TITLE Infocom INTERLOGIC interpreter disassembly, 5/27/84

PAGE

Infocom INTERLOGIC interpreter disassembly Apple II/6502 version, release 3 As used in interactive fiction games

The INTERLOGIC interpreter is copyrighted by Infocom, Inc.

This disassembly and the comments therof are copyright (C) 1984 by
Eric L. Smith
230 South 500 West Suite 133
Salt Lake City, Utah 84101
(801) 582-3371

* This disassembly represents well over 300 hours of intense study. It

* is intended for private, noncommercial use only. Any comments or

* questions about it should be addressed to the above address. There is

* no warranty, express or implied, as to the accuracy of this disassembly

* or its fitness for any particular purpose. I assume no liability for

* any damages, actual or alleged, direct or indirect, resulting from the

* use of, or inability to use this disassembly.

0.0							
28							•
29				.6502			
30				.SALL			
31				SECOND			
32				. SECOND			
33	0000		VERSN	EQU	0		O is old version, 1 is new
34	0000		RNGDBG	EQU	0		RNG debug
35	0000			EQU	1		40 column lower case
36	0001		LC40	EQU	•	•	40 COTUMN TOWER CASE
37			. dofin	e memory	116300		
38			; dei in	e memory	usage		
39	0100		LDORG	EQU	\$0100		where to load
40	0100		LDURG	LQU	\$0.100	,	where to road
41	0075		ZPORG	EQU	\$7F		origin of zero page usage
42	007F		BUFFER	EQU	\$0200		I/O buffer
43	0200			EQU	\$6200 \$E0		maximum size of stack in words
44	00E0		STCKMX STCKLC	EQU	\$03E8		base address of stack (works down)
45	03E8		STKLIM	EQU	STCKLC-2*STCKMX		lower limit of stack
46	0228		DIVITIM	EQU	STUREU Z+STURMA	,	TOWER TRIBLE OF STACK
47	0770		PRTWDT	EQU	\$0779		printer carriage width
48	0779		PRIWDI	EQU	\$0779	,	printer carriage width
49				IFF	VERSN		
50	0000		MATNOD				origin of main program
51	0800		MAINOR	EQU	\$0800		origin of wirtual memory tables
52	2200		VMTORG	EQU	MAINOR+\$1A00		
53 .	2400		RWTSOR	EQU	VMTORG+\$0200		origin of RWTS routines
54	2000		FIRFLC	EQU	RWTSOR+\$0800		first location available
55	BFFF		LSTFLC	EQU	\$C000-1	;	last potential location available
56				ENDIF			
57				=0			
58	2200		VMT1LC	EQU	VMTORG+\$0000	;	virtual memory page tables
59	2280		VMT2LC	EQU	VMTORG+\$0080		
60	2300		VMT3LC	EQU	VMTORG+\$0100		
61	2380		VMT4LC	EQU	∨MTORG+\$0180		
62					D. (T. 0.0.) - 0.0.0.0		
63	2900		RWTS	EQU	RWTSOR+\$0500	;	entry point of RWTS routines
64							i
65							
66			; Contr	ol chara	cters		
67					# O'C		
68	000D		CRCHAR	EQU	\$0D		carriage return
69	000A		LFCHAR	EQU	\$0A		line feed
70	0009		TBCHAR	EQU	\$09	•	horizontal tab
71	000C		FFCHAR	EQU	\$0C	;	form feed
72							
73							
74			; Apple	monitor	ROM's zero page loca	ations	
75							
76	0020		WNDLFT	EQU	\$20	;	screen window parameters
77	0021		WNDWDT	EQU	\$21		
78	0022		WNDTOP	EQU	\$22		
79	0023	;-	WNDBOT	EQU	\$ 23		
80							
81	0024		CURSRH	EQU	\$24	; ,	cursor position
82	0025		CURSRV	EQU	\$ 25		

Infocom	ARLOGIC int	erpreter disassemb	ly, 5/2	27/84 MACRO-80 3.	18-Sep-81 PAGE 1-2
83 84	0032	INVFLG	EQU	\$ 32	; inverse video output flag
85 86	0033	PROMPT	EQU	\$33	; line input prompt
87 88	0036	CSWL	EQU	\$36	; character output vector
- 89 90 91	004E	RNDLOC	EQU	\$4 E	; location randomized by keyboard input
92 93		; Apple	monitor	routines	
94 95	FC22	VTAB	EQU	\$FC22	; adjust video pointer after cursor move
96	FC58	HOME	EQU	\$FC58	; clear screen window
97	FC9C	CLREOL	EQU	\$FC9C	; clear to end of line
98	FD0C	RDKEY	EQU	\$FDOC	; get a key from keyboard
99	FD6F	GETLN1	EQU	\$FD6F	; get a line from keyboard
100	FDED	COUT	EQU	\$FDED	; output a char to current device
101	FDF0	COUT 1	EQU	\$FDF0	; output a char to screen
102					
103			IFT	RNGDBG	
104			ENDIF		
105					
106			PAGE		

107						
108			; defir	ne our o	wn zero page	e usage
109			,			
110	0000′	D		DSECT		
111		. D		ORG	ZPORG	
112		, D				
113	007F	D	SECPTK	DS	1	; number of sectors per track on disk
114		D				, , , , , , , , , , , , , , , , , , , ,
115	0080	D	OPCODE	DS	1	; opcode of current instruction
116	0081	D	ARGCNT	DS	1	; instruction arguments
117		D				•
118	0082	D	ARG1	DS	2	
119	0084	D	ARG2	DS	2	
120	0086	D	ARG3	DS	2	
121	0088	D	ARG4	DS	2	•
122		D				
123	008A	D	PRGIDX	DS	1	; PC low byte, index into page
124	008B	D	PRGLPG	DS	2	; PC logical page number
125	008D	D		DS	2	; PC mem loc of logical page
126	008F	D	PRGUPD	DS	1	; PC new page flag
127	0090	D	PRGPPG	DS	1	; PC physical page number
128		, D				
129	0091	D		DS	2	; AUX logical page number
130	0093	D		DS	1	; AUX low byte, index into page
131	0094	Ď		DS	2	; AUX mem loc of logical page
132	0096	D		DS	1	; AUX new page flag
133	0097	D	AUXPPG	DS	1	; AUX physical page number
134		D				
135	0098	D		DS	2	; pointer to global variables
136	009A	D	LOCVAR	DS	30	; storage of local variables
137		D				
138	0088	D	SWPMEM	DS	2	; address of first swappable page
139	OOBA	D		DS	2	; address of first frozen page
140	OOBC	D		DS	1	; number of frozen pages
141	00BD	D	SWPPGS	DS	1	; number of swappable phys. pages
142		D				
143	OOBE	D		DS	1	; phys. pg. # of most recently used page
144	00BF	D	LRUPAG	DS	1	; phys. pg. # of least recently used page
145		D				
146	0000	D	VMTAB1	DS	2	; virtual memory table pointers
147	00C2	D	VMTAB2	DS	2	
148	00C4	D	VMTAB3	DS	2	
149	0006	D	VMTAB4	DS	2	
150		D				
151	0008	D	STKCNT	DS	1	; # items on stack
152	00C9	, D	STKPNT	DS	2	; stack pointer
153	OOCB	D	STKPSV	DS	2	; stack ptr save during call
154	OOCD	D	STKCSV	DS	1	; stack cnt save during call
155		D				
156	00CE	D	TMPMOD	DS	1	; string output temporary char. mode
157	00CF	. D	PRMMOD	DS	1	; string output perm. char. mode
158	00D0	D	PNYBCN	DS	1	; string output nybble counter
159	00D1	D	PNYBBF	DS	2	; string output nybble buffer
160	00=5	D	.		_	
161	00D3	D	INWORD	DŞ	6	; word to be packed

160		D				
162 163	0009	Ď	LD9	DS	1	
	0005	D	LDJ	D3		
164	OODA	D	PKWORD	DS	4	; packed word
165	UUDA	Q Q	FRWORD	<i>D</i> 3	-	, pasked not a
166	0005	D	LDE	DS	1	
167	00DE	D	LDF	DS	1	
168	00DF	D	LEO	DS DS	1	
169	00E0			DS DS	1	
170	00E1	D	LE1	מס	ı	
171		D	COMOUT	DS	2	
172	00E2	D D	SBWDPT	סט	2	
173			4.00	0.0	2	
174	00E4	D	ACB	DS	2	
175	00E6	D	ACC	DS	2 2	
176	00E8	D	ACD	DS	2	
177		D		5.0		annative and count for mult/div
178	OOEA	D	MDFLAG	DS	1	; negative arg count for mult/div
179		D			_	aban aut buffen neinten
180	00EB	D	CHRPTR	DS	1	; char out buffer pointer
181	OOEC	D	CHRPT2	DS	1	; char out buffer pointer 2
182	00ED	D	LINCHT	DS	1	; output line counter
183	OOEE	D	PRCSWL	DS	2	; CSWL vector contents for printer
184		D				
185	00F0	D		DS	3	
186		D				
187	00F3	D	STLTYP	DS	1	; status line type (time vs. score)
188		D				
189	00F4			DEND		
190						
191				PAGE		

MACRO-80 3

PAGE

Infocom IN. ERLOGIC interpreter disassembly, 5/27/84

```
192
                               ; define offsets into game header
193
194
        0000'
                            D
                                        DSECT
195
                                        ORG
                                                0
196
                            D
                            D
197
                                                                         ; required interpreter release (should be 3)
                               HDRIRL
                                       DS
        0000
                            D
198
                            D
                               HDRTYP
                                                                         ; game type flags (score/time, etc.)
        0001
199
                                                                         ; game release
                            D
                               HDRREL
                                       DS
200
        0002
                                                                         ; log. addr. of end of frozen memory
201
        0004
                            D
                               HDRFRZ
                                       DS
                                                2
                                                2
                                                                          ; log. addr. of start of code
        0006
                            D
                               HDRSTR
                                       DS
202
                               HDRVCB
                                                2
                                                                          ; log. addr. of vocab. table
        8000
                            D
                                       DS
203
                               HDRTHG
                                                2
                                                                          ; log. addr. of thing table
                            D
204
        A000
                                                                          ; log. addr. of global variables
205
        000C
                            D
                               HDRGBV
                                                2
206
        000E
                            D
                               HDRIMP
                                       DS
                                                2
                                                                          ; log. addr. of end of impure storage
                            D
                               HDRFLG
                                       DS
                                                2
                                                                          ; flags (script, etc.)
207
        0010
                            D
                               HDRSER
                                       DS
                                                6
                                                                          ; game serial no. (release date)
208
        0012
                               HDRSBW
                                                2
                                                                          ; log. addr. of subword table
209
        0018
                            D
                                      DS
                                                                          ; half of last log. addr. to checksum
                            D
                               HDRCKA DS
                                                2
210
        001A
                                                2
                                                                          ; expected checksum value
                            D
                               HDRCKV DS
        001C
211
212
                            D
213
                            D
                            D
                               ; define thing table offsets
214
215
                            D
                                                0
                            Đ
                                        ORG
216
                            D
217
                            D
                               THGATT
                                        DS
                                                                          ; attribute bits
218
        0000
                               THGPAR
                                                                          ; parent thing number
                                        DS
219
        0004
                            D
                               THGSIB
                                        DS
                                                                         ; sibling thing number
220
        0005
                            D
                                                                         ; child thing number
221
        0006
                            D
                               THGCHD
                                        DS
222
        0007
                            D
                               THGPRP
                                        DS
                                                2
                                                                          ; property list pointer
223
                            D
        0009
                                        DEND
224
225
                             С
                                        INCLUDE ZIPMAC
226
                                        PAGE
227
```

```
C
C
228
229
230
                               С
                                 ; Some useful macros
231
                               С
                               C+DSTZ
                                          MACRO
                                                   ADDR
232
                                          LDA
                                                    #$00
                               C+
233
                                                    ADDR
234
                               C+
                                          STA
                               C+
                                                    ADDR+1
                                          STA
235
                               С
                                          ENDM
236
                               С
237
                               C+DASL
238
                                          MACRO
                                                    ADR1,ADR2
                               C+
                                          IFNB
                                                    <ADR2>
239
                               C+
                                                    ADR 1
                                          LDA
240
                               C+
                                          ASL
                                                    Α
241
                                                    ADR2
242
                               C+
                                          STA
243
                               C+
                                          LDA
                                                    ADR 1+1
                               C+
                                          ROL
244
                                                    ADR2+1
245
                               C+
                                          STA
                               C+
                                          ELSE
246
                                                    ADR1
247
                               C+
                                          ASL
248
                               C+
                                          ROL
                                                    ADR 1+1
249
                               C+
                                          ENDIF
                               С
                                          ENDM
250
251
                               С
                               C+DLSR
                                          MACRO
                                                    ADR1,ADR2
252
                                          IFNB
                                                    <ADR2>
253
                               C+
                                                    ADR 1+1
                               C+
                                          LDA
254
255
                               C+
                                          LSR
                               C+
                                          STA
                                                    ADR2+1
256
                                                    ADR1
257
                               C+
                                          LDA
258
                               C+
                                          ROR
                                                    Α
259
                               C+
                                          STA
                                                    ADR2
                               C+
                                          ELSE
260
261
                               C+
                                          LSR
                                                    ADR 1+1
262
                               C+
                                          ROR
                                                    ADR 1
263
                               C+
                                          ENDIF
                               C
                                          ENDM
264
265
                               С
                               C+DROR
                                          MACRO
                                                    ADR1, ADR2
266
                                                    <ADR2>
267
                               C+
                                          IFNB
                               C+
                                          LDA
                                                    ADR 1+1
268
                               C+
269
                                          ROR
270
                               C+
                                          STA
                                                    ADR2+1
271
                               C+
                                          LDA
                                                    ADR 1
                               C+
                                          ROR
272
                                                    Α
                                                    ADR2
                               C+
                                          STA
273
274
                               C+
                                          ELSE
275
                               C+
                                          ROR
                                                    ADR 1 + 1
                               C+
                                          ROR
                                                    ADR1
276
277
                               C+
                                          ENDIF
                               С
                                          ENDM
278
279
                               6 .
280
                               C+DROL
                                          MACRO
                                                    ADR1,ADR2
                               C+
                                          IFNB
                                                    <ADR2>
281
                               C+
                                                    ADR 1
282
                                          LDA
```

```
C+
                                          ROL
                                                    Α
283
                               C+
                                          STA
                                                    ADR2
284
                               C+
                                          LDA
                                                    ADR 1+1
285
                                          ROL
                               C+
                                                    Α
286
                                                    ADR2+1
                                          STA
287
                               C+
                               C+
                                          ELSE
288
                                                    ADR1
                               C+
                                          ROL
289
                               C+
                                          ROL
                                                    ADR 1+1
290
                               C+
                                          ENDIF
291
                                          ENDM
                               С
292
                               С
293
                                                    ADR1, ADR2, ADR3
                                          MACRO
                               C+DOR
294
                                                    ADR 1+1
                                C+
                                          LDA
295
                                C+
                                                    ADR2+1
296
                                          ORA
                                C+
                                           STA
                                                    ADR3+1
297
                                C+
                                          LDA
                                                    ADR 1
298
                                C+
                                           ORA
                                                    ADR2
299
                                C+
                                                    ADR3
                                           STA
300
                                С
                                           ENDM
301
                                С
302
                                           MACRO
                                                    ADR1, ADR2, ADR3
                                C+DAND
303
                                C+
                                           LDA
                                                    ADR 1+1
304
                                C+
                                           AND
                                                    ADR2+1
305
                                                    ADR3+1
                                C+
                                           STA
306
                                                    ADR 1
                                C+
                                           LDA
307
                                                    ADR2
308
                                C+
                                           AND
                                                    ADR3
309
                                C+
                                           STA
                                С
                                           ENDM
310
                                С
311
                                          MACRO
                                                    ADR1,ADR2
312
                                C+D1COMP
                                C+
                                           LDA
                                                    ADR 1
.313
                                C+
                                           EOR
                                                    #$FF
314
                                C+
                                           STA
                                                    ADR2
315
                                C+
                                                    ADR 1+1
316
                                           LDA
                                C+
                                           EOR
                                                    #$FF
317
                               Č+
                                                    ADR2+1
                                           STA
318
                                С
                                           ENDM
319
320
                                С
                               C+DADC
                                           MACRO
                                                    ADR1,ADR2,ADR3
321
                                                    ADR 1
                                           LDA
322
                                C+
                                           ADC
                                                    ADR2
                                C+
323
                                                    ADR, <ADR3>
324
                                C+
                                           IRP
                                                    ADR
325
                                C+
                                           STA
                                C+
                                           ENDM
326
                                C+
                                           LDA
                                                    ADR1+1
327
328
                                C+
                                           ADC
                                                    ADR2+1
                                C+
                                           IRP
                                                    ADR, <ADR3>
329
                                C+
                                                    ADR+1
                                           STA
330
                               C+
C
                                           ENDM
331
332
                                           ENDM
                                С
333
                                                    ADR1, ADR2, ADR3
                                C+DSBC
                                           MACRO
334
                                           LDA
                                                    ADR 1
                                C+
335
                                                    ADR2
                                C+
                                           SBC
336
                                           IRP
                                                    ADR, <ADR3>
337
                                C+
338
                                C+
                                           STA
                                                    ADR
```

220	C+	ENDM	
339	C+	LDA	ADR 1+1
340			
341	C+	SBC	ADR2+1
342	C+	IRP	ADR, <adr3></adr3>
343	C+	STA	ADR+1
344	C+	ENDM	
345	Ċ	ENDM	
	Č		
346		MACDO	ADD1 ADD2 ADD2
347	C+DADD	MACRO	ADR1,ADR2,ADR3
348	C+	CLC	
349	C+	DADC	<adr1>,<adr2>,<adr3></adr3></adr2></adr1>
350	С	ENDM	
351	С		
	C+DSUB	MACRO	ADR1,ADR2,ADR3
352	C+	SEC	
353	_		
354	C+	DSBC	<adr1>,<adr2>,<adr3></adr3></adr2></adr1>
355	С	ENDM	
356	С		
357	C+ADD	MACRO	ADR1,ADR2,ADR3
358	C+	IFNB	<adr1></adr1>
	C+	LDA	ADR 1
359			ADICI
360	. C+	ENDIF	•
361	C+	CLC	
362	C+	ADC	ADR2
363	C+	IFNB	<adr3></adr3>
364	C+	IRP	ADR, <adr3></adr3>
365	C+	STA	ADR
	C+	ENDM	
366			
367	C+	ENDIF	
368	С	ENDM	
369	С		
370	C+SUB	MACRO	ADR1,ADR2,ADR3
371	C+	IFNB	<adr 1=""></adr>
	Č+	LDA	ADR 1
372	C+	ENDIF	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
373			
374	C+	SEC	
375	C+	SBC	ADR2
376	C+	IFNB	<adr3></adr3>
377	C+	IRP	ADR, <adr3></adr3>
378	C+	STA	ADR
	C+	ENDM	
379			
380	C+	ENDIF	
381	С	ENDM	
382	С		
383	C+DADDB1	MACRO	ADDR, BYTE
384	C+	LOCAL	LABEL
	C+	CLC	
385		LDA	ADDR
386	C+		
387	C+	ADC	BYTE
388	C+	STA	ADDR
389	C+	BCC	LABEL
390	C+	INC	ADDR+1
391	C+LABEL:		
		ENDM	
392	C	ENDM	
393	С		
394	C+DSUBB1	MACRO	ADDR, BYTE

395	C+	LOCAL	LABEL
396	C+	SEC	
397	C+	LDA	ADDR
398	C+	SBC	BYTE
399	C+	STA	ADDR
	C+	BCS	LABEL
400	C+	DEC	ADDR+1
401		DEC	ADDRIT
402	C+LABEL:	E	
403	C	ENDM	
404	С		
405	C+DADDB2	MACRO	ADDR, BYTE
406	C+	LOCAL	LABEL
407	C+	IFNB	<byte></byte>
408	C+	ADD	ADDR, BYTE, ADDR
409	C+	ELSE	•
410	C+	ADD	, ADDR , ADDR
411	C+	ENDIF	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
	C+	BCC	LABEL
412			
413	C+	INC	ADDR+1
414	C+LABEL:		
415	С	ENDM	
416	С		
417	C+DSUBB2	MACRO	ADDR, BYTE
418	C+	LOCAL	LABEL
419	C+	IFNB	<byte></byte>
420	C+	SUB	ADDR, BYTE, ADDR
	C+	ELSE	NDON, 5112, NDON
421		SUB	ADDD ADDD
422	C+		, ADDR , ADDR
423	C+	ENDIF	
424	C+	BCS	LABEL
425	C+	DEC	ADDR+1
426	C+LABEL:		
427	С	ENDM	
428	С		
429	C+DINC	MACRO	ADDR
430	C+	LOCAL	LABEL
431	C+	INC	ADDR
	C+	BNE	LABEL
432			
433	C+	INC	ADDR+1
434	C+LABEL:		
435	С	ENDM	
436	С		
437	C+DDEC	MACRO	ADDR
438	C+	DSUBB2	ADDR,<#\$01>,ADDR
439	С	ENDM	
440	Č		
441	C+DDEC2	MACRO	ADDR
			ADDR, <#\$02>, ADDR
442	C+	DSUBB2	AUDR, ~# \$U2>, AUDR
443	C	ENDM	
444	С		
445	C+DMOV	MACRO	ADR1,ADR2
446	C+	LDA	ADR 1
447	C+	IRP	ADR, <adr2></adr2>
448	C+	STA	ADR
449	C+	ENDM	
450	C+	LDA	ADR 1+1
730	.	בטת	ASK 1 · 1

451 452 453 454 455	C+ C+ C+ C	IRP STA ENDM ENDM	ADR, <adr2> ADR+1</adr2>
456 457 458 459	C+DMOVI C+ C+ C+ C+	MACRO LDA IRP STA ENDM	DATA,ADR2 #<(DATA) ADR, <adr2> ADR</adr2>
460 461 462 463 464	C+ C+ C+	LDA IRP STA ENDM	#>(DATA) ADR, <adr2> ADR+1</adr2>
465 466 467 468 469	C C+DMOVI2 C+ C+	ENDM MACRO LDA IRP	DATA,ADR2 #>(DATA) ADR, <adr2></adr2>
470 471 472 473	C+ C+ C+ C+	STA ENDM LDA IRP	ADR+1 #<(DATA) ADR, <adr2></adr2>
474 475 476 477	C+ C+ C	STA ENDM ENDM	ADR
478 479 480 481	C+PUL C+ C+ C+ C+	MACRO IRP PLA STA ENDM	ADR1 ADR, <adr1> ADR</adr1>
482 483 484 485 486	C+ C C+PSH C+	ENDM ENDM MACRO IRP	ADR1 ADR, <adr1></adr1>
487 488 489 490	C+ C+ C+	LDA PHA ENDM ENDM	ADR
491 492 493 494 495	C C+DPUL C+ C+ C	MACRO PUL PUL ENDM	ADR ADR+1 ADR
496 497 498 499 500	C C+DPUL2 C+ C+ C	MACRO PUL PUL ENDM	ADR ADR ADR+1
501 502 503 504 505 506	C C+DPSH C+ C+ C	MACRO PSH PSH ENDM	ADR ADR ADR+1

	0.4001	MACDO	ADD 1 ADD 2
507	C+MOV C+	MACRO LDA	ADR1,ADR2 ADR1
508 509	C+	IRP	ADR, <adr2></adr2>
510	C+	STA	ADR
511	C+	ENDM	
512	С	ENDM	
513	С	•	
514	C+INCA	MACRO	
515	C+	ADD	,<#\$01>
516	C	ENDM	
517	C	MACBO	
518	C+DECA C+	MACRO SUB	.<#\$01>
519 520	C.	ENDM	,
521	č	LINDIN	
522	C+TSTA	MACRO	
523	C+	ORA	#\$ 00
524	С	ENDM	
525	С		
526	C+STR	MACRO	TEXT
527	C+	DB	TEXT
528	C	ENDM	
529	C	MA CDO	ADD
530	C+JEQ	MACRO LOCAL	ADR LABEL
531	C+	BNE	LABEL
532	C+	JMP	ADR
533 534	C+LABEL:	J.WIT	ADIX
535	C C	ENDM	
536	č		
537	C+JNE	MACRO	ADR
538	C+	LOCAL	LABEL
539	C+	BEQ	LABEL
540	C+	JMP	ADR
541	C+LABEL:		
542	C	ENDM	
543	C	44 A CD O	4 D.D.
544	C+JCC	MACRO LOCAL	ADR LABEL
545	C+	BCS	LABEL
546 547	C+	JMP	ADR
548	C+LABEL:	· · · · ·	
549	C	ENDM	
550	Ċ		
551	C+JCS	MACRO	ADR
552	C+	LOCAL	LABEL
553	C+	BCC	LABEL
554	C+	JMP ·	ADR
555	C+LABEL:		
556	C	ENDM	
557	C	*** 000	4.00
	C+JLT	MACRO	ADR
559 .	C+	LOCAL BGE	LABEL LABEL
560	C+	JMP	ADR
561	C+LABEL:	JMF	ADI
562	C. LADEL:		

563	C	ENDM	
564 565	C C+JGE	MACRO	ADR
566	C+	LOCAL	LABEL
567	C+	BLT	LABEL
568	C+	JMP	ADR
569	C+LABEL:	END!	
570 571	C C	ENDM	
572	C+JGT	MACRO	ADR
573	C+	LOCAL	LABEL
574	C+	BLT	LABEL
575	C+	BCC	LABEL
576	C+	JMP	ADR
577	C+LABEL:	ENDM	
578 579	C C	ENDM	
580	C+JPL	MACRO	ADR
581	C+	LOCAL	LABEL
582	C+	BMI	LABEL
583	C+	JMP	ADR ·
584	C+LABEL:		
585	C	ENDM	
586	C	*** 600	400
587	C+JMI C+	MACRO LOCAL	ADR LABEL
588 589	C+	BPL	LABEL
590	C+	JMP	ADR
591	C+LABEL:		
592	С	ENDM	
593	С		
594	C+JSREQ	MACRO	ADR, ADR2
595	C+	LOCAL	LABEL
596 597	C+ C+	BNE JSR	LABEL ADR
598	C+	IFNB	<adr2></adr2>
599	C+	JMP	ADR2
600	C+	ENDIF	
601	C+LABEL:		
602	C	ENDM	
603	C	*** 600	*DD *DD0
604 605	C+JSRNE C+	MACRO LOCAL	ADR,ADR2 LABEL
606	C+	BEQ	LABEL
607	C+	JSR	ADR
608	C+	IFNB	<adr2></adr2>
609	C+	JMP	ADR2
610	C+	ENDIF	
611	C+LABEL:		
612	C	ENDM	
613	C + 18000	MACDO	4DD 4DD3
614 615	C+JSRCC C+	MACRO LOCAL	ADR,ADR2 LABEL
616	C+	BCS	LABEL
617	C+	JSR	ADR
618	C+	IFNB	<adr2></adr2>

619 620	C+ C+	JMP Endif	ADR2
621 622	C+LABEL: C C	ENDM	
623 624 625 626 627 628 629 630	C+JSRCS C+ C+ C+ C+ C+ C+ C+ C+	MACRO LOCAL BCC JSR IFNB JMP ENDIF	ADR, ADR2 LABEL LABEL ADR <adr2> ADR2</adr2>
632 633	C C	ENDM	
634 635 636 637 638 639 640 641 642 643	C+JSRLT C+ C+ C+ C+ C+ C+ C+ C+LABEL: C	MACRO LOCAL BGE JSR IFNB JMP ENDIF	ADR, ADR2 LABEL LABEL ADR <adr2> ADR2</adr2>
644 645	C+JSRGE C+	MACRO LOCAL	ADR,ADR2 LABEL
646 647 648	C+ C+	BLT JSR IFNB	LABEL ADR <adr2></adr2>
649 650 651 652	C+ C+ C+LABEL: C	JMP ENDIF ENDM	ADR2
653 654 655 656 657 658 659 660 661	C C+JSRGT C+	MACRO LOCAL BLT BEQ JSR IFNB JMP ENDIF	ADR, ADR2 LABEL LABEL LABEL ADR <adr2> ADR2</adr2>
663 664	C C	ENDM	
665 666 667 668 669 670 671 672 673	C+JSRPL C+ C+ C+ C+ C+ C+ C+ C+LABEL: C	MACRO LOCAL BMI JSR IFNB JMP ENDIF	ADR, ADR2 LABEL LABEL ADR <adr2> ADR2</adr2>

675 676 677 678 679 680 681 682 683 684	C+JSRMI C+ C+ C+ C+ C+ C+ C+ C+LABEL: C	MACRO LOCAL BPL JSR IFNB JMP ENDIF	ADR,ADR2 LABEL LABEL ADR <adr2> ADR2</adr2>
685 686 687 688 689 690	C+RTSEQ C+ C+ C+ C+ C+LABEL: C	MACRO LOCAL BNE RTS	ADR LABEL LABEL
692 693 694 695 696 697 698	C+RTSNE C+ C+ C+ C+LABEL: C	MACRO LOCAL BEQ RTS	ADR LABEL LABEL
699 700 701 702 703 704 705	C+RTSCC C+ C+ C+ C+LABEL: C	MACRO LOCAL BCS RTS	ADR LABEL LABEL
706 707 708 709 710 711	C+RTSCS C+ C+ C+ C+ C+LABEL: C	MACRO LOCAL BCC RTS	ADR LABEL LABEL
712 713 714 715 716 717 718	C+RTSLT C+ C+ C+ C+ C+LABEL: C	MACRO LOCAL BGE RTS ENDM	ADR LABEL LABEL
719 720 721 722 723 724 725	C+RTSGE C+ C+ C+ C+ C+LABEL: C	MACRO LOCAL BLT RTS	ADR LABEL LABEL
727 728 729 730	C+RTSGT C+ C+ C+	MACRO LOCAL BLT BEQ	ADR LABEL LABEL LABEL

731	C+	RTS	
732 733	C+LABEL: C	ENDM	
734	Č		
735	C+RTSPL	MACRO	ADR
736	C+	LOCAL BMI	LABEL LABEL
737 738	C+	RTS	LADEL
739	C+LABEL:		
740	С	ENDM	
741	С		
742	C+RTSMI	MACRO	ADR LABEL
743 744	C+ C+	LOCAL. BPL	LABEL
745	C+	RTS	LAGEL
746	C+LABEL:		
747	C	ENDM	
748	C		
749	C+DTST	MACRO	ADDR
750 751	C+	LDA ORA	ADDR+1 ADDR
751 752	C.	ENDM	ADDIK
753	Č	,	
754	C+DTSTBE	MACRO	ADR1,ADR2
755	C+	DTST	ADR1
756	C+	BEQ	ADR2
757	C	ENDM	
758 750	C C+DTSTBN	MACRO	ADR1,ADR2
759 760	C+	DTST	ADR1
761	C+	BNE	ADR2
762	Ċ	ENDM	
763	С		
764	C+DTSTJE	MACRO	ADR1,ADR2
765	C+	DTST	ADR1 ADR2
766 767	C+ C	JEQ ENDM	ADR2
767 768	C	LINDIN	
769	C+DTSTJN	MACRO	ADR1,ADR2
770	C+	DTST	ADR 1
771	C+	JNE	ADR2
772	C	ENDM	
773	C	MACRO	ADR1
774 775	C+DTSTRE C+	MACRO DTST	ADR1
776	C+	RTSEQ	
777	C	ENDM	
778	С		
779	C+DTSTRN	MACRO	ADR1
780	C+	DTST	ADR 1
781 782	C+ C	RTSNE ENDM	
782 783	C	FIADIAI	
784	C+DTST2	MACRO	ADDR
785	C+	LDA	ADDR
786	C+	ORA	ADDR+1

787	C	ENDM	
788	C C+DTS2BE	MACRO	ADR1,ADR2
789 790	C+D132BE	DTST2	ADR 1
790 791	C+	BEQ	ADR2
792	Ċ.	ENDM	ADINZ
793	č		
794	C+DTS2BN	MACRO	ADR1,ADR2
795	C+	DTST2	ADR 1
796	C+	BNE	ADR2
797	С	ENDM	
798	С		
799	C+DTS2JE	MACRO	ADR1,ADR2
800	C+	DTST2	ADR 1
801	C+	JEQ	ADR2
802	C	ENDM	
803	C	MACRO	A001 A002
804	C+DTS2JN C+	DTST2	ADR1,ADR2 ADR1
805 806	C+	JNE	ADR 1
807	C.	ENDM	ADICZ
808	č	LITE	
809	C+DTS2RE	MACRO	ADR 1
810	C+	DTST2	ADR1
811	C+	RTSEQ	
812 .	С	ENDM .	
813	С		
814	C+DTS2RN	MACRO	ADR 1
815	C+	DTST2	ADR 1
816	C+	RTSNE	
817	C	ENDM	
818	C	MA CD 0	4 D.D.
819	C+DXBNE C+	MACRO DEX	ADR
820 821	C+	BNE	ADR
822	c.	ENDM	ADIX
823	Č	LITOM	
824	C+DYBNE	MACRO	ADR
825	C+	DEY	
826	C+	BNE	ADR
827	С	ENDM	
828	С		
829	C+DXBEQ	MACRO	ADR
830	C+	DEX	
831	C+	BEQ	ADR
832	С	ENDM	
833	C	MACRO	ADR
834 835	C+DYBEQ C+	DEY	ADK
836	C+	BEQ	ADR
837	Č.	ENDM	
838	∵č •		
839	C+DXBPL	MACRO	ADR
840	C+	DEX	
841	C+	BPL	ADR
842	С	ENDM	

