

%%
%%

M6800 MICROPROCESSOR CROSS-ASSEMBLER
[ASM68 REV.II (FORTRAN) BY M.J. BAUER, 1978]
DEAKIN UNIVERSITY DEC-SYSTEM-20/50

%%
%%

CHIPS

COMPACT HEX. INTERPRETIVE PROGRAMMING AND
OPERATING SYSTEM.

CHIP8 LANGUAGE INTERPRETER
PLUS OPERATING SYSTEM
FOR 6800 COMPUTERS

- (1) UPON RELOCATION, THE DATA AT LOC \$C133
MUST BE CHANGED ACCORDINGLY (SEE RANDOM);
- (2) IF DISPLAY AREA MOVED, THE DATA AT LOC \$C227
IS CHANGED TO HIGH-ORDER BYTE OF BUFFER ADRS.

SCRATCHPAD RAM ASSIGNMENTS (PAGE 0)

0000	IRQV	EQU	\$0000	INTERRUPT VECTOR
0002	BEGA	EQU	\$0002	BEGIN ADRS FOR LOAD/DUMP
0004	ENDA	EQU	\$0004	ENDING ADRS FOR LOAD/DUMP
0006	ADRS	EQU	\$0006	ADRS FOR GO AND MEMOD
0008	DDPAT	EQU	\$0008	DIGIT PATTERN TEMP (5 BYTES)
000D	RND	EQU	\$000D	RANDOM BYTE (SEED)
000E	N	EQU	\$000E	TEMP
000F	ATEMP	EQU	\$000F	TEMP
0012	XTEMP	EQU	\$0012	2-BYTE SAVE FOR X, SP
0014	ZHI	EQU	\$0014	TEMP ADRS
0015	ZLO	EQU	\$0015	
0017	KEYCOD	EQU	\$0017	KEYCODE TEMP
0018	BADRED	EQU	\$0018	KEY BAD-READ FLAG
001C	BLOC	EQU	\$001C	DISPLAY POINTER (BYTE LOC'N)
001E	PATNH	EQU	\$001E	PATTERN TEMP
001F	PATNL	EQU	\$001F	
0020	TIME	EQU	\$0020	RTC TIMER VALUE
0021	TONE	EQU	\$0021	DURATION COUNT FOR TONE
0022	PPC	EQU	\$0022	PSEUDO PRGM-COUNTER
0024	PSP	EQU	\$0024	PSEUDO STACK-PTR
0026	I	EQU	\$0026	CHIPS MEMORY POINTER
0028	PIR	EQU	\$0028	PSEUDO INST-REG
002A	VXLOC	EQU	\$002A	POINTS TO VX
002C	RNDX	EQU	\$002C	RANDOM POINTER
002E	VX	EQU	\$002E	VARIABLE X [ALSO X-COORD]
002F	VY	EQU	\$002F	VARIABLE Y [ALSO Y-COORD]

CHIPS VARIABLES (TABLE)

0030	VO	EQU	\$0030
003F	VF	EQU	\$003F

CHIPS SUBROUTINE STACK

005F	STACK	EQU	\$005F

*** OPERATING-SYSTEM STACK ***			
007F	STOP	EQU	\$007F
			STACK TOP (MONITOR)

*** CHIPS GRAPHIC DISPLAY AREA:

(1/4 K RAM BLOCK MAPPED ONTO T.V. SCREEN BY DMA,
IN FORMAT 64H*32V DOTS)

0100	DISBUF	EQU	\$0100	DISPLAY BUFFER AREA
0200	ENDBUF	EQU	\$0200	
8010	PIAA	EQU	\$8010	PORT-A FOR KEYPAD
8012	PIAB	EQU	\$8012	PORT-B FOR TAPE , RTC, TONE

*** CHIPS INTERPRETER MAINLINE ***

C000		ORG	\$C000	
C000 8D 77	CHIPS	BSR	ERASE	NORMAL ENTRY POINT
C002 CE 0200		LDX	#\$0200	RESET PSEUDO-PC
C005 DF 22		STX	PPC	
C007 CE 005F		LDX	#STACK	RESET STACK PTR
C00A DF 24		STX	PSP	
C00C DE 22	FETCH	LDX	PPC	POINT TO NEXT INSTR
C00E EE 00		LDX	0,X	COPY TO PIR
C010 DF 28		STX	PIR	
C012 DF 14		STX	ZHI	SAVE ADRS (MMM)
C014 BD C0D0		JSR	SKIP2	BUMP PRGM-CTR
C017 96 14		LDA A	ZHI	MASK OFF ADRS
C019 84 0F		AND A	#\$0F	
C01B 97 14		STA A	ZHI	
C01D 8D 21		BSR	FINDV	EXTRACT VX ALSO
C01F 97 2E		STA A	VX	STASH VX
C021 DF 2A		STX	VXLOC	SAVE LOCATION OF VX
C023 96 29		LDA A	PIR+1	FIND Y
C025 44		LSR A		
C026 44		LSR A		
C027 44		LSR A		
C028 44		LSR A		
C029 8D 15		BSR	FINDV	EXTRACT VY
C02B 97 2F		STA A	VY	STASH VY
C02D CE C048	EXEC	LDX	#JUMTAB-2	POINT TO JUMP TABLE
C030 96 28		LDA A	PIR	EXTRACT MSD
C032 84 F0		AND A	#\$F0	
C034 08	EXE1	INX		FIND ROUTINE ADRS
C035 08		INX		
C036 80 10		SUB A	#\$10	
C038 24 FA		BCC	EXE1	BRANCH IF HIGHER OR SAME
C03A EE 00		LDX	0,X	LOAD ROUTINE ADRS
C03C AD 00		JSR	0,X	PERFORM ROUTINE
C03E 20 CC		BRA	FETCH	NEXT INSTR....
C040 CE 002F	FINDV	LDX	#V0-1	POINT TO VARIABLES TABLE
C043 08	FIND1	INX		FIND LOCN VX
C044 4A		DEC A		
C045 2A FC		BPL	FIND1	
C047 A6 00		LDA A	0,X	FETCH VX FROM TABLE
C049 39		RTS		

