```
; *********************************************************
; *********************************************************
; ***                                                   ***
; ***          G C E     === V E C T R E X ===          ***
; ***                                                   ***
; ***                      E X E C U T I V E            ***
; ***                                                   ***
; *********************************************************
; *********************************************************
;
;
;
;==============================================================================
;
; REV     DATE     PROG     COMMENT(S)
; ---     ----     ----     ----------
;
;  C     MM/DD/YY JJH       CHANGE TO 'RASTER'
;                               RECOMMENDED BY MARK INDICTOR
;
;                           DOCUMENTATION CHANGES
;
;
;  B1    4/06/83   JJH      STROKE SEQUENCE IN 'JOYBIT'
;                               RECOMMENDED BY GARY LORENC
;
;                           CHANGE TO 'RASTER'
;                               RECOMMENDED BY GERRY KARR
;
;
;  B    12/28/82   JJH      TITLE PAGES CAN NOW BE ABORTED
;
;                           DOCUMENTATION CHANGES
;
;
;  A    09/29/82   JJH      LABEL AND DOCUMENTATION CHANGES
;
;
;   -   05/16/82   RELEASE FROM WESTERN TECHNOLOGIES
;
;==============================================================================
                        ; ************************************
                        ; ************************************
                        ; ***                              ***
                        ; ***         E Q U A T E S        ***
                        ; ***                              ***
                        ; ************************************
                        ; ************************************
                        ;
                        ;
  = 00CC                PZERO   EQU     $CC             ; ZERO INTEGRATORS
  = 00CE                PUNZRO  EQU     $CE             ; RELEASE INTEGRATORS
  = 007F                SCAL1X  EQU     $7F             ; TIMING PARMAMETERS FOR VECTOR WRITE
  = 00FF                SCAL2X  EQU     $FF             ; .
                        ;
  = 3075                MSEC20  EQU     $3075           ; FRAME RATE - 50 HERTZ (LSB, MSB)
                        ;
  = 7FA0                PSCOR1  EQU     $7FA0           ; POSITION OF PLAYER #1 SCORE
  = 7F10                PSCOR2  EQU     $7F10           ; POSITION OF PLAYER #2 SCORE
                        ;
                        ;
                        ;
                        ;
                        ; *************************************************
                        ; *************************************************
                        ; ***                                           ***
                        ; ***         G A M E   C A R T R I D G E       ***
                        ; ***                                           ***
                        ; *************************************************
                        ; *************************************************
                        ;
                        ;
  = 0000                GAMCRT  EQU     $0000
```

```
                        ; ********************************************************************
                        ; ********************************************************************
                        ; ***                                                              ***
                        ; ***         E X E C U T I V E   W O R K I N G   S T O R A G E    ***
                        ; ***                                                              ***
                        ; ********************************************************************
                        ; ********************************************************************
                        ;
0000  = C800                    ORG     $C800
                        ;       ===     =====
                        ;
                        ;
                        ; SOUND GENERATOR MIRROR
                        ; ======================
                        ;
C800  = 0001            REG0    DS      1               ; CHANNEL A: FINE TONE PERIOD
C801  = 0001            REG1    DS      1               ; CHANNEL A: COURSE TONE PERIOD
C802  = 0001            REG2    DS      1               ; CHANNEL B: FINE TONE PERIOD
C803  = 0001            REG3    DS      1               ; CHANNEL B: COURSE TONE PERIOD
C804  = 0001            REG4    DS      1               ; CHANNEL C: FINE TONE PERIOD
C805  = 0001            REG5    DS      1               ; CHANNEL C: COURSE TONE PERIOD
C806  = 0001            REG6    DS      1               ; NOISE PERIOD
C807  = 0001            REG7    DS      1               ; TONE / NOISE ENABLES
C808  = 0001            REG8    DS      1               ; CHANNEL A: AMPLITUDE
C809  = 0001            REG9    DS      1               ; CHANNEL B: AMPLITUDE
C80A  = 0001            REGA    DS      1               ; CHANNEL C: AMPLITUDE
C80B  = 0001            REGB    DS      1               ; FINE ENVELOPE PERIOD
C80C  = 0001            REGC    DS      1               ; COURSE ENVELOPE PERIOD
C80D  = 0001            REGD    DS      1               ; ENVELOPE SHAPE / CYCLE
C80E  = 0001            REGE    DS      1               ; I/O PORT DATA
                        ;
                        ;
                        ;
                        ;
                        ; CONTROLLER BUTTON RESULTS
                        ; =========================
                        ;
C80F  = 0002            TRIGGR  DS      2               ; COLLECTIVE BUTTON STATUS
C811  = 0001            EDGE    DS      1               ; .
                        ;
C812  = 0001            KEY0    DS      1               ; CONTROLLER #1 - LEFT MOST BUTTON
C813  = 0001            KEY1    DS      1               ; .
C814  = 0001            KEY2    DS      1               ; .
C815  = 0001            KEY3    DS      1               ; .             - RIGHT MOST BUTTON
                        ;
C816  = 0001            KEY4    DS      1               ; CONTROLLER #2 - LEFT MOST BUTTON
C817  = 0001            KEY5    DS      1               ; .
C818  = 0001            KEY6    DS      1               ; .
C819  = 0001            KEY7    DS      1               ; .             - RIGHT MOST BUTTON
```

```
                        ; CONTROLLER JOYSTICK RESULTS
                        ; ==========================
                        ;
C81A  = 0001            POTRES  DS      1               ; SUCCESSIVE APPROXIMATION ACCURACY
                        ;                                     $40 = 1 BIT
                        ;                                     $20 = 2 BITS
                        ;                                     $01 = 7 BITS
                        ;                                     $00 = 8 BITS (DEFAULT)
                        ;
C81B  = 0001            POT0    DS      1               ; JOYSTICK #1 - 'X' AXIS
C81C  = 0001            POT1    DS      1               ; .           - 'Y' AXIS
                        ;
C81D  = 0001            POT2    DS      1               ; JOYSTICK #2 - 'X' AXIS
C81E  = 0001            POT3    DS      1               ; .           - 'Y' AXIS
                        ;
C81F  = 0001            EPOT0   DS      1               ; ENABLE POT READ
C820  = 0001            EPOT1   DS      1               ; .
C821  = 0001            EPOT2   DS      1               ; .
C822  = 0001            EPOT3   DS      1               ; .
                        ;
                        ;
C823  = 0001            LIST    DS      1               ; NUMBER OF VECTORS IN LIST
C824  = 0001            ZSKIP   DS      1               ; SKIP INTEGRATOR ZEROING AND ACTIVE GROUND
                        ;
C825  = 0001                    DS      1               ; FRAME COUNTER (MSB)
C826  = 0001            FRAME   DS      1               ; .    LSB
                        ;
C827  = 0001            TENSTY  DS      1               ; CURRENT INTENSITY
C828  = 0001            DWELL   DS      1               ; DOT DWELL TIME
C829  = 0001            DASH    DS      1               ; DASHED VECTOR PATTERN
C82A  = 0002            SIZRAS  DS      2               ; RASTER MESSAGE SIZE
                        ;                                     SIZE     = 'Y' AXIS
                        ;                                     SIZE + 1 = 'X' AXIS
C82C  = 0002            MESAGE  DS      2               ; POINTER TO RASTER MESSAGE STRING
                        ;
                        ;
C82E  = 0001            XTMR0   DS      1               ; EXECUTIVE INTERVAL TIMERS
C82F  = 0001            XTMR1   DS      1               ; .