843	С		
844	C+DYBPL	MACRO	ADR
845	C+	DEY	• 0.0
846	C+ C	BPL	ADR
847	C	ENDM	
848	C+DXBMI	MACRO	ADR
849 850	C+	DEX	ADK
851	C+	BMI	ADR
852	Č	ENDM	
853	č		
854	C+DYBMI	MACRO	ADR
855	C+	DEY	
856	C+	BMI	ADR
857	С	ENDM	
858	С		
859	C+IXBNE	MACRO	ADR
860	C+	INX	
861	C+	BNE	ADR
862	C	ENDM	
863	C		
864	C+IYBNE	MACRO	ADR
865	C+	INY	4.D.D.
866	C+	BNE	ADR
867	C	ENDM	
868	C+DECBE	MACRO	ADR1,ADR2
869 870	C+DECBE	DEC	ADR1
871	C+	BEQ	ADR2
872	C.	ENDM	ADILL
873	č	L.110	-
874	C+DECBN	MACRO	ADR1,ADR2
875	C+	DEC	ADR1
876	C+	BNE	ADR2
877	С	ENDM	
878	С		
879	C+DECJE	MACRO	ADR1,ADR2
880	C+	DEC	ADR 1
881	C+	JEQ	ADR2
882	С	ENDM	
883	С		
884	C+DECJN	MACRO	ADR1,ADR2
885	C+	DEC	ADR 1
886	C+	JNE	ADR2
887	C	ENDM	
888	C	*** 600	A D D 1
889	C+DECABE	MACRO	ADR 1
890	C+	DECA	ADD 1
891	C+ C	BEQ ENDM	ADR 1
892 803	C	LNUM	
893 894	C+DECABN	MACRO	ADR 1
895	C+DECABN	DECA	APICI
896	C+	BNE	ADR 1
897	c.	ENDM	
898	C	-110111	
0.70	-		

899	C+DECABP	MACRO	ADR 1
900	C+	DECA	
901	C+	BPL	ADR1
902	С	ENDM	
903	С		
904	C+DECABM	MACRO	ADR 1
905	C+	DECA	
906	C+	BMI	ADR 1
	Č	ENDM	
907	Č	LITOIN	
908	C+TSTABE	MACRO	ADR1
909			ADRI
910	C+	TSTA	ADD 1
911	C+	BEQ	ADR 1
912	C	ENDM	
913	С		
914	C+TSTABN	MACRO	ADR 1
915	C+	TSTA	
916	C+	BNE	ADR 1
917	С	ENDM	
918	С		
919	C+TSTABP	MACRO	ADR 1
920	C+	TSTA	
921	C+	BPL	ADR1
922	Č	ENDM	
923	č		
924	C+TSTABM	MACRO	ADR 1
	C+	TSTA	ADICI
925	C+	BMI	ADR 1
926			ADKI
927	С	ENDM	
928	C		
929	C+TSTAJE	MACRO	ADR 1
930	C+	TSTA	
931	C+	JEQ	ADR 1
932	С	ENDM	
933	С		
934	C+TSTARP	MACRO	
935	C+	TSTA	
936	C+	RTSPL	
937	Ċ	ENDM	
938	č		
939	C+CMPBE	MACRO	ADR1,ADR2
940	C+	CMP	ADR1
	C+	BEQ	ADR2
941	C	ENDM	AURZ
942		CHDM	
943	C	*** 000	1001 1000
944	C+CMPBN	MACRO	ADR1,ADR2
945	C+	CMP	ADR 1
946	C+	BNE	ADR2
947	С	ENDM	
948	С		
949	C+CMPBL	MACRO	ADR1,ADR2
950	C+	CMP	ADR 1
951	C+	BLT	ADR2
952	Ċ	ENDM	
953	č		
	C+CMPBG	MACRO	ADR1,ADR2
954	C.CWPDG	MACKO	ADN I, ADNZ

Infocom	IN.	_RLOGIC	interpreter	disassem	bly, 5/2	7/84 MA
955				C+	СМР	ADR 1
956				C+	BGE	ADR2
957				С	ENDM	
958				С		
959				C+CMPBM	MACRO	ADR1,ADR2
960				C+	CMP	ADR 1
961				C+	BMI	ADR2
962				С	ENDM	
963				С		
964				C+CMPBP	MACRO	ADR1,ADR2
965				C+	CMP	ADR 1
966				C+	BPL	ADR2
967				Č	ENDM	
968				Č		
969				C+CMPJE	MACRO	ADR1,ADR2
970				C+	CMP	ADR 1
971				C+	JEQ	ADR2
972				Č	ENDM	,,_,,_
973				č		
974				C+CMPJL	MACRO	ADR1,ADR2
975				C+	CMP	ADR 1
976				C+	JLT	ADR2
977				Č	ENDM	
978				č		
979				C+CMPJSE	MACRO	ADR1,ADR2
980				C+	CMP	ADR1
981				C+	JSREQ	ADR2
982				C	ENDM	
983				č		
984				C+CMPJSN	MACRO	ADR1,ADR2
985				C+	CMP	ADR1
986				C+	JSRNE	ADR2
987			*	č	ENDM	
988				Č		
989				C+CMPJSG	MACRO	ADR1,ADR2
990				C+	CMP	ADR1
991				C+	JSRGE	ADR2
992				C	ENDM	
993				č		
994				C+CMPRE	MACRO	ADR 1
995				C+	CMP	ADR1
996				C+	RTSEQ	
997				Č	ENDM	
998				Č		
999				C+CPXBE	MACRO	ADR1,ADR2
1000				C+	CPX	ADR 1
1001				C+	BEQ	ADR2
1001				C	ENDM	
1002				C	LINDIN	
1003				C+CPXBG	MACRO	ADR1,ADR2
1004				C+	CPX	ADR1
1005				C+	BGE	ADR2
				C.	ENDM	
1007				C	LINDIN	
1008				C+CPXRGT	MACRO	ADR 1
1009				C+	CPX	ADR 1
1010		4			CFA	ADICI

Infocom In ERLOGIC interpre	eter disassen	mbly, 5/2	27/84 MACRO-80 3.	18-Sep-81	PAGE	1-20
1011 1012 1013 1014 1015 1016 1017 1018	C+ C ·C C+CPYBN C+ C+	RTSGT ENDM MACRO CPY BNE ENDM	ADR1,ADR2 ADR1 ADR2			

1020								
1021				; start	of inte	rpreter		
1022								
1023	0000′				ASEG	1,0000		load at one address
1024					ORG	LDORG		but assemble for another
1025					. PHASE	MAINOR .	;	but assemble for another
1026								wany important
1027	0800	D8		START:	CLD		;	very important
1028						## 0.0		alone ave continue of zone come
1029	0801		00		LDA	#\$00	;	clear our section of zero page
1030	0803		80		LDX	#\$8 0		
1031	0805	95	00	L0805:	STA	\$00,X		
1032				+	IXBNE	L0805		
1033						" * 5.5		init boodware stack
1034	A080		FF		LDX	#\$FF	;	init hardware stack
1035	080C	9 A			TXS			
1036						T.U.T.T.C.O.		init and along someon window
1037	Q080	20	1AF7		JSR	INITSC	;	init and clear screen window
1038						THE COLUMN TO STATE OF THE STAT		indicate no pages loaded
1039				+	MOV	<#\$00>, <prgupd,auxupd></prgupd,auxupd>	;	indicate no pages loaded
1040								i-itft.com stook
1041				+	MOV	<#\$01>,STKCNT	;	init software stack
1042				+	DMOVI	STCKLC, STKPNT		•
1043								
1044				+	MOV	<#\$FF>,LD9		
1045 ·						ANTAL C. MATARA		init wintual mamasy table pointers
1046				+	DMOVI	VMT1LC, VMTAB1	;	init virtual memory table pointers
1047				+	DMOVI	VMT2LC, VMTAB2		
1048				+	DMOVI	VMT3LC, VMTAB3		
1049				+	DMOVI	VMT4LC,VMTAB4		
1050								init winters) memory tables
1051	0846		00		LDY	#\$ 00	;	init virtual memory tables
1052	0848	A2	80		LDX	#\$80		ITADOL VSS
1053	084A			+L084A:	MOV	<#\$FF>,<<(VMTAB1),Y>,<(, V IV	11AD2), Y >>
1054	0850	98			TYA			
1055				+	ADD	,<#\$01>,<<(VMTAB3),Y>>		
1056	0856	98			TYA			
1057				+	SUB	,<#\$01>,<<(VMTAB4),Y>>		
1058	085C	С8			INY			
1059				+	DXBNE	L084A		
1060	0860	88			DEY			
1061				+	MO∨	<#\$FF>,<<(VMTAB3),Y>>		
1062						WERRY AND LIDAR		
1063				+	MOV	<#\$00>,MRUPAG		
1064				+	MOV	<#\$7F>,LRUPAG		
1065					~	STORE OF EDITHEN		init memony cointers
1066				+	DMOVI	FIRFLC, FRZMEM	;	; init memory pointers
1067					2427	EDZMEN ACC		mond log gags A to first frozen nage
1068				+	DMOV	FRZMEM, ACC	;	read log page 0 to first frozen page
1069				+	DMOVI	\$0000,ACB		
1070	0885	20	1E0D		JSR	DRDBKF		
1071				120	. 5.7	#UDDED7 . 1		setup frozen storage page count
1072	0888	ΑO	05		LDY	#HDRFRZ+1		; setup frozen storage page count ; bump up to page boundary-1
1073				· +	MOV	<#\$FF>,<<(FRZMEM),Y>>	•	, bump up to page boundary i
1074	088E	88			DEY			

1075				+	MOV	<(FRZMEM),Y>,FRZPGS	
1075	0893	E6	вс		INC	FRZPGS	
1077	0000						
1078	0895	Α9	00		LDA	#\$00	; read in rest of frozen memory
1079	0897			+L0897:	ADD	,<#\$01>	
1080	089A	AA			TAX	,	
1081	089B		в8		ADC	FRZMEM+1	
			E7		STA	ACC+1	
1082	089D	65	L /			FRZMEM, ACC	
1083				+	MOV	FRZWEM, ACC	
1084	8A80	88			TXA	507000 + 0000	
1085				+	CMPBE	FRZPGS,LO8B6	
1086	8A80	48			PHA		
1087	08A9	85	E4		STA	ACB	
1088				+	MOV	<#\$00>,ACB+1	
1089	08AF	20	1EOD		JSR	DRDBKF	•
1090	08B2	68			PLA		
1091	0883		0897		JMP	L0897	
1092	0000		000.				
	0886	Α.Ω	01	L08B6:	LDY	#HDRTYP	: setup for proper type of status line
1093			BA	LUGDO:	LDA	(FRZMEM),Y	, bottop for proper type of bratter time
1094	08B8					#\$02	
1095	08BA		02		AND	•	
1096	08BC	85	F3		STA	STLTYP	
1097							
1098	08BE	ΑO	07		LDY	#HDRSTR+1	; init PC
1099				+	MOV	<(FRZMEM),Y>,PRGIDX	
1100	08C4	88			DEY		
1101				+	MOV	<(FRZMEM),Y>,PRGLPG	
1102				+	MOV	<#\$00>,PRGLPG+1	
1103						,	
1104	08CD	۸٥	0D		LDY	#HDRGBV+1	: init global variable pointer
	0865	AU	UD	+	MOV	<(FRZMEM),Y>,GLBVAR	, 3,022,
1105	0000	0.0		T	DEY	((KEMEM), IF, GEBVAK	
1106	08D3	88		+		<(FRZMEM),Y>,FRZMEM+1,G	I BV/AD+1
1107				T	ADD	(FRZMEW), YZ, FRZMEW, I, G	LDVAR. I
1108		_	_			#UDD CBUL 1	init and models spintes
1109	08DB	ΑU	19		LDY	#HDRSBW+1	; init sub-word table pointer
1110				+	MOV	<(FRZMEM),Y>,SBWDPT	
1111	08E1	88			DEY		
1112				+	ADD	<(FRZMEM),Y>,FRZMEM+1,S	BWDPT+1
1113							·
1114				+	MOV	<#\$00>,SWPMEM	; swpmem := frzmem + 256 * frzpgs
1115				+	ADD	FRZPGS, FRZMEM+1, SWPMEM+	1
1116							
1117	08F4	20	1B1E		JSR	FNDMEM	; determine number of pages of memory
1118	001 4		10.2	+	SUB	, SWPMEM+1	: swppgs := (maxmem - swpmem) / 256
1119	08FA	0.0	0E	·	BCC	L090A	; if swppgs < 0 then fatal error
					TAY	LUSUA	, it suppose to their rates of the
1120	08FC	A8					
1121	08FD	C8			INY	CHIDDEE	
1122	08FE		BD		STY	SWPPGS	
1123	0900	8A			TAY	•	
1124	0901	84	BF		STY	LRUPAG	
1125				+	MOV	<#\$FF>,<<(VMTAB3),Y>>	
1126						•	
1127	0907	4C	098F		JMP	MNLOOP	; start the game!
1128					*		
1129	090A	20	21D1	L090A:	JSR	FATAL	
1130		_0	-	· · •			

PAGE 1-23

1131

```
1132
                                 ; class C instructions (implicit or no operand)
1133
1134
         090D
                 0018
                                 OPTAB1: DW
                                                  OPRTNT
                                                                            ; return with TRUE
1135
         090F
                  0C23
                                         DW
                                                  OPRINE
                                                                            ; return with FALSE
1136
                                                  OPPSI
                                                                            ; print string immediate
                 0C28
                                         DW
         0911
1137
                                         DW
                                                  OPPSIC
                                                                            ; print string immediate, CRLF, return true
         0913
                 0C54
1138
                                         DW
                                                  OPNULL
1139
         0915
                  0C53
                                                                            ; no-op
         0917
                  204B
                                         DW
                                                  OPSVGM
                                                                            ; save game status to disk
1140
                                                                            : restore game status from disk
                  20EB
                                         DW
                                                  OPRSGM
         0919
1141
                                                                            ; restart game
                                                  START
         091B
                  0800
                                         DW
1142
                                                                            ; return with value
                                                  OPRTNV
1143
         091D
                  0C64
                                         DW
                                                                            ; drop a word from the stack
         091F
                  1720
                                         DW
                                                  PULLWD
1144
                                         DW
                                                  OPENDS
                                                                            ; end the game
                  21EA
1145
         0921
                                                                            ; print CRLF
                                                  OPCRLF
                  0C72
                                         DW
1146
         0923
                                                                            ; print status line
                  1C8A
                                         DW
                                                  OPPRST
1147
         0925
         0927
                  0C7C
                                         DW
                                                  OPCKSM
                                                                            ; checksum the program
1148
                                                  (*-OPTAB1)/2
1149
         000E
                                 OPMAX1 EQU
1150
1151
1152
                                 ; class B instructions (single operand)
1153
                                                                            ; compare ARG1=0 (ARG1<>0)
         0929
                  OCDD
                                 OPTAB2: DW
                                                  OPTSTZ
1154
                                                  OPGTSB
                                                                            ; get thing's sibling
1155
         092B
                  OCE9
                                         DW
1156
         092D
                  0CF3
                                         DW
                                                  OPGTCH
                                                                            ; get thing's child
                  0D0E
                                         DW
                                                  OPGTPR
                                                                            ; get thing's parent
         092F
1157
                                                  OPGTPL
                                                                            ; get length of property (given addr)
                  0D20
                                         DW
1158
         0931
                                                                            ; increment variable
                                         DW
                                                  OPINC
1159
         0933
                  0D43
         0935
                  0D60
                                         DW
                                                  OPDEC
                                                                            ; decrement variable
1160
                                                  OPPSB
                                                                            ; print string at byte address
                                         DW
1161
         0937
                  0D73
1162
         0939
                  21D1
                                         DW
                                                  FATAL
                  0D81
                                         DW
                                                  OPDSTT
                                                                            ; destroy thing
1163
         093B
         093D
                  ODE2
                                         DW
                                                  OPPRTN
                                                                            ; print thing name
1164
                                                  OPRTN
         093F
                  0E06
                                         DW
                                                                            ; return
1165
                                                                            ; unconditional jump
         0941
                  0E7C
                                         DW
                                                  OPJUMP
1166
                                                                            ; print string at word address
                  0E92
                                         DW
                                                  OPPSW
1167
         0943
                                                  OPMOVE
                                                                            ; move var ARG1 to var
1168
         0945
                  0EA0
                                         DW
                                         DW
                                                  OPNOT
                                                                            ; 1's complement
1169
         0947
                  0EA8
                                 OPMAX2
                                         EQU
                                                  (*-OPTAB2)/2
1170
         0010
1171
                                         PAGE
1172
```

```
1173
                                 ; class A instructions (variable number of operands, may use short form
1174
                                 ; opcode)
1175
1176
                  21D1
                                 OPTAB3: DW
                                                  FATAL
         0949
1177
                  116B
                                         DW
                                                  OPMTCH
                                                                            ; match ARG1 against ARG2, ARG3, or ARG4
1178
         094B
                                                  LOEB7
                                                                            ; ??? compare ARG1<=ARG2 (ARG1>ARG2)
                  0EB7
                                         DW
1179
         094D
                                                                            ; ??? compare ARG1>=ARG2 (ARG1<ARG2)
                                                  L0ECF
1180
         094F
                  0ECF
                                         DW
                  0EE7
                                         DW
                                                  OPDECB
                                                                            : decrement variable and branch
1181
         0951
                                                                            : increment variable and branch
                                         DW
                                                  OPINCB
1182
         0953
                  0EF5
                                                                            ; is thing ARG1 in thing ARG2
                                         DW
                                                  OPTINT
1183
         0955
                  0F13
                                                  L0F23
1184
         0957
                  0F23
                                         DW
                  OF3B
                                         DW
                                                  OPOR
                                                                            ; logical OR
1185
         0959
                                         DW
                                                  OPAND
                                                                            ; logical AND
1186
         095B
                  OF4A
                                                  OPTSTA
                                                                            ; test thing attribute
                                         DW
1187
         0950
                  0F59
                                                  OPSETA
                                                                            ; set thing attribute
1188
         095F
                  OF6D
                                         DW
                  0F80
                                         DW
                                                  OPCLRA
                                                                            ; clear thing attribute
1189
         0961
                                                                            : move ARG2 into var ARG1
         0963
                  0F97
                                         DW
                                                  L0F97
1190
                                         DW
                                                  OPMOVT
                                                                            ; move thing ARG1 into thing ARG2
         0965
                  0FA4
1191
                                         DW
                                                  OPGTWD
                                                                            : get a word
1192
         0967
                  0FD2
                                                                            ; store a word
                  OFEC
                                         DW
                                                  OPGTBY
1193
         0969
                                                                            ; get thing property
                                         DW
                                                  OPGTP
1194
         096B
                  1008
                                                  OPGTPA
                                                                            ; get address of property
         096D
                  1069
                                         DW
1195
                                                  OPGTNP
                                                                            ; get next property
1196
         096F
                  109E
                                          DW
         0971
                  10C3
                                         DW
                                                  OPADD
                                                                            ; add
1197
                                         DW
                                                  OPSUB
                                                                            ; subtract
                  10D3
1198
         0973
                                                  OPMUL
                                                                            ; multiply
                                         DW
1199
         0975
                  10E3
                                                                            ; divide
                                         DW
                                                  OPDIV
1200
         0977
                  1118
                                         DW
                                                  OPRMD
                                                                            ; remainder
1201
         0979
                  114A
                                 OPMAX3 EQU
                                                  (*-OPTAB3)/2
1202
         0019
1203
1204
                                 ; class D instructions (variable number of operands)
1205
1206
                                                                            ; call procedure
                                 OPTAB4: DW
                                                  OPCALL
1207
         097B
                  11A3
                                                  OPPTWD
                                                                            ; store a word
         097D
                  125F
                                         DW
1208
                                         DW
                                                  OPPTBY
                                                                            ; store a byte
1209
         097F
                  1288
                                         DW
                                                  OPPTP
                                                                            ; store into thing property
1210
         0981
                  12A9
                                          DW
                                                  OPGTLN
                                                                            ; get a line of input
1211
         0983
                  12DC
                                                  OPPRCH
                                                                            : print a character
1212
         0985
                  14E5
                                          DW
                                                  OPPRNM
                                                                            ; print number
1213
         0987
                  14EA
                                          DW
1214
         0989
                  1536
                                          DW
                                                  OPRNDM
                                                                            ; generate random number
1215
         098B
                  1555
                                          DW
                                                  OPPUSH
                                                                            ; push ARG1 to stack
                                                  OPPULL
                                                                            ; pull var from stack
                                          DW
1216
         098D
                  1560
                                         EQU
                                                  (*-OPTAB4)/2
1217
         A000
                                 OPMAX4
1218
                                          PAGE
1219
```

1220								
1221								
1222	098F			+MNLOOP:	MOV	<#\$00>,ARGCNT	;	default no arguments
1223								
1224	0993		173E		JSR	FTPRBA	;	get opcode
1225	0996	85	80		STA	OPCODE		
1226			(•			
1227				+	CMPJL	#\$80,OPCGPA	;	is it class A (\$00-\$7F)?
1228				+	CMPJL	#\$BO,OPCGPB	:	how about class B (\$80-\$AF)?
1229				+	CMPBL	#\$CO,OPCGPC	:	perhaps class C (\$BO-\$BF)?
1230				;	JMP	OPCGPD		nope, it's class D (\$CO-\$FF).
1231				,	•		•	(411 411)
1232								
1233				. nrnce	ss ancad	e group D (\$CO-\$FF)		
1234				, proce.	33 opeou	c group b (wee with		
1235	09AA	20	173E	OPCGPD:	150	FTPRBA		get operand specification byte
1236	USAA	20	1736	OFCGFD:	J 3N	LIFKDA	,	ger operand specification byte
	0040	A 2	00		LDX	#\$00	_	igit aparand count
1237	09AD	AZ	00		LDX	#\$00	,	init operand count
1238	0045	40			DUIA			some the engaged appointing but
1239	09AF	48		LO9AF:	PHA		-	save the operand specification byte
1240	0980	8A			TAY		;	in Y and on stack
1241								
1242	09B1	8A			TXA		;	save operand count on stack
1243	09B2	48			PHA			
1244								
1245 .	09B3	98			TYA			get back operand specification byte
1246	09B4	29	CO		AND	#\$C0	;	look at top two bits
1247								
1248				+	JSREQ	FTPRWD,L09D7	;	if they're 00, operand is word immed.
1249				+	CMPJSE	#\$80, <gtvarp,l09d7></gtvarp,l09d7>	;	10? variable
1250				+	CMPJSE	#\$40, <ftprby,l09d7></ftprby,l09d7>	;	01? byte immediate
1251								
1252	09D2	68			PLA		;	must be 11, no more operands
1253	09D3	68			PLA		:	pull operand spec byte and count
1254	09D4	4C	09ED		JMP	L09ED	:	and finish up
1255							•	•
1256	09D7	68		L09D7:	PLA			get operand count back
1257	0908	AA			TAX			to use as index
1258							,	
1259				+	MOV	ACC, < <arg1, x="">></arg1,>		store operand in proper ARG location
1260				+	MOV	ACC+1, < <arg1+1, x="">></arg1+1,>	,	the control of the co
1261						7,000		
1262	09E1	E8			INX			increment ARG pointer
1263	09E2	E8			INX		,	merement And potition
1264	09E3	E6	Ο 1		INC	ARGCNT		and count
1265	0913	LO	01		TNC	ARGENT	,	and count
1265	09E5	68			PLA			null and anda buta
								pull arg spec byte
1267	09E6	38			SEC	٨		shift top two bits off left, while
1268	09E7	2A			ROL	Α		shifting 11 in from right (to
1269	09E8	38			SEC	•	;	indicate no more operands)
1270	09E9	2A			ROL	A		
1271								
1272	09EA	4C	09AF		JMP	L09AF	;	try for another
1273								
1274	09ED			+L09ED:	DMOVI	OPTAB4,ACC	ï	assume class D

Infocom	N ≟RLOG	IC in	nterpreter	disassemb	oly, 5/2	7/84 MACRO-80 3.	18-Sep-81 PAGE 1-27
1075	09F5	A5	an		LDA	OPCODE	; but if it's \$CO-\$DF then it's class A
1275 1276	USFS	AS	80	+	CMPJL	#\$E0,L0A98	,
1277						,	
1278	09FE	E9	E0		SBC	#\$E0	; adjust to \$00\$1F
1279				+	CMPBG	#OPMAX4,LOA2B	; make sure it's not illegal
1280							(hara in ACC)
1281	0A04	OA		GODOIT:		A	; get address from table (base in ACC)
1282	0A05	8A			TAY MOV	<(ACC),Y>,DSPTCH+1	; word indexed by A and execute
1283	0.4.00	CO		+	INY	(ACC), YZ, DSPTCHFT	
1284 1285	OAOB	С8		+	MOV	<(ACC),Y>,DSPTCH+2	
1285	OA 1 1	20	0A11	DSPTCH:		DSPTCH	
1287	0A14		098F	201 10111	JMP	MNLOOP	
1288	5 ,						
1289							
1290				; proce	ss opcod	le group C (\$BO-\$BF)	
1291							
1292	OA 17			+OPCGPC:		,<#\$B0>	; adjust to \$00\$0F
1293				+	CMPBG	#OPMAX1,LOA2B	; make sure it's not illegal
1294	OAIE	48			PHA	ODTAB1 ACC	; save it temp. ; get base address of proper table
1295				+	DMOVI	OPTAB1,ACC	; get base address of proper table
1296	0A27	68	0.4.0.4		PLA J m p	GODOIT	
1297	0A28	40	0A04		JMP	GODOTI	
1298 1299	0A2B	20	21D1	LOA2B:	JSR	FATAL	; oops! illegal opcode
1300	UAZB	20	2101	LUAZD.	0311	TATAL	,
1301							
1302				: proce	ss opcod	de group B (\$80-\$AF)	
1303				, ,		•	
1304	0A2E	29	30	OPCGPB:	AND	#\$30	; mask off operand type bits
1305							
1306				+	JSREQ	FTPRWD,LOA45	; 00? then it's word immediate
1307				+	CMPJSE		; 01? byte immediate
1308	0A42	20	OAE8		JSR	GTVARP	; must be 10, variable
1309	a			0 . 45	NAON (CHECKS ADCONT	- one acquiment:
1310	0A45			+L0A45:	MOV	<#\$01>,ARGCNT	; one argument
1311				+	DMOV	ACC, ARG1	
1312	0A51	۸۶	80		LDA	OPCODE	; adjust opcode to \$00\$0F
1313 1314	0A51		0F		AND	#\$0F	, dagate species to territor
1315	0.755	23	0.	+	CMPBG	#OPMAX2,LOA2B	; make sure it's not illegal
1316	0A59	48			PHA		; save temp.
1317	07.00			+	DMOVI	OPTAB2,ACC	; get appropriate table base addr
1318	0A62	68			PLA	·	
1319	0A63	4C	0A04		JMP	GODOIT	; and go do it!
1320							
1321							
1322				; proce	ss opcod	de group A (\$00-\$7F)	
1323							ADC1
1324	0A66	29	40	OPCGPA:		#\$40	; get type bit for ARG1
1325	2470		0455	+	JSREQ	FTPRBY, LOA73	; O: byte immediate ; 1: variable/stack
1326	0A70	20	OAE8	*1 U * 7 3	JSR DMOV	GTVARP	; i: variable/stack ; save it
1327	0A73			+L0A73:	DINIO A	ACC, ARG1	, dave it
1328 1329	OA7B	۸۶	80		LDA	OPCODE	; get type bit for ARG2
1329	0A7B		20		AND	#\$20	, get -, pe

0A7B 0A7D

1330

A5 80 29 20

AND

OPCODE #\$20

1331			+	JSREQ	FTPRBY, LOASA	,	0: byte immediate
1332	0A87	20 OAE8		JSR	GTVARP	,	1: variable/stack
1333	A8A0		+LOA8A:	DMOV	ACC,ARG2	;	save it
1334							
1335			+	MOV	<#\$02>,ARGCNT	:	indicate two operands
1336		<i>'</i>			,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	,	
	0.400	AF 00		1.04	OPCODE		act accords book
1337	0A96	A5 80		LDA	**	•	get opcode back
1338	0A98	29 1F	L0A98:	AND	#\$1F	;	adjust to \$00\$1F
1339			+	CMPBG	#OPMAX3,LOA2B		make sure it's not illegal
1340	0A9E	48		PHA		:	save temp.
1341			+	DMO∨I	OPTAB3.ACC		get base addr of appropriate table
1342	OAA7	68		PLA		,	
					COROLE		
1343	8AAO	4C 0A04		JMP	GODOIT	;	and go do it!
1344							
1345				PAGE			•
						•	

1-28

Infocom $I_{\rm res}$ _RLOGIC interpreter disassembly, 5/27/84

1346						
1347			; fetch	byte im	mediate into ACC	
1348						
1349	OAAB	20 173E	FTPRBY:		FTPRBA	; get a byte from program into A
1350	OAAE	85 E6		STA	ACC	; zero-fill to 16 bits in ACC
1351			+	MOV	<#\$00>,ACC+1	
1352	OAB4	60		RTS		; return
1353						
1354						
1355			; fetch	word im	mediate into ACC	
1356						
1357	OAB5	20 173E	FTPRWD:	JSR	FTPRBA	; get high byte from program into A
1358	OAB8	48		PHA		; save it temp.
1359	OAB9	20 173E		JSR	FTPRBA	; get low byte from program into A
1360	OABC	85 E6		STA	ACC	; store low byte
1361			+	PUL	ACC+1	; store high byte
1362	OAC1	60		RTS		; return
1363						
1364						
1365	OAC2		+GTVRA1:		LOADO	; fetch ACC from var in A, keep if stack
1366	OAC6	4C OAEF		JMP	GTVARA	
1367						
1368						
1369	OAC9		+PTVRA1:	TSTABE	LOAD6	; store ACC into var in A, replace if stack
1370	OACD	4C 0B46		JMP	PTVARA	
1371						
1372	OADO	20 1720	LOADO:	JSR	PULLWD	; read stack non-destructive
1373	0AD3	4C 16F4		JMP .	PUSHWD	
1374						
1375	OAD6		+LOAD6:	DPSH	ACC	; replace TOS w/ ACC
1376	OADC	20 1720		JSR	PULLWD	
1377			+	DPUL	ACC	
1378	OAE5	4C 16F4		JMP	PUSHWD	
1379						
1380				PAGE		

1381						~~cc.		£ a + a b	400	£		ind	hu	program
1382	OAE8	20	173E	GTVARP:		FTPRBA	,	ettn	ACC	110111	vai	ma.	Uy	pi ugi am
1383				+	TSTABE	LOB26		6 - 1 - 1-	• • •	6				
1384	OAEF			+GTVARA:		<#\$10>,L0B02	;	fetch	ACC	rrom	var	IN A		
1385				+	SUB	,<#\$01>								
1386	OAF6	OA			ASL	· A								
1387	OAF7	AA			TAX									
1388				+	MOV	<locvar,x>,ACC+1</locvar,x>								
1389	DAFC	E8			INX									
1390				+	MOV	<locvar,x>,ACC</locvar,x>								
1391	0B01	60			RTS	•								
1392	ODO I													
1393	0802			+L0B02:	SUB	,<#\$10>								
	0B02 0B05	OA		· LOBOL.	ASL	Α								
1394		85	E 4		STA	ACB								
1395	0B06				LDA	#\$ 00								
1396	0808	A9	00			# 3 00								
1397	OBOA	2A			ROL									
1398	0B0B	85	E5		STA	ACB+1								
1399				+	DADD	GLBVAR, ACB, ACB								
1400	0B1A	Α0	00		LDY	#\$00								
1401				+	MOV	<(ACB),Y>,ACC+1								
1402	0B20	С8			INY									
1403				+	MO∨	<(ACB),Y>,ACC								
1404	0B25	60			RTS									
1405														
1406	0B26	20	1720	L0B26:	JSR	PULLWD								
1407	0829	60			RTS									
1408	0010													
1409					PAGE									
1409														

1410								
1411	OB2A	A9	00	PTVRPZ:	LDA	#\$00	; store 0 in var.	. ind. by program
1412	OB2C	85	E6 .	PTVRPA:	STA	ACC	; store byte in A	in var. ind. by prog.
1413				+	MOV	<#\$00>,ACC+1		
1414	0B32	4C	0B35	PTVRP1:	JMP	PTVARP	; unnecessary!!!	
1415								
1416	0835			+PTVARP:	DPSH	ACC	; store ACC in va	ar. ind. by program
1417	0B3B	20	173E		JSR	FTPRBA		
1418	OB3E	AA			TAX			
1419	0			+	DPUL	ACC		
1420	0845	8A			TXA			
1421	0B46			+PTVARA:		PUSHWD	; store ACC in va	ar. in A
1422	02.0			+	CMPBG	<#\$10>,L0B60		
1423				+	DECA	,		
1424	0B54	OA			ASL	A		
1425	0B55	AA			TAX			
1426	0_00			+	MOV	ACC+1, < <locvar, x="">></locvar,>		
1427	OB5A	E8			INX	•		
1428				+	MOV	ACC, < <locvar, x="">></locvar,>		
1429	0B5F	60			RTS	•		
1430								
1431	0860			+L0B60:	SUB	,<#\$10>		
1432	0B63	AO.			ASL	À		
1433	0B64		E4		STA	ACB		
1434	0866		00		LDA	#\$00		
1435	0B68	2A			ROL			
1436	0869		E5		STA	ACB+1		
1437				+	DADD	GLBVAR, ACB, ACB		
1438	0878	ΑO	00		LDY	#\$00		
1439				+	MOV	ACC+1,<<(ACB),Y>>		
1440	0B7E	С8			INY			
1441				+	MOV	ACC,<<(ACB),Y>>		
1442	0B83	60			RTS			
1443								
1444					PAGE			

...