JUMP TABLE (ROUTINE ADDRESSES):

C04A C0 6A	JUMTAB	FDB	EXCALL	ERASE, RET, CALL, NOP
C04C C0 A2		FDB	GOTO	GOTO MMM
C04E C0 AC		FDB	DOSUB	DO MMM
C050 C0 BA		FDB	SKFEK	SKF VX=KK
C052 C0 C1		FDB	SKFNK	SKF VX#KK
C054 C0 C8		FDB	SKFEV	SKF VX=VY
C056 C0 EE		FDB	LETK	VX=KK
C058 C0 F2		FDB	LETVK	VX=VX+KK
C05A C0 FE		FDB	LETVV	VX=[VX][+&-&!]VY
C05C C0 CC		FDB	SKFNV	SKF VX#VY
C05E C0 A7		FDB	LETI	I=MMM

C060	C0	97	FDB	GOTOV	GOTO MMM+V0
C062	C0	F8	FDB	RANDV	VX=RND.KK
C064	C2	1F	FDB	SHOW	SHOW N@VX,VY
C066	C0	D7	FDB	SKFKEY	SKF VX[=#]KEY
C068	C1	5F	FDB	MISC	(MINOR JUMP TABL)

ERASE, RETURN, CALL (MLS), OR NOP INTRN:					
C06A	D6	28	EXCALL	LDA B PIR	GET INSTR REG
C06C	26	25		BNE CALL	
C06E	96	29		LDA A PIR+1	
C070	81	E0		CMP A #\$E0	
C072	27	05		BEQ ERASE	
C074	81	EE		CMP A #\$EE	
C076	27	0E		BEQ RETDO	
C078	39			RTS	NOP, FETCH
C079	4F		ERASE	CLR A	WRITE ZEROS TO SCREEN
C07A	CE	0100		LDX #DISBUF	POINT TO DISPLAY BUFF
C07D	A7	00	FILL	STA A 0,X	FILL SCREEN WITH ACC-A
C07F	08			INX	
C080	8C	0200		CPX #ENDBUF	DONE?
C083	26	F8		BNE FILL	
C085	39			RTS	
C086	30		RETDO	TSX	SAVE REAL SP
C087	9E	24		LDS PSP	
C089	32			PUL A	
C08A	97	22		STA A PPC	PULL PPC
C08C	32			PUL A	
C08D	97	23		STA A PPC+1	
C08F	9F	24		STS PSP	SAVE CHIP8 SP
C091	35			TXS	RESTORE SP
C092	39			RTS	
C093	DE	14	CALL	LDX ZHI	GET OPRND ADRS (MMM)
C095	6E	00		JMP 0,X	PERFORM MLS
C097	96	30	GOTOV	LDA A V0	16-BIT ADD V0 TO ADRS
C099	5F			CLR B	
C09A	9B	15		ADD A ZLO	
C09C	97	15		STA A ZLO	
C09E	D9	14		ADC B ZHI	
COA0	D7	14		STA B ZHI	
COA2	DE	14	GOTO	LDX ZHI	MOVE ADRS TO PPC
COA4	DF	22		STX PPC	
COA6	39			RTS	FETCH
COA7	DE	14	LETI	LDX ZHI	MOVE ADRS TO MI PTR
COA9	DF	26		STX I	
COAB	39			RTS	FETCH
COAC	30		DOSUB	TSX	SAVE SP
COAD	9E	24		LDS PSP	
COAF	96	23		LDA A PPC+1	PUSH PPC
COB1	36			PSH A	
COB2	96	22		LDA A PPC	
COB4	36			PSH A	
COB5	9F	24		STS PSP	SAVE CHIP SP
COB7	35			TXS	RESTORE REAL SP
COB8	20	E8		BRA GOTO	JUMP TO ADRS (MMM)

CONDITIONAL SKIP ROUTINES:

COBA 96 29	SKFEK	LDA A	PIR+1	GET KK
COBC 91 2E	SKFEQ	CMP A	VX	
COBE 27 10		BEQ	SKIP2	
COC0 39		RTS		
COC1 96 29	SKFNK	LDA A	PIR+1	GET KK
COC3 91 2E	SKFNE	CMP A	VX	
COC5 26 09		BNE	SKIP2	
COC7 39		RTS		
COC8 96 2F	SKFEV	LDA A	VY	GET VY
COCA 20 F0		BRA	SKFEQ	
COCC 96 2F	SKFNV	LDA A	VY	
COCE 20 F3		BRA	SKFNE	
COD0 DE 22	SKIP2	LDX	PPC	ADD 2 TO PPC
COD2 08		INX		
COD3 08		INX		
COD4 DF 22		STX	PPC	
COD6 39		RTS		
COD7 BD C297	SKFKEY	JSR	KEYINP	INTERROGATE KEYBOARD
CODA 7D 0018		TST	BADRED	KEY DOWN?
CODD 27 07		BEQ	SKFK1	
CODF C6 A1		LDA B	##A1	WHAT INSTRN?
COE1 D1 29		CMP B	PIR+1	SKF VX#KEY
COE3 27 EB		BEQ	SKIP2	
COE5 39		RTS		NO KEY; GO FETCH
COE6 C6 9E	SKFK1	LDA B	##9E	
COE8 D1 29		CMP B	PIR+1	WHAT INSTRN?
COEA 27 D0		BEQ	SKFEQ	
COEC 20 D5		BRA	SKFNE	

ARITHMETIC/LOGIC ROUTINES:

COEE 96 29	LETK	LDA A	PIR+1	GET KK
COF0 20 3B		BRA	PUTVX	
COF2 96 29	LETVK	LDA A	PIR+1	
COF4 9B 2E		ADD A	VX	
COF6 20 35		BRA	PUTVX	
COF8 8D 38	RANDV	BSR	RANDOM	GET RANDOM BYTE
COFA 94 29		AND A	PIR+1	
COFC 20 2F		BRA	PUTVX	
COFE 96 2E	LETVV	LDA A	VX	
C100 D6 29		LDA B	PIR+1	
C102 C4 0F		AND B	##0F	EXTRACT N
C104 26 02		BNE	LETV1	
C106 96 2F		LDA A	VY	VX=VY
C108 5A	LETV1	DEC B		
C109 26 02		BNE	LETV2	
C10B 9A 2F		ORA A	VY	VX=VX!VY (OR)
C10D 5A	LETV2	DEC B		
C10E 26 02		BNE	LETV4	
C110 94 2F		AND A	VY	VX=VX.VY

C112 5A	LETV4	DEC B	
C113 5A		DEC B	
C114 26 0A		BNE	LETV5
C116 7F 003F		CLR	VF
C119 9B 2F		ADD A	VY
C11B 24 03		BCC	LETV5
C11D 7C 003F		INC	VF
C120 5A	LETV5	DEC B	
C121 26 0A		BNE	PUTVX
C123 7F 003F		CLR	VF
C126 90 2F		SUB A	VY
C128 25 03		BCS	PUTVX
C12A 7C 003F		INC	VF
C12D DE 2A	PUTVX	LDX	VXLOC
C12F A7 00		STA A	0,X
C131 39		RTS	

VF=0
VX=VX+VY
RESULT < 256 ?
VF=1 (OVERFLOW)

VF=0
VX=VX-VY
VX<VY? (UNSIGNED)
NO, PUT VF=1
REPLACE VX

*** RANDOM BYTE GENERATOR ***

C132 86 C0	RANDOM	LDA A	#\$C0	** HIGH-ORDER BYTE OF RNDX =
C134 97 2C		STA A	RNDX	= MSB OF CHIP8 START ADRS.
C136 7C 002D		INC	RNDX+1	
C139 DE 2C		LDX	RNDX	POINT TO NEXT PROGRAM BYTE
C13B 96 0D		LDA A	RND	GET SEED (LAST VALUE)
C13D AB 00		ADD A	0,X	MANGLE IT
C13F A8 FF		EOR A	\$FF,X	
C141 97 0D		STA A	RND	STASH IT
C143 39		RTS		

JUMP TABLE FOR MISCELLANEOUS INSTRNS [FXZZ]

C144 07	MINJMP	FCB	\$07	VX=TIME
C145 C1 79		FDB	VTIME	
C147 0A		FCB	\$0A	VX=KEY
C148 C1 7D		FDB	VKEY	
C14A 15		FCB	\$15	TIME=VX
C14B C1 82		FDB	TIMEV	
C14D 18		FCB	\$18	TONE=VX
C14E C1 85		FDB	TONEV	
C150 1E		FCB	\$1E	I=I+VX
C151 C1 89		FDB	LETIV	
C153 29		FCB	\$29	I=DSPL,VX
C154 C1 93		FDB	LETDSP	
C156 33		FCB	\$33	MI=DEQ,VX
C157 C1 DE		FDB	LETDEQ	
C159 55		FCB	\$55	MI=V0:VX
C15A C1 FA		FDB	STORV	
C15C 65		FCB	\$65	V0:VX=MI
C15D C2 04		FDB	LOADV	