C830  = 0001            XTMR2   DS      1               ; .
C831  = 0001            XTMR3   DS      1               ; .
C832  = 0001            XTMR4   DS      1               ; .
C833  = 0001            XTMR5   DS      1               ; .
```

```
C834  = 0001       ABSY    DS      1               ; EXECUTIVE WORKING STORAGE
C835  = 0001       ABSX    DS      1               ; .
C836  = 0001       ANGLE   DS      1               ; .
C837  = 0002       WSINE   DS      2               ; .
C839  = 0002       WCSINE  DS      2               ; .
C83B  = 0001       LEC     DS      1               ; .
C83C  = 0001       LAC     DS      1               ; .
C83D  = 0002       FRMTIM  DS      2               ; BASE FRAME DURATION
                   ;
                   ;
                   ; TUNE / SOUND-EFFECT WORKING STOARGE
                   ; ==================================
                   ;
C83F  = 0001       REQ0    DS      1               ; ENVELOPE SHAPE / CYCLE
C840  = 0001       REQ1    DS      1               ; COURSE ENVELOPE PERIOD
C841  = 0001       REQ2    DS      1               ; FINE ENVELOPE PERIOD
C842  = 0001       REQ3    DS      1               ; CHANNEL C: AMPLITUDE
C843  = 0001       REQ4    DS      1               ; CHANNEL B: AMPLITUDE
C844  = 0001       REQ5    DS      1               ; CHANNEL A: AMPLITUDE
C845  = 0001       REQ6    DS      1               ; TONE / NOISE ENABLES
C846  = 0001       REQ7    DS      1               ; NOISE PERIOD
C847  = 0001       REQ8    DS      1               ; CHANNEL C: COURSE TONE PERIOD
C848  = 0001       REQ9    DS      1               ; CHANNEL C: FINE TONE PERIOD
C849  = 0001       REQA    DS      1               ; CHANNEL B: COURSE TONE PERIOD
C84A  = 0001       REQB    DS      1               ; CHANNEL B: FINE TONE PERIOD
C84B  = 0001       REQC    DS      1               ; CHANNEL A: COURSE TONE PERIOD
C84C  = 0001       REQD    DS      1               ; CHANNEL A: FINE TONE PERIOD
                   ;
                   ;
C84D  = 0002       DOREMI  DS      2               ; NOTE TABLE POINTER
C84F  = 0002       FADE    DS      2               ;; FOR TUNE PLAYER USE
C851  = 0002       VIBE    DS      2               ;; .
C853  = 0002       TUNE    DS      2               ;; .
C855  = 0001       NEWGEN  DS      1               ;; .
C856  = 0001       TSTAT   DS      1               ;; .
C857  = 0001       RESTC   DS      1               ;; .
C858  = 0001       RATEA   DS      1               ;; .
C859  = 0001       VIBA    DS      1               ;; .
C85A  = 0001       RATEB   DS      1               ;; VIBRATO
C85B  = 0001       VIBB    DS      1               ;; .
C85C  = 0001       RATEC   DS      1               ;; .
C85D  = 0001       VIBC    DS      1               ;; .
C85E  = 0001       FADEA   DS      1               ;; .
C85F  = 0001       FADEB   DS      1               ;; .
C860  = 0001       FADEC   DS      1               ;; .
C861  = 0002       TONEA   DS      2               ;; FREQ BEFORE VIBE
C863  = 0002       TONEB   DS      2               ;; .
C865  = 0002       TONEC   DS      2               ;; .
```

```
C867  = 0001        SATUS   DS      1               ;;  GAME-SOUNDS INTERFACE
C868  = 0001        LATUS   DS      1               ;;  LAST 'SATUS' VALUE
C869  = 0001        XATUS   DS      1               ;;  ", EXPLOSION TYPE
C86A  = 0001        GAP     DS      1               ;;  TIMES HOLD IN BACKGROUND SOUND
C86B  = 0002        B1FREQ  DS      2               ;;
C86D  = 0002        B2FREQ  DS      2               ;;  .
C86F  = 0002        F1FREQ  DS      2               ;;  .
C871  = 0001        FEAST   DS      1               ;;  FIRING STATE COUNTER
C872  = 0001        PEDGE   DS      1               ;;  POSITIVE EDGE OF 'SATUS'
C873  = 0001        NEDGE   DS      1               ;;  NEGATIVE EDGE OF 'SATUS'
C874  = 0002        K1FREQ  DS      2               ;;  BACKGROUND FREQ 1
C876  = 0001        BACON   DS      1               ;;  " SEQUENCER
C877  = 0001        XACON   DS      1               ;;  EXPLO SEQ
C878  = 0001        SPEKT   DS      1               ;;  EXPL SPECTR
                    ;
                    ;
C879  = 0001        PLAYRS  DS      1               ;  NUMBER OF PLAYERS IN GAME
C87A  = 0001        OPTION  DS      1               ;  GAME OPTION NUMBER
                    ;
C87B  = 0002        SEED    DS      2               ;  SEED FOR RANDOM NUMBER GENERATOR
C87D  = 0003        RANCID  DS      3               ;  .
                    ;
      = C880        LASRAM  EQU     *               ;  FIRST AVAILABE BYTE AFTER EXECUTIVE RAM
      = CA00        RAMMES  EQU     $CA00           ;  .
                    ;
                    ;
C880  = CBEA                ORG     $CBEA
                    ;       ===     =====
                    ;
CBEA  = 0001        STACK   DS      1               ;  EXECUTIVE STACK
CBEB  = 0007        HISCOR  DS      7               ;  HIGH-SCORE
                    ;
CBF2  = 0003        VSWI2   DS      3               ;  VECTOR FOR SOFTWARE INTERRUPTS #2 & #3
CBF5  = 0003        VFIRQ   DS      3               ;  VECTOR FOR FAST INTERRUPT
CBF8  = 0003        VIRQ    DS      3               ;  VECTOR FOR MASKABLE INTERRUPT
CBFB  = 0003        VSWI    DS      3               ;  VECTOR FOR SOFTWARE INTERRUPT #1
                    ;
CBFE  0000          RESTFLG DW      $0000           ;  COLD / WARM-BOOT RESTART FLAG
```

```
                      ; **********************************************************
                      ; **********************************************************
                      ; ***                                                    ***
                      ; ***          M I N E - S T O R M   S T O R A G E       ***
                      ; ***                                                    ***
                      ; **********************************************************
                      ; **********************************************************
                      ;
                      ;      ================================================================
                      ;      ===== THESE LOCATIONS ARE SHOWN FOR THE GENERAL USE SUBROUTINES =====
                      ;      ================================================================
                      ;
                      ;
CC00  = C880                 ORG     $C880
                      ;      ===     =====
                      ;
C880  00              SBTN   DB      0                  ; CONTROLLER DEBOUNCE FLAGS
C881  0000            SJOY   DW      0                  ; JOYSTICK 'BANG' FLAGS
                      ;
                      ;
C883  00              ETMP1  DB      0                  ; TEMPORARY WORKING STORAGE (FIRST LEVEL)
C884  00              ETMP2  DB      0                  ; .
C885  00              ETMP3  DB      0                  ; .
C886  00              ETMP4  DB      0                  ; .
C887  00              ETMP5  DB      0                  ; .
C888  00              ETMP6  DB      0                  ; .
C889  00              ETMP7  DB      0                  ; .
C88A  00              ETMP8  DB      0                  ; .