1 4 4 5								
1445 1446	0884	20 1	73F	PREDTR:	JSR	FTPRBA		fetch first displacement byte
1447	0004			+	TSTABM	LOB9C		complement condition if necessary
1448	088В	10 0	17		BPL	L0B94	,	,
1449	0000	, 0	•					
1450	088D	20 1	73F	PREDFL:	JSR	FTPRBA	:	fetch first displacement byte
1450	0000	20 1	,,,,	+	TSTABP	L0B9C		complement condition if necessary
1451				;	BMI	LOB94	,	,
1452				,	Dimi	2000 .		
1453	0894	29 4	10	L0B94:	AND	#\$40		branch not taken
1454	0094	23 -	•0	+	JSREQ	FTPRBA		fetch second displacement byte if
1455	0898	60		•	RTS	i ii kok		necessary and discard it
1450	0690	00			K13		,	noodally and around vi
1457	овэс	AA		LOB9C:	TAX			branch take, save first disp. byte
	0B9D	29 4	10	LUBSC.	AND	#\$4 0		do we need a second byte?
1459		F0 0	-		BEQ	LOBAD		yes
1460	0B9F	8A	,,		TXA	LUDAD		no, extend what we have w/ zeros
1461	OBA1		· -		AND	#\$3F	•	no, extend what we have we have
1462	OBA2	29 3			STA	ACC		
1463	OBA4	85 E	-6	+	MOV	<#\$00>,ACC+1		
1464	00	40.0	2000	т	JMP	LOBC3		and go do it!
1465	OBAA	4C C	BC3		JMP	LUBCS	,	and go do it:
1466	00.0				TVA			get rest of displacement
1467	OBAD	8A		LOBAD:	TXA	# # 0F	,	get rest of displacement
1468	OBAE	29 3	31-		AND	#\$3F		
1469	0880	48			PHA	STOOD A		
1470	0881		173E		JSR	FTPRBA		
1471	OBB4	85 E	=6		STA	ACC		
1472				+	PUL	ACC+1		
1473	0889	29 2			AND	#\$20		
1474	OBBB	F0 (BEQ	LOBC3		
1475	OBBD	A5 E			LDA	ACC+1		
1476	OBBF	09 (ORA	#\$C0		
1477	0BC1	85 E	E 7		STA	ACC+1		
1478								16 11 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1479	OBC3			+LOBC3:	DTSTBE	ACC, OPRTNF	•	if displacement = 0, return false
1480				+	DDEC	ACC		
1481				+	DTSTBE	ACC, OPRTNT	;	if displacement = 1, return true
1482	OBDA			+LOBDA:	DDEC	ACC		
1483								
1484				+	MOV	ACC+1,ACB		copy high byte of displacement to ACB
1485	OBE9	OA			ASL	Α	;	and sign extend to 17 bits
1486	OBEA	A9 (00		LDA	#\$00		
1487	OBEC	2A			ROL	Α		
1488	OBED	85 (Ē5		STA	ACB+1		
1489								
1490				+	ADD	PRGIDX,ACC		add low byte of displacement to PC
1491	OBF4	90 (06		BCC	LOBFC	;	increment high 9 bits of displacement
1492				+	DINC	ACB	;	if overflow
1493	OBFC	85 8	ВА	LOBFC:	STA	PRGIDX		
1494	-							
1495				+	DTSTBE	ACB,LOC17	;	if high 9 bits of disp. =0, all done
1496						•		
1497	0004	18			CLC		:	add high 9 bits of disp. to PC log page
1498	0C05	A5 I	E4		LDA	ACB	•	
1499	0007	65			ADC	PRGLPG		
1 7 3 3	000,					- -		

Infocom	INTERLOG	IC interpre	eter disassem	nbly, 5	/27/84	MACRO-80 3.4	18-Sep-81	PAGE	1-33
1500	0009	85 8B		STA	PRGLPG	.			
1501	0C0B	A5 E5		LDA	ACB+1				
1501	OCOD	65 BC		ADC	PRGLPG	i+1			
					#\$01	• •	; mod 2^17		
1503	OCOF	29 01		AND	• •		, mou 2//17		
1504	OC11	85 BC		STA	PRGLPG	i+1 _			
1505									
1506			+	MOV	<#\$00>	,PRGUPD	; indicate pag	e change	
1507									
1508	0C17	60	L0C17:	RTS			; all done		
1509									
1510				PAGE					

MACRO-80 3.4% 18-Sep-81 PAGE 1	MACRO-80 3. 4℃	18-Sep-81	PAGE	1-34
--------------------------------	----------------	-----------	------	------

1511								(001)
1512	0C18		01	OPRTNT:		#\$01		return true (\$01)
1513	OCIA	85	82	LOC1A:	STA	ARG1		return byte in A
1514				+	MOV	<#\$00>,ARG1+1	;	make high byte of return value \$00
1515	0C20	4C	0E06		JMP	OPRTN	;	and do the return!
1516								(200)
1517	0C23		00	OPRTNF:	LDA	#\$ 00	;	return false (\$00)
1518	0C25	4C	OC1A		JMP	LOC1A		
1519								
1520								
1521	0C28			+OPPSI:	MOV	PRGIDX, AUXIDX	;	copy PC to AUX
1522				+	D M OV	PRGLPG, AUXLPG		
1523				+	MOV	<#\$00>,AUXUPD	;	indicate new log. page
1524								
1525	0C38	20	18B4		JSR	PRNTST	;	print the string
1526								
1527				+	MO∨	AUXIDX, PRGIDX	;	copy AUX back to PC
1528				+	DMOV	AUXLPG, PRGLPG		
1529				+	MOV	AUXUPD, PRGUPD		
1530				+	DMOV	AUXMPT, PRGMPT		•
1531								
1532	0C53	60		OPNULL:	RTS		;	done
1533								
1534								
1535	0C54	20	0C28	OPPSIC:	JSR	OPPSI	;	print string immediate
1536								
1537	0C57		0D		LDA	#CRCHAR	;	print CRLF (could use JSR OPCRLF)
1538	0C59		1B3F		JSR	BECHAR		
1539	0C5C		OA		LDA	#LFCHAR		
1540	OC5E	20	1B3F		JSR	BFCHAR		
1541								
1542	0C61	4C	0C18		JMP	OPRTNT	;	return true
1543						•		
1544								
1545	0C64	20	1720	OPRTNV:		PULLWD		pull value off stack
1546				+	DMOV	ACC,ARG1		save it for posterity
1547	0C6F	4C	0E06		JMP	OPRTN	;	return with it
1548								
1549								
1550	0C72	Α9	OD	OPCRLF:	LDA	#CRCHAR	.;	print CRLF
1551	0C74	20	1B3F		JSR	BFCHAR		
1552	0C77		OA		LDA	#LFCHAR		
1553	0C79	4C	1B3F		JMP	BFCHAR	;	implicit RTS
1554								
1555					PAGE			

Infocom INTERLOGIC interpreter disassembly, 5/27/84

Infocom	INTERLOGIC	interpreter	disassembly,	5/27/8
---------	------------	-------------	--------------	--------

AΛ	CRO-80	3.2	18-Sen-8

•		^	_
- 1	_	٠.	n

0070	۸۵	18	ODCK SM +	LDV	#HDDCKA+1	; get checksum end log. address (word
0070	AU	1.5				; index.)
0000	0.0		•		(TREMEM), TO, ARGE	, mae »
0082	88				<(ED7MEM) V> ADC2+1	
			т	MOV	<(FRZMEM),Y/,ARGZTI	

			+	MOV	<#\$00>, < ARG3, ARG1, ARG1	+1,ACC+1,ARG4> ; initialize everything
			+	MOV	<#ARG4>,L1807+1	; patch VM routine to swap in all pages
						; convert end address to byte index
						,
0C 9C	26	86		ROL	ARG3	
			+	MOV	<#\$40>,ACC	; start at log. address \$00040
OCA2	20	17B8		JSR	SETAXB	
OCA5	20	17E8	LOCA5:	JSR	FTAXBA	; get a byte
			+	DADDB2	ARG1	; and add it to checksum
0CB1	A5	93		LDA	AUXIDX	; compare AUX to end address
			+	CMPBN	ARG2.LOCA5	; if not done, loop
0CB7	45	91				,
		- '	+			
OCBD	45	92				
OCDD		J.	+			
			•	CIMI DIV	ARGO, EUCAS	
			_	MOV	<pre><#ED7DC\$> 1807+1</pre>	; unpatch VM routine
			•	WOV	\#1 RZFG3/, L160/ · 1	; dipaten vm routine
0.000	• •	10		LDV	#UDDCKV+1	: compare computed vs. expected checksum
						; compare computed vs. expected checksum
UCCA	Б	DA				
0.000			+		ARGI, LUCDA	
					(555454)	
OCD1	Вı	ВА				
			+	CMPJE	ARG1+1, PREDIR	
				=		
OCDA	4C	088D	LOCDA:	JMP	PREDFL	
				PAGE		
		OC82 88 OC98 06 OC9A 26 OC9C 26 OCA2 20 OCA5 20 OCB1 A5 OCB7 A5 OCBD A5 OCC8 A0 OCCA B1 OCCA B1 OCDO 88 OCD1 B1	OC82 88 OC98 06 84 OC9A 26 85 OC9C 26 86 OCA2 20 17B8 OCA5 20 17E8 OCB1 A5 93 OCB7 A5 91 OCBD A5 92 OCC8 A0 1D OCCA B1 BA OCD0 88 OCD1 B1 BA	OC82 8B + + CC98 06 84 CC9A 26 85 CC9C 26 86 CCA2 20 17B8 CCA2 20 17E8 LOCA5: + CCB1 A5 93 CCB7 A5 91 CCBD A5 92 + CCC8 A0 1D CCCA B1 BA CCD0 88 CCD1 B1 BA + CCD0 88 CCD1 B1 BA + CCCB A0 1D CCCA B1 BA + CCD0 88 CCD1 B1 BA	+ MOV DEY + MOV + MOV + MOV 0C98	+ MOV C(FRZMEM), Y>, ARG2 + MOV C(FRZMEM), Y>, ARG2 + MOV C(FRZMEM), Y>, ARG2+1 + MOV C(FRZMEM), Y , ARG2+1 + MOV C(FRZMEM), Y , ARG2+1 + CMPBN ARG2+1, ARG1+1 + CMPBN ARG2+1 + CMPBN ARG2+1, LOCA5 LDA AUXLPG + CMPBN ARG2+1, LOCA5 LDA AUXLPG+1 + CMPBN ARG3, LOCA5 + MOV C(FRZMEM), Y , CMPBN + CMPBN ARG1, LOCDA + CMPBN ARG1+1, PREDTR + CMPBL ARG1+1, PREDTR + CMPBL

1595	OCDD			+OPTSTZ:	DTST IN	ARG1,PREDFL				
1596 1597	0CE6	40	0B84	LOCE6:	JMP	PREDTR				
1598	UCLU		0004	20020.	J	7 1125 111				
1599	OCE9	A5	82	OPGTSB:	LDA	ARG1		; get	sibling of thing, predicate	е
1600	OCEB		16A7		JSR	SETUPT				
1601	OCEE		05		LDY	#THGSIB				
1602	OCFO	4C	OCFA		JMP	LOCFA				
1603										
1604	OCF3	A5	82	OPGTCH:	LDA	ARG1		; get	child of thing, predicate	
1605	OCF5	20	16A7		JSR	SETUPT				
1606	OCF8	ΑO	06	*	LDY	#THGCHD				
1607	OCFA			+LOCFA:	PSH	<<(ACC), Y>>				
1608	OCFD	85	E6		STA	ACC				
1609				+	MOV	<#\$00>,ACC+1				
1610	0003	20	0B35		JSR	PTVARP				
1611	0D06	68			PLA					
1612				+	TSTABN	LOCE6				
1613	ODOB	4C	0B8D .		JMP	PREDFL				
1614										
1615	ODOE	A5	82	OPGTPR:	LDA	ARG1		; get	parent of thing	
1616	0D10		16A7		JSR	SETUPT				
1617	0D13	A O	04		LDY	#THGPAR				
1618				+	MOV	<(ACC),Y>,ACC				
1619				+	MOV	<#\$00>,ACC+1				
1620 .	0D1D	4 C	0B32		JMP	PTVRP1				
1621										
1622	0D20			+OPGTPL:		ARG1,FRZMEM,ACC				
1623				+	DDEC	ACC				
1624	0D38		00		LDY	#\$00				
1625	OD3A	20	1693		JSR	GTPLEN				
1626				+	ADD	, <#\$01>				
1627	0D40	4C	OB2C		JMP	PTVRPA				
1628										
1629										
1630						riable ARG1				
1631	0D43		82	OPINC:	LDA	ARG1				
1632	0D45	20	OAC2		JSR	GTVRA1				
1633				+	DINC	ACC				
1634	OD4E			+L0D4E:	DPSH	ACC				
1635	0D54		82		LDA	ARG1				
1636	0D56	20	OAC9		JSR	PTVRA 1				
1637	0055	0.0		+	DPUL	ACC				
1638	0D5F	60			RTS					
1639										
1640						1.0.1. 1001				
1641				; aecre	ment var	riable ARG1				
1642	0000	• -	0.2	ODDEC	1.04	ABC 1				
1643	0D60		82	OPDEC:	LDA	ARG1				
1644	0D62	∠0	OAC2	_	JSR	GTVRA1 ACC				
1645	0070	4.0	0045	+	DDEC					
1646	0D70	40	0D4E	/ **	JMP	L0D4E				
1647										
1648				. nnint	ctoico	at buta address in	ABG1			
1649				; print	String	at byte address in	ARUI			

Infocom N _RLOGIC interpreter disassembly, 5/27/84 1650 1651 ; set AUX to point to string at
; byte address
; and print it! +OPPSB: DMOV ARG1,ACC SETAXB 0D73 0D7B 0D7E 20 17B8 JSR 1652 L0E9D 4C 0E9D JMP 1653 1654 PAGE 1655

18-Sep-81

PAGE

1656			docto	ou thing	ARG1 (move to location	0)
1657			; desti	by thing	ARGI (move to rocation	3,
1658		۸۳. ۵۵	OPDSTT:	1.00	ARG1	
1659	0Đ81	A5 82	050211:	JSR	SETUPT	
1660	0D83	20 16A7		LDY 13K	#THGPAR	
1661	0D86	A0 04			(ACC),Y	
1662	0D88	B1 E6		LDA	(ACC), T	
1663			+	RTSEQ TAX		
1664	0D8D	AA		DPSH	ACC	
1665			+		ACC	
1666	0D94	8A		AXT	SETUPT	
1667	0D95	20 16A7		JSR	#THGCHD	
1668	0D98	AO 06		LDY	(ACC),Y	
1669	OD9A	B1 E6		LDA	• • •	
1670			+	CMPBN	ARG1,LODB7	
1671			+	DPUL	ACB	
1672			+	DPSH	ACB	
1673	ODAC	AO 05		LDY	#THGSIB	
1674	ODAE	B1 E4		LDA	(ACB),Y	
1675	0D B 0	AO 06		LDY	#THGCHD	
1676	ODB2	91 E6		STA	(ACC),Y	
1677	ODB4	4C 0DD2		JMP	LODD2	
1678	0DB7	20 16A7	LODB7:	JSR	SETUPT	
1679	ODBA	AO 05		LDY	#THGSIB	
1680	ODBC	B1 E6		LDA	(ACC),Y	
1681			+	CMPBN	ARG1,LODB7	
1682			+	DPUL	ACB	
1683			+	DPSH	ACB	
1684			+	MOV	<(ACB), Y>, <<(ACC), Y>>	
1685	ODD2		+LODD2:	DPUL	ACC	
1686	0DD8	AO 04		LDY	#THGPAR	
1687			· +	MO∨	<#\$00>,<<(ACC),Y>>	
1688	ODDE	C8		INY		; to THGSIB
1689	ODDF	91 E6		STA	(ACC),Y	
1690	ODE 1	60		RTS		
1691						
1692				PAGE		

1693 1694 1695	0DE2 0DE4	A5 20	82 16A7	OPPRTN: LODE4:	LDA JSR	ARG1 SETUPT	<pre>; print thing name ; set up pointer to thing</pre>
1696 1697 1698	ODE7	ΑO	07	+	LDY MOV	#THGPRP <(ACC),Y>,ACB+1	; get address of thing's property list
1699 1700 1701	ODED	С8		+ +	MOV DMOV	<(ACC),Y>,ACB ACB,ACC	
1702 1703 1704				+	DINC	ACC	; skip name length byte
1705 1706	0E00 0E03		17B8 18B4		JSR JMP	SETAXB PRNTST	<pre>; set AUX to point to it ; and print it and return</pre>
1707 1708					PAGE		

1-39

1709						
1710 1711	0E06		+OPRTN:	DMOV	STKPSV,STKPNT	: restore pre-call stack pointer, count
1711	0500		+	MOV	STKCSV.STKCNT	•
1713						
1714	0E12	20 1720		JSR	PULLWD	; are there any local variables to restore?
1715	0E15	A5 E6		LDA	ACC	·
1716	0E17	FO 33		BEQ	L0E4C	'; no, skip it
1717	OL 17	10 00				• •
1717			+	DMOVI	LOCVAR-2.ACB	; yes, calc. addr. of last var to restore
1719			+	MOV	ACC.ACD	
1719	0E25	OA		ASL	Α	•
1721	ULZJ	UA.	+	DADDB2	ACB	
1721				0.10000		
1723	0E2F	20 1720	LOE2F:	JSR	PULLWD	; pull the value of the var
1724	0E32	AQ 01		LDY	#\$01	: store it in the var
1725	ULU2	A0 01	+	MOV	ACC, << (ACB), Y>>	
1725	0E38	88		DEY	,	
1726	0236	00	+	MOV	ACC+1,<<(ACB),Y>>	
			+	DDEC2	ACB	: decrement the var pointer
1728 1729			+	DECBN	ACD.LOE2F	and the count and loop if more to do
1729	*		•	DECON	, 2022.	,,
1730	0E4C	20 1720	LOE4C:	JSR	PULLWD	; pull the PC log. page
1731	UEAC	20 1720	+	DMOV	ACC.PRGLPG	, , , ,
			•	DINIO	7,00,111,021,0	
1733 1734	0E57	20 1720		JSR	PULLWD	: pull the stack pointer save
1734	UESI	20 1720	+	DMOV	ACC, STKPSV	, , , , , , , , , , , , , , , , , , , ,
1735			•	DING V	700,5711 51	
1736	0E62	20 1720		JSR	PULLWD	; pull the stack count save and PC
1737	UEUZ	20 1720	+	MOV	ACC+1, PRGIDX	; low byte
1738			+	MOV	ACC, STKCSV	,
1739			•	100	A00,0111001	
			+	MOV	<#\$00>,PRGUPD	; indicate need to locate new page
1741 1742				mo v	1#400°,1 NGOLD	,
			+	DMO∨	ARG1,ACC	: store the return value and return.
1743	0E79	4C 0B32	•	JMP	PTVRP1	•
1744	05/9	4C 003Z		JIMIF		
1745				PAGE		
1746				PAGE		

1747					
1748			; jump 1	o addre	ss ARG1
1749			. 00	DMOV	ADC1 ACC
1750	0E7C		+OPJUMP:	DDEC	ARG1,ACC ACC
1751	0505	40 0004	т	JMP	LOBDA
1752	0E8F	4C OBDA		JIMIT	EODDA
1753					
1754 1755	0E92		+OPPSW:	DMOV	ARG1,ACC
1756	0E9A	20 1709		JSR	SETAXW
1757	0E9D	4C 18B4	LOE9D:	JMP	PRNTST
1758	0200	.0 .00 .			
1759					
1760	OEAO	A5 82	OPMOVE:	LDA	ARG1
1761	OEA2	20 OAC2		JSR	GTVRA1
1762	0EA5	4C 0B32		JMP	PTVRP1
1763					
1764					
1765	OEA8		+OPNOT:	D1COMP	ARG1,ACC
1766	0EB4	4C OB32		JMP	PTVRP1
1767					
1768	OEB7		+L0EB7:	DMOV	ARG1,ACC
1769			+	DMO∨	ARG2,ACB
1770	OEC7	20 16DE		JSR	L16DE
1771	OECA	90 44		BCC	LOF10
1772 •	OECC	4C OB8D		JMP	PREDFL
1773					
1774	OECF		+LOECF:	DMOV	ARG1,ACB
1775			+	DMOV	ARG2,ACC
1776	OEDF	20 16DE		JSR	L16DE
1777	0EE2	90 2C		BCC	LOF10
1778	OEE4	4C OB8D		JMP	PREDFL
1779			000000	.65	00050
1780	OEE7	20 OD60	OPDECB:		OPDEC
1781		.0.0500	+	DMOV	ARG2,ACB
1782	OEF2	4C 0F08		JMP	L0F08
1783	0555	00 00 40	ODINCE	JSR	OPINC
1784	OEF5	20 OD43	OPINCB:	DMOV	ACC, ACB
1785			+	DMOV DMOV	ARG2,ACC
1786	0.00	20 1605	L0F08:	JSR	L16DE
1787	0F08	20 16DE	+	JCS	PREDFL
1788	0F10	4C 0B84	LOF10:	JMP	PREDTR
1789 1790	OFIO	40 0004	LOI TO.	O IVII	T ICES T.
1790	0F13	A5 82	OPTINT:	LDA	ARG1
1791	0F15	20 16A7	01 11111.	JSR	SETUPT
1793	0F18	AO 04		LDY	#\$04
	0F1A	A5 84		LDA	ARG2
1794 1795	ULTA	NO 04	+	CMPBE	<(ACC),Y>,LOF10
1796	0F20	4C 0B8D		JMP	PREDFL
1797	01 20	.0 0000			–
1798	0F23		+L0F23:	MOV	ARG2+1,ACC+1
1799	0F27	25 83	, ,	AND	ARG1+1
1800	0F29	85 E5		STA	ACB+1
1801	5, 25	-0 -0	+	MOV	ARG2,ACC
.00.					•

; setup to jump into middle of ; predicate routine

; and do it!

18-Sep-81

; set AUX to point to string at

; word address ; and print it!

; get number of first variable

; get its contents

; store into another variable

Infocom	¶ ≟RLOG	IC in	nterpret	ter disassem	bly, 5/	27/84 MACRO-80 3.	18-Sep-81	PAGE	1-42
1802	0F2F	25	82		AND	ARG1			
1803	0F31	85	E4		STA	ACB			
1804	0F33	20			JSR	L16E9			
1805	0F36	FO	D8		BEQ	L0F10			
1806	0F38	4C	088D		JMP	PREDFL			
1807									
1808	OF3B			+OPOR:	DOR	ARG2,ARG1,ACC			
1809	0F47	4C	0B32		JMP	PTVRP1			
1810									
1811	OF4A			+OPAND:	DAND	ARG2,ARG1,ACC			
1812	0F56	4C	0B32		JMP	PTVRP1			
1813									
1814					PAGE				

1815 1816 1817			; test a	ittribute	e bit ARG2 of thing ARG1
1818	0F59	20 1629	OPTSTA:	JSR	SETUPA
1819	0F5C	A5 E5	01 1 0 171.	LDA	ACB+1
	OF5E	25 E9		AND	ACD+1
1820				STA	ACB+1
1821	0F60	85 E5		LDA	ACB
1822	0F62	A5 E4		AND	ACD
1823	0F64	25 E8			
1824	0F66	05 E5		ORA	ACB+1
1825	0F68	DO A6		BNE	LOF10
1826	OF6A	4C OB8D		JMP	PREDFL
1827					
1828					
1829			; set at	ttribute	bit ARG2 of thing ARG1
1830					
1831	OF6D	20 1629	OPSETA:		SETUPA
1832	0F70	AO 01		LDY	#\$ 01
1833	0F72	A5 E4		LDA	ACB
1834	0F74	05 E8		ORA	ACD
1835	0F76	91 E6		STA	(ACC),Y
1836	0F78	88		DEY	
1837	0F79	A5 E5 .		LDA	ACB+1
1838	OF7B	05 E9		ORA	ACD+1
1839	OF7D	91 E6		STA	(ACC),Y
1840	OF7F	60		RTS	
1841					
1842					
1843			: clear	attribu	te bit ARG2 of thing ARG1
1844			,		
1845	0F80	20 1629	OPCLRA:	JSR	SETUPA
1846	0F83	AO 01		LDY	#\$01
1847	0F85	A5 E8		LDA	ACD
1848	0F87	49 FF		EOR	#\$FF
1849	0F89	25 E4		AND	ACB
1850	0F8B	91 E6		STA	(ACC),Y
1851	0F8D	88		DEY	()
1852	OF8E	A5 E9		LDA	ACD+1
1852	0F90	49 FF		EOR	#\$FF
		49 FF 25 E5		AND	ACB+1
1854	0F92	25 E5 91 E6		STA	(ACC),Y
1855	0F94			RTS	(100),1
1856	0F96	60		K12	
1857				PAGE	
1858				FAUL	

1859	0507		+L0F97:	DMOV	ARG2,ACC
1860	0F97 0F9F	A5 82	LUI 37.	LDA	ARG1
1861	OFA1	4C 0AC9	LOFA1:	JMP	PTVRA1
1862	UFAT	4C UACS	LUINI.	O.W.	
1863 1864	OFA4	20 0D81	OPMOVT:	JSR	OPDSTT
1865	OFA7	A5 82	•	LDA	ARG1
1866	OFA9	20 16A7		JSR	SETUPT
1867	01 73	20 10/17	+	DPSH	ACC
1868	OFB2	AO 04		LDY	#THGPAR
1869	01 52		+	MOV	ARG2, << (ACC), Y>>
1870	OFB8	20 16A7		JSR	SETUPT
1871	OFBB	AO 06		LDY	#THGCHD
1872	OFBD	B1 E6		LDA	(ACC),Y
1873	OFBF	AA		TAX	
1874	0		+	MOV	ARG1, << (ACC), Y>>
1875			+	DPUL	ACC
1876	OFCA	8A		TXA	
1877	OFCB	FO 04		BEQ	LOFD1
1878	OFCD	AO 05		LDY	#THGSIB
1879	OFCF	91 E6		STA	(ACC),Y
1880	0FD1	60	LOFD1:	RTS	
1881					
1882	OFD2		+OPGTWD:	DASL	ARG2
1883			+	DADD	ARG2,ARG1,ACC
1884	OFE3	20 17B8		JSR	SETAXB
1885	OFE6	20 17DB		JSR	FTAXWD
1886	OFE9	4C 0B32		JMP	PTVRP1
1887					
1888	OFEC		+OPGTBY:	DADD	ARG2,ARG1,ACC
1889	OFF9	20 17B8		JSR	SETAXB
1890	0FFC	20 17E8		JSR	FTAXBA
1891	OFFF	85 E6		STA	ACC
1892			+	MOV	<#\$00>,ACC+1
1893	1005	4C 0B32		JMP	PTVRP1
1894					·
1895	•			PAGE	

1934 1935			; get a	ddress (of property ARG2 o	of thing ARG1
1936	1000	20 1669	OPGTPA:	JSR	SETUPP	
1937	1069		L106C:	JSR	GTPNUM	
1938	106C	20 168E				
1939			+	CMPBE	ARG2,L107E	
1940			+	JCC	PTVRPZ	
1941	1078	20 169D		JSR	ADVPPT	
1942	107B	4C 106C		JMP	L106C	
1943	107E		+L107E:	DINC	ACC	
1944	1084	18		CLC		
1945	1085	98		TYA		
1946	1086	65 E6		ADC	ACC	
1947	1088	85 E6		STA	ACC	
1948	108A	90 02		BCC	L108E	
1949	108C	E6 E7		INC	ACC+1	
1950	108E		+L108E:	DSUB	ACC, FRZMEM, ACC	
1951	109B	4C OB32		JMP	PTVRP1	
1952						
1953				PAGE		

1954 1955 1956				; get n	umber of	next property	of	thing	ARG1	after	property	ARG2
1957	109E	20 1	1669	OPGTNP:	JSR	SETUPP						
1958	10A1	A5 8			LDA	ARG2						
1959	10A3	F0 1			BEQ	L10B7						
1960	10A5		168E	L10A5:	JSR	GTPNUM						
1961	10			+	CMPBE	ARG2,L10BD						
1962				+	JCC	PTVRPZ						
1963	10B1	20 1	169D		JSR	ADVPPT						
1964	10B4	4C 1	10A5		JMP	L10A5						
1965	10B7	20	168E	L10B7:	JSR	GTPNUM						
1966	10BA	4C (OB2C		JMP	PTVRPA						
1967	10BD	20	169D	L10BD:	JSR	ADVPPT						
1968	10C0	4C	10B7		JMP	L10B7						
1969												
1970					PAGE							

1971						
1972				; add Af	RG1 and A	ARG2
1973						
1974	10C3			+OPADD:	DADD	ARG1,ARG2,ACC
1975	10D0	4C OB32	2		JMP	PTVRP1
1976						•
1977						
1978				; subtra	act ARG2	from ARG1
1979						
1980	10D3			+OPSUB:	DSUB	ARG1, ARG2, ACC
1981	10E0	4C 0B32	2		JMP	PTVRP1
1982						
1983						
1984				; multip	oly ARG1	by ARG2
1985						
1986	10E3			+OPMUL:	DMOV	ARG1,ACC
1987				+	DMOV	ARG2,ACB
1988	10F3	20 15FE	3		JSR	L15FB
1989	10F6	A5 E5			LDA	ACB+1
1990	10F8	DO OA			BNE	L1104
1991	10FA	A5 E4			LDA	ACB
1992				+	CMPBE	<#\$02>,L1111
1993				+	CMPBE	<#\$04>,L110D
1994	1104	20 1568		L1104:	JSR	L1568
1995	1107	20 160A		L1107:	JSR	L160A
1996	110A	4C 0B32	2		JMP	PTVRP1
1997	110D			+L110D:	DASL	ACC
1998	1111			+L1111:	DASL	ACC
1999	1115	4C 1107	'		JMP	L1107
2000						
2001					PAGE	

2002					
2003			; divid	le ARG1 by	y ARG2
2004					
2005	1118		+OPDIV:	DMO∨	ARG1,ACC
2006			+	DMOV	ARG2,ACB
2007	1128	20 15FB		JSR	L15FB
2008	112B	A5 E5		LDA	ACB+1
2009	112D	DO 0A		BNE	L1139
2010	112F	A5 E4		LDA	ACB
2011			+	CMPBE	<#\$02>,L1143
2012			+	CMPBE	<#\$04>,L113F
2013	1139	20 15AD	L1139:	JSR	DIVIDE
2014	113C	4C 1107		JMP	L1107
2015	113F		+L113F:	DLSR	ACC
2016	1143		+L1143:	DLSR	ACC
2017	1147	4C 1107		JMP	L1107
2018					
2019					
2020			; get r	emainder	of ARG1 divided by ARG2
2021					
2022	114A		+OPRMD:	DMOV	ARG1,ACC
2023	•		+	DMOV	ARG2,ACB
2024	115A	20 15FB		JSR	L15FB
2025	115D	20 15AD		JSR	DIVIDE
2026			+	DMO∨	ACB, ACC
2027	1168	4C 0B32		JMP	PTVRP1
2028					
2029				PAGE	

2030							
2031			; test	whether	ARG1 is equal	to any	of the other args
2032							
2033	116B	A6 81	OPMTCH:		ARGCNT		
2034			+	DXBNE	L1173		
2035	1170	20 21D1		JSR	FATAL		
2036	1173	A5 82	L1173:	<u>LDA</u>	ARG1		
2037			+	CMPBN	ARG2,L117F		
2038	. 1179	A5 83		LDA	ARG1+1		
2039			+	CMPBE	ARG2+1,L11A0		
2040	117F		+L117F:	DXBEQ	L119D		
2041	1182	A5 82		LDA	ARG1		
2042			+	CMPBN	ARG3,L118E		
2043	1188	A5 83		LDA	ARG1+1		
2044			+	CMPBE	ARG3+1,L11AO		
2045	118E		+L118E:	DXBEQ	L119D		
2046	1191	A5 82		LDA	ARG1		
2047			+	CMPBN	ARG4,L1173		
2048	1197	A5 83		LDA	ARG1+1		
2049			+	CMPBE	ARG4+1,L11AO		
2050	119D	4C OB8D	L119D:	JMP	PREDFL		
2051	11AO	4C 0B84	L11A0:	JMP	PREDTR		
2052							
2053				PAGE			

.