C15F CE C144	MISC	LDX	#MINJMP	POINT TO TABLE
C162 C6 09		LDA B	#9	DO 9 TIMES....
C164 A6 00	MIS1	LDA A	0,X	GET TABLE OPCODE
C166 91 29		CMP A	PIR+1	
C168 27 09		BEQ	MIS2	
C16A 08		INX		
C16B 08		INX		
C16C 08		INX		
C16D 5A		DEC B		
C16E 26 F4		BNE	MIS1	
C170 7E C360		JMP	START	BAD OPCODE, RETURN TO MON.
C173 EE 01	MIS2	LDX	1,X	GET ROUTINE ADRS FROM TABLE
C175 96 2E		LDA A	VX	GET VX
C177 6E 00		JMP	0,X	GO TO ROUTINE
<hr/>				
C179 96 20	VTIME	LDA A	TIME	
C17B 20 B0		BRA	PUTVX	
C17D BD C2C4	VKEY	JSR	GETKEY	
C180 20 AB		BRA	PUTVX	
C182 97 20	TIMEV	STA A	TIME	
C184 39		RTS		
C185 16	TONEV	TAB		SET DURATION=VX
C186 7E C2E1		JMP	BTONE	
C189 5F	LETIV	CLR B		16-BIT ADD VX TO I
C18A 9B 27		ADD A	I+1	
C18C 97 27		STA A	I+1	
C18E D9 26		ADC B	I	
C190 D7 26		STA B	I	
C192 39		RTS		
<hr/>				
*** COPY COMPRESSED DIGIT PATTERN (FROM TABLE)				
TO 5-BYTE ARRAY (DDPAT), & SET I FOR 'SHOW'.				
C193 CE C1BC	LETDSP	LDX	#HEXTAB-2	POINT TO HEX DIGIT PATTERNS
C196 84 0F		AND A	##0F	ISOLATE LS DIGIT
C198 08	LDSP1	INX		SEARCH TABLE.....
C199 08		INX		
C19A 4A		DEC A		(A=VX)
C19B 2A FB		BPL	LDSP1	
C19D EE 00		LDX	0,X	MOVE PATNH
C19F DF 1E		STX	PATNH	
C1A1 CE 0008		LDX	#DDPAT	POINT PATRN ARRAY(5)
C1A4 DF 26		STX	I	SET MI POINTER
C1A6 C6 05		LDA B	#5	DO 5 TIMES.....
C1A8 96 1E	LDSP5	LDA A	PATNH	
C1AA 84 E0		AND A	##E0	EXTRACT 3 BITS
C1AC A7 04		STA A	4,X	
C1AE 09		DEX		
C1AF 86 03		LDA A	#3	DO 3 TIMES...
C1B1 79 001F	LDSP3	ROL	PATNL	MOVE NEXT 3 BITS
C1B4 79 001E		ROL	PATNH	
C1B7 4A		DEC A		
C1B8 26 F7		BNE	LDSP3	CONT(3)....
C1BA 5A		DEC B		
C1BB 26 EB		BNE	LDSP5	CONT(5)....
C1BD 39		RTS		

*** HEXADECIMAL DIGIT PATTERNS (3*5 MATRIX):

C1BE	F6	DF	HEXTAB	FDB	\$F6DF	0
C1C0	49	25		FDB	\$4925	1
C1C2	F3	9F		FDB	\$F39F	2
C1C4	E7	9F		FDB	\$E79F	3
C1C6	3E	D9		FDB	\$3ED9	4
C1C8	E7	CF		FDB	\$E7CF	5
C1CA	F7	CF		FDB	\$F7CF	6
C1CC	24	9F		FDB	\$249F	7
C1CE	F7	DF		FDB	\$F7DF	8
C1D0	E7	DF		FDB	\$E7DF	9
C1D2	B7	DF		FDB	\$B7DF	A
C1D4	D7	DD		FDB	\$D7DD	B
C1D6	F2	4F		FDB	\$F24F	C
C1D8	D6	DD		FDB	\$D6DD	D
C1DA	F3	CF		FDB	\$F3CF	E
C1DC	93	4F		FDB	\$934F	F

C1DE	DE	26	LETDEQ	LDX	I	GET MI POINTER
C1E0	C6	64		LDA B	#100	N=100
C1E2	8D	06		BSR	DECI	CALC 100'S DIGIT
C1E4	C6	0A		LDA B	#10	N=10
C1E6	8D	02		BSR	DECI	CALC 10'S DIGIT
C1E8	C6	01		LDA B	#1	
C1EA	D7	0E	DECI	STA B	N	
C1EC	5F			CLR B		
C1ED	91	0E	LDEQ1	CMP A	N	DO UNTIL ACN ...
C1EF	25	05		BCS	LDEQ2	BRANCH IF LOWER; NOT SAME.
C1F1	5C			INC B		
C1F2	90	0E		SUB A	N	
C1F4	20	F7		BRA	LDEQ1	END-DO...
C1F6	E7	00	LDEQ2	STA B	0,X	STASH
C1F8	08			INX		FOR NEXT DIGIT
C1F9	39			RTS		