C88B  00              ETMP9  DB      0                  ; .
C88C  00              ETMP10 DB      0                  ; .
                      ;
C88D  0000                   DW      0                  ; .    WORKING STORAGE SLOP
                      ;
C88F  00              TEMP1  DB      0                  ; TEMPORARY WORKING STORAGE (SECOND LEVEL)
C890  00              TEMP2  DB      0                  ; .
C891  00              TEMP3  DB      0                  ; .
C892  00              TEMP4  DB      0                  ; .
C893  00              TEMP5  DB      0                  ; .
C894  00              TEMP6  DB      0                  ; .
C895  00              TEMP7  DB      0                  ; .
C896  00              TEMP8  DB      0                  ; .
C897  00              TEMP9  DB      0                  ; .
C898  00              TEMP10 DB      0                  ; .
                      ;
C899  0000                   DW      0                  ; .    WORKING STORAGE SLOP
```

```
C89B  00            ACTPLY  DB      0             ;  ACTIVE PLAYER FLAG ($00 / $02)
                    ;
                    ;
C89C  00            TMR1    DB      0             ;  TIMER #1 - DOWN COUNTER
C89D  0000                  DW      0             ;  .        - TIME-OUT ROUTINE POINTER
                    ;
C89F  00            TMR2    DB      0             ;  TIMER #2 - DOWN COUNTER
C8A0  0000                  DW      0             ;  .        - TIME-OUT ROUTINE POINTER
                    ;
C8A2  00            TMR3    DB      0             ;  TIMER #3 - DOWN COUNTER
C8A3  0000                  DW      0             ;  .        - TIME-OUT ROUTINE POINTER
                    ;
C8A5  00            TMR4    DB      0             ;  TIMER #4 - DOWN COUNTER
C8A6  0000                  DW      0             ;  .        - TIME-OUT ROUTINE POINTER
                    ;
                    ;
C8A8  = 0007        SCOR1   DS      7             ;  PLAYER #1 SCORE
C8AF  = 0007        SCOR2   DS      7             ;  PLAYER #2 SCORE
                    ;
                    ;
      = CB71        FSTR    EQU     $CB71         ;  STAR FIELD TABLES
      = CB81        ZSTR    EQU     $CB81         ;  .
```

```
                        ;  ************************************************
                        ;  ************************************************
                        ;  ***                                          ***
                        ;  ***          I / O   R E G I S T E R S       ***
                        ;  ***                                          ***
                        ;  ************************************************
                        ;  ************************************************
                        ;
                        ;
 = D000                 CNTRL    EQU     $D000           ;  CONTROL REGISTER
                        ;                                     BIT 7  =  RAMP ENABLE
                        ;                                     BIT 6  =  CARTRIDGE EXTERNAL FUNCTION
                        ;                                     BIT 5  =  COMPARE
                        ;                                     BIT 4  =  BDIR (TO PSG)
                        ;                                     BIT 3  =  BC1 (TO PSG)
                        ;                                     BIT 2  =  SAMPLE / HOLD SELECT, BIT 1
                        ;                                     BIT 1  =  .                     BIT 0
                        ;                                     BIT 0  =  SAMPLE / HOLD ENABLE
                        ;
 = D001                 DAC      EQU     $D001           ;  D/A DATA
                        ;
 = D002                 DCNTRL   EQU     $D002           ;  DIRECTION FOR CONTROL REGISTER
 = D003                 DDAC     EQU     $D003           ;  DIRECTION FOR DAC DATA
                        ;
 = D004                 T1LOLC   EQU     $D004           ;  TIMER #1
 = D005                 T1HOC    EQU     $D005           ;  .
 = D006                 T1LOL    EQU     $D006           ;  .
 = D007                 T1HOL    EQU     $D007           ;  .
                        ;
 = D008                 T2LOLC   EQU     $D008           ;  TIMER #2
 = D009                 T2HOC    EQU     $D009           ;  .
                        ;
 = D00A                 SHIFT    EQU     $D00A           ;  SHIFT REGISTER
 = D00B                 ACNTRL   EQU     $D00B           ;  .
 = D00C                 PCNTRL   EQU     $D00C           ;  .
                        ;
 = D00D                 IFLAG    EQU     $D00D           ;  INTERRUPT FLAG REGISTER
 = D00E                 IENABL   EQU     $D00E           ;  INTERRUPT ENABLE REGISTER
```

```
                        ; ****************************************
                        ; ****************************************
                        ; ***                                  ***
                        ; ***        M I N E - S T O R M       ***
                        ; ***                                  ***
                        ; ****************************************
                        ; ****************************************
                        ;
                        ;
      = E000            STORM   EQU     $E000
                        ;
                        ; ===== REFER TO 'MINE-STORM' LISTING =====
                        ;
                        ;
                        ; ***************************************************************
                        ; ***************************************************************
                        ; ***                                                         ***
                        ; ***        M I N E - S T O R M   S U B R O U T I N E S      ***
                        ; ***                                                         ***
                        ; ***************************************************************
                        ; ***************************************************************
                        ;
                        ;
                        ; FORM 'YX' DISPLACEMENTS (X8)
                        ; ============================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          A  = SPEED VECTOR
                        ;          B  = DIRECTION (ANGLE)
                        ;          DP = $C8
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          X  = 'X' DISPLACEMENT VALUE (MSB/LSB)
                        ;          Y  = 'Y' DISPLACEMENT VALUE (MSB/LSB)
                        ;
C8B6  = E7B5                    ORG     $E7B5
E7B5  = 00C8                    SETDP   $C8
                        ;       =====   =====
                        ;
E7B5  3436              MLTY8   PSHS    A,B,X,Y         ; SAVE ENTRY VALUES
                        ;
E7B7  BDF601                    JSR     LNROT           ; ROTATE LINE (SPEED VECTOR)
E7BA  A764                      STA     4,S             ; .
                        ;
E7BC  1D                        SEX                     ; FORM 'X' DISPLACEMENT (8X)
E7BD  58                        ASLB                    ; .    MULTIPLY BY EIGHT
E7BE  49                        ROLA                    ; .    .
E7BF  58                        ASLB                    ; .    .
E7C0  49                        ROLA                    ; .    .
E7C1  58                        ASLB                    ; .    .
E7C2  49                        ROLA                    ; .    .
```

```
E7C3  ED62                      STD     2,S             ; .     .
                        ;
E7C5  E664                      LDB     4,S             ; FORM 'Y' DISPLACEMENT (8X)
E7C7  1D                        SEX                     ; .     EXTEND SIGN
E7C8  58                        ASLB                    ; .     MULTIPLY BY EIGHT
E7C9  49                        ROLA                    ; .     .
E7CA  58                        ASLB                    ; .     .
E7CB  49                        ROLA                    ; .     .
E7CC  58                        ASLB                    ; .     .
E7CD  49                        ROLA                    ; .     .
E7CE  ED64                      STD     4,S             ; .     .