2054 2055 2056				; call ;		e at addr. ARG1 and	d optionally pass ARG2, ARG3, and ARG4
2057 2058	11A3			+OPCALL:	DTS2BN	ARG1,L11B4	; if argument 1 (call address/2) is
2059 2060 2061	11B1	4C 0	B32	+	JMP	\$0000,ACC PTVRP1	<pre>; zero, just put zero in var ; these three lines could be replaced ; with "DTS2BE PTVRPZ"</pre>
2062 2063	11B4			+L11B4:	MOV	STKCSV, ACC	; push the stack count save and low byte
2064	1704			+	MOV	PRGIDX, ACC+1	; of the PC
2065	11BC	20 1	6F4		JSR	PUSHWD	
2066					DMO14	CTUDEN ACC	; push the stack pointer save
2067	1167	20 1	CE 4	+	DMOV JSR	STKPSV,ACC PUSHWD	; push the stack pointer save
2068 2069	11C7	20 1	6F4		JSK	PUSHWD	
2009				+	DMOV	PRGLPG.ACC	; push the PC logical page
2071	11D2	20 1	6F4		JSR	PUSHWD	
2072							
2073				+	MOV	<#\$00>,PRGUPD	; indicate need to search for new page
2074					5461	ABC1 DBCIBY	; make new PC := ARG1 * 2
2075		•0 0		+	DASL LDA	ARG1,PRGIDX #\$00	; make new PC := ARGI - 2
2076	11E3 11E5	A9 C 2A	JU		ROL	# 5 00	
2077 2078	11E6	85 8	ır.		STA	PRGLPG+1	
2079	1120	00 0	, .		•		
2080	11E8	20 1	173E		JSR	FTPRBA	; get first byte of routine
2081	11EB	48			PHA		; and save it
2082							if itte on local vanishles
2083				+	TSTABE	L1220	; if it's zero, no local variables
2084				- puch	the loca	l variables the rou	utine will use
2085 2086				; pasii	the roca	variables the re-	
2087	11F0	A2 0	00		LDX	#\$00	
2088	11F2	48		L11F2:	PHA	•	
2089				+	MOV	<locvar,x>,ACC+1</locvar,x>	
2090	11F7	E8			INX		
2091				+	MOV	<locvar,x>,ACC</locvar,x>	
2092	11FC	CA			DEX		
2093	11FD	8A			TXA PHA		
2094	11FE 11FF	48 20 1	1654		JSR	PUSHWD	
2095 2096	1202		173E		JSR	FTPRBA	
2097	1205	48			PHA		
2098	1206	20 1	173E		JSR	FTPRBA	
2099	1209	85 E	6		STA	ACC	
2100				+	PUL	ACC+1	
2101	120E	68			PLA		
2102	120F	AA			TAX MOV	ACC+1,< <locvar,x>></locvar,x>	· •
2103 2104	1214	E8		+	INX	ACC. I, CLUCVAN, X22	.
2104	1214			+	MOV	ACC. < <locvar, x="">></locvar,>	
2106	1219	E8		•	INX		
2107	121A	68			PLA -		
2108				+	SUB	, <#\$01>	

Infocom	ERLOG	IC interpret	er disassem	bly, 5/2	27/84 MACRO-80 3.	18-Seρ-81 PAGE 1-52
2109	121E	DO D2		BNE	L11F2	
2110						
2111	1220		+L1220:	MOV	ARGCNT, ACD	; do we pass any parameters?
2112			+	DECBE	ACD,L124C	; no
2113						
2114			+	MOV	<#\$00>,ACB	; yes, copy them in
2115			+	MOV	<#\$00>,ACC	
2116	1230	A6 E4	L1230:	LDX.	ACB	
2117	1232	B5 85		LDA	ARG2+1,X	
2118	1234	A6 E6		LDX	ACC	
2119	1236	95 9A		STA	LOCVAR,X	
2120	1238	E6 E6		INC	ACC	
2121	12 3A	A6 E4		LDX	ACB	
2122	123C	B5 84		LDA	ARG2,X	
2123	123E	A6 E6		LDX	ACC	
2124	1240	95 9A		STA	LOCVAR,X	
2125	1242	E6 E6		INC	ACC	
2126	1244	E6 E4		INC	ACB	
2127	1246	E6 E4		INC	ACB	
2128						• • • • • • • • • • • • • • • • • • • •
2129			+	DECBN	ACD,L1230	; loop if more paramaters to pass
2130						
2131	124C		+L124C:	PUL	ACC	; get the first program byte again
2132	124F	20 16F4		JSR	PUSHWD	; and push it so return can restore
2133						; the local variables
2134				***	CTWONE STROSH	and the steel estates and sount
2135			+	MOV	STKCNT, STKCSV	; save the stack pointer and count
2136			+	DMOV	STKPNT, STKPSV	
2137	.055			DTC		; all done!
2138	125E	60		RTS		; all dolle:
2139				DACE		
2140				PAGE		

18-Sep-81

Sen-81	PAGE	1-54

2184					*DCO - F + L-1-	and ADC1
2185			; store	ARG3 as	property ARG2 of thin	IS ARGI
2186						
2187	12A9	20 1669	OPPTP:	JSR	SETUPP	; setup for thing property operations
2188						
2189	12AC	20 168E	L12AC:	JSR	GTPNUM	; get the property number
2190			+	CMPBE	ARG2,L12BE	; if it is the one, go do it!
2191						
2192			+	JSRCC	FATAL	; oops! past it!
2193						
2194	1288	20 169D		JSR	ADVPPT	; advance pointer
2195	1288	4C 12AC		JMP	L12AC	; and try again
2196	1200	40 /2/10		•		
2197			· ant t	he orope	erty we wand	
			, 90	по р. орс		
2198	12BE	20 1693	L12BE:	JSR	GTPLEN	; get property length
2199	12BE	C8	LIZDL.	INY	#	, , , , ,
2200	1201	Co	+	CMPBE	₩\$00,L12D7	: if it is byte sized, go store it
2201			+	CMPJSN	#\$01.FATAL	; if it isn't word sized, fatal error
2202			7	CMPJSN	#301,1A1AL	, 11 12 1311 2 1131 3 1 1 1 1 1 1 1 1 1 1
2203				MOV	ARG3+1,<<(ACC),Y>>	; yes, store high byte
2204		0.0	+		ARGS11, << (ACC), 122	, yes, store man byto
2205	12D1	C8		INY		
2206				***	ABCO ((ACC) V>>	: these two lines are unnecessary
2207			+	MOV	ARG3,<<(ACC),Y>>	; these two times are dimecessary
2208	12D6	60		RTS		
2209						
2210	12D7		+L12D7:	MOV	ARG3,<<(ACC),Y>>	; store low byte
2211	12DB	60		RTS		; and return
2212						
2213				PAGE		
_						

2214					
2214	12DC	20 1C8A	OPGTLN:	JSR	OPPRST
2215 2216	1200	20 100%	+	DADD	ARG1, FRZMEM, ARG1
			+	DADD	ARG2, FRZMEM, ARG2
2217	1250	20 1D65	•	JSR	GETLIN
2218	12F9			STA	ACD+1
2219	12FC	85 E9			
2220			+	, MOV	<#\$00>,ACD
2221	1302	AO 01		LDY	#\$01
2222			+	MOV	<#\$00>,<<(ARG2),Y>>
2223			+	MOV	<#\$02>,LEO
2224			+	MO∨	<#\$01>,LE1
2225	1310	AO 00	L1310:	LDY	#00
2226	1312	B1 84		LDA	(ARG2),Y
2227	1314	C8		INY	
2228			. +	CMPRE	<(ARG2),Y>
2229			+	DTSTRE	ACD
2230	1321	A5 E8		LDA	ACD
2231			+	CMPJSE	<#\$06>,L13BA
2232	132A	A5 E8		LDA	ACD
2233	132C	DO 2E		BNE	L135C
2234	132E	AO 06		LDY	#\$06
2235	1330	A2 00		LDX	#\$00
2236	1332	A2 00	+L1332:	MOV	<#\$00>,<<\$D3,X>>
	1336	E8		INX	, 4 20,
2237	1330	LO	+	DYBNE	L1332
2238		45 51	•	LDA	LE1
2239 .	133A	A5 E1			LEO
2240	133C	A4 E0		LDY	LEO
2241	133E	C8		INY	
2242	133F	C8		INY	
2243	1340	С8		INY	(
2244	1341	91 84		STA	(ARG2),Y
2245	1343	A4 E1		LDY	LE1
2246	1345	B1 82		LDA	(ARG1),Y
2247	1347	20 13F1		JSR	L13F1
2248	134A	BO 2E		BCS	L137A
2249	134C	A4 E1		LDY	LE1
2250	134E	B1 82		LDA	(ARG1),Y
2251	1350	20 13E0		JSR	L13E0
2252	1353	90 07	,	BCC	L135C
2253	1355	E6 E1		INC	LE1
2254	1357	C6 E9		DEC	ACD+1
2255	1359	4C 1310		JMP	L1310
2256	135C	A5 E9	L135C:	LDA	ACD+1
2257	135E	F0 22		BEQ	L1382
2258	1360	A4 E1		LDY	LE1
2259	1362	B1 82		LDA	(ARG1),Y
2260	1364	20 13DA		JSR	L13DA
	1367	BO 19		BCS	L1382
2261		A4 E1		LDY	LE1
2262	1369			LDA	(ARG1),Y
2263	136B	B1 82		LDX	ACD
2264	136D	A6 E8		STA	
2265	136F	95 D3			INWORD,X
2266	1371	C6 E9		DEC	ACD+1
2267	1373	E6 E8	:	INC	ACD
2268	1375	E6 E1	•	INC	LE1

2269	1377	4C 1310		JMP	L1310
2270	137A	85 D3	L137A:	STA	INWORD
2271	137C	E6 E8		INC	ACD
2272	137E	C6 E9		DEC	ACD+1
2273	1380	E6 E1		INC	LE1
2274	1382	A5 E8	L1382:	LDA	ACD
		FO 8A	£1002.	BEQ	L1310
2275	1384	FU BA			ACD+1
2276			+	PSH	
2277	1389	A4 E0		LDY	LEO
2278	138B	C8	,	INY	
2279	138C	C8		INY	
2280			+	MOV	ACD, <<(ARG2), Y>>
2281	1391	20 1A05		JSR	CRNWRD
2282	1394	20 141F		JSR	L141F
2283	1397	A4 E0		LDY	LE0
2284	1001	// LO	+	MOV	ACB+1,<<(ARG2),Y>>
	1200	С8	•	INY	ACD: 1,(ARGZ), 1
2285	139D	Co			ACR <<(ADC2) V>>
2286		• •	+	MOV	ACB, << (ARG2), Y>>
2287	13A2	C8		INY	
2288	13A3	C8		INY	
2289	13A4	C8		INY	
2290	13A5	84 E0		STY	LEO
2291	13A7	AO 01		LDY	#\$01
2292			+	ADD	<(ARG2),Y>,<#\$01>,<<(ARG2),Y>>
2293			+	PUL	ACD+1
2294			+	MOV	<#\$00>,ACD
	1207	AC 1210	•	JMP	L1310
2295	13B7	4C 1310		JWIF	L1310
2296		45 50	1 10DA	1.04	ACD (1
2297	13BA	A5 E9	L13BA:	LDA	ACD+1
2298			+	RTSEQ	
2299	13BF	A4 E1		LDY	LE1
2300	13C1	B1 82		LDA	(ARG1),Y
2301	13C3	20 13DA		JSR	L13DA
2302			+	RTSCS	
23113	1309	F6 F1			LE1
2303	13C9	E6 E1		INC	LE1 ACD+1
2304	13CB	C6 E9		DEC	ACD+1
2304 2305	13CB 13CD	C6 E9 E6 E8		DEC INC	ACD+1 ACD
2304 2305 2306	13CB	C6 E9		DEC	ACD+1
2304 2305 2306 2307	13CB 13CD 13CF	C6 E9 E6 E8 4C 13BA		DEC INC JMP	ACD+1 ACD L13BA
2304 2305 2306	13CB 13CD	C6 E9 E6 E8 4C 13BA	SEPTAB:	DEC INC JMP	ACD+1 ACD
2304 2305 2306 2307	13CB 13CD 13CF	C6 E9 E6 E8 4C 13BA	SEPTAB:	DEC INC JMP	ACD+1 ACD L13BA
2304 2305 2306 2307 2308	13CB 13CD 13CF	C6 E9 E6 E8 4C 13BA	SEPTAB:	DEC INC JMP	ACD+1 ACD L13BA
2304 2305 2306 2307 2308 2309 2310	13CB 13CD 13CF 13D2 13D5	C6 E9 E6 E8 4C 13BA 20 2E 2C 3F OD OA	SEPTAB:	DEC INC JMP	ACD+1 ACD L13BA
2304 2305 2306 2307 2308 2309 2310 2311	13CB 13CD 13CF 13D2 13D5 13D8	C6 E9 E6 E8 4C 13BA 20 2E 2C 3F OD OA 09 OC		DEC INC JMP DB	ACD+1 ACD L13BA '.,?',CRCHAR,LFCHAR,TBCHAR,FFCHAR
2304 2305 2306 2307 2308 2309 2310 2311 2312	13CB 13CD 13CF 13D2 13D5	C6 E9 E6 E8 4C 13BA 20 2E 2C 3F OD OA	L13DA:	DEC INC JMP DB	ACD+1 ACD L13BA
2304 2305 2306 2307 2308 2309 2310 2311 2312 2313	13CB 13CD 13CF 13D2 13D5 13D8	C6 E9 E6 E8 4C 13BA 20 2E 2C 3F 0D 0A 09 0C 20 13F1	L13DA:	DEC INC JMP DB	ACD+1 ACD L13BA '.,?',CRCHAR,LFCHAR,TBCHAR,FFCHAR L13F1
2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314	13CB 13CD 13CF 13D2 13D5 13D8 13DA	C6 E9 E6 E8 4C 13BA 20 2E 2C 3F 0D 0A 09 0C 20 13F1 A0 00	L13DA:	DEC INC JMP DB JSR RTSCS LDY	ACD+1 ACD L13BA '.,?',CRCHAR,LFCHAR,TBCHAR,FFCHAR L13F1 #\$00
2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315	13CB 13CD 13CF 13D2 13D5 13D8 13DA 13E0 13E2	C6 E9 E6 E8 4C 13BA 20 2E 2C 3F 0D 0A 09 0C 20 13F1	L13DA: + L13EO:	DEC INC JMP DB JSR RTSCS LDY LDX	ACD+1 ACD L13BA '.,?',CRCHAR,LFCHAR,TBCHAR,FFCHAR L13F1 #\$00 #\$08
2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316	13CB 13CD 13CF 13D2 13D5 13D8 13DA 13E0 13E2 13E4	C6 E9 E6 E8 4C 13BA 20 2E 2C 3F 0D 0A 09 0C 20 13F1 A0 00 A2 08	L13DA:	DEC INC JMP DB JSR RTSCS LDY LDX CMPBE	ACD+1 ACD L13BA '.,?',CRCHAR,LFCHAR,TBCHAR,FFCHAR L13F1 #\$00
2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317	13CB 13CD 13CF 13D2 13D5 13D8 13DA 13E0 13E2	C6 E9 E6 E8 4C 13BA 20 2E 2C 3F 0D 0A 09 0C 20 13F1 A0 00	L13DA: + L13EO: +L13E4:	DEC INC JMP DB JSR RTSCS LDY LDX CMPBE INY	ACD+1 ACD L13BA '.,?',CRCHAR,LFCHAR,TBCHAR,FFCHAR L13F1 #\$00 #\$08 <septab,y>,L13EF</septab,y>
2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316	13CB 13CD 13CF 13D2 13D5 13D8 13DA 13E0 13E2 13E4	C6 E9 E6 E8 4C 13BA 20 2E 2C 3F 0D 0A 09 0C 20 13F1 A0 00 A2 08 C8	L13DA: + L13EO: +L13E4:	DEC INC JMP DB JSR RTSCS LDY LDX CMPBE INY DXBNE	ACD+1 ACD L13BA '.,?',CRCHAR,LFCHAR,TBCHAR,FFCHAR L13F1 #\$00 #\$08
2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317	13CB 13CD 13CF 13D2 13D5 13D8 13DA 13E0 13E2 13E4	C6 E9 E6 E8 4C 13BA 20 2E 2C 3F 0D 0A 09 0C 20 13F1 A0 00 A2 08 C8 18	L13DA: + L13EO: +L13E4:	DEC INC JMP DB JSR RTSCS LDY LDX CMPBE INY DXBNE CLC	ACD+1 ACD L13BA '.,?',CRCHAR,LFCHAR,TBCHAR,FFCHAR L13F1 #\$00 #\$08 <septab,y>,L13EF</septab,y>
2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318	13CB 13CD 13CF 13D2 13D5 13D8 13DA 13E0 13E2 13E4 13E9	C6 E9 E6 E8 4C 13BA 20 2E 2C 3F 0D 0A 09 0C 20 13F1 A0 00 A2 08 C8	L13DA: + L13EO: +L13E4:	DEC INC JMP DB JSR RTSCS LDY LDX CMPBE INY DXBNE	ACD+1 ACD L13BA '.,?',CRCHAR,LFCHAR,TBCHAR,FFCHAR L13F1 #\$00 #\$08 <septab,y>,L13EF</septab,y>
2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319	13CB 13CD 13CF 13D2 13D5 13D8 13DA 13E0 13E2 13E4 13E9	C6 E9 E6 E8 4C 13BA 20 2E 2C 3F 0D 0A 09 0C 20 13F1 A0 00 A2 08 C8 18	L13DA: + L13EO: +L13E4:	DEC INC JMP DB JSR RTSCS LDY LDX CMPBE INY DXBNE CLC	ACD+1 ACD L13BA '.,?',CRCHAR,LFCHAR,TBCHAR,FFCHAR L13F1 #\$00 #\$08 <septab,y>,L13EF</septab,y>
2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321	13CB 13CD 13CF 13D2 13D5 13D8 13DA 13E0 13E2 13E4 13E9 13ED 13EE 13EF	C6 E9 E6 E8 4C 13BA 20 2E 2C 3F 0D 0A 09 0C 20 13F1 A0 00 A2 08 C8 18 60 38	L13DA: + L13EO: +L13E4: + L13ED: L13EF:	DEC INC JMP DB JSR RTSCS LDY LDX CMPBE INY DXBNE CLC RTS SEC	ACD+1 ACD L13BA '.,?',CRCHAR,LFCHAR,TBCHAR,FFCHAR L13F1 #\$00 #\$08 <septab,y>,L13EF</septab,y>
2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321 2322	13CB 13CD 13CF 13D2 13D5 13D8 13DA 13E0 13E2 13E4 13E9 13ED 13EE	C6 E9 E6 E8 4C 13BA 20 2E 2C 3F 0D 0A 09 0C 20 13F1 A0 00 A2 08 C8 18 60	L13DA: + L13EO: +L13E4: + L13ED:	DEC INC JMP DB JSR RTSCS LDY LDX CMPBE INY DXBNE CLC RTS	ACD+1 ACD L13BA '.,?',CRCHAR,LFCHAR,TBCHAR,FFCHAR L13F1 #\$00 #\$08 <septab,y>,L13EF</septab,y>
2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321	13CB 13CD 13CF 13D2 13D5 13D8 13DA 13E0 13E2 13E4 13E9 13ED 13EE 13EF	C6 E9 E6 E8 4C 13BA 20 2E 2C 3F 0D 0A 09 0C 20 13F1 A0 00 A2 08 C8 18 60 38	L13DA: + L13EO: +L13E4: + L13ED: L13EF:	DEC INC JMP DB JSR RTSCS LDY LDX CMPBE INY DXBNE CLC RTS SEC	ACD+1 ACD L13BA '.,?',CRCHAR,LFCHAR,TBCHAR,FFCHAR L13F1 #\$00 #\$08 <septab,y>,L13EF</septab,y>

2325 2326 2327 2328 2329 2330 2331 2332	13F2 13F5 13F7 13F9 13FA 13FB 13FD	20 1406 A0 00 B1 E6 AA 68 F0 F0	L13FB:	JSR LDY LDA TAX PLA BEQ INY CMPBE	GTVCBA #\$00 (ACC),Y L13ED <(ACC),Y>,L13EF
2333 2334 2335	1402 1403	CA 4C 13FB		DEX JMP	L13FB
2336 2337	1406	AO 08	GTVCBA:	MOV	#HDRVCB <(FRZMEM),Y>,ACC+1
2338 2339 2340	140C	C8	+ +	INY MOV DADD	<(FRZMEM),Y>,ACC ACC,FRZMEM,ACC
2341 2342 2343	141E 141F	60 20 1406	L141F:	RTS JSR	GTVCBA
2344 2345 2346	1422 1424 1426	AO OO B1 E6 A8	21411.	LDY LDA TAY	#\$00 (ACC),Y
2347 2348 2349	1427 1428 142A	C8 B1 E6		INY LDA ASL	(ACC),Y
2350 2351 2352	142B 142C 142D	OA OA OA		ASL ASL ASL	A A A
2353 2354 2355	142E 1430	85 E8 C8	+	STA INY MOV	ACD <(ACC),Y>,ACB+1
2356 2357	1435	C8	~+	INY MOV	<(ACC), Y>, ACB
2358 2359 2360	143A 143B	C8 98	+	INY TYA ADD	,ACC,ACC
2361 2362 2363	1441 1443 1445	90 02 E6 E7 A0 00	L1445:	BCC INC LDY	L1445 ACC+1 #\$00
2364 2365 2366	1447 144A	4C 1450 B1 E6	L144A:	JMP LDA CMPBG	L1450 (ACC),Y PKWORD+1,L1470
2367 2368 2369	1450		+L1450: +	DADDB 1 DSUBB 1	ACC,ACD,ACC ACB,<#\$10>,ACB
2370 2371 2372 2373 2374	1466 1468 146A 146C 146E	A5 E5 30 O6 D0 DE A5 E4 D0 DA		LDA BMI BNE LDA BNE	ACB+1 L1470 L144A ACB L144A
2375 2376 2377 2378 2379 2380	1470 1486 1488 1489 148A	A5 E8 4A 4A 4A	+L1470: +	DSUBB1 DADDB1 LDA LSR LSR LSR	ACC,ACD,ACC ACB,<#\$10>,ACB ACD A A A

Infocom	ERLOG	IC interpret	er disassem	bly, 5/2	7/84 MACRO-80 3	18-Sep-81	PAGE
0001	148B	4A		LSR	A		
2381	148C	85 E8		STA	ACD		
2382	148E	AO 00	L148E:	LDY	#\$00		
2383	1490	A5 DB	_,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	LDA	PKWORD+1		
2384	1490	AS DD	+	CMPBL	<(ACC),Y>,L14D0		
2385	1.406	DO 1C		BNE	L14B4		
2386	1496	C8		INY			
2387	1498	A5 DA		LDA	PKWORD		
2388	1499	AS DA	+	CMPBL	<(ACC),Y>,L14D0		
2389	149F	DO 13		BNE	L14B4		
2390	1497	AO 02		LDY	#\$02		
2391		A5 DD		LDA	PKWORD+3		
2392	14A3	43 DD	+	CMPBL	<(ACC),Y>,L14D0		
2393	1440	DO 09		BNE	L14B4		
2394	1449	C8		INY			
2395	14AB	A5 DC		LDA	PKWORD+2		
2396	14AC	AS DC	+	CMPBL	<(ACC),Y>,L14D0		
2397	1.40.0	F0 23	·	BEQ	L14D7		
2398	14B2	FU 23	+L14B4;	DADDB 1	ACC,ACD,ACC		
2399	14B4		+	DDEC	ACB		
2400			+	DTS2BN	ACB,L148E		
2401	1.400		+L14D0:	MOV	<#\$00>, <acb+1,acb></acb+1,acb>		
2402	14D0	60	. [1 7 5 0 .	RTS	,		
2403	14D6	60	+L14D7:	DSUB	ACC, FRZMEM, ACB		
2404	14D7	60		RTS			
2405	14E4	90					
2406				PAGE			
2407							

2408 2409						
2410 2411			; print	ASCII c	haracter ARG1	
2412	14E5	A5 82	OPPRCH:		ARG1	
2413	14E7	4C 1B3F		JMP	BFCHAR	
2414 2415						-
2415			· orint	decimal	number ARG1	•
2417			, p	ace ma	mainbo. And.	
2418	14EA		+OPPRNM:	DMOV	ARG1,ACC	
2419	14F2	4C 14F5		JMP	PRNTNM	; unnecessary
2420						
2421						
2422			; print	decimal	number in ACC	
2423			BO.17.114			
2424	14F5	A5 E7	PRNTNM:	JSRMI	ACC+1 L152E	; negative? ; yes, print '-' and negate
2425			+	MOA	<#\$00>,ACD	; yes, print - and negate ; initialize digit count to 0
2426 2427	1500		+L1500:	DTSTBE	ACC,L1519	; if the remainder is zero, print it now
2427	1500		+	DMOVI	\$000A,ACB	; set up divisor of 10
2429	150E	20 15AD	•	JSR	DIVIDE	: divide
2430	1502	20 10/10	+	PSH	ACB	: push remainder onto stack
2431	1514	E6 E8		INC	ACD	; increment digit count
2432	1516	4C 1500		JMP	L1500	; do it again
2433						
2434	1519	A5 E8	L1519:	LDA	ACD	; is digit count zero?
2435	151B	FO OC		BEQ	L1529	; yes, just print a '0' and return
2436	151D	68	L151D:	PLA		; pull a digit off stack
2437			+	ADD	,<#'0'>	; convert to ASCII
2438	1521	20 1B3F		JSR	BFCHAR	; print it
2439	*500	60	+	DECBN RTS	ACD,L151D	; decrement digit count, loop if more : return to caller
2440 2441	1528	60		KIS		; return to carrer
2441	1529	A9 30	L1529:	LDA	#'O'	; get code for 'O'
2443	1528	4C 1B3F	L1323.	JMP	BFCHAR	; print it and return to caller
2444	1320	40 1501		J	Di di iiti	, print it and rotain to darior
2445	152E	A9 2D	L152E:	LDA	#'-'	; get code for '-'
2446	1530	20 1B3F		JSR	BFCHAR	; print it
2447	1533	4C 1611		JMP	L1611	; negate the number, return
2448						
2449				PAGE		

; save range ; get the "random" number

; get the remainder ; increment it (base of result is 1)

; divide by range

; and store it

2450					number from 1 to ARG1	
2451			; get a	random	Number 110m 1 to Akar	
2452					DNCDDC	
2453				IFF	RNGDBG	
2454				544017	ADC1 ACD	
2455	1536		+OPRNDM:	DMOV	ARG1,AČB	
2456	153E	20 21AO		JSR	L21AO	
2457	1541	20 15AD		JSR	DIVIDE	
2458			+	DMOV	ACB, ACC	
2459			+	DINC	ACC	
2460	1552	4C OB32		JMP	PTVRP1	
2461						
2462				ENDIF		
2463						
2464			; push	ARG1 on	stack	
2465						
2466	1555		+OPPUSH:		ARG1,ACC	
2467	155D	4C 16F4		JMP	PUSHWD	
2468						
2469						
2470			; pull	stack ir	nto variable ARG1	
2471						
2472	1560	20 1720	OPPULL:	JSR	PULLWD	
2473	1563	A5 82		LDA	ARG1	
2474	1565	4C OFA1		JMP	LOFA1	
2475 .						
2476						
2477	1568		+L1568:	DPSH	ACD	
2478			+	DMOVI	\$0000,ACD	
2479	1576	A2 10		LDX	#\$10	
2480	1578	A5 E4	L1578:	LDA	ACB	
2481	157A	18		CLC		
2482	157B	29 01		AND	#\$ 01	
2483	157D	F0 0C		BEQ	L158B	
2484	1315		+	DADC	ACC, ACD, ACD	
2485	158B		+L158B:	DROR	ACD	
2486	1300		+	DROR	ACB	
2487			+	DXBNE	L1578	
2488			+	DMOV	ACB, ACC	
2489			+	DMOV	ACD, ACB	
2489			+	DPUL	ACD	
	15AC	60	•	RTS		
2491	IDAC	UU				
2492				PAGE		
2493				. AGE		

```
2494
                                  ; divide ACC by ACB, quotient to ACC, remainder to ACB
2495
2496
                                +DIVIDE: DPSH
                                                   ACD
          15AD
2497
                                          DMOV
                                                   ACC, ACD
2498
                                          DMOVI
                                                   $0000,ACC
2499
                  A2 11
                                          LDX
                                                   #$11
2500
          15C3
                                 L15C5: SEC
                  38
2501
         15C5
                                                   ACC
                                          LDA
2502
         15C6
                  A5 E6
                                                   ACB
                  E5 E4
                                          SBC
2503
          15C8
                  A8
                                          TAY
2504
          15CA
                                          LDA
                                                   ACC+1
                  A5 E7
2505
          15CB
                                          SBC
                                                   ACB+1
          15CD
                  E5 E5
2506
                                          BCC
                                                   L15D6
2507
          15CF
                  90 05
                                                   ACC+1
2508
          15D1
                  85 E7
                                          STA
         15D3
                  98
                                          TYA
2509
                                                   ACC
                  85 E6
                                          STA
          15D4
2510
                                                   ACD
                                +L15D6:
                                          DROL
2511
          15D6
                                          DROL
                                                   ACC
2512
                                          DXBNE
                                                   L15C5
2513
                                          CLC
2514
          15E1
                   18
                                                   ACC, ACB
                                          DROR
2515
                                                   ACD, ACC
                                          DMOV
2516
                                                   ACD
                                          DPUL
2517
                                          RTS
2518
          15FA
                  60
2519
2520
                                 +L15FB:
                                          MOV
                                                   <#$00>, MDFLAG
          15FB
                                          LDA
                                                   ACC+1
2521
          15FF
                  A5 E7
                                                   L161F
                  20 161F
                                          JSR
2522
          1601
                                          LDA
                                                   ACB+1
2523
          1604
                  A5 E5
                                                   L161F
2524
          1606
                  20 161F
                                          JSR
                                          RTS
          1609
                  60
2525
2526
                                  L160A: LDA
                                                   MDFLAG
2527
          160A
                   A5 EA
                                                   #$01
                   29 01
                                          AND
2528
          160C
                                          RTSEQ
2529
                                  L1611: SEC
2530
          1611
                   38
                                                   #$00
                                          LDA
2531
          1612
                   A9 00
          1614
                   E5 E6
                                           SBC
                                                   ACC
2532
                                           STA
                                                   ACC
2533
          1616
                   85 E6
                                          LDA
                                                   #$00
2534
          1618
                   A9 00
                                                   ACC+1
                   E5 E7
                                           SBC
2535
          161A
                                                   ACC+1
                   85 E7
                                          STA
2536
          161C
                                          RTS
2537
          161E
                   60
2538
                                                                              ; if positive, return
          161F
                                 +L161F:
                                          TSTARP
2539
                                                   MDFLAG
2540
          1624
                   E6 EA
                                           INC
                   4C 1611
                                           JMP
                                                   L1611
2541
          1626
2542
                                           PAGE
2543
```

2544			. satur	stuff f	or thing attribute bit operations
2545 2546			; setup	Stair i	di tilling atti ibate bit operations
2547	1629	A5 82	SETUPA:	LDA	ARG1
2548	162B	20 16A7		JSR	SETUPT
2549	162E	A5 84		LDA	ARG2
2550	1022	A5 54		CMPBL	<#\$10>,L1643
2551				SUB	,<#\$10>
2552				DINC	ACC
2553				DINC	ACC
2554	1643	85 E4		STA	ACB
2555	1043	00 L-1		DMOVI	\$0001,ACD
2556				SUB	<#\$0F>,ACB
2557	1652	AA		TAX	,
2558	1653	F0 08		BEQ	L165D
2559	1033	10 00		DASL	ACD
2560	1659	CA		DEX	
2561	165A	4C 1653		JMP	L1653
2562	165D	AO 00		LDY	#\$00
2563	1000	A0 00		MOV	<(ACC),Y>,ACB+1
2564	1663	C8		INY	(1100)
2565	1000	CO		MOV	<(ACC),Y>,ACB
2566	1668	60		RTS	(
2567	1000	00		5	
2568					
2569			· setun	stuff f	for thing property operations
2570			, ootop		or annual by ober 1) the same of
2571	1669	A5 82	SETUPP:	I.DA	ARG1
2572	166B	20 16A7		JSR	SETUPT
2573	166E	AO 07		LDY	#THGPRP
2574	1002	NO 07		MOV	<(ACC),Y>,ACB+1
2575	1674	C8		INY	, ,
2576		••	+	MOV	<(ACC),Y>,ACB
2577				DADD	ACB, FRZMEM, ACC
2578	1686	AO OO		LDY	#\$ 00
2579	1688	B1 E6		LDA	(ACC),Y
2580	168A	OA OA		ASL	À
2581	168B	A8		TAY	
2582	168C	C8		INY	
2583	168D	60		RTS	
2584					
2585					
2586			; get nu	umber of	f property pointed to by ACC
2587					
2588	168E	B1 E6	GTPNUM:	LDA	(ACC),Y
2589	1690	29 1F		AND	#\$1F
2590	1692	60		RTS	
2591					
2592					
2593			; get le	enght of	f property pointed to by ACC
2594					
2595	1693	B1 E6	GTPLEN:	LDA	(ACC),Y
2596			+	REPT	5
2597			+	ROR	A
2598			+	ENDM	

2599	169A	29 07	AND #\$07
2600	169C	60	RTS
2601			
2602			
2603			; advance ACC to point to next property
2604			
2605	169D	20 1693	ADVPPT: JSR GTPLEN
2606	16A0	AA	TAX
2607	16A1	C8	L16A1: INY
2608			+ DXBPL L16A1
2609	16A5	CB	INY
2610	16A6	60	RTS
2611			
2612			
2613			; setup stuff for thing operations
2614			200
2615	16A7	85 E6	SETUPT: STA ACC
2616			+ MOV <#\$00>,ACC+1
2617	16AD	A5 E6	LDA ACC
2618			+ REPT 3
2619			+ DASL ACC
2620			+ ENDM
2621			+ ADD ,ACC BCC L16C3
2622	16BE	90 03	INC ACC+1
2623	16C0	E6 E7	CLC
2624	1602	18	L16C3: ADC #\$35
2625	16C3	69 35	STA ACC
2626	1605	85 E6	BCC L16CB
2627	16C7	90 02	INC ACC+1
2628	16C9	E6 E7	L16CB: LDY #HDRTHG+1
2629	16CB	AO 0B	+ ADD <(FRZMEM),Y>,ACC,ACC
2630	1604	88	DEY
2631 2632	16D4 16D5	B1 BA	LDA (FRZMEM),Y
2632	16D7	65 E7	ADC ACC+1
2633 2634	16D7	65 BB	ADC FRZMEM+1
263 4 2635	16DB	85 E7	STA ACC+1
2636	16DD	60	RTS
2637	1000	50	
2638			PAGE
2000			

Infocom 1...ERLOGIC interpreter disassembly, 5/27/84

PAGE

MACRO-80 3.43 18-Sep-81	PAGE	1-6
-------------------------	------	-----

2639					
2640	16DE	A5 E5	L16DE:	LDA	ACB+1
2641	16E0	45 E7		EOR	ACC+1
2642	16E2	10 05		BPL	L16E9
2643	16E4	A5 E5		LDA	ACB+1
2644	16E6	C5 E7		CMP	ACC+1
2645	16E8	60		RTS	
2646	16E9	A5 E7	L16E9:	LDA	ACC+1
2647			+	CMPBN	ACB+1,L16F3
2648	16EF	A5 E6		LDA	ACC
2649	16F1	C5 E4		CMP	ACB
2650	16F3	60	L16F3:	RTS	
2651					
2652	√16F4		+PUSHWD:	DDEC	STKPNT
2653	16FF	AO 00		LDY	#\$ 00
2654			+	MOV	ACC, << (STKPNT), Y>>
2655			+	DDEC	STKPNT
2656			÷	MO∨	ACC+1,<<(STKPNT),Y>>
2657	1714	E6 C8		INC	STKCNT
2658	1716	A5 C8		LDA	STKCNT
2659			+	CMPJSG	<#STCKMX>,FATAL
2660	171F	60		RTS	
2661					
2662	1720	AO 00	PULLWD:	LDY	#\$ 00
2663			+	MOV	<(STKPNT),Y>,ACC+1
2664			+	DINC	STKPNT
2665			+	MOV	<(STKPNT),Y>,ACC
2666			+	DINC	STKPNT
2667	1736	C6 C8		DEC	STKCNT
2668			+	JSREQ	FATAL
2669	173D	60		RTS	
2670					
2671				PAGE	

Infocom ... ERLOGIC interpreter disassembly, 5/27/84

2672				fatab		to from DC into A		
2673				; retth	next by	te from PC into A		
2674			2-	CTODDA		DOCUDD		and to find a new page?
2675	173E		8F	FTPRBA:		PRGUPD	•	need to find a new page?
2676	1740	FO	15		BEQ	L1757	;	yes, go do it!
2677								
2678	1742	A4	8A		LDY	PRGIDX	;	get the byte
2679	1744	В1	8D		LDA	(PRGMPT),Y		
2680								
2681	1746	C8			INY		;	increment the low byte of the PC
2682	1747	84	8A		STY	PRGIDX		
2683				+	RTSNE		;	return unless we've entered a new page
2684								
2685	174C	ΑO	00		LDY	#\$0 0	;	unnecessary!
2686	174E		8F		STY	PRGUPD	;	indicate new page
2687			-	+	DINC	PRGLPG	:	increment page number
2688	1756	60			RTS			return
2689	1730						•	
2690	1757	۸5	8C	L1757:	LDA	PRGLPG+1	:	is the page we're looking for frozen?
2690	1757		06	2,737.	BNE	L1761	,	The time page are to training that the training
	175B		8B		LDA	PRGLPG		
2692	1/50	CA	80	+	CMPBL	FRZPGS,L1778		
2693				т	CMPDL	TRZFG3,E1776		
2694	. 70.			1701	DMOV	DDCI DC ACC		no, see if it is swapped in
2695	1761			+L1761:	DMOV	PRGLPG, ACC	,	no, see ii it is swapped iii
2696	1769		1897		JSR	FNDPAG		save abus page as
2697 .	176C		90		STA	PRGPPG		save phys. page no.
2698	176E	В0	18		BCS	L1788	;	not found
2699								
2700				; we ha	ve the s	swappable page, fix up	the p	pointers, etc.
2701								
2702	1770	20	1862	L1770:	JSR	MRKPAG	;	indicate that we're using this page
2703								
2704	1773	18			CLC			add phys. page number to number
2705	1774	A5	90		LDA	PRGPPG	;	of frozen pages
2706	1776	65	BC		ADC	FRZPGS		
2707								
2708				; fix t	he memo	ry pointers		
2709								
2710	1778			+L1778:	ADD	,FRZMEM+1,PRGMPT+1	;	add base of frozen memory
2711				+	MO∨	<#\$00>,PRGMPT		
2712								
2713				+	MOV	<#\$FF>.PRGUPD	;	indicate that we have the page
2714	1785	40	173E		JMP	FTPRBA	:	and go get the byte
2715	1705		., 02		•		•	,
2716				· we ne	ed to 1	oad the page from disk		
2717				, 46 110	CG CG I	oud the page tham area		
2718	1788			+L1788:	CMPBN	AUXPPG,L1790		if we are about to load a new logical
	1700			+	MOV	<#\$00>,AUXUPD		page into the physical page AUX points
2719				T	WO V	·#WOO', NONOFD		to, mark it as new page
2720							•	to, mark it as non page
2721	. 700			1700	014014	CHIDMEN ACC		setup to read the page
2722	1790			+L1790:	DMOV	SWPMEM, ACC	;	secup to read the page
2723				7 4	ADD	PRGPPG,ACC+1,ACC+1		
2724				· +	DMOV	PRGLPG, ACB		
2725				•	100	6000KE		and the same (die if cases)
2726	17A7	20	1E0D		JSR	DRDBKF	;	read the page (die if error)

Infocom .	ERLOGI	C interpreter	disassembly,	5/27/84	MACRO-80 3	18-Sep-81	PAGE	1-66
2727 2728 2729 2730	17AA	A4 90	+ MOV + MOV		<<(VMTAB1),Y>> ,<<(VMTAB2),Y>>	; copy the new ; the VM table	log. page	e number into
2731 2732 2733 2734 2735	1784 1785	98 4C 1770	TYA JMP PAG	L1770 E		; go fix up the	pointers	s and fetch the byte

Infocom N. ERLOGIC interpreter disassembly, 5/27/84

MACRO-80 3.