C1FA	0F		STORV	SEI		KILL IRQ FOR DATA STACK
C1FB	9F	12		STS	XTEMP	SAVE SP
C1FD	8E	002F		LDS	#V0-1	POINT TO VARIABLES TABLE
C200	DE	26		LDX	I	POINT MI
C202	20	09		BRA	MOVX	TRANSFER NB BYTES
C204	0F		LOADV	SEI		KILL IRQ
C205	9F	12		STS	XTEMP	
C207	9E	26		LDS	I	POINT MI
C209	34			DES		
C20A	CE	0030		LDX	#V0	POINT TO V0
C20D	D6	2B	MOVX	LDA B	VXLOC+1	CALC. X (AS IN VX)
C20F	C4	0F		AND B	##0F	LOOP (X+1) TIMES.....
C211	32		MOVX1	PUL A		GET NEXT V
C212	A7	00		STA A	0,X	COPY IT
C214	08			INX		
C215	7C	0027		INC	I+1	I=I+X+1 (ASSUMES SAME PAGE)
C218	5A			DEC B		
C219	2A	F6		BPL	MOVX1	CONTINUE....
C21B	9E	12		LDS	XTEMP	RESTORE SP
C21D	0E			CLI		RESTORE IRQ
C21E	39			RTS		

*** DISPLAY ROUTINES *****

C21F D6 29	SHOW	LDA B	PIR+1	GET N (OPCODE LSB)
C221 7F 003F		CLR	VF	CLEAR OVERLAP FLAG
C224 DE 26	SHOWI	LDX	I	POINT TO PATTERN BYTES
C226 86 01	SHOWX	LDA A	#\$01	** SET DISPLAY ADRS MSB =
C228 97 1C		STA A	BLOC	= DISBUF HIGH-ORDER BYTE.
C22A C4 0F		AND B	#\$0F	COMPUTE NO. OF BYTES (N)
C22C 26 02		BNE	SHOW2	IF N=0, MAKE N=16
C22E C6 10		LDA B	#16	
C230 37	SHOW2	PSH B		DO N TIMES.....
C231 DF 14		STX	ZHI	SAVE MI POINTER
C233 A6 00		LDA A	0,X	FETCH NEW PATTERN BYTE
C235 97 1E		STA A	PATNH	
C237 7F 001F		CLR	PATNL	
C23A D6 2E		LDA B	VX	DETERMINE OFFSET BIT COUNT
C23C C4 07		AND B	#7	
C23E 27 09	SHOW3	BEQ	SHOW4	SHIFT INTO MATCHING POS'N
C240 74 001E		LSR	PATNH	
C243 76 001F		ROR	PATNL	
C246 5A		DEC B		
C247 26 F5		BNE	SHOW3	
C249 D6 2E	SHOW4	LDA B	VX	GET X COORD
C24B 8D 28		BSR	DISLOC	FIND WHERE IS FIRST DISP BYT
C24D 96 1E		LDA A	PATNH	
C24F 8D 15		BSR	SHOWUP	
C251 D6 2E		LDA B	VX	
C253 CB 08		ADD B	#8	FIND WHERE IS ADJACENT BYTE
C255 8D 1E		BSR	DISLOC	
C257 96 1F		LDA A	PATNL	
C259 8D 0B		BSR	SHOWUP	
C25B 7C 002F		INC	VY	
C25E DE 14		LDX	ZHI	POINT NEXT PATTERN BYTE
C260 08		INX		
C261 33		PUL B		
C262 5A		DEC B		
C263 26 CB		BNE	SHOW2	CONT.....
C265 39		RTS		
C266 16	SHOWUP	TAB		UPDATE DISPLAY BYTE
C267 E8 00		EOR B	0,X	X-OR WITH EXISTING DISPLAY
C269 AA 00		ORA A	0,X	'OR' ALSO, FOR O' LAP TEST
C26B E7 00		STA B	0,X	STORE XOR'ED BYTE
C26D 11		CBA		
C26E 27 04		BEQ	SHOWR	'XOR' SAME AS 'OR', ELSE...
C270 86 01		LDA A	#1	SET OVERLAP FLAG (VF)
C272 97 3F		STA A	VF	
C274 39	SHOWR	RTS		

```

*** COMPUTE ADRS OF DISPLAY BYTE AT COORDS (B,VY):
C275 96 2F DISLOC LDA A VY      FETCH Y COORD
C277 84 1F      AND A #$1F      MASK TO 5 BITS FOR WRAP-ROUND
C279 48      ASL A      LEFT JUSTIFY
C27A 48      ASL A
C27B 48      ASL A
C27C C4 3F      AND B #$3F      MASK X COORD TO 6 BITS
C27E 54      LSR B      DROP 3 LS BITS
C27F 54      LSR B
C280 54      LSR B
C281 1B      ABA      BUILD BYTE
C282 97 1D      STA A BLOC+1     DISP LOC'N LSB COMPLETED
C284 DE 1C      LDX BLOC        POINT TO DISP BYTE AT (VX,VY)
C286 39      RTS