                        ;
E7D0  35B6                      PULS    A,B,X,Y,PC      ; RETURN TO CALLER
                        ;
                        ;
                        ;  FORM 'YX' DISPLACEMENTS (X16)
                        ;  ============================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;
                        ;          A  = SPEED VECTOR
                        ;          B  = DIRECTION (ANGLE)
                        ;          DP = $C8
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;
                        ;          X  = 'X' DISPLACEMENT VALUE (MSB/LSB)
                        ;          Y  = 'Y' DISPLACEMENT VALUE (MSB/LSB)
                        ;
E7D2  = E7D2                    ORG     $E7D2
E7D2  = 00C8                    SETDP   $C8
                        ;       =====   =====
                        ;
E7D2  3436              MLTY16  PSHS    A,B,X,Y         ; SAVE ENTRY VALUES
                        ;
E7D4  8DDF        ^E7B5         BSR     MLTY8           ; CALCULATE 8X DISPLACEMENTS
                        ;
E7D6  EC7C                      LDD     -4,S            ; FORM 'Y' DISPLACEMENT (16X)
E7D8  58                        ASLB                    ; .     MULTIPLY BY TWO
E7D9  49                        ROLA                    ; .     .
E7DA  ED64                      STD     4,S             ; .     .
                        ;
E7DC  EC7A                      LDD     -6,S            ; FORM 'X' DISPLACEMENT (16X)
E7DE  58                        ASLB                    ; .     MULTIPLY BY TWO
E7DF  49                        ROLA                    ; .     .
E7E0  ED62                      STD     2,S             ; .     .
                        ;
E7E2  35B6                      PULS    A,B,X,Y,PC      ; RETURN TO CALLER
                        ;
                        ;
                        ;  INITIALIZE STAR FIELDS
                        ;  ======================
                        ;
```

```
                         ;      ENTRY VALUES
                         ;      ------------
                         ;         NONE REQUIRED
                         ;
                         ;      RETURN VALUES
                         ;      -------------
                         ;         A  = .
                         ;         B  = .
                         ;         X  = .
                         ;         Y  = .
                         ;         U  = .
                         ;
                         ;
E7E4  = E8E3                    ORG     $E8E3
E8E3  = 0000                    SETDP   $00
                         ;      =====   =====
                         ;
E8E3  8EEDE0             ISTARS LDX     #STAR1
E8E6  108ECB71                  LDY     #FSTR
E8EA  CECB81                    LDU     #ZSTR
                         ;
E8ED  C608                      LDB     #$08
E8EF  8616                      LDA     #$16
                         ;
E8F1  AFA1               ST101  STX     Y++
E8F3  3008                      LEAX    8,X
E8F5  A7C0                      STA     U+
E8F7  8B0F                      ADDA    #$0F
E8F9  5A                        DECB
E8FA  26F5      ^E8F1           BNE     ST101
E8FC  39                        RTS
                         ;
                         ;
                         ;  ZOOM STAR FIELDS FORWARD AND DISPLAY
                         ;  ====================================
                         ;
                         ;      ENTRY VALUES
                         ;      ------------
                         ;         A  = STAR FIELD LIMIT
                         ;         B  = ZOOM VALUE
                         ;         DP = $D0
                         ;
                         ;      RETURN VALUES
                         ;      -------------
                         ;         SAME AS ENTRY VALUES
                         ;
E8FD  = E8FD                    ORG     $E8FD
E8FD  = 00D0                    SETDP   $D0
                         ;      =====   =====
                         ;
E8FD  341E               FSTARS PSHS    A,B,X,DP        ; SAVE ENTRY VALUES
                         ;
E8FF  8ECB81                    LDX     #ZSTR           ; BUMP ZOOM VALUES
```

```
E902  8608                      LDA     #$08            ; .
E904  6C80             ST201    INC     X+              ; .
E906  4A                        DECA                    ; .
E907  26FB      ^E904           BNE     ST201           ; .
                       ;
E909  2002      ^E90D           BRA     DSTRS1          ; DISPLAY NEW STAR FIELDS
                       ;
                       ;
                       ;  DISPLAY STAR FIELDS
                       ;  ===================
                       ;
                       ;        ENTRY VALUES
                       ;        ------------
                       ;
                       ;           A  = STAR FIELD INNER LIMIT
                       ;           B  = ZOOM VALUE
                       ;           DP = $D0
                       ;
                       ;        RETURN VALUES
                       ;        -------------
                       ;
                       ;           SAME AS ENTRY VALUES
                       ;
E90B  = E90B                    ORG     $E90B
E90B  = 00D0                    SETDP   $D0
                       ;        =====   =====
                       ;
E90B  341E             DSTARS   PSHS    A,B,X,DP        ; SAVE ENTRY VALUES
                       ;
E90D  86D0             DSTRS1   LDA     #$D0            ; SET "DP" REGISTER TO I/O
E90F  1F8B                      TFR     A,DP            ; .
                       ;
E911  8609                      LDA     #$09            ; SET FIELD COUNT
E913  3402                      PSHS    A               ; .
                       ;
E915  6AE4             ST000    DEC     ,S              ; MOVE TO NEXT STAR FIELD
E917  2607      ^E920           BNE     ST010           ; .
                       ;
E919  BDF354                    JSR     ZERGND          ; ZERO INTEGRATORS
                       ;
E91C  3502                      PULS    A               ; RETURN TO CALLER
E91E  359E                      PULS    A,B,X,DP,PC     ; .
                       ;
                       ;
E920  BDF354           ST010    JSR     ZERGND          ; TURN-OFF CRT GUN AND ZERO INTEGRATORS
                       ;
E923  8603                      LDA     #$03            ; SET DOT COUNT
E925  B7C823                    STA     LIST            ; .
                       ;
E928  A6E4                      LDA     ,S              ; FETCH ZOOM VALUE FOR THIS FIELD
E92A  4A                        DECA                    ; .
E92B  8ECB81                    LDX     #ZSTR           ; .
E92E  E686                      LDB     A,X             ; .
E930  C47F                      ANDB    #$7F            ; .
                       ;
```

```
E932  E161                     CMPB    1,S            ; HAS STAR FIELD REACHED ITS LIMIT ?
E934  23DF      ^E915          BLS     ST000          ; .    IF SO, FETCH NEXT STAR FIELD
E936  E062                     SUBB    2,S            ; .    MODIFY VECTOR LENGTH WITH ZOOM VALUE
E938  2FDB      ^E915          BLE     ST000          ; .    .
E93A  D704                     STB     T1LOLC         ; .    SET VECTOR LENGTH
                      ;
E93C  8ECB71                   LDX     #FSTR          ; FETCH STAR FIELD POINTER
E93F  48                       LSLA                   ; .
E940  AE86                     LDX     A,X            ; .
                      ;
E942  BDF2A9                   JSR     INTMAX         ; SET BRIGHTNESS
E945  BDF2D5                   JSR     DIFDOT         ; DRAW STAR FIELD
E948  20CB      ^E915          BRA     ST000          ; SET-UP FOR NEXT STAR-FIELD
                      ;
                      ;
                      ; DETERMINE RANDOM 'Y:X' POSITION
                      ; ===============================
                      ;
                      ;        ENTRY VALUES
                      ;        ------------
                      ;           NONE REQUIRED
                      ;
                      ;        RETURN VALUES
                      ;        -------------
                      ;           A  = 'Y' AXIS VALUE ($00 - $FF)
                      ;           B  = 'X' AXIS VALUE ($60 - $7F, $A0 - $FF)
                      ;
E94A  = E98A                   ORG     $E98A
E98A  = 0000                   SETDP   $00
                      ;        =====   =====
                      ;
E98A  3406            RANPOS   PSHS    D              ; SAVE ENTRY VALUES
                      ;
E98C  BDF517                   JSR     RANDOM         ; 'Y' POSITION
E98F  A7E4                     STA     ,S             ; .
                      ;
E991  BDF517          RANPS1   JSR     RANDOM         ; 'X' POSITION
E994  8160                     CMPA    #$60           ; .
