 01000101 01000000			
1218.	007753L 11010010 00001100	BRA	SIIEXIT,FZ	QUIT IF EOB
1219.				
1220.	007754L 00110001 00110000	LDTP	SRVREQ	CHECK FOR SERVICE REQUEST
1221.	007755L 11010010 00001110	BRA	SIISRV,FZ	HANDLE IT IF IT'S THERE
1222.				
1223.	007756L 11010100 00010001	MWAIT	,CMEMPF	
	007757L 11010111 00000101			
1224.	007760L 11011111 01000000	BRA	SII	AND DO IT AGAIN
1225.				
1226.	>007761L 01011001 11111111	SIISRV	BRAX	SRVRPT
	>007762L 11001111 11111111			
1227.				
1228.	007763L 00110001 11011100	SIIEXIT	DOPIP	PSW0,ND,-1-SWRPT,PSWI CLEAR THE REPEAT FLAG
	007764L 01010101 11011111			
	007765L 00110111 10001100			
1229.	>007766L 01011001 11111111	BRAX	FETCHW	
	>007767L 11001111 11111111			
1230.				

1233.
1234.
1235.
1236.
1237. 007770L 01010001 00000000
007771L 01101111 11111001
1238.
1239. >007772L 01011001 11111111
>007773L 11001111 11111111
1240.
1241. >007774L 01011001 11111111
>007775L 11001111 11111111
1242.
1243. >007776L 01011001 11111111
>007777L 11001111 11111111
1244.
1245. 001000
1246. 007000
1247. 007000
1248.

*
PAD -S-010,AND,0377
.CMEMPK
LDRI COMMODE,0 TURN COMM OFF IF MEMPF
.CMEMPF
BRAX MEMPFS
.CRETURN
BRAX SRVNXT
.CIVIOI
BRAX IVIOLS
CDORLEN EQU S=CDORP
USE CDORL
SKIP CDORLEN
END