```

```

-----
*** KEYPAD ROUTINES *****
C287 C6 F0 PAINZ LDA B #$F0      INITIALIZE PORT
C289 CE 8010 PAINV LDX #PIAA      (ENTRY PT FOR INV. DDR)
C28C 6F 01      CLR 1,X          RESET & SELECT DDR
C28E E7 00      STA B 0,X        SET DATA DIRECTION
C290 C6 06      LDA B #$06      SEL O/P REG & SETUP CTRL
C292 E7 01      STA B 1,X
C294 6F 00      CLR 0,X        OUTPUT ZEROS & RESET FLAGS
C296 39      RTS

```

```

*** KEYPAD INPUT SERVICE ROUTINE *****
C297 8D EE KEYINP BSR PAINZ      RESET KEYPAD PORT
C299 7F 0018      CLR BADRED     RESET BAD-READ FLAG
C29C 8D 55      BSR DEL333      DELAY FOR DEBOUNCE
C29E E6 00      LDA B 0,X      INPUT ROW DATA
C2A0 8D 15      BSR KBILD      FORM CODE BITS 0,1
C2A2 97 17      STA A KEYCOD
C2A4 C6 0F      LDA B #$0F      SET DDR FOR....
C2A6 8D E1      BSR PAINV      INVERSE ROW/COL DIR'N
C2A8 E6 00      LDA B 0,X      INPUT COLUMN DATA
C2AA 54      LSR B      RIGHT JUSTIFY
C2AB 54      LSR B
C2AC 54      LSR B
C2AD 54      LSR B
C2AE 8D 07      BSR KBILD      FORM CODE BITS 2,3
C2B0 48      ASL A
C2B1 48      ASL A
C2B2 9B 17      ADD A KEYCOD
C2B4 97 17      STA A KEYCOD      BUILD COMPLETE KEY CODE
C2B6 39      RTS

C2B7 C1 0F KBILD CMP B #$0F      CHECK KEY STATUS
C2B9 26 02      BNE KBILD0      KEY IS DOWN, GO DECODE IT
C2BB D7 18      STA B BADRED     NO KEY, SET BAD-READ FLAG
C2BD 86 FF KBILD0 LDA A #-1
C2BF 4C KBILD1 INC A      (A=RESULT)
C2C0 54      LSR B      SHIFT DATA BIT TO CARRY
C2C1 25 FC      BCS KBILD1      FOUND ZERO BIT ?
C2C3 39      RTS

```

*** GETKEY WAITS FOR KEYDOWN, THEN INPUTS *****

```

C2C4 DF 12  GETKEY STX      XTEMP      SAVE X FOR CALLING ROUTINE
C2C6 8D BF  GETK1  BSR      PAINZ      RESET PORT, CLEAR FLAGS
C2C8 A6 01  GETK2  LDA A    1,X        INPUT STATUS (HEX KEY DOWN?)
C2CA 2B 07          BMI      HEXKEY     YES; FETCH IT IN
C2CC 48          ASL A                TRY CA2 FLAG
C2CD 2A F9          BPL      GETK2      FN KEY DOWN ? (AC0?)
C2CF 6D 00  FNKEY  TST      0,X        YES; RESET FLAG IN PIA
C2D1 20 07          BRA      HEXK1      RETURN WITHOUT CODE
C2D3 8D C2  HEXKEY BSR      KEYINP     DECODE THE KEYPAD
C2D5 7D 0018 TST      BADRED          WAS IT A BAD READ ?
C2D8 26 EC          BNE      GETK1      YES, TRY AGAIN
C2DA 8D 03  HEXK1  BSR      BLEEP      O.K. ACKNOWLEDGE
C2DC DE 12          LDX      XTEMP     RESTORE CALLER'S X-REG
C2DE 39          RTS                  RETURN (WITH AC0 FOR FN KEY)

```

*** TONE GENERATING ROUTINES

```

C2DF C6 04  BLEEP  LDA B #4
C2E1 D7 21  BTONE  STA B  TONE      SET DURATION (RTC CYCLES)
C2E3 C6 41          LDA B #$41      TURN AUDIO ON
C2E5 F7 8012        STA B  PIAB
C2E8 7D 0021  BTON1 TST      TONE      WAIT FOR RTC TIME-OUT
C2EB 26 FB          BNE      BTON1
C2ED C6 01          LDA B #1        TURN AUDIO OFF
C2EF F7 8012        STA B  PIAB
C2F2 39          RTS