E996  2EF9      ^E991          BGT     RANPS1         ; .
E998  81A0                     CMPA    #$A0           ; .
E99A  2DF5      ^E991          BLT     RANPS1         ; .
E99C  A761                     STA     1,S            ; .
                      ;
E99E  3506                     PULS    D              ; RETURN TO CALLER
E9A0  39                       RTS                    ; .
                      ;
                      ;
                      ; SELECT DIRECTION WITHIN LIMIT CONES
                      ; ===================================
                      ;
                      ;        ENTRY VALUES
                      ;        ------------
                      ;           NONE REQUIRED
```

```
                           ;
                           ;        RETURN VALUES
                           ;        -------------
                           ;
                           ;          B  = RANDOM ANGLE WITHIN LIMIT CONES
                           ;
E9A1 = EA3E                         ORG    $EA3E
EA3E = 0000                         SETDP  $00
                           ;        =====  =====
                           ;
EA3E 3406                  CONE     PSHS   A,B               ; SAVE ENTRY VALUES
                           ;
EA40 BDF517                         JSR    RANDOM            ; FETCH RANDOM NUMBER
EA43 1F89                           TFR    A,B               ; .    SET-UP FOR CONE TESTS
EA45 8430                           ANDA   #$30              ; .    .
EA47 A761                           STA    1,S               ; .    .
                           ;
EA49 C40F                           ANDB   #$0F              ; LIMIT DIRECTION WITHIN CONE
EA4B C104                           CMPB   #$04              ; .    TEST AGAINST LOW-END LIMIT
EA4D 2402        ^EA51              BHS    CONE1             ; .    .
EA4F CB04                           ADDB   #$04              ; .    .    MOVE LOW-END UP
                           ;
EA51 C10C                  CONE1    CMPB   #$0C              ; .    TEST AGAINST UPPER-END LIMIT
EA53 2302        ^EA57              BLS    CONE2             ; .    .
EA55 C004                           SUBB   #$04              ; .    .    MOVE UPPER-END DOWN
                           ;
EA57 EB61                  CONE2    ADDB   1,S               ; ADD QUADRANT TO DIRECTION
EA59 E761                           STB    1,S               ; .
EA5B 3586                           PULS   A,B,PC            ; .    RETURN TO CALLER
                           ;
                           ;
                           ;  POSITION AND DRAW DOT
                           ;  =====================
                           ;
                           ;        ENTRY VALUES
                           ;        ------------
                           ;
                           ;          Y  = ABSOLUTE 'YX' POSITION
                           ;          DP = $D0
                           ;
                           ;        RETURN VALUES
                           ;        -------------
                           ;
                           ;          SAME AS ENTRY VALUES
                           ;
EA5D = EA5D                         ORG    $EA5D
EA5D = 00D0                         SETDP  $D0
                           ;        =====  =====
                           ;
EA5D 3406                  ADOT     PSHS   A,B               ; SAVE ENTRY VALUES
                           ;
EA5F 867F                           LDA    #$7F              ; POSITION DOT
EA61 9704                           STA    T1LOLC            ; .    SET 1X VECTOR LENGTH
EA63 1F20                           TFR    Y,D               ; .    SET 'YX' POSITION
EA65 BDF2C3                         JSR    DOTAB             ; .    .
                           ;
```

```
EA68  BDF354                   JSR     ZERGND          ;  ZERO INTEGRATORS
                   ;
EA6B  3586                     PULS    A,B,PC          ;  RETURN TO CALLER
                   ;
                   ;
                   ;  POSITION WITH 16-BIT VALUES AND DRAW DOT
                   ;  ========================================
                   ;
                   ;        ENTRY VALUES
                   ;        ------------
                   ;           Y  = POINTER TO 32-BIT ABSOLUTE 'YX' POSITION
                   ;           DP = $D0
                   ;
                   ;        RETURN VALUES
                   ;        -------------
                   ;           SAME AS ENTRY VALUES
                   ;
                   ;
EA6D  = EA6D                   ORG     $EA6D
EA6D  = 00D0                   SETDP   $D0
                   ;           =====   =====
                   ;
EA6D  3406         DDOT        PSHS    A,B             ;  SAVE ENTRY VALUES
                   ;
EA6F  867F                     LDA     #$7F            ;  POSITION DOT
EA71  9704                     STA     T1LOLC          ;  .    SET 1X VECTOR LENGTH
                   ;
EA73  A6A4                     LDA     ,Y              ;  .    SET 'YX' POSITION
EA75  E622                     LDB     2,Y             ;  .    .
EA77  BDF2C3                   JSR     DOTAB           ;  .    .
                   ;
EA7A  BDF354                   JSR     ZERGND          ;  ZERO INTEGRATORS
                   ;
EA7D  3586                     PULS    A,B,PC          ;  RETURN TO CALLER
                   ;
                   ;
                   ;  POSITION AND DRAW PACKET
                   ;  ========================
                   ;
                   ;        ENTRY VALUES
                   ;        ------------
                   ;           B  = ZOOM VALUE
                   ;           X  = PACKET ADDRESS
                   ;           Y  = ABSOLUTE 'YX' POSITION
                   ;           DP = $D0
                   ;
                   ;        RETURN VALUES
                   ;        -------------
                   ;           SAME AS ENTRY VALUES
                   ;
EA7F  = EA7F                   ORG     $EA7F
EA7F  = 00D0                   SETDP   $D0
                   ;           =====   =====
                        ;
EA7F  3416              APACK   PSHS   A,B,X            ; SAVE ENTRY VALUES
                        ;
EA81  1F20                      TFR    Y,D              ; .    SET 'YX' POSITION
EA83  BDF2FC                    JSR    POSITD           ; .    .
                        ;
EA86  E661                      LDB    1,S              ; DRAW PACKET
EA88  BDF40E                    JSR    TPACK            ; .    DRAW PACKET
                        ;
EA8B  3596                      PULS   A,B,X,PC         ; RETURN TO CALLER
                        ;
                        ;
                        ; POSITION WITH 16-BIT VALUES AND DRAW PACKET
                        ; ===========================================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;
                        ;          B  = ZOOM VALUE
                        ;          X  = ADDRESS OF PACKET
                        ;          Y  = POINTER TO 32-BIT ABSOLUTE 'YX' POSITION
                        ;          DP = $D0
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;
                        ;          SAME AS ENTRY VALUES
                        ;
EA8D  = EA8D                    ORG    $EA8D
EA8D  = 00D0                    SETDP  $D0
                        ;       =====  =====
                        ;
EA8D  3416              DPACK   PSHS   A,B,X            ; SAVE ENTRY VALUES
                        ;
EA8F  1F21                      TFR    Y,X              ; POSITION PACKET
EA91  BDF2F2                    JSR    POSWID           ; .