PAGE

18-Sep-81

2770			C - 1 - 1		to form Ally into A	
2771			; retci	next by	te from AUX into A	
2772	4750	A5 96	FTAXBA	LDA	AUXUPD	; need to find a new page?
2773	17E8	FO 15		BEQ	L1801	; yes, go do it!
2774	17EA	FU 15	•	DLQ	L1001	, , , , , , , , , , , , , , , , , , , ,
2775	1750	A4 93	•	LDY	AUXIDX	; get the byte
2776	17EC	B1 94		LDA	(AUXMPT),Y	, 301 1 2,10
2777	17EE	B1 94	•	LDA	(AOAMI I), I	
2778	17F0	СВ		INY	,	; increment the low byte of AUX
2779	17F0 17F1	84 93	•	STY	AUXIDX	•
2780	1761	64 93	, +	RTSNE	NONZEN	; return unless we've entered a new page
2781			·	KISHE		,
2782 2783	17F6	AO 00)	LDY	#\$00	; unnecessary!
2783	17F8	84 96		STY	AUXUPD	; indicate new page
	1756	64 50	, +	DINC	AUXLPG	; increment page number
2785 2786	1800	60	·	RTS	, none: a	return
2787	1800	00				•
2788	1801	A5 92	L1801:	LDA	AUXLPG+1	; is the page we're looking for frozen?
2789	1803	DO 06		BNE	L180B	,
2790	1805	A5 91		LDA	AUXLPG	
2790	1807	A3 31	+L1807:		FRZPGS,L1822	
2792	1807		21007.	· · · · · ·		
2793	180B		+L180B:	DMOV	AUXLPG, ACC	; no, see if it is swapped in
2794	1813	20 18		JSR	FNDPAG	•
2795	1816	85 97		STA	AUXPPG	; save phys. page no.
2796	1818	BO 18		BCS	L1832	; not fount
	1010	50 10	•			•
2797			: we h	ave the	swappable page, fix up 1	the pointers, etc.
2797 2798			; we h	ave the	swappable page, fix up f	
2797 2798 2799	1814	20 18			swappable page, fix up f	the pointers, etc. ; indicate that we're using this page
2797 2798 2799 2800	181A	20 18				
2797 2798 2799 2800 2801		20 18 18				; indicate that we're using this page ; add phys. page number to number of
2797 2798 2799 2800 2801 2802	181D		362 L181A:	JSR		; indicate that we're using this page
2797 2798 2799 2800 2801 2802 2803		18	362 L181A:	JSR CLC	MRKPAG	; indicate that we're using this page ; add phys. page number to number of
2797 2798 2799 2800 2801 2802 2803 2804	181D 181E	18 A5 97	362 L181A:	JSR CLC LDA	MRKPAG AUXPPG	; indicate that we're using this page ; add phys. page number to number of
2797 2798 2799 2800 2801 2802 2803	181D 181E	18 A5 97	362 L181A:	JSR CLC LDA ADC	MRKPAG AUXPPG	; indicate that we're using this page ; add phys. page number to number of
2797 2798 2799 2800 2801 2802 2803 2804 2805	181D 181E	18 A5 97	362 L181A:	JSR CLC LDA ADC	MRKPAG AUXPPG FRZPGS	; indicate that we're using this page ; add phys. page number to number of ; frozen pages
2797 2798 2799 2800 2801 2802 2803 2804 2805 2806	181D 181E	18 A5 97	362 L181A:	JSR CLC LDA ADC the memo	MRKPAG AUXPPG FRZPGS ry pointers ,FRZMEM+1,AUXMPT+1	; indicate that we're using this page ; add phys. page number to number of
2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807	181D 181E 1820	18 A5 97	362 L181A:	JSR CLC LDA ADC the memo	MRKPAG AUXPPG FRZPGS ry pointers	; indicate that we're using this page ; add phys. page number to number of ; frozen pages
2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808	181D 181E 1820	18 A5 97	362 L181A: 7 C ; fix +L1822:	JSR CLC LDA ADC the memo ADD MOV	MRKPAG AUXPPG FRZPGS ry pointers ,FRZMEM+1,AUXMPT+1 <#\$00>,AUXMPT	; indicate that we're using this page ; add phys. page number to number of ; frozen pages ; add base of memory
2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809	181D 181E 1820	18 A5 97	362 L181A: 7 C ; fix +L1822:	JSR CLC LDA ADC the memo ADD MOV MOV	MRKPAG AUXPPG FRZPGS ry pointers ,FRZMEM+1,AUXMPT+1 <#\$00>,AUXMPT <#\$FF>,AUXUPD	<pre>; indicate that we're using this page ; add phys. page number to number of ; frozen pages ; add base of memory ; indicate that we have the page</pre>
2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809 2810	181D 181E 1820	18 A5 97	362 L181A: 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	JSR CLC LDA ADC the memo ADD MOV	MRKPAG AUXPPG FRZPGS ry pointers ,FRZMEM+1,AUXMPT+1 <#\$00>,AUXMPT	; indicate that we're using this page ; add phys. page number to number of ; frozen pages ; add base of memory
2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809 2810 2811	181D 181E 1820	18 A5 97 65 B0	362 L181A: 7 ; fix +L1822: +	JSR CLC LDA ADC the memo ADD MOV MOV JMP	MRKPAG AUXPPG FRZPGS ry pointers ,FRZMEM+1,AUXMPT+1 <#\$00>,AUXMPT <#\$FF>,AUXUPD FTAXBA	<pre>; indicate that we're using this page ; add phys. page number to number of ; frozen pages ; add base of memory ; indicate that we have the page</pre>
2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809 2810 2811 2812	181D 181E 1820	18 A5 97 65 B0	362 L181A: 7 ; fix +L1822: +	JSR CLC LDA ADC the memo ADD MOV MOV JMP	MRKPAG AUXPPG FRZPGS ry pointers ,FRZMEM+1,AUXMPT+1 <#\$00>,AUXMPT <#\$FF>,AUXUPD	<pre>; indicate that we're using this page ; add phys. page number to number of ; frozen pages ; add base of memory ; indicate that we have the page</pre>
2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809 2810 2811 2811 2813	181D 181E 1820	18 A5 97 65 B0	362 L181A: 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 8 7 8 8 7 8 8 9 1 8 1 8 1 8 1 8 1 8 1 8 1 8 1 8 1	JSR CLC LDA ADC the memo ADD MOV JMP eed to 1	MRKPAG AUXPPG FRZPGS ry pointers ,FRZMEM+1,AUXMPT+1 <#\$00>,AUXMPT <#\$FF>,AUXUPD FTAXBA oad the page from disk	; indicate that we're using this page ; add phys. page number to number of ; frozen pages ; add base of memory ; indicate that we have the page ; and go get the byte
2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809 2810 2811 2812 2813 2814	181D 181E 1820	18 A5 97 65 B0	362 L181A: 7 7 7 7 7 7 7 7 7 7 7 7 7 7 8 8 9 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	JSR CLC LDA ADC the memo ADD MOV MOV JMP eed to 1 CMPBN	MRKPAG AUXPPG FRZPGS ry pointers ,FRZMEM+1,AUXMPT+1 <#\$00>,AUXMPT <#\$FF>,AUXUPD FTAXBA oad the page from disk PRGPPG,L183A	; indicate that we're using this page ; add phys. page number to number of ; frozen pages ; add base of memory ; indicate that we have the page ; and go get the byte ; if we are about to load a new logical
2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809 2810 2811 2812 2813 2814 2815	181D 181E 1820 1822	18 A5 97 65 B0	362 L181A: 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 8 7 8 8 7 8 8 9 1 8 1 8 1 8 1 8 1 8 1 8 1 8 1 8 1	JSR CLC LDA ADC the memo ADD MOV JMP eed to 1	MRKPAG AUXPPG FRZPGS ry pointers ,FRZMEM+1,AUXMPT+1 <#\$00>,AUXMPT <#\$FF>,AUXUPD FTAXBA oad the page from disk	; indicate that we're using this page ; add phys. page number to number of ; frozen pages ; add base of memory ; indicate that we have the page ; and go get the byte ; if we are about to load a new logical ; page into the physical page the PC
2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809 2810 2811 2812 2813 2814 2815 2816	181D 181E 1820 1822	18 A5 97 65 B0	362 L181A: 7 7 7 7 7 7 7 7 7 7 7 7 7 7 8 8 9 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	JSR CLC LDA ADC the memo ADD MOV MOV JMP eed to 1 CMPBN	MRKPAG AUXPPG FRZPGS ry pointers ,FRZMEM+1,AUXMPT+1 <#\$00>,AUXMPT <#\$FF>,AUXUPD FTAXBA oad the page from disk PRGPPG,L183A	; indicate that we're using this page ; add phys. page number to number of ; frozen pages ; add base of memory ; indicate that we have the page ; and go get the byte ; if we are about to load a new logical
2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809 2810 2811 2812 2813 2814 2815 2816 2817 2818 2819	181D 181E 1820 1822 182F	18 A5 97 65 B0	; fix +L1822: + 7E8 ; we n +L1832:	JSR CLC LDA ADC the memo ADD MOV JMP eed to 1 CMPBN MOV	MRKPAG AUXPPG FRZPGS ry pointers ,FRZMEM+1,AUXMPT+1 <#\$00>,AUXMPT <#\$FF>,AUXUPD FTAXBA oad the page from disk PRGPPG,L183A <#\$00>,PRGUPD	; indicate that we're using this page ; add phys. page number to number of ; frozen pages ; add base of memory ; indicate that we have the page ; and go get the byte ; if we are about to load a new logical ; page into the physical page the PC ; points to, mark it as a new page
2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809 2810 2811 2812 2813 2814 2815 2816 2817 2816 2817	181D 181E 1820 1822	18 A5 97 65 B0	362 L181A: 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 8 7 8 9 9 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	JSR CLC LDA ADC the memo ADD MOV MOV JMP eed to 1 CMPBN MOV DMOV	MRKPAG AUXPPG FRZPGS ry pointers ,FRZMEM+1,AUXMPT+1 <#\$00>,AUXMPT <#\$FF>,AUXUPD FTAXBA oad the page from disk PRGPPG,L183A <#\$00>,PRGUPD SWPMEM,ACC	; indicate that we're using this page ; add phys. page number to number of ; frozen pages ; add base of memory ; indicate that we have the page ; and go get the byte ; if we are about to load a new logical ; page into the physical page the PC
2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809 2810 2811 2812 2813 2814 2815 2816 2817 2818 2819 2819 2819 2819	181D 181E 1820 1822 182F	18 A5 97 65 B0	362 L181A: 7 7 7 7 7 7 7 7 7 7 7 7 7 7 8 7 8 8 9 9 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	JSR CLC LDA ADC the memo ADD MOV JMP eed to 1 CMPBN MOV ADD	MRKPAG AUXPPG FRZPGS ry pointers ,FRZMEM+1,AUXMPT+1 <#\$00>,AUXMPT <#\$FF>,AUXUPD FTAXBA oad the page from disk PRGPPG,L183A <#\$00>,PRGUPD SWPMEM,ACC AUXPPG,ACC+1,ACC+1	; indicate that we're using this page ; add phys. page number to number of ; frozen pages ; add base of memory ; indicate that we have the page ; and go get the byte ; if we are about to load a new logical ; page into the physical page the PC ; points to, mark it as a new page
2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809 2810 2811 2812 2813 2814 2815 2816 2817 2818 2819 2819 2820 2821 2821	181D 181E 1820 1822 182F	18 A5 97 65 B0	362 L181A: 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 8 7 8 9 9 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	JSR CLC LDA ADC the memo ADD MOV MOV JMP eed to 1 CMPBN MOV DMOV	MRKPAG AUXPPG FRZPGS ry pointers ,FRZMEM+1,AUXMPT+1 <#\$00>,AUXMPT <#\$FF>,AUXUPD FTAXBA oad the page from disk PRGPPG,L183A <#\$00>,PRGUPD SWPMEM,ACC	; indicate that we're using this page ; add phys. page number to number of ; frozen pages ; add base of memory ; indicate that we have the page ; and go get the byte ; if we are about to load a new logical ; page into the physical page the PC ; points to, mark it as a new page
2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809 2810 2811 2812 2813 2814 2815 2816 2817 2818 2819 2819 2819 2819	181D 181E 1820 1822 182F	18 A5 97 65 B0	; fix +L1822: + 7E8 ; we n +L1832: + +L183A: +	JSR CLC LDA ADC the memo ADD MOV JMP eed to 1 CMPBN MOV ADD	MRKPAG AUXPPG FRZPGS ry pointers ,FRZMEM+1,AUXMPT+1 <#\$00>,AUXMPT <#\$FF>,AUXUPD FTAXBA oad the page from disk PRGPPG,L183A <#\$00>,PRGUPD SWPMEM,ACC AUXPPG,ACC+1,ACC+1	; indicate that we're using this page ; add phys. page number to number of ; frozen pages ; add base of memory ; indicate that we have the page ; and go get the byte ; if we are about to load a new logical ; page into the physical page the PC ; points to, mark it as a new page

2825 2826 2827 2828	1854	A4 97	+ +	LDY MOV MOV	AUXPPG AUXLPG,<<(VMTAB1),Y>> AUXLPG+1,<<(VMTAB2),Y>>	;		ppy the new log. page number into ne VM table
2829 2830 2831 2832 2833	185E 185F	98 4C 181A		TYA JMP PAGE	L181A	÷	go	o fix up the pointers and fetch the byte

```
2834
2835
                                 ; we've just started using a new logical page, move it to front of our list
                                 ; this make least recently used pages first candidates to be removed
2836
2837
                                +MRKPAG: CMPBE
2838
         1862
                                                   MRUPAG, L1891
                  A6 BE
                                          LDX
                                                   MRUPAG
2839
         1866
                  85 BE
                                          ·STA
                                                   MRUPAG
2840
         1868
                                          TAY
2841
         186A
                  8A
                                          MOV
2842
                                                   <(VMTAB3),Y>,ACC
                  88
                                          TXA
2843
         186F
                  91 C4
                                          STA
                                                   (VMTAB3),Y
2844
         1870
                                          MOV
                                                   <(VMTAB4), Y>, ACC+1
2845
2846
                                          MOV
                                                   <#$FF>, << (VMTAB4), Y>>
2847
         187A
                  A4 E7
                                          LDY
                                                   ACC+1
                                          MOV
                                                   ACC, << (VMTAB3), Y>>
2848
                                          TXA
2849
         1880
                  88
2850
         1881
                  8A
                                          TAY
                                          MOV
                                                   MRUPAG, << (VMTAB4), Y>>
2851
                                          LDA
2852
         1886
                  A5 E6
                                                   ACC
                                                   <#$FF>,L1892
                                          CMPBE
2853
2854
                                          TAY
         188C
                  8A
2855
                                          MOV
                                                   ACC+1, << (VMTAB4), Y>>
                                 L1891:
                                          RTS
2856
         1891
                  60
                                +L1892:
2857
         1892
                                          MOV
                                                   ACC+1, LRUPAG
         1896
                                          RTS
2858
                  60
2859 .
2860
2861
                                 ; search virtual memory table for logical page # in ACC
2862
2863
         1897
                  A6 BD
                                 FNDPAG: LDX
                                                   SWPPGS
2864
         1899
                  A0 00
                                          LDY
                                                   #$00
2865
         189B
                  A5 E6
                                          LDA
                                                   ACC
                                +L189D:
                                          CMPBN
                                                   <(VMTAB1), Y>, L18A9
2866
         189D
2867
         18A1
                  A5 E7
                                          LDA
                                                   ACC+1
2868
                                          CMPBE
                                                   <(VMTAB2), Y>, L18B1
                                                   ACC
2869
                  A5 E6
                                          LDA
         18A7
                                 L18A9:
                                          INY
2870
         18A9
                  C8
                                          DXBNE
                                                   L189D
2871
2872
         18AD
                  A5 BF
                                          LDA
                                                   LRUPAG
2873
                                          SEC
         18AF
                  38
2874
         18B0
                  60
                                          RTS
2875
         1881
                  98
                                 L18B1:
                                          TYA
2876
         18B2
                  18
                                          CLC
                                          RTS
2877
         1883
                  60
2878
2879
                                          PAGE
```

2880					
2881			; print	string	at AUX
2882					
2883	1884		+PRNTST:	_	<#\$00>, <prmmod,pnybcn></prmmod,pnybcn>
2884			+	MOV	<#\$FF>,TMPMOD
2885	18BE	20 1989	DONEXT:		GETNYB
2886		0= 50	. +	RTSCS	100
2887	18C4	85 E8		STA	ACD
2888	18C6	FO 48		BEQ	DOSPAC
2889			+	CMPBL	<#\$04>,DOSBWD
2890			+	CMPBL	<#\$06>, NEWMOD
2891	18D0	20 19AD		JSR	TSTMOD
2892		==	+	TSTABN	L18E2
2893	18D7	A9 5B		LDA	#\$5B
2894	18D9	00 4005	+L18D9:	ADD	,ACD
2895	1BDC	20 1B3F	L18DC:	JSR	BECHAR
2896	18DF	4C 18BE		JMP	DONEXT
2897	18E2		+L18E2:	CMPBN	<#\$01>,DOSPCL
2898	18E6	A9 3B		LDA	#\$3B
2899	18E8	4C 18D9		JMP	L18D9
2900				0.10	100 1000
2901	18EB		+DOSPCL:		ACD, <#\$07>
2902	18F0	90 OA		BCC	DOASCI
2903	18F2	F0 21		BEQ	DOCRLF
2904	18F4	A8		TAY	
2905	18F5	88		DEY	6001 611 14
2906	18F6	B9 1995		LDA	SPCLCH, Y
2907	18F9	4C 18DC		JMP	L18DC
2908	1050	20 1000	DOASCI:	JSR	GETNYB
2909	18FC	20 1989	+ DOASCI:	REPT	5
2910			+	ASL	5 A
2911			+	ENDM	A
2912	1004	48	т	PHA	
2913 2914	1904 1905	20 19B9		JSR	GETNYB
				STA	ACD
2915	1908	85 E8		PLA	ACD
2916	190A	68		ORA	ACD
2917	190B	05 E8 4C 18DC		JMP	L 18DC
2918	1900	46 1806		JMP	LIODC
2919				PAGE	
2920				PAGE	

P	Δ	G	F	1	-7	2

Infocom In ERLOGIC interpreter disa	assembly, 5/2//	84
-------------------------------------	-----------------	----

MACRO-80	3 📜 💄
----------	-------

CR0-80	3	18-Sep-81
ICRO DO	J . ~. ~	10 3cp 0.

2921					
2922	1910	A9 20	DOSPAC:	LDA	#''
2923	1912	4C 18DC		JMP	L18DC
2924					
2925	1915	A9 OD	DOCRLF:	LDA	#CRCHAR
2926	1917	20 1B3F		JSR	BFCHAR
2927	191A	A9 OA		LDA	#LFCHAR
2928	191C	4C 18DC		JMP	L18DC
2929					
2930	191F		+NEWMOD:	SUB	,<#\$03>
2931	1922	A8		TAY	
2932	1923	20 19AD		JSR	TSTMOD
2933	1926	DO 05		BNE	L192D
2934	1928	84 CE		STY	TMPMOD
2935	192A	4C 18BE		JMP	DONEXT
2936	192D	84 CF	L192D:	STY	PRMMOD
2937			+	CMPBE	PRMMOD, L1937
2938	1933	AO 00		LDY	#\$00
2939	1935	84 CF		STY	PRMMOD
2940	1937	4C 18BE	L1937:	JMP	DONEXT
2941					
2942				PAGE	

2943 2944 2945	193A	00	L193A:	DB	\$00
2945 2946 2947 2948	193B		+DOSBWD: + +	REPT ASL	6 A
2949 2950	1944	8D 193A	+	ENDM STA	L193A GETNYB
2951	1947	20 19B9		JSR ASL	A
2952	194A	0A 69 01		ADC	#\$ 01
2953	194B	6D 193A		ADC	L193A
2954	194D 1950	A8		TAY	
2955 2956	1930	AU	+	MOV	<(SBWDPT),Y>,ACC
2957	1955	88		DEY	•
2958	1000		+	MOV	<(SBWDPT),Y>,ACC+1
2959			+	PSH	<prmmod, pnybcn=""></prmmod,>
2960			+	DPSH	PNYBBF
2961			+	PSH	AUXIDX
2962			+	DPSH	AUXLPG
2963	196F	20 1709		JSR	SETAXW
2964	1972	20 18B4		JSR	PRNTST
2965			+	DPUL	AUXLPG
2966			+	PUL	AUXIDX
2967			+	MOV	<#\$00>,AUXUPD
2968			+	DPUL	PNYBBF <pnybcn.prmmod></pnybcn.prmmod>
2969			+	PUL MOV	<pre><pre></pre></pre>
2970		40 1005	+	JMP	DONEXT
2971	1992	4C 18BE		JMP	DONEAT
2972	1005	30 31 32	SPCLCH:	DB	'0123456789.,!? <u>_</u> #''"/\-:()'
2973	1995 1998	33 34 35	SPULCIT.	56	0,20,00,001,711_
2974 2975	1998	36 37 38			
2975 2976	199E	39 2E 2C			
2977	19A1	21 3F 5F			
2978	19A4	23 27 22			
2979	19A7	2F 5C 2D			
2980	19AA	3A 28 29			
2981					
2982	19AD	A5 CE	TSTMOD:		TMPMOD
2983	19AF	10 03	•	BPL	L19B4
2984	19B1	A5 CF		LDA	PRMMOD
2985	19B3	60		RTS	
2986	19B4	AO FF	L19B4:	LDY	#\$FF
2987	19B6	84 CE		STY	TMPMOD
2988	19B8	60		RTS	
2989 2990				PAGE	

2991					
2992	1989	A5 D0	GETNYB:	LDA	PNYBCN
	19BB	10 02	acinib.	BPL	L19BF
2993		38		SEC	E 1351
2994	19BD	60		RTS	
2995	19BE		L19BF:	BNE	L19D6
2996	19BF	DO 15	LISEF:	INC	PNYBCN
2997	1901	E6 D0			FTAXWD
2998	19C3	20 17DB		JSR	ACC.PNYBBF
2999			+	DMOV	
3000	19CE	A5 D2		LDA	PNYBBF+1
3001	1900	4A	*	LSR	A
3002	19D1	44		LSR	Α
3003	19D2	29 1F		AND	#\$1F
3004	19D4	18		CLC	
3005	19D5	60		RTS	
3006	19D6		+L19D6:	DECABN	L19F3
3007			+	MOV	<#\$02>,PNYBCN
3008	19DF	A5 D2		LDA	PNYBBF+1
3009	19E1	4A		LSR	Α
3010	19E2	A5 D1		LDA	PNYBBF
3011	19E4	6A		ROR	Α
3012	19E5	8A		TAY	
3013	19E 6	A5 D2		LDA	PNYBBF+1
3014	19E8	4A		LSR	A
3015	19E9	4A		LSR	Α
3016	19EA	98		TYA	
3017	19EB	6A		ROR	Α
3018	19EC	4A		LSR	Α
3019	19ED	4A		LSR	Α
3020	19EE	4A		LSR	Α
3021	19EF	29 1F		AND	#\$1F
3022	19F1	18		CLC	
3023	19F2	60	•	RTS	
3024	19F3		+L19F3:	MOV	<#\$00>,PNYBCN
3025	19F7	A5 D2		LDA	PNYBBF+1
3026	19F9	10 04		BPL	L19FF
3027			+	MOV	<#\$FF>,PNYBCN
3028	19FF	A5 D1	L19FF:	LDA	PNYBBF
3029	1401	29 1F		AND	#\$1F
3030	1A03	18		CLC	÷
3031	1A04	60		RTS	
3032	1707	00			
3033				PAGE	
3033				. Au	

3036	3034					
1007 1005 1007 1008 1007 1008 1009	3035			; crunc	h word t	o compare with vocab table entries
1038	3036					
1009				CRNWRD:		
100			AO 06			
100 100				+L1A09:		<#\$U5>,< <pkword,x>></pkword,x>
3042		1A0D	E8			1.1400
3043 3044 3044 3045 1A1D 66 66 4 1NC ACC 3046 3047 1A23 00 05 3048 1A25 3049 1A27 4C 1A52 3050 1A2A 3051 1A2A 3051 1A2A 3051 1A2A 3051 3051 3052 4 4 ADD 3053 3054 1A36 3054 1A36 3054 1A36 3054 1A37 3058 1A38 3059 1A38 3051 3058 1A43 3059 1A45 3060 1A54 3060 1A54 3060 1A54 3060 1A55 3060 1A55 3060 1A55 3060 1A55 3060 1A55 3060 1A56 3060 1A57 3060 1A57 3060 1A58 3060 1A58 3060 1A58 3060 1A58 3060 1A58 3060 1A59 3060 1A59 3060 1A59 3060 1A50 3060 1A50 1A50 3060 1A50 1A50 1A50 3060 1A50 1A50 3060 1A50 1A50 1A50 1A50 1A50 1A50 1A50 1A5						
1418						
3045 1A1D E6 E6			.0 50			
1				LIAID:		
1A23		TAID	E0 E0	_		
1		1422	DO 05	т		
1427						
1						
1				11121.		
Total				LIAZA.		
3053		TAZC	20 TARD	+		
3054 1A36 A6 E4 LDX ACB 3055 1A38 95 DA STA PKWORD, X 3056 1A3A E6 E4 INC ACB 3057						
3055		1436	46 F4	-		
3056 1A3A BE E4 TNC ACB						
3057						·
1443		1404	20 21	+		
3059		1443	A5 E8	L1A43:		·
3060 3061 3062 3062 3063 3064 3064 3064 3065 3066 3066 3066 3067 3066 3067 3062 3068 3068 3069 3069 3060 3060 3060 3060 3060 3060						
3061 3062 3063 3063 1A52 A6 E4 L1A52: LDX ACB 3064 3064 1A56 BE E4 SUB ACD, <#\$5B> ACD, <#\$\$5B> 3063 3064 1A56 BE E4 SUB ACB ACB 3065 + DECJN ACB 3066 3066 1A5F 4C 1ACA 3067 1A62 D0 0B L1A62: BNE L1A6C 3068 3069 1A69 4C 1A52 3070 1A6C A5 E8 L1A6C: LDA ACD 3071 1A6E 20 1A99 3072 1A71 D0 DF BNE L1A52 3073 1A73 A9 06 LDA #\$06 3074 1A75 A6 E4 LDX ACB 3075 1A77 95 DA STA PKWORD, X ACB 3077 + DECBE ACD+1, L1A1B 3080 ACD **\$3B> ** ** ** ** ** ** ** ** **				+		
3062 1A52 A6 E4 L1A52: LDX ACB 3063 1A54 95 DA STA PKWORD,X 3064 1A56 E6 E4 INC ACB 3065 + DECJN ACD+1,L1A1B 3066 1A5F 4C 1ACA JMP L1ACA 3067 1A62 D0 0B L1A62: BNE L1A6C 3068 + SUB ACD, <#\$3B> 3069 1A69 4C 1A52 JMP L1A52 3070 1A6C A5 E8 L1A6C: LDA ACD 3071 1A6E 20 1A99 JSR L1A99 3072 1A71 D0 DF BNE L1A52 3073 1A73 A9 06 LDA #\$06 3074 1A75 A6 E4 LDX ACB 3075 1A77 95 DA STA PKWORD,X 3076 1A79 E6 E4 INC ACB 3079 + REPT 5 3080 + REPT 5 3080 1A88 A6 E4 LDX ACB 3084 1A8A 95 DA STA PKWORD,X 3086 1A8C E6 E4 INC ACB 3087 1A92 A5 E8 LDX ACD				+		
3063 1A54 95 DA STA PKWORD,X 3064 1A56 E6 E4 INC ACB 3065		1A52	A6 E4	L1A52:		
1A56						
Table Tabl						
3066 1A5F 4C 1ACA 3067 1A62 D0 08 L1A62: BNE L1A6C 3068				+		ACD+1,L1A1B
3067		1A5F	4C 1ACA		JMP	LIACA
3068 3069 1A69 4C 1A52 3070 1A6C A5 E8		1A62	DO 08	L1A62:	BNE	L1A6C
3070 1A6C A5 E8 L1A6C: LDA ACD 3071 1A6E 20 1A99 JSR L1A99 3072 1A71 D0 DF BNE L1A52 3073 1A73 A9 06 LDA #\$06 3074 1A75 A6 E4 LDX ACB 3075 1A77 95 DA STA PKWORD, X 3076 1A79 E6 E4 INC ACB 3077 + DECBE ACD+1,L1ACA 3078 1A7F A5 E8 LDA ACD 3079 + REPT 5 3080 + LSR A 3081 + ENDM 3082 1A86 29 03 AND #\$03 3083 1A88 A6 E4 LDX ACB 3084 1A8A 95 DA STA PKWORD, X 3086 + LDX ACB 3087 1A92 A5 E8 LDX ACB 3087 1A92 A5 E8 LDX ACB				+	SUB	ACD,<#\$3B>
3071 1A6E 20 1A99 JSR L1A99 3072 1A71 D0 DF BNE L1A52 3073 1A73 A9 06 LDA #\$06 3074 1A75 A6 E4 LDX ACB 3075 1A77 95 DA STA PKWORD,X 3076 1A79 E6 E4 INC ACB 3077 + DECBE ACD+1,L1ACA 3078 1A7F A5 E8 LDA ACD 3080 + REPT 5 3080 + LSR A 3081 + ENDM 3082 1A86 29 03 AND #\$03 3083 1A88 A6 E4 LDX ACB 3084 1A8A 95 DA STA PKWORD,X 3086 + LDX ACB 3087 1A92 A5 E8 LDX ACB	3069	1A69	4C 1A52		JMP	L1A52
3072 1A71 D0 DF BNE L1A52 3073 1A73 A9 06 LDA #\$06 3074 1A75 A6 E4 LDX ACB 3075 1A77 95 DA STA PKWORD, X 3076 1A79 E6 E4 INC ACB 3077 + DECBE ACD+1,L1ACA 3078 1A7F A5 E8 LDA ACD 3080 + REPT 5 3080 + LSR A 3081 + ENDM 3082 1A86 29 03 AND #\$03 3083 1A88 A6 E4 LDX ACB 3084 1A8A 95 DA STA PKWORD, X 3086 + DECBE ACD+1,L1ACA 3086	3070	1A6C	A5 E8	L1A6C:	LDA	ACD
3073 1A73 A9 06 LDA #\$06 3074 1A75 A6 E4 LDX ACB 3075 1A77 95 DA STA PKWORD,X 3076 1A79 E6 E4 INC ACB 3077 + DECBE ACD+1,L1ACA 3078 1A7F A5 E8 LDA ACD 3079 + REPT 5 3080 + LSR A 3081 + ENDM 3082 1A86 29 03 AND #\$03 3083 1A88 A6 E4 LDX ACB 3084 1A8A 95 DA STA PKWORD,X 3086 + DECBE ACD+1,L1ACA LDA ACD 3086 - DECBE ACD+1,L1ACA LDA ACB 3087 1A92 A5 E8 LDA ACD	3071	1A6E	20 1A99		JSR	L1A99
3074 1A75 A6 E4 LDX ACB 3075 1A77 95 DA STA PKWORD,X 3076 1A79 E6 E4 INC ACB 3077 + DECBE ACD+1,L1ACA 3078 1A7F A5 E8 LDA ACD 3079 + REPT 5 3080 + LSR A 3081 + ENDM 3082 1A86 29 03 AND #\$03 3083 1A88 A6 E4 LDX ACB 3084 1A8A 95 DA STA PKWORD,X 3085 1A8C E6 E4 INC ACB 3086 + DECBE ACD+1,L1ACA 3086 DECBE ACD+1,L1ACA 3087 1A92 A5 E8 LDA ACD	3072	1A71	DO DF	•	BNE	L1A52
3075 1A77 95 DA STA PKWORD, X 3076 1A79 E6 E4 INC ACB 3077 + DECBE ACD+1, L1ACA 3078 1A7F A5 E8 LDA ACD 3080 + REPT 5 3080 + LSR A 3081 + ENDM 3082 1A86 29 03 AND #\$03 3083 1A88 A6 E4 LDX ACB 3084 1A8A 95 DA STA PKWORD, X 3085 1A8C E6 E4 INC ACB 3086 + DECBE ACD+1, L1ACA 3087 1A92 A5 E8 I DA ACD	3073	1A73				· ·
3076 1A79 E6 E4 INC ACB 3077 + DECBE ACD+1,L1ACA 3078 1A7F A5 E8 LDA ACD 3079 + REPT 5 3080 + LSR A 3081 + ENDM 3082 1A86 29 03 AND #\$03 3083 1A88 A6 E4 LDX ACB 3084 1A8A 95 DA STA PKWORD,X 3085 1A8C E6 E4 INC ACB 3086 + DECBE ACD+1,L1ACA 3087 1A92 A5 E8 LDA ACD	3074	1A75	A6 E4			
3077						· · · · · · · · · · · · · · · · · · ·
3078 1A7F A5 E8 LDA ACD 3079 + REPT 5 3080 + LSR A 3081 + ENDM 3082 1A86 29 03 AND #\$03 3083 1A88 A6 E4 LDX ACB 3084 1A8A 95 DA STA PKWORD,X 3085 1A8C E6 E4 INC ACB 3086 + DECBE ACD+1,L1ACA 3087 1A92 A5 E8 LDA ACD	3076	1A79	E6 E4			
3079				+		
3080		1A7F	A5 E8			
3081						
3082 1A86 29 03 AND #\$03 3083 1A88 A6 E4 LDX ACB 3084 1A8A 95 DA STA PKWORD,X 3085 1A8C E6 E4 INC ACB 3086 + DECBE ACD+1,L1ACA 3087 1A92 A5 E8 LDA ACD						A
3083 1A88 A6 E4 LDX ACB 3084 1A8A 95 DA STA PKWORD,X 3085 1A8C E6 E4 INC ACB 3086 + DECBE ACD+1,L1ACA 3087 1A92 A5 E8 LDA ACD				+		
3084 1A8A 95 DA STA PKWORD,X 3085 1A8C E6 E4 INC ACB 3086 + DECBE ACD+1,L1ACA 3087 1A92 A5 E8 LDA ACD						
3085 1A8C E6 E4 INC ACB 3086 + DECBE ACD+1,L1ACA 3087 1A92 A5 E8 LDA ACD						
3086						
3087 1A92 A5 E8 : LDA ACD		1A8C	E6 E4	17		
		1.00	45 50	. +		
3088 1894 29 1F AND #\$1F						
	3088	1494	29 IF		ANU	# ⊅ 1L