```

*** SOFTWARE DELAY ROUTINE FOR SERIAL I/O:

```

C2F3 8D 00  DEL333 BSR      DEL167     DELAY FOR 3.33 MILLISEC
C2F5 37      DEL167 PSH B
C2F6 C6 C8          LDA B #200        DELAY FOR 1.67 MILLISEC
C2F8 5A      DEL    DEC B
C2F9 01          NOP
C2FA 26 FC          BNE      DEL
C2FC 33          PUL B
C2FD 39          RTS

```

*** TAPE INPUT/OUTPUT ROUTINES *****

*** INITIALIZE TAPE, TONE, RTC, & DMA ***

*** A=\$3F FOR DISPLAY/DMA ON; A=\$37 FOR OFF:

```

C2FE CE 8012  PBINZ  LDX      #PIAB
C301 C6 3B          LDA B #$3B      SELECT DDR (DMA ON)
C303 E7 01          STA B  1,X
C305 C6 7F          LDA B #$7F      WRITE DDR
C307 E7 00          STA B  0,X
C309 A7 01          STA A  1,X      WRITE CTRL REG
C30B C6 01          LDA B #1        OUTPUT FOR TONE OFF, AND...
C30D E7 00          STA B  0,X      TAPE DATA-OUT HIGH (MARKING)
C30F 39          RTS

```

```

*** INPUT ONE BYTE FROM TAPE PORT ***
C310 8D 13 INBYT BSR XCHG EXCHANGE X FOR PIA ADR'S
C312 A6 00 IN1 LDA A 0,X
C314 2B FC BMI IN1 LOOK FOR START BIT
C316 8D DD BSR DEL167 DELAY HALF BIT-TIME (300BD)
C318 C6 09 LDA B #9 DO 9 TIMES....
C31A 0D IN2 SEC ENSURE PBO MARKING
C31B 69 00 ROL 0,X INPUT & SHIFT NEXT BIT
C31D 46 ROR A INTO ACC-A
C31E 8D D3 BSR DEL333 WAIT 1 BIT-TIME
C320 5A DEC B
C321 26 F7 BNE IN2 CONT.....
C323 20 17 BRA OUTX RESTORE X AND RETURN

C325 DF 12 XCHG STX XTEMP SAVE X-REG
C327 CE 8012 LDX #PIAB
C32A 39 RTS

*** OUTPUT ONE BYTE TO TAPE PORT ***
C32B 8D F8 OUTBYT BSR XCHG
C32D 36 PSH A
C32E 6A 00 DEC 0,X RESET START BIT
C330 C6 0A LDA B #10 DO 10 TIMES.....
C332 8D BF OUT1 BSR DEL333 DELAY 1 BIT-TIME
C334 A7 00 STA A 0,X NEXT BIT TO OUT LINE (PBO)
C336 0D SEC
C337 46 ROR A
C338 5A DEC B
C339 26 F7 BNE OUT1 CONT.....
C33B 32 PUL A RESTORE A.
C33C DE 12 OUTX LDX XTEMP RESTORE X.
C33E 39 RTS

C33F 20 83 GETKEE BRA GETKEY FOR INTERLINKING

*** TAPE LOAD AND DUMP ROUTINES *****
C341 86 37 LODUMX LDA A #37 KILL DISPLAY (DMA OFF)
C343 8D B9 BSR PBINZ
C345 DE 02 LDX BEGA POINT TO FIRST LOAD/DUMP ADR
C347 39 RTS
C348 8D F7 DUMP BSR LODUMX
C34A A6 00 DUMP1 LDA A 0,X FETCH RAM BYTE
C34C 8D DD BSR OUTBYT
C34E 08 INX
C34F 9C 04 CPX ENDA (ENDA = LAST ADRS + 1)
C351 26 F7 BNE DUMP1
C353 20 0B BRA START
C355 8D EA LOAD BSR LODUMX
C357 8D B7 LOAD1 BSR INBYT
C359 A7 00 STA A 0,X STASH BYTE IN RAM
C35B 08 INX
C35C 9C 04 CPX ENDA DONE?
C35E 26 F7 BNE LOAD1 CONT....
*** (BRA START) ***