                        ;
EA94  E661                      LDB    1,S              ; DRAW PACKET
EA96  AE62                      LDX    2,S              ; .    FETCH PACKET POINTER
EA98  BDF40E                    JSR    TPACK            ; .    DRAW PACKET
                        ;
EA9B  3596                      PULS   A,B,X,PC         ; RETURN TO CALLER
                        ;
                        ;
                        ; DRAW COMPACT RASTER MESSAGE
                        ; ===========================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;
                        ;          U  = ADDRESS OF MESSAGE STRING
                        ;          DP = $D0
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;
                        ;          SAME AS ENTRY VALUES
                        ;
EA9D  = EA9D                  ORG     $EA9D
EA9D  = 00D0                  SETDP   $D0
                        ;     =====   =====
                        ;
EA9D  3456              SMESS PSHS    A,B,X,U         ; SAVE ENTRY VALUES
                        ;
EA9F  867F                    LDA     #$7F            ; POSITION PACKET
EAA1  9704                    STA     T1LOLC          ; .    SET 1X VECTOR LENGTH
                        ;
EAA3  BDF373                  JSR     RSTSIZ          ; DRAW RASTER MESSAGE
                        ;
EAA6  35D6                    PULS    A,B,X,U,PC      ; RETURN TO CALLER
                        ;
                        ;
                        ; POSITION AND DRAW RASTER MESSAGE
                        ; ===============================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          Y  = ABSOLUTE 'YX' POSITION
                        ;          U  = ADDRESS OF MESSAGE
                        ;          DP = $D0
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          SAME AS ENTRY VALUES
                        ;
EAA8  = EAA8                  ORG     $EAA8
EAA8  = 00D0                  SETDP   $D0
                        ;     =====   =====
                        ;
EAA8  3456              ASMESS PSHS   A,B,X,U         ; SAVE ENTRY VALUES
                        ;
EAAA  1F20                    TFR     Y,D             ; .    SET 'YX' POSITION
EAAC  BDF2FC                  JSR     POSITD          ; .    .
                        ;
EAAF  BDF495                  JSR     RASTER          ; .    DRAW PACKET
                        ;
EAB2  35B6                    PULS    A,B,X,Y,PC      ; RETURN TO CALLER
                        ;
                        ;
                        ; DRAW ACTIVE PLAYER'S SCORES
                        ; ===========================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          DP = $D0
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          A  = DESTROYED
                        ;          B  = DESTROYED
                        ;          X  = .
                        ;          Y  = DESTROYED
                        ;          U  = DESTROYED
                        ;
                        ;
EAB4  = EAB4                    ORG     $EAB4
EAB4  = 00D0                    SETDP   $D0
                        ;       =====   =====
                        ;
EAB4  BDF2A9            SCRMES  JSR     INTMAX          ; SET MAXIMUM INTENSITY
                        ;
EAB7  CCFC38                    LDD     #$FC38          ; SET RASTER SIZE
EABA  FDC82A                    STD     SIZRAS          ; .
                        ;
EABD  B6C89B                    LDA     ACTPLY          ; FETCH POSITION OF SCORE
EAC0  108EEDA3                  LDY     #PSCRPTR        ; .
EAC4  10AEA6                    LDY     A,Y             ; .
                        ;
EAC7  CEED9F                    LDU     #SCRPTR         ; FETCH ADDRESS OF SCORE
EACA  EEC6                      LDU     A,U             ; .
EACC  8DDA       ^EAA8          BSR     ASMESS          ; .
                        ;
EACE  39                        RTS                     ; RETURN TO CALLER
                        ;
                        ;
                        ;  DRAW BOTH PLAYER'S SCORES
                        ;  ========================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          X  = .
                        ;          DP = $D0
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          A  = DESTROYED
                        ;          B  = DESTROYED
                        ;          X  = .
                        ;          Y  = DESTROYED
                        ;          U  = DESTROYED
                        ;
                        ;
EACF  = EACF                    ORG     $EACF
EACF  = 00D0                    SETDP   $D0
                        ;       =====   =====
                        ;
EACF  BDF2A9            SCRBTH  JSR     INTMAX          ; SET MAXIMUM INTENSITY
                        ;
EAD2  CCFC38                    LDD     #$FC38          ; SET RASTER SIZE
EAD5  FDC82A                    STD     SIZRAS          ; .
                        ;
EAD8  108E7FA0                  LDY     #PSCOR1         ; DRAW PLAYER #1 SCORE
EADC  CEC8A8                    LDU     #SCOR1          ; .
```

```
EADF 8DC7     ^EAA8               BSR    ASMESS        ; .
                    ;
EAE1 B6C879                       LDA    PLAYRS        ; ONE OR TWO PLAYERS ?
EAE4 2709     ^EAEF               BEQ    BOTH9         ; .
EAE6 108E7F10                     LDY    #PSCOR2       ; .   DRAW PLAYER #2 SCORE
EAEA CEC8AF                       LDU    #SCOR2        ; .   .
EAED 8DB9     ^EAA8               BSR    ASMESS        ; .   .
                    ;
EAEF 39             BOTH9         RTS                  ; RETURN TO CALLER
                    ;
                    ;
                    ;  WAIT FOR FRAME BOUNDARY AND INPUT FROM CONTROLLER
                    ;  ==========================================
                    ;
                    ;       ENTRY VALUES
                    ;       ------------
                    ;         NONE REQUIRED
                    ;
                    ;       RETURN VALUES
                    ;       -------------
                    ;         A  = .
                    ;         B  = .
                    ;         X  = .
                    ;         Y  = .
                    ;         U  = .
                    ;         DP = $D0
                    ;
                    ;
EAF0 = EAF0                       ORG    $EAF0
EAF0 = 0000                       SETDP  $00
                    ;             =====  =====
                    ;
EAF0 BDF192         WAIT          JSR    FRWAIT        ; WAIT FOR FRAME BOUNDARY
EAF3 = 00D0                       SETDP  $D0           ; .   SET DP = I/O
EAF3 3408                         PSHS   DP            ; .
                    ;
EAF5 BDF2E6                       JSR    DEFLOK        ; PREVENT SCAN COLLAPSE
EAF8 BDEAB4                       JSR    SCRMES        ; DRAW PLAYER'S SCORES
                    ;
EAFB B6C880                       LDA    SBTN          ; INPUT CONSOLE SWITCHES
EAFE BDF1B4                       JSR    DBNCE         ; .
EB01 FCC881                       LDD    SJOY          ; READ JOYSTICK
EB04 FDC81F                       STD    EPOT0         ; .   ENABLE BOTH POTS ON JOYSTICK #1
EB07 FDC821                       STD    EPOT2         ; .   ENABLE BOTH POTS ON JOYSTICK #2
EB0A BDF1F8                       JSR    JOYBIT        ; .
                    ;
EB0D 86C8                         LDA    #$C8          ; SET "DP" REGISTER TO RAM
EB0F 1F8B                         TFR    A,DP          ; .
EB11 = 00C8                       SETDP  $C8           ; .
                    ;
EB11 969C           TIMER         LDA    TMR1          ; DOWN-COUNT TIMER #1
EB13 2708     ^EB1D               BEQ    DCT2          ; .   IS TIMER INHIBITED ?
EB15 0A9C                         DEC    TMR1          ; .
```

```
EB17  2604       ^EB1D            BNE     DCT2           ;  .
EB19  AD9FC89D                    JSR     [TMR1+1]       ;  .    EXECUTE THE USER PROGRAM
                          ;
EB1D  969F                DCT2    LDA     TMR2           ;  DOWN-COUNT TIMER #2
EB1F  2708       ^EB29            BEQ     DCT3           ;  .    IS TIMER INHIBITED ?
EB21  0A9F                        DEC     TMR2           ;  .
EB23  2604       ^EB29            BNE     DCT3           ;  .
EB25  AD9FC8A0                    JSR     [TMR2+1]       ;  .    EXECUTE THE USER PROGRAM
                          ;
EB29  96A2                DCT3    LDA     TMR3           ;  DOWN-COUNT TIMER #3
EB2B  2708       ^EB35            BEQ     DCT4           ;  .    IS TIMER INHIBITED ?