3089	1A96	4C	1A52		JMP	L1A52
3090 3091 3092 3093	1A99 1A9B	A2	24	L1A99: +L1A9B:	LDX CMPBE DXBPL	#\$24 <spclch,x>,L1AA6 L1A9B</spclch,x>
3094	1AA3	ΑO	00		LDY	#\$00
3095	1AA5	60			RTS	
3096	1446	8A		L1AA6:	TXA	
3097				+	ADD	,<#\$08>
3098	1AAA	60			RTS	
3099						
3100	1AAB			+TSTCHR:	CMPBL	<#'a'>,L1AB6
3101				+	CMPBG	<#'z'+1>,L1AB6
3102	1AB3		00		LDA	#\$00
3103	1AB5	60			RTS	
3104	1AB6			+L1AB6:	CMPBL	<#'A'>,L1AC1
3105				+	CMPBG	<#'Z'+1>,L1AC1
3106	1ABE		01		LDA	#\$01
3107	1ACO	60			RTS	
3108	1AC1			+L1AC1:	TSTABE	L1AC9
3109	1AC5		02		BMI	L1AC9
3110	1AC7		02		LDA	#\$ 02
3111	1AC9	60		L1AC9:	RTS	
3112			00	1.1000		PKWORD+1
3113	1ACA	A5	DB	L1ACA:	LDA REPT	4
3114				+	ASL	Ā
3115				+	ENDM	^
3116 3117	1ADO	26	DA	•	ROL	PKWORD
3118	1AD2	0A	DA		ASL	A
3119	1AD2	-	DA		ROL	PKWORD
3119	1AD5		DA		LDX	PKWORD
3121	1AD7		DB		STX	PKWORD+1
3122	1AD9		DC		ORA	PKWORD+2
3123	1ADB		DA		STA	PKWORD
3124	1ADD		DE		LDA	LDE
3125				+	REPT	4
3126				+	ASL	Α
3127				+	ENDM	
3128	1AE3	26	DD		ROL	PKWORD+3
3129	1AE5	0 A			ASL	Α
3130	1AE6	26	DD		ROL	PKWORD+3
3131	1AE8	Α6	DD		LDX	PKWORD+3
3132	1AEA		DD		STX	PKWORD+3
3133	1AEC		DF		ORA	LDF
3134	1AEE		DC		STA	PKWORD+2
3135	1AFO		DD		LDA	PKWORD+3
3136	1AF2		80		ORA	#\$80
3137	1AF4		DD		STA	PKWORD+3
3138	1AF6	60			RTS	
3139						
3140					PAGE	

3141						4							
3142			; init (output r	outine and screen wind	MOR							
3143													
3144	1AF7		+INITSC:	MOV	<#\$C1>,PRCSWL+1								
3145													
3146				IFF	RNGDBG	;	if	RNG	debug,	save	2 lin	es at	top!
3147			+	MOV	<#\$01>,WNDTOP								
3148				ENDIF									
3149													
3150			+	MOV	<#\$00>, <wndlft,l1ba0></wndlft,l1ba0>	>							
3151			+	MO∨	<#\$28>,WNDWDT								
3152			+	MO∨	<#\$18>,WNDBOT								
3153			+	MO∨	<#\$BE>,PROMPT								
3154			+	MOV	<#\$FF>,INVFLG								
3155											•		
3156			; clear	the scr	een								
3157			•										_
3158	1B16	20 FC58	CLRSCR:	JSR	HOME								
3159			+	MOV	WNDTOP, LINCHT								
3160	1B1D	60		RTS	•								
3161													
3162													
3163			· find	the high	est usable page of mem	norv							
3164			,										
3165	181E		+FNDMEM:	DMOVT2	LSTFLC+\$0100,ACC								
3166	1826	AO 00	· · · · · · · · · · · · · · · · · · ·	LDY	#\$00								
3167	1B28	C6 E7	L1B28:	DEC	ACC+1								
3168	1B2A	B1 E6	LIBEO.	LDA	(ACC),Y								
3169	IDZA	טו בט	+	CMPBN	<(ACC),Y>,L1B28								
3170	1B30	49 FF	•	EOR	#\$FF								
	1B30	91 E6		STA	(ACC),Y								
3171	1632	91 60	. +	CMPBN	<(ACC), Y > , L1B28								
3172	* 0.00	40 55	, т		#\$FF								
3173	1B38	49 FF		EOR									
3174	1B3A	91 E6		STA	(ACC),Y								
3175	1B3C	A5 E7		LDA	ACC+1								
3176	1B3E	60		RTS									
3177				5.05									
3178				PAGE .									

MACRO-80 3.45 18-Sep-8

3179 3180			; buffe	r a char	acter for output	
3181	1B3F	A6 EB	BFCHAR:	I DX	CHRPTR	; get buffer pointer
3182 3183	ופטר	AO LD	DI CHAR.	LDA		, -
3184			+	CMPJE	<#CRCHAR>,PRNTBF	; if char is a CR, flush buffer
3185			+	CMPBL	<#' '>,L1B61	; if it is a control character, discard it ; if it is in 64 char subset, buffer it as is
3186			+	CMPBL	<#\$60>,L1B57	; IT IT IS III 04 Char subset, butler it as is
3187				IFT	LC40	
3188		24 22		BIT	INVFLG	; if inverse, convert LC to UC
3189	1850 1852	24 32 30 03		BMI	L1B57	, , , , , , , , , , , , , , , , , , , ,
3190 3191	1652	30 03		ENDIF		
3192						
3193			+	SUB	,<#\$20>	; yes, convert to upper case
3194						
3195	1B57	09 80	L1B57:	ORA	#\$80	; set high bit for Apple
3196					OUECED V	; store it in buffer
3197	1B59	9D 0200		STA	BUFFER,X	; store it in burrer ; if buffer is full, print some of it
3198			+	CPXBG	WNDWDT,L1B64	; if builter is full, print some of it
3199	1B60	E8		INX		; increment pointer
3200 3201	1B61	86 EB	L1B61:		CHRPTR	; save pointer
3201	1B63	60	215011	RTS	<u> </u>	; return
3203	1500					
3204			; find	last spa	ce in buffer, if any	
3205						
3206	1B64	A9 A0	L1B64:	LDA	#" "	; load a space for comparison
3207			1000	044005	(DUEEED V) 11870	; if this is one, we've got it
3208	1866		+L1B66:		<buffer,x>,L1B70 L1B66</buffer,x>	no, loop if more characters in buffer
3209			+	DXBNE	LIBOO	, 110, 100p 11 more characters we serve
3210 3211	1B6E	A6 21		LDX	WNDWDT	: no space use last character
3211	IBOL	A0 21				, ,
3213	1870	86 EC	L1B70:	STX	CHRPT2	; save pointer
3214	1B72	86 EB		STX	CHRPTR	
3215						
3216	1874	20 1010		JSR	PRNTBF	; print line up to this point
3217						as of huffor
3218			; move	rest or	line back to beginning	ig of buffer
3219	1877	E6 EC	L1B77:	INC	CHRPT2	: get pointer to next char
3220 3221	1B77	A6 EC	LID//:	LDX	CHRPT2	, got pointer
3222	1079	AO LC	+	CPXRGT		; if it is past the last char, return
3223						
3224	1882	BD 0200		LDA	BUFFER,X	; get the character
3225	1885	A6 EB		LDX	CHRPTR	; get the pointer to the new loc
3226	1B87	9D 0200		STA	BUFFER,X	; store the character there
3227	1B8A	E6 EB		INC	CHRPTR	; and increment the pointer
3228				. 5.7	CLIODES	- unnocossary!
3229	1B8C	A6 EC		LDX	CHRPT2	; unnecessary! : try for another one
3230	188E	4C 1B77		JMP	L1B77	, cry for another one
3231				PAGE		
3232				r AGE		

2222								
3233 3234				- output	t the bu	ffer to the screen, and	+ 0	the esister if emphiled
				; output	t the bu	itel to the screen, and	U	the printer is enabled
3235		• •		OUTOUE	. 5.7	#UDDELC14		
3236	1B91	ΑO		OUTBUF:		#HDRFLG+1		
3237	1B93	В1			LDA	(FRZMEM),Y		
3238	1B95	29	01		AND	#\$01		
3239				+	JSRNE	PRTBUF		
3240	1B9C	20	1BF5		JSR	DSPBUF	,	
3241	1B9F	60			RTS			
3242	1551	-						
3243								
3244				; outpu	t the bu	ffer to the printer		
3245								
3246	1BA0	00		L1BAO:	DB	\$00	;	printer initialization flag
3247								
3248	1BA1			+PRTBUF:	DPSH	CSWL	:	save our output vector
3249				+	PSH	CURSRH		and cursor column
3250				7		001101111	•	2.1.d 00.00. 00.10
				+	DMOV	DDCCMI CCMI	_	act vector for printer
3251				т	DMO	PRCSWL,CSWL	,	get vector for printer
3252		_						
3253	1BB2	A 2	00		LDX	#\$00	;	start with position 0 in buffer
3254						•		
3255	1BB4	AD	1BA0		LDA	L1BAO	;	is printer initialized?
3256	1BB7	DO	1C		BNE	L1BD5	•	yes, go print it
3257	1889		1BA0		INC	L1BAO		no, but now it will be
3258	1003		IDAO		1110	EIBAO	•	no, but now it will be
	1000	• •	00		LDA	#\$89		output ^I80N
3259	1BBC		89			•		•
3260	1BBE		FDED		JSR	COUT	;	
3261	1BC1		91		LDA	#\$91	;	
3262	1BC3	8D	0779		STA	PRTWDT	;	screen echo (we hope!))
3263	1BC6	Α9	B8		LDA	#\$B8		
3264	1BC8	20	FDED		JSR	COUT		
3265	1BCB		В0		LDA	#\$B0		
3266	1BCD		FDED		JSR	COUT		
						#\$CE		
3267	1BD0		CE		LDA			
3268	1BD2	20	FDED		JSR	COUT		
3269								
3270	1BD5			+L1BD5:	CPXBE	CHRPTR,L1BE3	;	are we done yet?
3271								
3272	1BD9	BD	0200		LDA	BUFFER, X	:	no, get character
3273	1BDC		FDED		JSR	COUT		and output it
3274	IBDC		1020		0011		,	
	1005				TNIN			incoment maintag
3275	1BDF	E8			INX			increment pointer
3276	1BE0	4C	1BD5		JMP	L1BD5	;	and go for another one
3277								
3278	1BE3			+L1BE3:	DMOV	CSWL, PRCSWL	;	save print vector again (may have changed)
3279								
3280				+	PUL	CURSRH		restore cursor column
3281				+	DPUL	CSWL		and display vector
	1BF4	60		•	RTS	CSWL		and return
3282	IDF4	60			KIS		•	and return
3283								
3284								
3285				; outpu	t the bu	ffer to the display		
3286				•				
3287	1BF5	A2	00	DSPBUF:	LDX	#\$00	:	start with position 0 in buffer
		,		3 .		·· •	,	

Infocom	N., ERLOG	IC interpret	er disassem	bly, 5/	27/84 MACRO-80 3.	18-Sep-81 PAGE 1-80
3288						_
3289	1BF7		+L1BF7:	CPXBE	CHRPTR,L1C05	; are we done yet?
3290						
3291	1BFB	BD 0200		LDA	BUFFER,X	; get the character
3292	18FE	20 FDF0		JSR	COUT1	; and output it
3293						
3294	1001	E8		INX		; increment pointer
3295	1C02	4C 1BF7		JMP	L1BF7	; and go for another one
3296	•					
3297	1005	A2 00	L1C05:	LDX	#\$00	; reset pointer to beginning
3298	1C07	86 EB		STX	CHRPTR	
3299	1C09	60		RTS		; and return
3300						
3301				PAGE		

3302						
3303	1 C O A	58	4D 4F	MOREMS:	DB	'[MORE]'
3304	1C0D		45 5D			_
3305	0006	-		MRMSLN	EQU	*-MOREMS
3306	0000					
3307	1C10	E6	EĐ	PRNTBF:	INC	LINCNT
3308	1C12	A5			LDA	LINCNT
3309				+	CMPBL	WNDBOT,L1C40
3310				+	DMOVI	MOREMS,ACC
3311	1C20	A2	06		LDX	#MRMSLN
3312				+	MOV	<#\$3F>,INVFLG
3313	1C26	20	1D57		JSR	SHWMSG
3314				+	MOV	<#\$FF>,INVFLG
3315	1C2D	20	FD0C		JSR	RDKEY
3316				+	SUB	CURSRH,<#\$06>,CURSRH
3317	1C37	20	FC9C		JSR	CLREOL
3318				+	MOV	WNDTOP, LINCNT
3319	1C3E	E6	ED		INC	LINCNT
3320	1C40			+L1C40:	PSH	CHRPTR
3321	1C43	20	1B91		JSR	OUTBUF
3322	1C46	68			PLA	
3323				+	CMPBE	WNDWDT,L1C50
3324	1C4B	Α9	8D		LDA	#\$8D
3325	1C4D	20	FDF0		JSR	COUT 1
3326	1C50	A0	11	L1C50:	LDY	#HDRFLG+1
3327	· 1C52	B1	BA		LDA	(FRZMEM),Y
3328	1C54	29	01		AND	#\$01
3329	1C56	F0	21		BEQ	L1C79
3330				+	DPSH	CSWL
3331				+	DMOV	PRCSWL,CSWL
3332	1066	Α9	8D		LDA	#\$8D
3333	1C68	20	FDED		JSR	COUT
3334				+	DMOV	CSWL, PRCSWL
3335				+	DPUL	CSWL
3336	1C 79	Α2	00	L1C79:	LDX	#\$00
3337	1C7B	4C	1B61		JMP	L1B61
3338						
3339					PAGE	

3340								
3341	1C7E	53	43 4F	SCORMS:	DB	'SCORE:'		
3342	1C81		45 3A					
3343	0006			SCMSLN	EQU	*-SCORMS		
3344	0000				•			
3345	1C84	54	49 4D	TIMEMS:	DΒ	'TIME:'		
3346	1087		3A	12111211101				
	0005	73	JA.	TMMSLN	EQU	*-TIMEMS		
3347	0005			IMMOLIT	Luo			
3348	1089	00		L1C89:	DB	\$00		
3349	1009	00		LICOJ.	00	400		
3350	1001	20	1891	OPPRST:	150	OUTBUF	· orint w	hat's in the buffer
3351	1C8A	20	1091	+	PSH	<cursrh, cursrv=""></cursrh,>		e cursor position
3352				+	MOV	<#\$00>, <cursrh,cursrv></cursrh,cursrv>	, home th	
3353		00	5000	т		VTAB	, 1101116 211	c car so.
3354	1 C 9 9	20	FC22		JSR		. set inv	erse mode
3355				+	MOV	<#\$3F>,INVFLG	, set illv	er se mode
3356			_			## 1 O		ual uas O
3357	1 C A O		10		LDA	#\$ 10	; get gro	val var O
3358	1CA2		OAC2		JSR	GTVRA1		14 +i2
3359	1CA5	A5	E6		LDA	ACC		ave as last time?
3360				+	CMPBE	L1C89,L1CB8		n't print it
3361	1 CAC		1C89		STA	L1C89		e for next time's compare
3362	1CAF	20	ODE4		JSR	LODE4		thing name
3363	1CB2	20	1BF5		JSR	DSPBUF		to display
3364	1 CB5	20	FC9C		JSR	CLREOL	; clear r	est of line
3365								
3366	1CB8			+L1CB8:	MO∨	<#\$19>,CURSRH	; tab ove	
3367	1CBC	A5	F3		LDA	STLTYP	; score o	r time?
3368	1CBE	DO	1B		BNE	L1CDB	; time	
3369				+	DMOVI	SCORMS, ACC	; score,	print "SCORE:"
3370	1008	A 2	06		LDX	#SCMSLN		
3371	1 C C A		1D57		JSR	SHWMSG		
3372	1CCD		24		INC	CURSRH	; one spa	ice
3373	1 C C F		11		LDA	#\$11	: get glo	bal var 1 (score)
3374	1CD1		DAC2		JSR	GTVRA1	,	
3374	1CD1	-	14F5		JSR	PRNTNM	: output	it as decimal number
			2F		LDA	#'/'	; seperat	
3376	1CD7		2A		BNE	L1D05	; always	
3377	1CD9	טט	ZA		DIVE	LIDOS .	, almays	Cancil
3378	1.000			+L1CDB:	DMOVI	TIMEMS,ACC	: print "	TIME."
3379	1 CDB		0.5	TLICUB:	LDX	#TMMSLN	, p	
3380	1CE3		05					
3381	1CE5		1D57		JSR	SHWMSG		
3382	1CE8		24		INC	CURSRH	; one spa	
3383	1CEA		11		LDA	#\$11	; get gio	obal var 1 (time)
3384	1CEC		OAC2		JSR	GTVRA1		
3385	1CEF		E6		LDA	ACC	; is it z	ero?
3386	1 C F 1	D0	02		BNE	L1CF5		
3387	1CF3	Α9	18		LDA	#\$18		ike it 24:00
3388	1CF5			+L1CF5:	CMPBM	<#\$0C>,L1D00	; is it A	A.M. or P.M.?
3389	1CF9	F0	05		BEQ	L1D00		
3390	1CFB	38			SEC			convert to 1-12 range
3391	1CFC	E9	OC		SBC	#\$0C	; by sub	stracting 12
3392	1CFE		E6		STA	ACC		
3393	1D00		14F5	L1D00:	JSR '	PRNTNM	; print o	out hours
3394	1D03		3A	•	LDA	#':'		
303.4		,.0						

		20 1B3F	L1D05:	JSR	BECHAR	;	print the seperator
3395	1D05			LDA	#\$12	;	get global var 2 (turns/minutes)
3396	1D08	A9 12 20 OAC2		JSR	GTVRA1		
3397	1 DOA			LDA	STLTYP		time?
3398	1 DOD	A5 F3		BEQ	L1D40	;	no, go print turns
3399	1D0F	FO 2F		LDA	ACC		yes, are minutes < 10?
3400	1D11	A5 E6		CMPBG	<#\$0A>,L1D1C	:	no
3401			+		#\$B0		yes, print a space (?)
3402	1D17	A9 B0		LDA	BFCHAR	•	
3403	1D19	20 1B3F	_	JSR			print the minutes
3404	1D1C	20 14F5	L1D1C:	JSR	PRNTNM	,	print a space
3405	1D1F	A9 A0		LDA	#\$AO	,	prime a space
3406	1D21	20 1B3F		JSR	BECHAR	_	get global var 1 (hours)
3407	1024	A9 11		LDA	#\$11	•	get grobal var , (meet s)
3408	1D26	20 OAC2		JSR	GTVRA1		is it A.M. or P.M.?
	1D20	A5 E6		LDA	ACC		
3409	1029	A3 L0	+	CMPBP	<#\$0C>,L1D33		P.M.
3410		A9 C1		LDA	#"A"	;	A.M.
3411	1D2F	DO 02		BNE	L1D35		
3412	1D31		L1D33:	LDA	#"P"		
3413	1D33	A9 D0	L1D35:	JSR	BFCHAR	;	; print the "A" or "P"
3414	1D35	20 1B3F	LID35:	LDA	#"M"		
3415	1D38	A9 CD		JSR	BFCHAR		; print the "M"
3416	1D3A	20 1B3F			L1D43		•
3417	1D3D	4C 1D43		JMP	EID43		
3418					DONTHA		; print the score
3419	1D40	20 14F5	L1D40:	JSR	PRNTNM		display the buffer
3420	1D43	20 1BF5	L1D43:	JSR	DSPBUF		: clear out the line
3421	1D46	20 FC9C		JSR	CLREOL		: back to normal video mode
3422			+	MOV	<#\$FF>,INVFLG		and the old cursor loc
3423			+	PUL	<cursrv,cursrh></cursrv,cursrh>		; and the did carsor roo
	1D53	20 FC22		JSR	VTAB		
3424	1D56	60		RTS			; return to caller
3425	1056	00					
3426		AO OO	SHWMSG:	LDY	#\$00		
3427	1D57		L1D59:	LDA	(ACC),Y		
3428	1D59	B1 E6	L1033.	ORA	#\$80		
3429	1D5B	09 80		JSR	COUT 1		
3430	1D5D	20 FDF0		INY	20011		
3431	1D60	C8			L1D59		
3432			+	DXBNE	LIDOS		
3433	1D64	60		RTS			
3434							
3435				PAGE			
5455							

3436	1005	20 1891	GETLIN:	ISB.	OUTBUF
3437	1D65	20 1091	+	MOV	WNDTOP, LINCHT
3438	1000	20 FD6F	•	JSR	GETLN1
3439	1D6C	E6 ED		INC	LINCNT
3440	1D6F	EQ ED	+	MOV	<#\$8D>.< <buffer,x>></buffer,x>
3441	1076	- 0	т	INX	, Lo,
3442	1D76	E8		TXA	
3443	1D77,	8A		PHA	
3444	1D78	48		LDY	#HDRFLG+1
3445	1D79	AO 11		LDA	(FRZMEM),Y
3446	1D7B	B1 BA		AND	#\$01
3447	1D7D	29 01		BEO	L1D8B
3448	1D7F	FO OA		TXA	LIDOD
3449	1D81	8A		STA	CHRPTR
3450	1D82	85 EB			PRTBUF
3451	1D84	20 1BA1		JSR	<pre><#\$00>,CHRPTR</pre>
3452			+	MOV	\#\$UU>,CHRPIK
3453	1D8B	68	L1D8B:	PLA	# # 00
3454	1D8C	AO 00		LDY	#\$00
3455			+	CMPBL	<(ARG1), Y>, L1D94
3456	1D92	B1 82		LDA	(ARG1),Y
3457	1D94	48	L1D94:	PHA	
3458	1D95	FO 1A		BEQ	L1DB1
3459	1D97	AA		TAX	21.5552 1/
3460	1D98	B9 0200	L1D98:	LDA	BUFFER, Y
3461	1D9B	29 7F		AND	#\$7F
3462			+	CMPBL	<#'A'>,L1DA7
3463			+	CMPBG	<#'Z'+1>,L1DA7
3464	1DA5	09 20		ORA	#\$ 20
3465	1DA7	C8	L1DA7:	INY	
3466	1DA8	91 82		STA	(ARG1),Y
3467			+	CMPBE	<pre><#CRCHAR>,L1DB1</pre>
3468			+	DXBNE	L1D98
3469	1DB1	68	L1DB1:	PLA	
3470	1DB2	60		RTS	
3471					
3472				PAGE	
-					

: Device Characteristics Table

; IOB type ; Slot * 16 ; Drive ; Volume ; Track ; Sector

; I/O buffer ; unused ; Command ; Status ; Actual volume ; Previous slot * 16 ; Previous drive

3473 3474 3475 3476 3477 3478 3479 3480 3481 3482 3483 3484 3485 3486 3487 3488	1DB3 1DB4 1DB5 1DB6 1DB7 1DB8 1DB9 1DBB 1DBD 1DBF 1DC0 1DC1 1DC2 1DC3	01 60 01 00 00 00 1DC4 0000 0000 00 00 00	IOB: IOBSLT: IOBDRV: IOBTRK: IOBSCT: IOBBUF: IOBCMD:	DB DB DB DW DW DW	\$01 \$60 \$01 \$00 \$00 \$00 DCT \$0000 \$0000 \$000 \$00 \$00 \$00 \$00
3490	1DC4	00 01 EF D8	DCT:	DB	\$00,\$01,\$EF,\$D8
3491 3492 3493 3494	1DC7	8D 1DBF	DISKIO:	DMOV	IOBCMD ACC, IOBBUF
3495 3496	1DDA	A5 E4	+	MOV LDA	#\$03,IOBTRK ACB ACB+1
3497 3498	1DDC 1DDE	A6 E5 38		LDX SEC	ACB+ I
3499 3500 3501	1DDF 1DE1	E5 7F B0 04	L1DDF:	SBC BCS DXBMI	SECPTK L1DE7 L1DED
3502 3503	1DE6 1DE7	38 EE 1DB7	L1DE7:	SEC INC	IOBTRK
3504 3505	1DEA 1DED	4C 1DDF	+L1DED:	JMP ADD LDA	L1DDF ,SECPTK,IOBSCT #>IOB
3506 3507 3508	1DF3 1DF5 1DF7	A9 1D A0 B3 4C 2900		LDY JMP	# <iob RWTS</iob
3509 3510	1DFA	40 2300	+DRDBUF:		BUFFER, ACC
3511 3512	1E02 1E08	A9 01	+DRDNXT: DRDBLK:		ACB #\$01
3513 3514	1E0A	4C 1DC8	DNDBLW.	JMP	DISKIO
3515 3516	1E0D	20 1E08	DRDBKF:	JSR JSRCS	DRDBLK FATAL
3517 3518	1E15	60		RTS	
3518 3519 3520 3521	1E16 1E1E 1E24	A9 02	+DWRBUF: +DWRNXT:		BUFFER,ACC ACB #\$02
3522 3523 3524	1E26	4C 1DC8		JMP PAGE	DISKIO

Infocom	I ERLOG	IC interpreter d	isassembly, 5/27/84	MACRO-80 3.	. 18-Sep-81	PAGE	1-87
3580 3581 3582 3583 3584 3585 3586 3587	1EA6 1EA9 1EAC 1EAF 1EB2 1EB5 1EB8	45 54 55 52 4E 27 20 4B 45 59 20 54 4F 20 42 45 47 49 4E 20 2D 2D 2D					
3588 3589	ICDD		PAGE				

3590	1EBD	20 1816	L1EBD:	JSR	CLRSCR
3591		20 1010	LILDO.	JSR	PRNTBF
3592	1EC0	20 1010		JSR	PRNTBF
3593	1EC3	20 1010	+	DMO∨I	L1E3D.ACC
3594	. = 0 =	40.10	•	LDX	#\$1C
3595	1ECE	A2 1C		JSR	OUTMSG
3596	1ED0	20 1E29		JSR	PRNTBF
3597	1ED3	20 1C10		MOV	<#\$24>,L1E59
3598			+	JSR	L1F4C
3599	1EDB	20 1F4C		-	L1E8D
3600	1EDE	8D 1E8D		STA	BFCHAR
3601	1EE1	20 1B3F		JSR	<#\$00>,L1E59
3602	_		+	MOV	L1F4C
3603	1EE9	20 1F4C		JSR	L1F4C
3604	1EEC	AA		TAX	##O7
3605	1 EED	29 07		AND	#\$07
3606			+	REPT	4
3607			+	ASL	A
3608			+	ENDM	TORGLE
3609	1EF3	8D 1DB4		STA	IOBSLT
3610	1EF6	8 A		TXA	
3611	1EF7	8D 1E69		STA	L1E69
3612	1EFA	20 1B3F		JSR	BECHAR
3613			+	MOV	<#\$12>,L1E59
3614	1F02	20 1F4C		JSR	L1F4C
3615	1F05	AA		TAX	
3616	1F06	29 03		AND	#\$03
3617	1F08	8D 1DB5		STA	IOBDRV
3618	1F0B	8 A		TXA	
3619	1F0C	8D 1E7B		STA	L1E7B
3620	1F0F	20 1B3F		JSR	BFCHAR
3621	1F12	20 1010	L1F12:	JSR	PRNTBF
3622			+	DMOVI	L1E9A,ACC
3623	1F1D	A2 23	•	LDX	#\$23
3624	1F1F	20 1E29		JSR	OUTMSG
3625	1F22	20 1B91		JSR	OUTBUF
3626	1F25	20 FD0C		JSR	RDKEY
3627			+	CMPBN	<#\$8D>,L1F12
3628			+	MOV	<#\$FF>, <acb,acb+1></acb,acb+1>
3629	1F32	AD 1EBD		LDA	L1E8D
3630	1F35	29 07		AND	#\$07
3631	1F37	FO OF		BEQ	L1F48
3632	1F39	A8		TAY	
3633	1F3A	A.O	+L1F3A:	DADDB2	ACB, <#\$40>
	11.04		+	DYBNE	L1F3A
3634	1F48	20 1010	L1F48:	JSR	PRNTBF
3635		60	LII TO.	RTS	
3636	1F4B	งบ			
3637				PAGE	
3638				FAGE	

D A	GE	1	ı —	89
PA	GE.		_	צס

3639					
3640	1F4C	20 1010	L1F4C:	JSR	PRNTBF
3641			+	DMO∨I	L1E5A,ACC
3642			+	DADDB2	ACC,L1E59
3643	1F63	A2 OF		LDX	#\$0F
3644	1F65	20 1E29		JSR	OUTMSG
3645	1F68	20 1891		JSR	OUTBUF
3646			+	MOV	<#\$19>,CURSRH
3647			+	MOV	<#\$3F>,INVFLG
3648			+	DMOVI	L1E90,ACC
3649	1F7B	A2 0A		LDX	#\$0A
3650	1F7D	20 1D57		JSR	SHWMSG
3651			+	DMO∨I	L1E69,ACC
3652			+	DADDB2	ACC,L1E59
3653	1F94	A2 01		LDX	#\$01
3654	1F96	20 1D57		JSR	SHWMSG
3655			+	MOV	<#\$FF>,INVFLG
3656	1F9D	20 FD0C		JSR	RDKEY
3657	1FAO	48		PHA	
3658			+	MOV	<#\$19>,CURSRH
3659	1FA5	20 FC9C		JSR	CLREOL
3660	1FA8	68		PLA	
3661	1FA9	AC 1E59		LDY	L1E59
3662			+	CMPBN	<#\$8D>,L1FB3
3663	1FB0	89 1E69		LDA	L1E69,Y
3664	1FB3	29 7F	L1FB3:	AND	#\$7F
3665			+	CMPBL	<l1e69+1,y>,L1F4C</l1e69+1,y>
3666			+	CMPBG	<l1e69+2,y>,L1F4C</l1e69+2,y>
3667	1FBF	60		RTS	
3668					
3669				PAGE	