```

***** MONITOR ENTRY POINT *****

C360 8E 007F	START	LDS	#STOP	RESET SP TO TOP
C363 CE C3E9		LDX	#RTC	SETUP IRQ VECTOR FOR RTC
C366 DF 00		STX	IRQV	
C368 86 3F		LDA A	#3F	SETUP I/O PORT; DISPLAY ON.
C36A 8D 92		BSR	PBINZ	
C36C 8D 43		BSR	SHOADR	PROMPT
C36E 0E		CLI		ENABLE RELATIVE TIME CLOCK
C36F 8D CE	COMAND	BSR	GETKEE	INPUT SOMETHING
C371 4D		TST A		
C372 2A 10		BPL	INADRS	IF HEX, GET AN ADDRESS
C374 8D C9		BSR	GETKEE	IF FN, GET A COMMAND
C376 84 03		AND A	#3	
C378 27 23		BEQ	MEMOD	0 = MEMORY MODIFY
C37A 4A		DEC A		
C37B 27 D8		BEQ	LOAD	1 = TAPE LOAD
C37D 4A		DEC A		
C37E 27 C8		BEQ	DUMP	2 = TAPE DUMP
C380 DE 06	GO	LDX	ADRS	FETCH ADRS FOR GO
C382 6E 00		JMP	0,X	
C384 8D 0C	INADRS	BSR	BYT1	BUILD ADRS MS BYTE
C386 97 06		STA A	ADRS	
C388 8D 06		BSR	BYTIN	INPUT & BUILD LSB
C38A 97 07		STA A	ADRS+1	
C38C 8D 23		BSR	SHOADR	DISPLAY RESULTANT ADRS
C38E 20 DF		BRA	COMAND	
C390 8D AD	BYTIN	BSR	GETKEE	INPUT 2 HEX DIGITS
C392 48	BYT1	ASL A		LEFT JUSTIFY FIRST DIGIT
C393 48		ASL A		
C394 48		ASL A		
C395 48		ASL A		
C396 97 0F		STA A	ATEMP	HOLD IT
C398 8D A5		BSR	GETKEE	INPUT ANOTHER DIGIT
C39A 9B 0F		ADD A	ATEMP	BUILD A BYTE
C39C 39		RTS		
*** MEMORY MODIFY ROUTINE				
C39D 8D 12	MEMOD	BSR	SHOADR	SHOW CURRENT ADRS
C39F DE 06		LDX	ADRS	SHOW DATA AT ADRS
C3A1 8D 25		BSR	SHODAT	
C3A3 8D 9A		BSR	GETKEE	WAIT FOR INPUT
C3A5 4D		TST A		
C3A6 2B 04		BMI	MEM1	FN KEY; NEXT ADRS
C3A8 8D E8		BSR	BYT1	HEX KEY; NEW DATA BYTE
C3AA A7 00		STA A	0,X	DEPOSIT IT
C3AC 08	MEM1	INX		
C3AD DF 06		STX	ADRS	BUMP ADRS
C3AF 20 EC		BRA	MEMOD	

C3B1 86 10	SHOADR	LDA A	#\$10	SET CURSOR HOME POSN
C3B3 8D 2B		BSR	CURS1	
C3B5 CE 01C8		LDX	#DISBUF+200	ILLUMINATE LAST 7 ROWS
C3B8 86 FF		LDA A	#\$FF	
C3BA BD C07D		JSR	FILL	
C3BD CE 0006		LDX	#ADRS	POINT TO ADRS MS BYTE
C3C0 8D 06		BSR	SHODAT	
C3C2 08		INX		POINT TO ADRS LS BYTE
C3C3 8D 03		BSR	SHODAT	
C3C5 8D 15		BSR	CURS1	MOVE CURSOR RIGHT
C3C7 39		RTS		
C3C8 A6 00	SHODAT	LDA A	0,X	FETCH DATA @ X
C3CA 36	SHOBYT	PSH A		
C3CB 44		LSR A		ISOLATE MS DIGIT
C3CC 44		LSR A		
C3CD 44		LSR A		
C3CE 44		LSR A		
C3CF 8D 01		BSR	DIGOUT	SHOW ONE DIGIT
C3D1 32		PUL A		
C3D2 DF 12	DIGOUT	STX	XTEMP	SAVE X
C3D4 BD C193		JSR	LETDSP	POINT TO DIGIT PATTERN
C3D7 C6 05		LDA B	#5	SHOW 5-BYTE PATTERN
C3D9 BD C224		JSR	SHOWI	
C3DC 86 04	CURS1	LDA A	#4	SHIFT CURSOR RIGHT 4 DOTS
C3DE 9B 2E		ADD A	VX	
C3E0 97 2E	CURS1	STA A	VX	SET X COORD
C3E2 86 1A		LDA A	#\$1A	SET Y COORD
C3E4 97 2F		STA A	VY	
C3E6 DE 12		LDX	XTEMP	RESTORE X-REG
C3E8 39		RTS		

 *** REAL TIME CLOCK INTERRUPT SERVICE ROUTINE:

C3E9 7A 0020	RTC	DEC	TIME	
C3EC 7A 0021		DEC	TONE	
C3EF 7D 8012		TST	PIAB	CLEAR IRQ FLAG IN PIA
C3F2 3B		RTI		
C3F3 DE 00	IRQ	LDX	IRQV	INDIRECT JUMP VIA IRQV
C3F5 6E 00		JMP	0,X	
C3F7 00		FCB	0	

*** RESTART AND INTERRUPT TRAPS:

C3F8 C3 F3	FDB	IRQ	(ALLOWS USER-WRITTEN ISR)
C3FA 00 80	FDB	\$0080	SWI ROUTINE AT \$0080 (OPTION
C3FC 00 83	FDB	\$0083	NMI ROUTINE AT \$0083 (OPTION
C3FE C3 60	FDB	START	

END CHIPOS INTERPRETER/MONITOR