EB2D  0AA2                        DEC     TMR3           ;  .
EB2F  2604       ^EB35            BNE     DCT4           ;  .
EB31  AD9FC8A3                    JSR     [TMR3+1]       ;  .    EXECUTE THE USER PROGRAM
                          ;
EB35  96A5                DCT4    LDA     TMR4           ;  DOWN-COUNT TIMER #4
EB37  2708       ^EB41            BEQ     WAIT9          ;  .    IS TIMER INHIBITED ?
EB39  0AA5                        DEC     TMR4           ;  .
EB3B  2604       ^EB41            BNE     WAIT9          ;  .
EB3D  AD9FC8A6                    JSR     [TMR4+1]       ;  .    EXECUTE THE USER PROGRAM
                          ;
EB41  3588                WAIT9   PULS    DP,PC          ;  RETURN TO CALLER
                          ;
                          ;
                          ;
                          ;
EB43   = ED9F                     ORG     $ED9F
                          ;       ===     =====
                          ;
ED9F  C8A8                SCRPTR  DW      SCOR1          ;  POINTERS TO PLAYER SCORES
EDA1  C8AF                        DW      SCOR2          ;  .
                          ;
EDA3  7FA0                PSCRPTR DW      PSCOR1         ;  SCREEN POSITIONS OF PLAYER SCORES
EDA5  7F10                        DW      PSCOR2         ;  .
                          ;
                          ;
                          ;
                          ;  STAR-FIELD TABLES
                          ;  =================
                          ;
EDA7   = EDE0                     ORG     $EDE0
                          ;       ===     =====
                          ;
EDE0  C840                STAR1   DB      $C8,$40        ;  STAR FIELD #1
EDE2  3F00                        DB      $3F,$00        ;  .
EDE4  2080                        DB      $20,$80        ;  .
EDE6  101F                        DB      $10,$1F        ;  .
                          ;
EDE8  3F3F                STAR2   DB      $3F,$3F        ;  STAR FIELD #2
EDEA  00BF                        DB      $00,$BF        ;  .
EDEC  BFBF                        DB      $BF,$BF        ;  .
EDEE  C020                        DB      $C0,$20        ;  .
                          ;
```

```
EDF0  4808             STAR3   DB     $48,$08          ;  STAR FIELD #3
EDF2  F830                     DB     $F8,$30          ;  .
EDF4  A810                     DB     $A8,$10          ;  .
EDF6  D0A0                     DB     $D0,$A0          ;  .
                       ;
EDF8  BFBF             STAR4   DB     $BF,$BF          ;  STAR FIELD #4
EDFA  003F                     DB     $00,$3F          ;  .
EDFC  3F48                     DB     $3F,$48          ;  .
EDFE  2080                     DB     $20,$80          ;  .
                       ;
EE00  00B0             STAR5   DB     $00,$B0          ;  STAR FIELD #5
EE02  4838                     DB     $48,$38          ;  .
EE04  FB38                     DB     $FB,$38          ;  .
EE06  8028                     DB     $80,$28          ;  .
                       ;
EE08  3048             STAR6   DB     $30,$48          ;  STAR FIELD #6
EE0A  8080                     DB     $80,$80          ;  .
EE0C  45F0                     DB     $45,$F0          ;  .
EE0E  287F                     DB     $28,$7F          ;  .
                       ;
EE10  3FBF             STAR7   DB     $3F,$BF          ;  STAR FIELD #7
EE12  A500                     DB     $A5,$00          ;  .
EE14  D060                     DB     $D0,$60          ;  .
EE16  2028                     DB     $20,$28          ;  .
                       ;
EE18  B840             STAR8   DB     $B8,$40          ;  STAR FIELD #8
EE1A  1580                     DB     $15,$80          ;  .
EE1C  40F8                     DB     $40,$F8          ;  .
EE1E  4018                     DB     $40,$18          ;  .
                       ;
                       ;
                       ;
EE20  = EE2F                   ORG    $EE2F
                       ;       ===    =====
                       ;
EE2F  FA38             M.END   DW     $FA38
EE31  E0D8                     DW     $E0D8
EE33  47414D45204F5645         DB     'GAME OVER',$80
```

```
                ; ************************************
                ; ************************************
                ; ***                              ***
                ; ***     E X E C U T I V E   R O M ***
                ; ***                              ***
                ; ************************************
                ; ************************************
                ;
EE3D  = EFE7                 ORG     $EFE7
                ;            ===     =====
                ;
                ;==================================================JJH
                ;                                            ======JJH
EFE7  = 00D0                 SETDP   $D0         ; CODE ADDED - REV. B CHANGES   ======JJH
                ;            =====   ===                         ======JJH
                ;                                                ======JJH
EFE7  BDF1BA    REVB.0       JSR     INPUT       ; .   ABORT OPENING PAGE ?      ======JJH
EFEA  BDF1AF                 JSR     DPRAM       ; .   .   SET DP = RAM          ======JJH
EFED  = 00C8                 SETDP   $C8         ; .   .   .                     ======JJH
EFED  960F                   LDA     TRIGGR      ; .   .                         ======JJH
EFEF  10260079   ^F06C       LBNE    WARM        ; .   .                         ======JJH
EFF3  7EF01F                 JMP     RRVR01      ; .   .                         ======JJH
                ;==================================================JJH
                ;
                ;
EFF6  = F000                 ORG     $F000
                ;            ===     =====
                ;
                ; POWER-UP HANDLER
                ; ================
                ;
F000  = 0000                 SETDP   $00
                ;            =====   ===
                ;
F000  10CECBEA  PWRUP        LDS     #STACK      ; INITIALIZE STACK-POINTER
F004  BDF18B                 JSR     INTALL      ; INITIALIZE EXECUTIVE
F007  = 00D0                 SETDP   $D0         ; .   SET DP = I/O
                ;
F007  CC7321                 LDD     #$7321      ; WARM OR COLD BOOT ?
F00A  10B3CBFE               CMPD    RESTFLG     ; .   HAS RESTART FLAG BEEN SET ?
F00E  275C       ^F06C       BEQ     WARM        ; .   .
                ;
F010  FDCBFE    COLD         STD     RESTFLG     ; COLD-BOOT PROCEDURE
F013  7CC83B                 INC     LEG         ; .   SET RESTART FLAG - INHIBIT HI-SCORE
F016  8ECBEB                 LDX     #HISCOR     ; .   CLEAR HI-SCORE
F019  BDF84F                 JSR     SCLR        ; .   .
                ;==================================================JJH
                ;COLD0  JSR     DPRAM            ; CODE DELETED - REV. B CHANGES  ======JJH
                ;==================================================JJH
                ;
                ;==================================================JJH
F01C  7EEFE7    COLD0        JMP     REVB.0      ; CODE ADDED - REV. B CHANGES   ======JJH
                ;==================================================JJH
```

```
F01F  = 00C8                   SETDP  $C8           ;  .    ,
F01F  DC25            RKVB01   LDD    FRAME-1       ;  .    WAIT FOR START OF TUNE
F021  10830101                 CMPD   #$101         ;  .    ,
F025  2602      ^F029          BNE    COLD1         ;  .    ,
F027  D756                     STB    TSTAT         ;  .    ,    SET TUNE START FLAG
F029  57              COLD1    ASRB                 ;;  .
F02A  C403                     ANDB   #3            ;;  .
F02C  8EF0FD                   LDX    #MARK1        ;;  .