3670							
	.500	F0 4	0 45				
3671	1FCO		C 45	L1FCO:	DB	'PLEASE RE-INS	ERT GAME DISKETTE,'
3672	1FC3		3 45				
3673	1FC6		2 45				
3674	1FC9	2D 4	9 4E				
3675	1FCC	53 4	5 52				
3676	1FCF	54 2	0 47				
3677	1FD2	41 4	D 45				· ·
3678	1FD5	20 4	4 49				
3679	1FD8		B 45				
3680	1FDB		4 45				
3681	1FDE	2C	7 70				
3682		20					
3683	1FDF	20. 2	D 2D	1.1505	5.5		
3684				L1FDF:	DB	' PRESS ''R	ETURN'' KEY TO CONTINUE'
	1FE2		0 52				
3685	1FE5	45 5					
3686	1FE8		7 52				
3687	1FEB	45 5					
3688	1FEE	52 4					
3689	1FF1	20 4					,
3690	1FF4	59 2					
3691	1FF7	4F 2	0 43				
3692	1FFA	4F 4	E 54				
3693	1FFD	49 4	E 55				
3694	2000	45 2	0 2D				
3695	2003	2D 2					
3696			-				
3697	2005	AD 11	DB4	L2005:	LDA	IOBSLT	
3698	2000	70 1	004	+	CMPBN		
3699	200C	AD 11	nee	•		<#\$60>,L2040	
3700	2000	AD II	000		LDA	IOBDRV	
3700	2012	20.1	C 1 D	+	CMPBN	<#\$01>,L2040	
3701	2013	20 1	CIU		JSR	PRNTBF	
	2015		_	+	DMOVI	L1FCO,ACC	
3703	201E	A2 11			LDX	#\$1F	
3704	2020	20 11			JSR	OUTMSG	
3705	2023	20 10	C10	L2023:	JSR	PRNTBF	
3706				+	DMO∨I	L1FDF,ACC	
3707	202E	A2 20			LDX	#\$ 26	
3708	2030	20 1	E29		JSR	OUTMSG	
3709	2033	20 18	391		JSR	OUTBUF	
3710	2036	20 F	00C		JSR	RDKEY	
3711				+	CMPBN	<#\$8D>,L2023	
3712	203D	20 10	010		JSR	PRNTBF	
3713	2040		- · -	+L2040:	MOV	<#\$60>,IOBSLT	
3714				+	MOV		
3715	204A	60		-	RTS	<#\$01>,IOBDRV	
3716	20 17				KI2		
3717					0465		
9717					PAGE		

3718									
3719									
3720	204B	20	1EBD	OPS	VGM:	JSR	L1EBD	:	setup for disk I/O
3721									P
3722	204E		00			LDX	#\$ 00	;	copy game release # to buffer
3723	2050	ΑO	02			LDY	#HDRREL		,, 0
3724				+		MOV	<(FRZMEM),Y>,< <buffer,x< td=""><td>>></td><td></td></buffer,x<>	>>	
3725	2057	E8				INX			
3726	2058	C8				INY			
3727	0055			+		MOV	<(FRZMEM),Y>,< <buffer,x< td=""><td>>></td><td></td></buffer,x<>	>>	
3728 3729	205E	E8				INX			
3730						D.1101.17			
3731	2067	۸.0	0.2	+		DMOVI	PRGIDX,ACC	;	copy PC to buffer
3732	2067		03 20DF			LDY	#\$03		
3732	2009	20	20DF			JSR	SVGMMV		
3734				+		DMOVIT	1.00//10 1.00		
3735	2074	۸٥	1 E	т		DMOVI LDY	LOCVAR,ACC	;	copy local variables to buffer
3736	2076		20DF				#\$1E		
3737	2070	20	2001			JSR	SVGMMV		
3738				+		DMOVI	CTUCHT AGO		
3739	2081	۸٥	06	т		DMOVI LDY	STKCNT, ACC	;	copy SP and SP save to buffer
3740	2083		20DF				#\$06		
3741	2000	20	2001			JSR	SVGMMV		
3742	2086	20	1E16			JSR	DWRBUF		
3743 ·	2089		4E			BCS	SVGMFL		write it out
3744	2000	-	-T b			DCS	SVGWIFL	;	fail if error
3745	208B	Δ2	00			LDX	#\$00		
3746	2005		•	+		DMOVI	·	;	copy lowest 256 bytes of stack
3747	2095	ΔΩ	00	•		LDY	STKLIM,ACC #\$00	;	to buffer
3748	2097	_	20DF			JSR	SVGMMV		
3749						0311	3 V CIMINI V		
3750	209A	20	1E16			JSR	DWRBUF		write it out
3751	209D		3A			BCS	SVGMFL		fail if error
3752							o v dim L	,	Tall II ellor
3753	209F	A 2	00			LDX	#\$00		copy bigh 102 bytes of start
3754				+		DMOVI	STKLIM+\$0100,ACC	:	copy high 192 bytes of stack to buffer
3755	20A9	ΑO	CO			LDY	#\$C0	,	to builter
3756	20AB	20	20DF			JSR	SVGMMV-		
3757									
3758	20AE	20	1E16			JSR	DWRBUF		write it out
3759	20B1	B0	26			BCS	SVGMFL		fail if error
3760								,	14.11.11.01
3761				+		DMOV	FRZMEM, ACC	,	figure out how many pages of
3762	20BB	ΑO	0E			LDY	#HDRIMP	•	impure storage there are to be
3763				+		MOV	<(FRZMEM),Y>,ACD		written out, and set up for first
3764	20C1	E6	E8			INC	ACD		one
3765								,	
3766	20C3	20	1E1E	L200	03:	JSR	DWRNXT		write one page of impure storage
3767	20C6	В0			ı	BCS	SVGMFL		fail if error
3768	20C8	E6	E7			INC	ACC+1		increment buffer address
3769				j=c+	- 1	DECBN	ACD, L20C3		decrement page count, loop if more
3770								•	February, 1998 11 more
3771	20CE		1E1E	:		JSR	DWRNXT	;	write final page
3772	20D1	во	06		1	BCS	SVGMFL		fail if error

Infocom	ERLOG	IC interpret	er disasseml	oly, 5/	27/84 MACRO-80 33	18-Sep-81	PAGE 1	-92
3773								
3774	20D3	20 2005		JSR	L2005	; make sure w	e have game	disk
3775	20D6	4C 0B84		JMP	PREDTR	; return true	(no error)	
3776								
3777	20D9	20 2005	SVGMFL:	JSR	L2005	; make sure w		disk
3778	20DC	4C 0B8D		JMP	PREDFL	; return fals	e (error)	
3779								
3780								
3781	20DF	88	SVGMMV:	DEY		; copy memory	into buffer	to write
3782			+	MOV	<(ACC),Y>,< <buffer,x>></buffer,x>			
3783	20E5	E8		INX	•			
3784			+	CPYBN	<#\$00>,SVGMMV	; if more, lo	ор	
3785	20EA	60		RTS		; no, return		
3786								
3787				PAGE				

3788 3789								
3790 3791	20EB	20	1EBD	OPRSGM:	JSR	L1EBD	;	setup for disk I/O
3792 3793	20EE	20	1DFA	+	JSR JCS	DRDBUF RSGMFL		read in a bufferful
3794	0056	• •	0.0					fail if error
3795 3796	20F6 20F8		00 02		LDX LDY	#\$00 #HDRREL	;	check release of game, fail if wrong
3797	20FA		BA		LDA	(FRZMEM),Y		
3798				+	CMPBN	<buffer, x="">, L210A</buffer,>		
3799	2101	E8			INX	2011 211,77 , 221071		
3800	2102	С8			INY			
3801	2103	B1	BA		LDA	(FRZMEM),Y		
3802				+	CMPBE	<buffer,x>,L210D</buffer,x>		
3803	210A	4C	218E	L210A:	JMP	RSGMFL.		
3804 3805	210D	ΑO	1 1	1.04.00				
3806	2100	AU	1 1	L210D:	LDY MOV	#HDRFLG+1	;	preserve SCRIPT flag
3807				7	MUV	<(FRZMEM),Y>,MDFLAG		
3808	2113	E8			INX			
3809				+	DMOVI	PRGIDX.ACC	;	restore PC
3810	211C	ΑO	03		LDY	#\$03		
3811	211E	20	2194		JSR	RSGMMV		
3812				+	MOV	<#\$00>, PRGUPD		
3813						•		
3814				+	DMOVI	LOCVAR, ACC	;	restore local variables
3815	212D	A0			LDY	#\$1E		
3816 3817	212F	20	2194		JSR	RSGMMV		
3818				+	D1401/7	CT C T C C		
3819	213A	ΑO	06	Ŧ	DMOVI LDY	STKCNT, ACC	;	restore SP and SP save
3820	213C		2194	•	JSR	#\$06 RSGMMV		
3821					33K	Kadmim A		
3822	213F	20	1DFA		JSR	DRDBUF		read a bufferful
3823	2142	В0	4A		BCS	RSGMFL.		fail if error
3824							,	
3825	2144	A 2	00		LDX	#\$00	;	restore first 256 bytes of stack
3826	0145	• •	0.0	+	DMOVI	STKLIM, ACC		, , , , , , , , , , , , , , , , , , , ,
3827 3828	214E 2150	A0			LDY	#\$00		
3829	2130	20	2194		JSR	RSGMMV		
3830	2153	20	1DFA		JSR	DRDBUF		
3831	2156	B0			BCS	RSGMFL		read a bufferful
3832			,00		D C3	KSGMI L	;	fail if error
3833	2158	A2	00		LDX	#\$00		restore last 102 butes of start
3834				+	DMOVI	STKLIM+\$0100,ACC	,	restore last 192 bytes of stack
3835	2162	Α0			LDY	#\$C0		
3836	2164	20	2194		JSR	RSGMMV		
3837								
3838	0165	• •	ó.c	+	DMOV	FRZMEM,ACC	;	figure out how many pages of
3839 3840	216F	ΑO	UE		LDY	#HDRIMP	;	impure storage there are to be
3840 3841	2175	E6	EΩ	+	MOV	<(FRZMEM),Y>,ACD	;	read in, and set up to read first
3842	2110	LU	LU		INC	ACD	;	one

						16 3ep-81 PAGE 1-94
3843 3844 3845 3846 3847	2177 217 A 217C	20 1E02 B0 12 E6 E7	L2177:	JSR BCS INC DECBN	DRDNXT RSGMFL ACC+1 ACD,L2177	; read in next page of impure storage ; fail if error ; increment buffer pointer ; decrement page count, loop if more
3848 3849 3850 3851	2182 2184 2186	A5 EA A0 11 91 BA		LDA LDY STA	MDFLAG #HDRFLG+1 (FRZMEM),Y	; restore SCRIPT flag
3852 3853 3854	2188 218B	20 2005 4C 0B84		JSR J m p	L2005 PREDTR	; make sure we have game disk ; return true (no error)
3855 3856 3857 3858	218E 2191	20 2005 4C 0B8D	RSGMFL:	JSR JMP	L2005 PREDFL	; make sure we have game disk ; return false (error)
3859 3860	2194	88	RSGMMV:	DEY MOV	<buffer,x>,<<(ACC),Y>></buffer,x>	; copy buffer to memory (read)
3861 3862	219A	E8		INX	15011 ER, X2, \(\(\frac{1}{400}\), \(\frac{1}{2}\)2	
3863 3864	219F	60	+	CPYBN RTS	<#\$00>,RSGMMV	
3865	÷			PAGE		

Infocom $r_{\rm col}$ ERLOGIC interpreter disassembly, 5/27/84

3866							
3867							
3868	21AO	E6 4E	L:	21AO:	INC	RNDLOC	; get a 'random' number
3869	21A2	E6 4F			INC	RNDLOC+1	
3870			+		DMOV	RNDLOC, ACC	
3871	21AC	60			RTS		
3872							
3873	21AD	2D 2D	20 F	NDMSG:	DB	' END OF SESSION'	
3874	21B0	45 4E				2.10 0. 020010.1	
3875	21B3	20 4F					
3876	2186	20 53					
3877	21B9	53 53					•
3878	21BC	4F 4E	20				
3879	21BF	2D 2D					
3880	0014		E	NMSLN	EQU	*-ENDMSG	
3881							
3882	2101	49 4E	54 F	TLMSG:	DB	'INTERNAL ERROR #'	
3883	21C4	45 52	4E				
3884	2107	41 4C	20				
3885	21CA	45 52					
3886	21CD	4F 52					
3887	2100	23					
3888	0010	20	E-	TMSLN	EQU	*-FTLMSG	
3889	0010		•	IMSEN	LQU	· I I EMISO	
3890	2101	20 101		ATAL:	100	DONTRE	63b
	Z 1 D 1	20 101	U F	AIAL:	JSR	PRNTBF	; flush anything left in buffer
3891					5		
3892			+		DMOVI	FTLMSG, ACC	; output fatal message
3893	21DC	A2 10			LDX	#FTMSLN	
3894	21DE	20 1E2	29		JSR	OUTMSG	
3895							
3896			+		DPUL2	ACC	; output address where error detected
3897	21E7	20 14F	5		JSR	PRNTNM	
3898							
3899	21EA	20 101	0 0	PENDS:	JSR	PRNTBF	; flush anything left in buffer
3900							,
3901			+		DMOVI	ENDMSG, ACC	; output end of session message
3902	21 F5	A2 14			LDX	#ENMSLN	, barpar cha or session message
3903	21F7	20 1E2	, a		JSR	OUTMSG	
3904	2117	20 122	. 3		3310	COTINISC	
	21FA	20 101	Λ.		ICD	PRNTBF	fluch the buffer
3905	ZIFA	20 161			JSR	FRITIOI	; flush the buffer
3906	0.450						
3907	21FD	4C 21F	יט H.	ALT:	JMP	HALT	; die horribly
3908						_	
3909					.DEPHASE	Ē	
3910							
3911					END	START	

Macros: ADD CMPBN CMPJSG CPXRGT DADDB1 DDEC2 DECABP DINC DOR DROR DROR DSUBB2 DTS2RE DTSTBN DXBEQ DYBMI IYBNE JGT JSRCC JSRLT PSH RTSGE RTSPL TSTABM	CMPBE CMPBP CMPJSN CPYBN DADDB2 DECA DECBE DLSR DPSH DSBC DTS2BE DTS2RN DTSTJE DXBMI DYBNE JCC JLT JSRCS JSRMI PUL RTSGT STR TSTABN	CMPBG CMPJE CMPRE D1COMP DAND DECABE DECBN DMOV DPUL DSTZ DTSZBN DTST DTSTJN DXBNE DYBPL JCS JMI JSREQ JSRNE RTSCC RTSLT SUB TSTABP	CMPBL CMPJL CPXBE DADC DASL DECABM DECJE DMOVI DPUL2 DSUB DTS2JE DTST2 DTSTRE DXBPL INCA JEQ JNE JSRGE JSRPL RTSCS RTSMI TSTAJE	CMPBM CMPJSE CPXBG DADD DDEC DECABN DECJN DMOVI2 DROL DSUBB1 DTS2JN DTSTBE DTSTRN DYBEQ IXBNE JGE JPL JSRGT MOV RTSEQ RTSNE TSTABE TSTABE
Symbols: 099F0000 09C80003 0A380006 0A870009 0BD4000C 0CB1000F 0D380012 0D8D0015 0E480018 101A001B 1084001E 12CD0021 132A0024 13E00027 147B002A 14CA002D 16110030 16430033 171F0036 173D0039 17F6003C 1A43003F 1B820042 1E150045 1F630048 0CE4 ACB 169D ADVPPT 0086 ARG3	09A6 09D2 0A42 0B4D 0BE5 0CDA 0D4E 0E00 0E8F 104A 10B1 131A 13BF 145B 145B 145C 1624 16FF 172C 174C 1800 1A5F 1B9C 1E24 1F94 00E6 0082	000100040007000A000D0010001300160019001C001F002200250028002800280028003100340037003A003A003A003A003A	09BE 09FE 0A70 0B9B 0BFC 0CE6 0D70 0E2F 0F10 1078 12B8 1321 13C9 1466 14BF 1552 163D 1710 1736 1756 18C4 1B48 1E08 1F45 20F6 00E8 0084	0002 0005 0008 0008 000E 0011 0014 0017 001A 001D 0020 0023 0026 0029 0025 0035 0038 0038 0038 0038 0038 0038 0041 0044 0047 0044 0047 0044 0047 0044 0044 0047 0044 0047 0044 0044 0047 0044 0047 0044 0044 0047 0044 0047 0044 0047 0044 0047 0044 0047 0044 0047 0044 0047 0048 0048 0048 0048 0048 0048 0048 0048 0048 0047 0044 0047 0044 0047 0048

PAGE

0097	AUXPPG	0096	AUXUPD	1B3F	BFCHAR
0200	BUFFER	00EC	CHRPT2	00EB	CHRPTR
FC9C	CLREOL	1B16	CLRSCR	FDED	COUT
FDF0	COUT1	000D	CRCHAR	1A05	CRNWRD
0036	CSWL	0024	CURSRH	0025	CURSRV
1DC4	DCT	1DC8	DISKIO	15AD	DIVIDE
18FC	DOASCI	1915	DOCRLF	18BE	DONEXT
193B	DOSBWD	1910	DOSPAC	18EB	DOSPCL
1EOD	DRDBKF	1E08	DRDBLK	1DFA	DRDBUF
1E02	DRDNXT	1BF5	DSPBUF	0A11	DSPTCH
1E16	DWRBUF	1E1E	DWRNXT	21AD	ENDMSG
0014	ENMSLN	21D1	FATAL	000C	FFCHAR
2C00	FIRFLC	1B1E	FNDMEM	1897	FNDPAG
00BA	FRZMEM	00BC	FRZPGS	17E8	FTAXBA
17DB	FTAXWD	2101	FTLMSG	0010	FTMSLN
173E	FTPRBA	OAAB	FTPRBY	0AB5	FTPRWD
1D65	GETLIN	FD6F	GETLN1	19B9	GETNYB
0098	GLBVAR	0A04	GODOIT	1693	GTPLEN
168E	GTPNUM	OAEF	GTVARA	0AE8	GTVARP
1406	GTVCBA	OAC2	GTVRA1	21FD	HALT
001A	HDRCKA	001C	HDRCKV	0010	HDRFLG
0004	HDRFRZ	000C	HDRGBV	000E	HDRIMP
0000	HDRIRL	0002	HDRREL	0018	HDRSBW
0012	HDRSER	0002	HDRSTR	000A	HDRTHG
0001	HDRTYP	0008	HDRVCB	FC58	HOME
1AF7	INITSC	0032	INVFLG		
1DB3	108	1DBB	IOBBUF	00D3	INWORD
1DB5	IOBDRV	1DBB 1DB8	IOBBUT	1DBF	IOBCMD
1DB3	IOBURV	0805		1DB4	IOBSLT
0897	L0897		L0805	084A	L084A
0897 09AF		08B6	L08B6	090A	L090A
OA2B	LO9AF	09D7	L09D7	09ED	L09ED
	LOA2B	0A45	L0A45	0A73	LOA73
A8A0	LOA8A	0A98	LOA98	OADO	LOADO
OAD6	LOAD6	0B02	L0B02	0B26	LOB26
0B60	LOB60	0B94	L0B94	OB9C	LOB9C
OBAD	LOBAD	OBC3	LOBC3	OBDA	LOBDA
OBFC	LOBFC	0C17	LOC17	OC1A	LOCIA
OCA5	LOCA5	OCDA	LOCDA	OCE6	LOCE6
OCFA	LOCFA	OD4E	LOD4E	ODB7	LODB7
ODD2	LODD2	ODE4	LODE4	0E2F	LOE2F
0E4C	LOE4C	0E9D	L0E9D	0EB7	LOEB7
0ECF	LOECF	0F08	LOFO8	0F10	LOF10
0F23	LOF23	0F97	L0F97	OFA1	LOFA1
OFD1	LOFD1	100B	L100B	103B	L103B
105E	L105E	106C	L106C	107E	L107E
108E	L108E	10A5	L10A5	10B7	L10B7
10BD	L10BD	1104	L1104	1107	L1107
110D	L110D	1111	L1111	1139	L1139
113F	L113F	1143	L1143	1173	L1173
117F	L117F	118E	L118E	119D	L119D
11A0	L11A0	11B4	L11B4	11F2	L11F2
1220	L1220	1230	L1230	124C	L124C
12AC	L12AC	12BE	L12BE	1207	L12D7
1310	L1310	1332	L1332	135C	L135C
137A	L137A	1382	L1382	13BA	L13BA
13DA	L 13DA	13E0	L13E0	13E4	L13E4

1D98

1DDF

1E2F

1E5A

1E7B

1E90

1F12

1F4C

1FDF

2040

210D

0001

00DF

00E1

009A

0800

1COA

00BE

OF4A

0A2E

L1D98

L1DDF

L1E2F

L1E5A

L1E7B

L1E90

L1F12

L1F4C

L1FDF

L2040

L210D

LC40

LDF

LE1

LOCVAR

MAINOR

MOREMS

MRUPAG

OPCGPB

OPAND

1D8B

1DA7

1DE7

1E3D

1E69

1E7E

1E9A

1F3A

1FB3

2005

20C3

2177

00D9

0100

000A

00BF

00EA

1862

191F

11A3

L1D88

L1DA7

L1DE7

L1E3D

L1E69

L1E7E

L1E9A

L1F3A

L1FB3

L2005

L20C3

L2177

LDORG

LFCHAR

LRUPAG

MDFLAG

MRKPAG

NEWMOD

OPCALL

LD9

1D94

1DB1

1DED

1E59

1E6C

1E8D

1EBD

1F48

1FC0

2023

210A

21A0

OODE

00E0

00ED

BFFF

098F

0006

10C3

0A66

L1D94

L1DB1

L1DED

L1E59

L1E6C

L1E8D

L1EBD

L1F48

L1FC0

L2023

L210A

L21A0

LINCNT

LSTFLC

MNLOOP

MRMSLN

OPADD

OPCGPA

LDE

LE0

PAGE

S-2

0	000000	25			
0A17	OPCGPC	09AA	OPCGPD	0C7C	OPCKSM
0F80	OPCLRA	0080	OPCODE	0C72	OPCRLF
0D60	OPDEC	OEE7	OPDECB	1118	OPDIV
0D81	OPDSTT	21EA	OPENDS	OFEC	OPGTBY
OCF3	OPGTCH	12DC	OPGTLN	109E	OPGTNP
1008	OPGTP	1069	OPGTPA	0D20	OPGTPL
ODOE	OPGTPR	OCE9	OPGTSB	0FD2	OPGTWD
0D43	OPINC	0EF5	OPINCB	OE7C	OPJUMP
000E	OPMAX1	0010	OPMAX2	0019	OPMAX3
A000	OPMAX4	OEAO	OPMOVE	OFA4	OPMOVT
116B	OPMTCH	10E3	OPMUL	0EA8	OPNOT
0C53	OPNULL	0F3B	OPOR	14E5	OPPRCH
14EA	OPPRNM	1C8A	OPPRST	0DE2	OPPRTN
0D73	0PPSB	0C28	OPPSI	0C54	OPPSIC
0E92	OPPSW	. 1288	OPPTBY	12A9	OPPTP
125F	OPPTWD	1560	OPPULL	1555	OPPUSH
114A	OPRMD	1536	OPRNDM	20EB	OPRSGM
0E06	OPRTN	0C23	OPRTNF	0C18	
0064	OPRTNV	0F6D	OPSETA	10D3	OPRTNT
204B	OPSVGM	090D	OPTAB1	0929	OPSUB
0949	OPTAB3	097B	OPTAB4		OPTAB2
0F59	OPTSTA	OCDD	OPTSTZ	0F13	OPTINT
1E29	OUTMSG	OODA	PKWORD	1B91	OUTBUF
00D0	PNYBCN	OOEE	PRCSWL	00D1	PNYBBF
0B84	PREDTR	008A	PRGIDX	0B8D	PREDFL
008D	PRGMPT	0090		00 8 8	PRGLPG
00CF	PRMMOD	1C10	PRGPPG	008F	PRGUPD
18B4	PRNTST		PRNTBF	14F5	PRNTNM
0779	PRTWDT	0033	PROMPT	1BA1	PRTBUF
0AC9	PTVRA1	0B46	PTVARA	0B35	PTVARP
OB2A	PTVRPZ	0B32	PTVRP1	0B2C	PTVRPA
FD0C	RDKEY	1720	PULLWD	16F4	PUSHWD
218E		004E	RNDLOC	0000	RNGDBG
2400	RSGMFL	2194	RSGMMV	2900	RWTS
	RWTSOR	00E2	SBWDPT	0006	SCMSLN
1C7E	SCORMS	007F	SECPTK	13D2	SEPTAB
1788	SETAXB	1709	SETAXW	1629	SETUPA
1669	SETUPP	16A7	SETUPT	1D57	SHWMSG
1995	SPCLCH	0800	START	03E8	STCKLC
00E0	STCKMX	00C8	STKCNT	00CD	STKCSV
0228	STKLIM	0009	STKPNT	00CB	STKPSV
00F3	STLTYP	20D9	SVGMFL	20DF	SVGMMV
00B8	SWPMEM	00BD	SWPPGS	0009	TBCHAR
0000	THGATT	0006	THGCHD	0004	THGPAR
0007	THGPRP	0005	THGSIB	1C84	TIMEMS
0005	TMMSLN	OOCE	TMPMOD	1AAB	TSTCHR
19AD	TSTMOD	0000	VERSN	2200	VMT1LC
2280	VMT2LC	2300	VMT3LC	2380	VMT4LC
0000	VMTAB1	00C2	VMTAB2	00C4	VMTAB3
00C6	VMTAB4	2200	VMTORG	FC22	VTAB
0023	WNDBOT	0020	WNDLFT	0022	WNDTOP
0021	WNDWDT	007F	ZPORG		

0000	1228	1228#
0001	1229	1229#
0002	1249	1249#
0003	1250	1250#
0004	1251	1251#
0005	1277	1277#
0006	1307	1307#
0007	1308	1308#
0008	1326	1326#
0009	1332	1332#
000.	1422	1422#
000A	1456	1456#
000C	1481	1481#
000D	1483	1483#
000E	1493	1493#
000E	1575	1575#
	1591	1573#
0010	1591	1591#
		1624#
0012	1624 1634	1634#
0014	1646	1646#
0015	1664	1664#
0016	1704	1704#
0017	1722	1722#
0018	1729	1729#
0019	1752	1752#
001A	1789	1789#
001B	1903	1903#
001C	1924	1924#
001D	1941	1941#
001E	1944	1944#
001F	1963	1963#
0020	2193	2193#
0021	2203	2203#
0022	2229	2229#
0023	2230	2230#
0024	2232	2232#
0025	2299	2299#
0026	2303	2303#
0027	2314	2314#
0028	2369	2369#
0029	2370	2370#
002A	2376	2376#
002B	2377	2377#
002C	2400	2400#
002D	2401	2401#
002E	2426	2426#
002F	2460	2460#

	0500	0500#												
0030	2530	2530#												
0031	2540	2540#												
0032	2553	2553#												
0033	2554	2554#												
0034	2653	2653#												
0035	2656	2656#												
0036	2660	2660#												
	2665	2665#												
0037		2667#												
0038	2667				•									
0039	2669	2669#												
003A	2684	2684#												
003B	2688	2688#												
003C	2782	2782#												
003D	2786	2786#												
003E	2887	2887#												
003F	3058	3058#												
0040	3066	3066#												
0041	3185	3185#												
	3223	3223	3223#											
0042		3240#	3223#											
0043	3240													
0044	3512	3512#												
0045	351 7	3517#												
0046	3521	3521#												
0047	3634	3634#												
0048	3643	3643#												
0049	3653	3653#												
004A	3794	3794#												
ACB -	174#	1070	1070	1087	1089	1395	1398	1400	1400	1400	1400	1402		
ACD .	1404	1433	1436	1438	1438	1438	1438	1440	1442	1485	1488	1493		
	1493	1496	1496	1498	1501	1672	1672	1673	1673	1674	1683	1683		
	1684	1684	1685	1699	1701	1702	1702	1719	1719	1722	1722	1722		
		1728	1729	1729	1729	1770	1770	1775	1775	1782	1782	1786		
	1726			1819	1821	1822	1824	1833	1837	1849	1854	1907	•	
	1786	1800	1803			1928	1928	1988	1988	1989	1991	2007	-	
	1911	1917	1919	1925	1927				2116	2121	2126	2127		
	2007	2008	2010	2024	2024	2027	2027	2115		2377	2377	2377		
	2285	2287	2356	2358	2370	2370	2370	2370	2373			2431		
	2401	2401	2401	2402	2402	2403	2403	2405	2405	2429	2429			
	2456	2456	2459	2459	2480	2487	2487	2489	2489	2490	2490	2503		
	250 6	2516	2516	2523	2554	2557	2564	2566	2575	2577	2578	2578		
	2640	2643	2648	2649	2725	2725	2823	2823	3044	3054	3056	3062		/
	3064	3074	3076	3083	3085	3496	3497	3512	3512	3521	3521	3629		
	3629	3634	3634	3634										
ACC	175#	1069	1069	1082	1084	1260	1261	1275	1275	1284	1286	1296		
ACC	1296	1312	1312	1318	1318	1328	1328	1334	1334	1342	1342	1350		
				1376	1376	1378	1378	1389	1391	1402	1404	1412		
	1352	1360	1362		1420	1427	1429	1440	1442	1463	1465	1471		
	1414	1417	1417	1420			1429	1481	1482	1482	1483	1483		
	1473	1475	1477	1480	1480	1481			1608	1610	1619	1619		
	1483	1485	1491	1547	1547	1563	1571	1608			1638	1638		
	1620	1623	1623	1624	1624	1624	1634	1634	1635	1635	1030	1036		

, "

								4					
	1646	1646	1646	1652	1652	1662	1666	1666	1669	1676	1680	1685	
	1686	1686	1688	1689	1699	1701	1702	1702	1704	1704	1715	1720	
	1726	1728	1733	1733	1736	1736	1739	1740	1744	1744	1751	1751	
	1752	1752	1752	1756	1756	1766	1766	1769	1769	1776	1776	1786	
	1786	1787	1787	1796	1799	1802	1809	1809	1812	1812	1835	1839	
	1850	1855	1861	1861	1868	1868	1870	1872	1875	1876	1876	1879	
	1884	1884	1889	1889	1891	1893	1917	1919	1925	1927	1928	1928	
	1930	1930	1931	1944	1944	1946	1947	1949	1951	1951	1951	1951	
	1975	1975	1981	1981	1987	1987	1998	1998	1999	1999	2006	2006	
	2016	2016	2017	2017	2023	2023	2027	2027	2060	2060	2064	2065	
	2068	2068	2071	2071	2090	2092	2099	2101	2104	2106	2116	2118	
	2120	2123	2125	2132	2150	2153	2156	2156	2156	2156	2159	2161	
	2171	2174	2177	2177	2177	2177	2180	2205	2208	2211	2327	2333	
	2338	2340	2341	2341	2341	2341	2345	2348	2356	2358	2361	2361	
	2362	2366	2369	2369	2369	2376	2376	2376	2386	2390	2394		
	2400	2400	2400	2405	2405	2419	2419	2424	2428	2428		2398	
	2460	2460	2467	2467	2485	2485	2489	2489	2499	2428	2459	2459	
	2502	2505	2508	2510	2513	2513	2516	2516	2517		2500	2500	
	2533	2535	2536	2553	2553	2554	2554	2564		2517	2521	2532	
	2578	2579	2588	2595	2615	2617			2566	2575	2577	2578	
	2621	2622	2623	2626	2628	2631	2617	2621	2621	2621	2621	2621	
	2648	2655	2657	2664	2666	2696	2631	2633	2635	2641	2644	2646	
	2741	2748	2751	2765	2767	2794	2696	2723	2723	2724	2724	2740	
	2846	2847	2849	2852	2856	2/94	2794	2821	2821	2822	2822	2843	
	3000	3044	3044	3045	3166	2858	2865	2867	2869	2957	2959	3000	
	3175	3311	3311	3359	3370	3166	3167	3168	3170	3171	3173	3174	
	3428	3495	3495			3370	3380	3380	3385	3392	3400	3409	
	3642	3642		3511	3511	3520	3520	3531	3595	3595	3623	3623	
	3703	3703	3643 3707	3643	3643	3649	3649	3652	3652	3653	3653	3653	
	3755	3755		3707	3731	3731	3735	3735	3739	3739	3747	3747	
	3827	3827	3762	3762	3768	3783	3810	3810	3815	3815	3819	3819	
	3897	3897	3835	3835	3839	3839	3845	3861	3871	3871	3893	3893	
4.00	3697 176#		3902	3902									
ACD		1720	1730	1820	1823	1834	1838	1847	1852	2112	2113	2130	
	2219	2221	2230	2230	2230	2232	2254	2256	2264	2266	2267	2271	
	2272	2274	2277	2281	2294	2295	2297	2304	2305	2353	2369	2376	
	2377	2382	2400	2427	2431	2434	2440	2478	2478	2479	2479	2485	
	2485	2485	2485	2486	2486	2490	2490	2491	2491	2498	2498	2499	
	2499	2512	2512	2517	2517	2518	2518	2556	2556	2560	2560	2887	
	2895	2902	2915	2917	3043	3047	3050	3058	3058	3062	3066	3069	
	3070	3078	3078	3087	3087	3527	3529	3530	3533	3535	3764	3764	
• • • •	3770	3841	3841	3847									
ADD	357#	1055	1079	1107	1112	1115	1490	1575	1626	1722	2292	2360	
	2437	2621	2630	2710	2723	2808	2821	2894	3053	3097	3505	3634	
	3643	3653				7							
ADVPPT	1903	1941	1963	1967	2194	2605#							
ARG1	118#	1260	1261	1312	1312	1328	1328	1513	1515	1547	1547	1563	
	1563	1575	1575	1575	1588	1591	1597	1597	1599	1604	1615	1623	
	1623	1631	1635	1643	1652	1652	1659	1671	1682	1694	1744	1744	
	1751	1751	1756	1756	1760	1766	1766	1769	1769	1775	1775	1791	

	•							•						Ţ	
	• 700	1802	1809	1809	1812	1812	1861	1865	1875	1884	1884	1889			
	1799		1975	1981	1981	1987	1987	2006	2006	2023	2023	2036			
	1889	1975			2048	2059	2059	2076	2076	2149	2152	2170			
	2038	2041	2043	2046		2246	2250	2259	2263	2300	2412	2419			
	2173	2217	2217	2217	2217		2547	2571	3456	3456	3466				
	2419	2456	2456	2467	2467	2473			1580	1770	1770	1776			
ARG2	119#	1334	1334	1559	1561	1566	1567	1578			1812	1812			
	1776	1782	1782	1787	1787	1794	1799	1802	1809	1809		1940			
	1861	1861	1870	1883	1883	1884	1884	1889	1889	1902	1912				
	1958	1962	1975	1975	1981	1981	1988	1988	2007	2007	2024	2024			
	2038	2040	2117	2122	2145	2147	2151	2168	2172	2191	2218	2218			
	2218	2218	2223	2226	2229	2244	2281	2285	2287	2293	2293	2549			
	120#	1563	1568	1582	2043	2045	2159	2161	2180	2205	2208	2211			
ARG3		1563	1565	2048	2050										
ARG4	121#		1264	1311	1336	2033	2112								
ARGCNT	116#	1223		1576	2740	2750	2776	2780	2962	2967					
AUXIDX	130#	1522	1528		1529	1578	1580	2741	2742	2753	2756	2786			
AUXLPG	129#	1523	1523	1529		2823	2823	2828	2829	2963	2963	2966			
	2786	2788	2790	2794	2794	2623	2023	2020	2020	2000					
	2966					0010									
AUXMPT	131#	1531	1531	2777	2809	2810									
AUXPPG	133#	2719	2795	2803	2822	2826	0770	2784	2812	2968					
AUXUPD	132#	1040	1524	1530	2720	2743	2773	2446	2895	2926	3182#	3395			
BFCHAR	1538	1540	1551	1553	2413	2438	2443		2090	2320	0102#	0000			
	3403	3406	3414	3416	3532	3601	3612	3620	3460	3511	3511	3520			
BUFFER	43#	3197	3209	3224	3226	3272	3291	3442	3460	3311	3311	0320			
	3520	3725	3728	3783	3799	3803	3861								
CHRPT2	181#	3213	3220	3221	3229			0000	0000	2221	3450	3453			
CHRPTR	180#	3182	3201	3214	3225	3227	3271	3290	3298	3321	3450	3433			
CLREOL	97#	3317	3364	3421	3659										
CLRSCR	3158#	3591								0011	2012	2039			
CMPBE	939#	1085	1795	1901	1922	1939	1961	1992	1993	2011					
· · · · · · ·	2044	2049	2190	2201	2316	2332	2838	2853	2868	2937	3092	3208			
	3323	3360	3467	3802								0.400			
CMPBG	954#	1279	1293	1315	1339	1384	1422	2367	3101	3105	3401	3463			
CMPBG	3666	1210	.200												
CMPBL	949#	1229	2385	2389	2393	2397	2550	2693	2791	2889	2890	3100			
CMPDL	3104	3185	3186	3309	3455	3462	3665								
СМРВМ	959#	3388	0.00												
	944#	1577	1579	1581	1587	1670	1681	2037	2042	2047	2647	2718			
CMPBN	2816	2866	2897	3169	3172	3627	3662	3698	3700	3711	3798				
		3410	2037	0.00	0.7-										
CMPBP	964#	1590	3184												
CMPJE	969#		1228	1276			•								
CMPJL	974#	1227		1307	2231										
CMPJSE	979#	1249	1250	1307	2201										
CMPJSG	989#	2659	0000												
CMPJSN	984#	1923	2202												
CMPRE	994#	2228	0000	2000	2260	3273	3333								
COUT	100#	3260	3264	3266	3268	3213	3333								
COUT 1	101#	3292	3325	3430											
CPXBE	999#	3270	3289												

	Ψ,											
CPXBG	1004#	3198										
CPXRGT	1009#	3222										
CPYBN	1014#	3784	3862				0.460					
CRCHAR	68#	1537	1550	2309	2925	3185	3468					
CRNWRD	2281	3037#					0070	0000	2202	3331	3331	3332
CSWL	88#	3249	3249	3252	3252	3279	3279	3282	3282	3331	3331	0002
03112	3332	3335	3335	3336	3336				0070	0000	2424	3647
CURSRH	8.1#	3250	3281	3317	3317	3353	3354	3367	3372	3382	3424	3041
CONSINI	3659											
CURSRV	82#	3353	3354	3424								
DICOMP	312#	1765									0010	0041
DADC	321#	1400	1438	1623	1884	1889	1975	2156	2177	2217	2218	2341
DADC	2484	2578										0040
D 4 D C	240 4 347#	1399	1437	1622	1883	1888	1974	2155	2176	2216	2217	2340
DADD	2577	1000	, ., .									
0.4500.1	2577 383#	2368	2376	2399								
DADDB1			1721	3633	3642	3652						
DADDB2	405#	1574	1/41	3033	00-12							
DAND	303#	1811	1007	1998	2075	2559	2621	2621	2621			
DASL	238#	1882	1997	1990	2013	2000						
DCT	3481	3490#	1.400	1623	1645	1751	2400	2652	2655			
DDEC	437#	1480	1482	1623	1040	1751	2-100					
DDEC2	441#	1728		0007	2061							
DECA	518#	1423	2946	3007	3061							
DECABE	889#											
DECABM	904#											
DECABN	894#	3006										
DECABP	899#	3060										
DECBE	869#	2112	3077	3086			00.46					
DECBN	874#	1729	2129	2439	3534	3769	3846					
DECJE	879#	3057										
DECJN	884#	3065							2004	2006	2687	2785
DINC	429#	1492	1633	1703	1943	2459	2552	2553	2664	2666	2007	2700
51110	3511	3520										
DISKIO	3493#	3513	3522									
DIVIDE	2013	2025	2429	2457	2497#							
DLSR	252#	2015	2016								.70.	1711
DMOV	445#	1068	1311	1327	1333	1522	1528	1530	1546	1651	1701	1711
DIMOA	1732	1735	1743	1750	1755	1768	1769	1774	1775	1781	1785	1786
	1860	1927	1986	1987	2005	2006	2022	2023	2026	2067	2070	2136
		2455	2458	2466	2488	2489	2498	2516	2695	2722	2724	2793
	2418	2822	2999	3251	3278	3331	3334	3494	3761	3838	3870	
5440147	2820		1046	1047	1048	1049	1066	1069	1274	1295	1317	1341
DMOVI	456#	1042		2478	2499	2555	3310	3369	3379	3510	3519	3594
	1718	2059	2428		3702	3706	3730	3734	3738	3746	3754	3809
	3622	3641	3648	3651	3892	3901	0.00		- · - -			
	3814	3818	3826	3834	3692	3901						
DMOVI2	467#	3165										
DOASCI	2902	2909#										
DOCRLF	2903	2925#		00.40	2071							
DONEXT	2885#	2896	2935	2940	2971							

.

	₹.,							~.				
DOR	294#	1808										
DOSBWD	2890	2946#										
DOSPAC	2888	2922#										
DOSPCL	2898	2901#								0.407	0000	20.62
DPSH	502#	1375	1416	1634	1665	1672	1683	1867	2477	2497	2960	2962
DI 311	3248	3330										
DPUL	492#	1377	1419	1637	1671	1682	1685	1875	2490	2517	2965	2968
DPUL	3281	3335										
DPUL2	497#	3896										
	1070	1089	2726	2824	3515#			4				
DRDBKF		3515	2720	202.	00.0"							
DRDBLK	3512#		3822	3830								
DRDBUF	3510#	3792	3622	3030								
DRDNXT	3511#	3843	0510									
DROL	280#	2511	2512	0515								
DROR	266#	2485	2486	2515								
DSBC	334#	1951	1981	2405								
DSPBUF	3240	3287#	3363	3420								
DSPTCH	1284	1286	1286#	1286								
DSTZ	232#											
DSUB	352#	1950	1980	2404								
DSUBB 1	394#	2369	2375						00=0	0050		
DSUBB2	417#	1481	1483	1624	1646	1729	1752	2401	2653	2656		
DTS2BE	789#	•										
DTS2BN	794#	2058	2401									
DTS2JE	799#											
DTS2JN	804#											
DTS2RE	809#											
DTS2RN												
DTST	749#	1480	1482	1496	1597	2230	2428					
		2059	2402	50								
DTST2	784#		1481	1495	2427							
DTSTBE		1479	1401	1433	4741							
DTSTBN												
DTSTJE		1500										
DTSTJN		1596										
DTSTRE		2 229										
DTSTRN			0750	0750								
DWRBUF		3742	3750	3758								
DWRNXT		3766	3771									
DXBEQ	829#	2040	2045									
DXBMI	849#	3501				05.0	2071	2200	3432	3468		
DXBNE	819#	1059	2034	2318	2487	2513	2871	3209	3432	3400		
DXBPL	839#	2608	3093									
DYBEQ	834#											
DYBMI	854#											
DYBNE	824#	2238	3041	3634								
DYBPL	844#											
ENDMSG		3880	3902	3902								
ENMSLN		3902									0545	0000 "
FATAL	1129	1162	1177	1299	1924	2035	2193	2203	2660	2669	3517	3890#

									٠,,				
F	FCHAR	71#	2310										
F	IRFLC	54#	1067	1067									
F	NDMEM	1117	3165#										
F	NDPAG	2696	2794	2863#									
F	RZMEM	139#	1067	1067	1069	1069	1074	1076	1081	1084	1094	1100	1102
		1106	1108	1108	1111	1113	1113	1116	1559	1561	1586	1589	1623
		1623	1905	1906	1909	1910	1951	1951	2156	2156	2177	2177	2217
		2217	2218	2218	2338	2340	2341	2341	2405	2405	2578	2578	2631
		2632	2634	2711	2809	. 3237	3327	3446	3725	3728	3762	3762	3764
		3797	3801	3807	3839	3839	3841	3850					
F	RZPGS	140#	1076	1076	1 086	1116	1584	2694	2706	2792	2804		
	TAXBA	1573	1890	2762	2764	2773#	2812						
F	TAXWD	1885	2762#	2998									
	TLMSG	3882#	3888	3893	3893								
	TMSLN	3888#	3893										
	TPRBA	1224	1235	1349	1357	1359	1382	1417	1446	1450	1456	1470	2080
		2096	2098	2675#	2714								
F	TPRBY	1251	1308	1326	1332	1349#							
F	TPRWD	1249	1307	1357#									
	ETLIN	2218	3437#										
G	ETLN1	99#	3439										
G	ETNYB	2885	2909	2914	2951	2992#							
G	LBVAR	135#	1106	1108	1400	1400	1438	1438					
G	ODOIT	1281#	1297	1319	1343								
G	TPLEN	1625	1920	2199	2595#	2605							
	TPNUM	1900	1938	1960	1965	2189	2588#						
G	TVARA	1366	1384#										
G	TVARP -	1250	1308	1326	1332	1382#							
G	TVCBA	2325	2336#	2343									
G	TVRA1	1365#	1632	1644	1761	3358	3374	3384	3397	3408			
۲	IALT	3907#	3907										
۲	IDRCKA	210#	1557										
H	IDRCKV	211#	1585										
H	IDRFLG	207#	3236	3326	3445	3805	3849						
F	IDRFRZ	201#	1072										
H	IDRGBV	205#	1104										
	IDRIMP	206#	3762	3839									
	IDRIRL	198#											
	IDRREL	200#	3723	3796									
	IDRSBW	209#	1109										
	IDRSER	208#			•								
	IDRSTR	202#	1098										
	IDRTHG	204#	1903	2629									
	IDRTYP	199#	1093										
	IDRVCB	203#	2336										
	IOME	96#	3158										
	NCA	514#											
	NITSC	1037	3144#			00.5	0050	0.400	2040	3656			
1	NVFLG	84#	3155	3189	3313	3315	3356	3423	3648	3656			

. ...

	*							· •			
INWORD	161#	2265	2270	3047							
IOB	3475#	3506	3507								
IOBBUF	3482#	3495	3495								
IOBCMD	3484#	3493									
IOBDRV	3477#	3617	3699	3715							
IOBSCT	3480#	3506		•							
IOBSLT	3476#	3609	3697	3714							
IOBTRK	3479#	3496	3503	3714							
IXBNE	859#	1032	3303								
IYBNE	864#	1032									
JCC	544#	1940	1962								
JCS	551#	1788	3793								
JEQ		1422		2050	0.105						
JGE	530#	1422	1591	3058	3185						
	565#										
JGT	572#	1000	1000								
JLT	558#	1228	1229	1277							
JMI	587#	4507									
JNE	537#	1597	3066								
JPL	580#										
JSRCC	614#	2192									
JSRCS	624#	1902	3516								
JSREQ	594#	1248	1250	1251	1306	1308	1325	1331	1455	2232	2668
JSRGE	644#	2660				•					
JSRGT	654#										
JSRLT	634#										
JSRMI	675#	2425									
JSRNE	604#	1924	2203	3239							
JSRPL	665#										
L0805	1031#	1033									
L084A	1053#	1060									
L0897	1079#	1091									
L08B6	1086	1093#									
L090A	1119	1129#									
L09AF	1239#	1272									
L09D7	1249	1250	1251	1256#							-
L09ED	1254	1274#									
LOA2B	1280	1294	1299#	1316	1340						
L0A45	1307	1308	1310#								
LOA73	1326	1327#									
LOA8A	1332	1333#									
L0A98	1277	1338#									
LOADO	1366	1372#									
LOAD6	1370	1375#									
L0B02	1385	1393#									
LOB26	1384	1406#									
L0B60	1423	1431#									
L0B94	1448	1454#									
LOB9C	1448	1452	1458#								
LOBAD	1460	1467#									

LOBC3	1465	1474	1479#								
LOBDA	1482#	1752									
LOBER	1491	1493#									
	1496	1508#									
LOC17											
LOC1A	1513#	1518	1500	1500							
LOCA5	1573#	1578	1580	1582							
LOCDA	1588	1592#									
LOCE6	1597#	1613									
LOCFA	1602	1607#									
LOD4E	1634#	1646									
LODB7	1671	1678#	1682								
LODD2	1677	1685#									
LODE4	1695#	3362									
LOE2F	1723#	1730									
LOE4C	1716	1731#									
LOE9D	1653	1757#									
LOEB7	1179	1768#									
LOECF	1180	1774#									
L0F08	1782	1787#									
LOF 10	1771	1777	1789#	1796	1805	1825					
L0F23	1184	1798#									
L0F97	1190	1860#									
LOFA1	1862#	2474									
LOFD1	1877	1880#									
L100B	1900#	1903									
L103B	1902	1920#									
L105E	1923	1929#									
L106C	1938#	1942									
L100C	1940	1943#									
L107E	1948	1950#									
	1960#	1964									
L10A5	1950#	1965#	1968								
L10B7	1962	1967#	1900								
L10BD	1902	1994#									
£1104	1990	1994#	2014	2017							
L1107		1999	2014	2017							
L110D	1994 1993	1997#									
L1111		2013#									
L1139	2009	2015#									
L113F	2013	2015#									
L1143	2012	2016#	2048							•	
L1173	2035	2030#	2046								
L117F	2038	2040# 2045#									
L118E	2043		2050#								
L119D	2041	2046	2050# 2050	2051#							
L11A0	2040	2045	2050	ZU51#							
L11B4	2059	2063#									
L11F2	2088#	2109									
L1220	2084	2111#									
L1230	2116#	2130									

.

		0101#					
L124C	2113	2131#					
L12AC	2189#	2195					
L12BE	2191	2199#					
L12D7	2202	2210#	0000	2275	2295		
L1310	2225#	2255	2269	2275	2293		
L1332	2236#	2239	0056#				
L135C	2233	2252	2256#				
L137A	2248	2270#					
L1382	2257	2261	2274#				
L13BA	2232	2297#	2306				
L13DA	2260	2301	2312#				
L13E0	2251	2314#					
L13E4	2316#	2319					
L13ED	2319#	2330					
L13EF	2317	2321#	2333				
L13F0	2322#						
L13F1	2247	2312	2324#				
L13FB	2330#	2334					
L141F	2282	2343#					
L1445	2361	2363#					
L144A	2366#	2372	2374				
L1450	2364	2368#					
L1470	2368	2371	2375#				
L148E	2383#	2402					
L14B4	2386	2390	2394	2399#			
L14D0	2386	2390	2394	2398	2402#		
L14D7	2398	2404#					
L1500	2427#	2432					
L1519	2428	2434#					
L151D	2436#	2440					
L1529	2435	2442#					
L152E	2426	2445#					
L1568	1994	2477#					
L1578	2480#	2488					
L158B	2483	2485#					
L15C5	2501#	2514					
L15D6	2507	2511#					
L15FB	1988	2007	2024	2520#			
L160A	1995	2527#					
L1611	2447	2530#	2541				
L161F	2522	2524	2539#				
L1643	2551	2554#					
L1653	2558#	2561					
L165D	2558	2562#					
L16A1	2607#	2609					
L16C3	2622	2625#					
L16CB	2627	2629#					
L16DE	1770	1776	1787	2640#			
L16E9	1804	2642	2646#				

•

	*.				
L16F3	2648	2650#			
L1757	2676	2690#			
L1761	2691	2695#			
L1770	2702#	2733			
L1778	2694	2710#			
L1788	2698	2718#			
L1790	2719	2722#			
L17C4	2742#	2757			
L1801	2774	2788#			
L1807	1565	1584	2791#		
L180B	2789	2793#			
L181A	2800#	2831			
L1822	2792	2808#			
L1832	2796	2816#			
L183A	2817	2820#			
L1891	2839	2856#			
L1892	2854	2857#			
L189D	2866#	2872			
L18A9	2867	2870#			
L18B1	2869	2875#			
L18D9	2894#	2899			
L 18DC	2895#	2907	2918	2923	2928
L18E2	2893	2897#			•
L192D	2933	2936#			
L1937	2938	2940#			
L193A	2944#	2950	2954		
L1984	2983	2986#			
L19BF	2993	2996#			
L19D6	2996	3006#			
L19F3	3007	3024#			
L19FF	3026	3028#			
L1A09	3039#	3042			
L1A1B	3044#	3066			
L1A2A	3047	3050#			
L1A43	3053	3058#			
L1A52	3049	3062#	3069	3072	3089
L1A62	3061	3067#			
L1A6C	3067	3070#			
L1A99	3071	3091#			
L1A9B	3092#	3094			
L1AA6	3093	3096#			
L1AB6	3101	3102	3104#		
L1AC1	3105	3106	3108#		
L1AC9	3109	3109	3111#		
LIACA	3058	3066	3078	3087	3113#
L1B28	3167#	3170	3173		
L1B57	3187	3190	3195#		
L1B61	3186	3201#	3337		
L1B64	3199	3206#			

L1B66 L1B70	3208# 3209	3210 3213#					
L1B77 L1BA0	3220# 3151	3230 3246#	3255	3257			
L1BD5	3256	3270#	3276				
L1BE3	3271	3278#					
L1BF7	3289#	3295					
L1C05	3290	3297#					
L1C40	3310	3320#			•		
L1C50	3324	3326# 3336#					
L1C79 L1C89	3329 3349#	3361	3361				
L1CB8	3361	3366#	3001				
L 1 CDB	3368	3379#					
L1CF5	3386	3388#					
L1D00	3389	3389	3393#				
L1D05	3377	3395#					
L1D1C	3402	3404#					
L1D33	3411	3413#					
L1D35	3412	3414#					
L1D40	3399	3419#					
L1D43	3417	3420#					
L1D59	3428#	3433					
L1D8B	3448	3453#					
L1D94	3456 3460#	3457# 3469					
L1D98 L1DA7	3463	3464	3465#				
L1DB1	. 3458	3468	3469#				
L 1DDF	3499#	3504	0.100#				
L1DE7	3500	3503#					
L 1DED	3502	3505#					
L1E2F	3530#	3535					
L1E3D	3537#	3595	3595				
L1E59	3548#	3599	3603	3614	3643	3653	3661
L1E5A	3550#	3642	3642				
L1E69	3555#	3611	3652	3652	3663	3666	3667
L1E6C	3557#						
L1E7B	3562#	3619					
L1E7E	3564#	3600	3629				
L1E8D L1E90	3569# 3571#	3649	3649				
L1E9A	3576#	3623	3623				
L 1EBD	3591#	3720	3790				
L1F12	3621#	3628	2.00				
L1F3A	3633#	3635					
L1F48	3631	3635#					
L1F4C	3599	3603	3614	3640#	3666	3667	
L1FB3	3663	3664#					
L1FC0	3671#	3703	3703				

L1FDF	3683#	3707	3707									
L2005	3697#	3774	3777	3852	3855							
L2023	3705#	3712										
L2040	3699	3701	3713#									
L20C3	3766#	3770										
L210A	3799	3803#										
L210D	3803	3805#										
L2177	3843#	3847										
L21A0	2456	3868#										
LC40	36#	3188										
LD9	163#	1045										
LDE	167#	3124										
LDF	168#	3133										
LDORG	40#	1024										
LE0	169#	2224	2240	2277	2283	2290		~				
LE1	170#	2225	2239	2245	2249	2253	2258	2262	2268	2273	2299	2303
LFCHAR	69#	1539	1552	2309	2927							
LINCNT	182#	3160	3307	3308	3319	3319	3439	3440				
LOCVAR	136#	1389	1391	1427	1429	1719	1719	2090	2092	2104	2106	2119
	2124	3735	3735	3815	3815							
LRUPAG	144#	1065	1124	2858	2872							
LSTFLC	55#	3166	3166									
MAINOR	51#	52	1025			-						
MDFLAG	178#	2521	2527	2540	3807	3848						
MNLOOP	1127	1222#	1287									
MOREMS	3303#	3305	3311	3311								
MOV	507#	1039	1041	1044	1053	1061	1063	1064	1073	1075	1083	1088
	1099	1101	1102	1105	1110	1114	1125	1222	1259	1260	1283	1285
	1310	1335	1351	1388	1390	1401	1403	1413	1426	1428	1439	1441
	1464	1484	1506	1514	1521	1523	1527	1529	1558	1560	1562	1564
	1570	1583	1609	1618	1619	1684	1687	1698	1700	1712	1719	1725
	1727	1738	1739	1741	1798	1801	1869	1874	1892	1916	1918	1924
	1926	1929	1930	2063	2064	2073	2089	2091	2103	2105	2111	2114
	2115	2135	2158	2160	2179	2204	2207	2210	2220	2222	2223	2224
	2236	2280	2284	2286	2294	2337	2339	2355	2357	2402	2426	2520
	2563	2565	2574	2576	2616	2654	2656	2663	2665	2711	2713	2719
	2729	2730	2739	2740	2741	2742	2809	2811	2817	2827	2828	2842
	2845	2846	2848	2851	2855	2857	2883	2884	2956	2958	2967	2970
	3007	3024	3027	3039	3042	3043	3046	3144	3147	3150	3151	3152
	3153	3154	3159	3312	3314	3318	3353	3355	3366	3422	3438	3441
	3452	3495	3598	3602	3613	3628	3646	3647	3655	3658	3713	3714
	3724	3727	3763	3782	3806	3812	3840	3860				
MRKPAG	2702	2800	2838#									
MRMSLN	3305#	3311										
MRUPAG	143#	1064	2839	2839	2840	2852						
NEWMOD	2891	2930#										
OPADD	1197	1974#										
OPAND	1186	1811#										
OPCALL	1207	2058#										

OPCGPA	1228	1324#				
OPCGPB	1229	1304#				
OPCGPC	1230	1292#				
OPCGPD	1235#					
OPCKSM	1148	1557#				
OPCLRA	1189	1845#				
OPCODE	115#	1225	1275	1313	1329	1337
OPCRLF	1146	1550#				
OPDEC	1160	1643#	1780			
OPDECB	1181	1780#				
OPDIV	1200	2005#				
OPDSTT	1163	1659#	1864			
OPENDS	1145	3899#				
OPGTBY	1193	1888#				
OPGTCH	1156	1604#				
OPGTLN	1211	2215#				
OPGTNP	1196	1957#				
OPGTP	1194	1899#				
OPGTPA	1195	1937#				
OPGTPL	1158	1622#				
OPGTPR	1157	1615#				
OPGTSB	1155	1599#				
OPGTWD	1192	1882#				
OPINC	1159	1631#	1784			
OPINCB	1182	1784#	•			
OPJUMP	1166	1750#				
OPMAX1	1149#	1294				
OPMAX2	1170#	1316				
OPMAX3	1202#	1340				
OPMAX4	1217#	1280				
OPMOVE	1168	1760#				
OPMOVT	1191	1864#				
OPMTCH	1178	2033#				
OPMUL	1199	1986#				
OPNOT	1169	1765#				
OPNULL	1139	1532#				
OPOR	1185	1808#				
OPPRCH	1212	2412#				
OPPRNM	1213	2418#				
OPPRST	1147	2215	3351#			
OPPRTN	1164	1694#				
OPPSB	1161	1651#				
OPPSI	1137	1521#	1535			
OPPSIC	1138	1535#				
OPPSW	1167	1755#				
OPPTBY	1209	2168#				
OPPTP	1210	2187#				
OPPTWD	1208	2145#				
OPPULL	1216	2472#				

OPPUSH	1215	2466#												~	
OPRMD	1201	2022#													
OPRNDM	1214	2455#													
OPRSGM	1141	3790#		4-444											
OPRTN	1165	1515	1547	1711#											
OPRTNE	1136	1480	1517#	1540											
OPRTNT	1135	1482	1512#	1542											
OPRTNV	1143	1545# 1831#													
OPSETA OPSUB	1188 1198	1980#													
OPSVGM	1140	3720#													
OPTAB1	1135#	1149	1296	1296											
OPTAB1	1154#	1170	1318	1318											
OPTAB3	1177#	1202	1342	1342											
OPTAB4	1207#	1217	1275	1275											
OPTINT	1183	1791#	1275	1275											
OPTSTA	1187	1818#													
OPTSTZ	1154	1596#													
OUTBUF	3236#	3321	3351	3437	3625	3645	3709								
OUTMSG	3527#	3596	3624	3644	3704	3708	3894	3903							
PKWORD	165#	2368	2384	2388	2392	2396	3040	3055	3063	3075	3084	3113			
	3117	3119	3120	3121	3122	3123	3128	3130	3131	3132	3134	3135			
	3137							• • • • • • • • • • • • • • • • • • • •			0.0.	0.00			
PNYBBF	159#	2961	2961	2969	2969	3000	3000	3000	3008	3010	3013	3025			
	3028														
PNYBCN	158#	2884	2960	2970	2992	2997	3008	3025	3028						
PRCSWL	183#	3145	3252	3252	3279	3279	3332	3332	3335	3335					
PREDFL	1450#	1592	1597	1613	1772	1778	1789	1796	1806	1826	2050	3778			
	3856														
PREDTR	1446#	1591	1597	1789	2051	3775	3853								
PRGIDX	123#	1100	1491	1493	1522	1528	1739	2065	2076	2076	2678	2682			
220120	3731	3731	3810	3810											
PRGLPG	124#	1102	1103	1499	1500	1502	1504	1523	1523	1529	1529	1733			
	1733	2071	2071	2078	2688	2688	2690	2692	2696	2696	2725	2725			
DDCMDT	2730	2731	1501	0.070	0711	0710									
PRGMPT	125#	1531	1531	2679	2711	2712									
PRGPPG PRGUPD	127#	2697 1040	2705	2724	2728	2817	2675	2000	0714	0010	0010				
PRGUPD	126# 157#		1507	1530	1742	2074	2675	2686	2714	2818	3813				
PRNTBF	3185	2884 3216	2936 3307#	2938 3592	2939 3593	2960 3597	2970	2984 3635	2040	0701	0705	0710			
PRIVIDI	3890	3899	3905	3592	3393	3591	3621	3635	3640	3701	3705	3712			
PRNTNM	2419	2424#	3375	3393	3404	3419	3897								
PRNTST	1525	1706	1757	2883#	2964	3419	3697								
PROMPT	86#	3154	1757	2003#	2904										
PRTBUF	3240	3248#	3451												
PRTWDT	48#	3262	3431												
PSH	485#	1376	1376	1417	1417	1607	1635	1635	1666	1666	1673	1673			
. 3.,	1684	1684	1868	1868	2276	2430	2478	2478	2498	2498	2959	2961			
	2961	2961	2963	2963	3249	3249	3249	3320	3331	3331	3352	2301			
					J	J	02.0	3020	000.	000.	0002				

 ϵ

PTVARA PTVARP	1370 1414	1421# 1416#	1610											
PTVRA1	1369#	1636	1862											
PTVRP1	1414#	1620	1744	1762	1766	1809	1812	1886	1893	1919	1928	1931		
	1951	1975	1981	1996	2027	2060	2460							
PTVRPA	1412#	162 7	1966											
PTVRPZ	1411#	1941	1963											
PUL	478#	1361	1378	1378	1420	1420	1472	1638	1638	1672	1672	1683		
	1683	1686	1686	1876	1876	2100	2131	2293	2491	2491	2518	2518		
	2766	2966	2966	2966	2969	2969	2969	3280	3282	3282	3336	3336	40	
	3423	3897	3897				4700	.701	.70.	. 707	0.470	0000#		
PULLWD	1144	1372	1376	1406	1545	1714	1723	1731	1734	1737	2472	2662#		
PUSHWD	1373	1378	1422	2065	2068	2071	2095	2132	2467	2652#				
RDKEY	98#	3315	3626	3656	3710									
RNDLOC	90#	3868	3869	3871	3871									
RNGDBG	35#	103	2453	3146										
RSGMFL	3794	3803	3823	3831	3844	3855#	0000							
RSGMMV	3811	3816	3820	3828	3836	3859#	3863							
RTSCC	699#		0010											
RTSCS	706#	2302	2313	2886	0000	25.00								
RTSEQ	685#	1663	2229	2230	2298	2529								
RTSGE	720#	0000												
RTSGT	727#	3223												
RTSLT	713#													
RTSMI	742#	2602	2781											
RTSNE	692#	2683 2540	2781											
RTSPL RWTS	735# 63#	2540 3508												
	53#	54	63											
RWTSOR SBWDPT	172#	1111	1113	2957	2959									
SCMSLN	3343#	3370	1113	2931	2555									
SCORMS	3341#	3343	3370	3370										
SECPTK	113#	3499	3506	3370										
SEPTAB	2308#	2317	3300											
SETAXB	1571	1652	1705	1884	1889	2739#								
SETAXW	1756	2748#	2963	1004	1000	2700%								
SETUPA	1818	1831	1845	2547#										
SETUPP	1899	1937	1957	2187	2571#									
SETUPT	1600	1605	1616	1660	1667	1678	1695	1792	1866	1870	2548	2572		
321311	2615#													
SHWMSG	3313	3371	3381	3427#	3650	3654								
SPCLCH	2906	2973#	3093											
START	1027#	1142	3911											
STCKLC	45#	46	1043	1043										
STCKMX	44#	46	2660											
STKCNT	151#	1042	1713	2136	2657	2658	2667	3739	3739	3819	3819			
-STKCSV	154#	1713	1740	2064	2136									
STKLIM	46#	3747	3747	3755	3755	3827	3827	3835	3835					
STKPNT	152#	1043	1043	1712	1712	2137	2137	2653	2653	2653	2655	2656		

		2656	2656	2657	2664	2665	2665 2068	2666 2068	2667 2137	2667 2137			
	TKPSV	153#	1712	1712	1736	1736	2000	2000	2101	2107			
	TLTYP	187#	1096	3367	3398								
	TR	526#			1292	1385	1393	1424	1431	1481	1483	1624	1646
SI	UB	370#	1057	1118	2401	2551	2556	2653	2656	2901	2930	2947	3007
		1729	1752	2108	3193	3316	2550	2000	2000				
		3061	3061	3068	3767	3772	3777#						
	VGMFL	3743	3751	3759		3772	3777# 3781#	3785					
	VGMMV	3732	3736	3740	3748	2723	2723	2821	2821				
	WPMEM	138#	1115	1116	1119	2123	2123	2021	2021				
	WPPGS	141#	1122	2863									
	BCHAR	70#	2309										
	HGATT	218#		4000	1075	1871							
	HGCHD	221#	1606	1668	1675	1868							
	HGPAR	219#	1617	1661	1686	1000							
	HGPRP	222#	1697	2573	1070	1070							
	HGSIB	220#	1601	1673	1679	1878							
	IMEMS	3345#	3347	3380	3380								
	MMSLN	3347#	3380		0071	2002	2987						
	MPMOD	156#	2885	2934	2971	2982	1448	1452	1613	2084	2540	2893	3053
Т	STA	522#	1366	1370	1384	1422	1446	1452		2004	2546	2000	0000
		3109			4000	0000	3052	3108					
	STABE	909#	1365	1369	1383	2083	3052	3106					
	STABM	924#	1447										
	STABN	914#	1612	2892									
	STABP	919#	1451										
T	STAJE	929#	1421										
	STARP		2539										
	STCHR	3051	3059	3100#									
	STMOD	2891	2932	2982#									
	/ERSN	34#	50										
	MT1LC	58#	1047	1047									
	/MT2LC	59#	1048	1048									
	/MT3LC	60#	1049	1049									
	/MT4LC	61#	1050	1050	4054	0700	2828	2867					
	/MTAB1	146#	1047	1047	1054	2730	2829	2869		•			
	/MTAB2	147#	1048	1048	1054	2731	1126	2843	2844	2849			
	/MTAB3	148#	1049	1049	1056	1062	2847	2852	2856	2043			
	/MTAB4	149#	1050	1050	1058	2846		2052	2830				
	/MTORG	52#	53	58	59	60	61						
	/TAB	95#	3354	3424									
	WNDBOT	79#	3153	3310									
	WNDLFT	76#	3151		0016	0.400							
	NDTOP	78#	3148	3160	3319	3439	2224						
	TOWDAN	77#	3152	3199	3211	3223	3324						
2	ZPORG	42#	111										