F02F  E685                     LDB    B,X           ;;  ,
F031  D729                     STB    DASH          ;;  .
F033  C602                     LDB    #2            ;;  ,
F035  D724                     STB    ZSKIP         ;;  .   PREPARE DASH LOOP
F037  CEFD0D                   LDU    #VCTRX        ;  .    UPDATE VECTREX TUNE
F03A  BDF687                   JSR    REPLAY        ;  .    .
F03D  BDF192                   JSR    FRWAIT        ;  ,    WAIT FOR FRAME BOUNDARY
F040  = 00D0                   SETDP  $D0           ;  .    .    SET DP = I/O
F040  BDF289                   JSR    REQOUT        ;  .    UPDATE SOUND GENERATOR WITH TUNE
F043  BDF2A9                   JSR    INTMAX        ;  .    SET INTENSITY TO MAXIMUM
F046  B6C826                   LDA    FRAME         ;  .    BLINK TITLE
F049  CEF10C                   LDU    #LOGO0        ;  .    ,
F04C  8520                     BITA   #$20          ;  .    ,
F04E  2702      ^F052          BEQ    COLD2         ;  .    ,
F050  334C                     LEAU   12,U          ;  .    ,
F052  BDF385          COLD2    JSR    TXTSIZ        ;;  .
F055  8EF0E9                   LDX    #MARK0        ;;  .   DISPLAY TITLE BLOCK
F058  BDF308          COLD3    JSR    POSIT2        ;;  .
F05B  8603                     LDA    #3            ;;  .
F05D  BDF434                   JSR    DSHDF         ;;  .
F060  7AC824                   DEC    ZSKIP         ;;  ,
F063  26F3      ^F058          BNE    COLD3         ;;  .
F065  B6C825                   LDA    FRAME-1       ;;  .
F068  8101                     CMPA   #1            ;;  .   FOR 10 SEC TIMEOUT
F06A  23B0      ^F01C          BLS    COLD0         ;;  .
                      ;
                      ;
F06C  BDF1AF          WARM     JSR    DPRAM         ; WARM-BOOT PROCEDURE
F06F  = 00C8                   SETDP  $C8           ;  ,    SET DP = RAM
F06F  86CC                     LDA    #$CC          ;  .    SET DASH PATTERN
F071  9729                     STA    DASH          ;  .    .
F073  CCF101                   LDD    #CRTKEY       ;  .    SET-UP TO TEST CARTRIDGE KEY
F076  DD39                     STD    WCSINE        ;  .    ,
F078  0F25                     CLR    FRAME-1       ;  .    ,    CLEAR FRAME COUNTER
F07A  0F26                     CLR    FRAME         ;  .    ,    .
F07C  CE0000                   LDU    #GAMCRT       ;  .    ,
F07F  8EF101                   LDX    #CRTKEY       ;  .    .
F082  C60B                     LDB    #$0B          ;  .    ,    CARTRIDGE KEY EXCLUDES DATE
```

```
F084  A6C0              WARM0    LDA     U+                 ;  .    TEST CARTRIDGE KEY
F086  A180                       CMPA    X+                 ;  .    .
F088  270D       ^F097           BEQ     WARM2              ;  .    .
F08A  C101                       CMPB    #1                 ;  .    .
F08C  2704       ^F092           BEQ     WARM1              ;  .    .
F08E  C105                       CMPB    #5                 ;  .    .
F090  2305       ^F097           BLS     WARM2              ;  .    .
F092  CEE000            WARM1    LDU     #STORM             ;  .    CARTRIDGE NOT VALID - USE MINE-STORM
F095  2007       ^F09E           BRA     WARM3              ;  .    .
                        ;
F097  5A                WARM2    DECB                       ;;   .
F098  26FA       ^F084           BNE     WARM0              ;;   .
F09A  D739                       STB     WCSINE             ;;   .   INDICATE CART SELECTED
F09C  D73A                       STB     WCSINE+1           ;;   .
F09E  0C56              WARM3    INC     TSTAT              ;;   .   PICK UP TUNE START
F0A0  DF37                       STU     WSINE              ;;   .
F0A2  EEC4                       LDU     ,U                 ;;   .
F0A4  BDF1AF            WARM4    JSR     DPRAM              ;  .    SET DP = RAM
F0A7  = 00C8                     SETDP   $C8                ;  .    .
F0A7  CCF848            RKVR11   LDD     #$F848             ;;   .   SET LEAGL SIZ
F0AA  DD2A                       STD     SIZRAS             ;;   .
F0AC  BDF687                     JSR     REPLAY             ;  .    UPDATE GAME TUNE
F0AF  BDF192                     JSR     FRWAIT             ;  .    WAIT FOR FRAME BOUNDARY
F0B2  = 00D0                     SETDP   $D0                ;  .    .    SET DP = I/O
F0B2  BDF289                     JSR     REQOUT             ;  .    UPDATE SOUND GENERATOR WITH TUNE
F0B5  BDF2A9                     JSR     INTMAX             ;  .    SET INTENSITY TO MAXIMUM
F0B8  CCC0C0                     LDD     #$C0C0             ;;   .
F0BB  FEC839                     LDU     WCSINE             ;;   .
F0BE  BDF37A                     JSR     MSSPOS             ;;   .   DISPLAY LEGAL MESS
F0C1  B6C83B                     LDA     LEG                ;  .    SKIP HI-SCORE DISPLAY ON COLD-START
F0C4  260C       ^F0D2           BNE     WARM5              ;  .    .
F0C6  4A                         DECA                       ;  .    .
F0C7  CECBEB                     LDU     #HISCOR            ;  .    DISPLAY PLAYER HI-SCORE
F0CA  A746                       STA     6,U                ;  .    .    SET HI-SCORE STRING TERMINATOR
F0CC  CC68D0                     LDD     #$68D0             ;  .    .    SET RASTER SIZE
F0CF  BDF37A                     JSR     MSSPOS             ;  .    .
F0D2  FEC837            WARM5    LDU     WSINE              ;;   .   GAME TEXT
F0D5  3342                       LEAU    2,U                ;;   .
F0D7  BDF385                     JSR     TXTSIZ             ;;   .
F0DA  B6C856                     LDA     TSTAT              ;  .    TUNE OVER ?
F0DD  26C5       ^F0A4           BNE     WARM4              ;  .    .
F0DF  BEC825                     LDX     FRAME-1            ;  .    .    DISPLAY FOR AT LEAST 25 SECONDS
F0E2  8C007D                     CMPX    #125               ;  .    .    .
F0E5  23BD       ^F0A4           BLS     WARM4              ;  .    .    .
F0E7  6E41              WARM6    JMP     1,U                ;  .    EXECUTE SELECTED GAME
```

```
                         ;  MARQUEE DATA
                         ;  ============
                         ;
F0E9  40D6               MARK0   DW     $40D6            ; POSITION OF BLOCK
F0EB  00568100                   DW     $0056,$8100      ; DASHED LINES
F0EF  00A97E00                   DW     $00A9,$7E00      ; .
                         ;
F0F3  39DC                       DW     $39DC            ; SMALLER DASHES, OPPOSITE DIRECTION
F0F5  8E00004A                   DW     $8E00,$004A      ; .
F0F9  720000B6                   DW     $7200,$00B6      ; .
                         ;
F0FD  E0380E03           MARK1   DB     $E0,$38,$0E,$03  ; FOR MOVIE PATTERN
                         ;
                         ;       CARTRIDGE KEY
                         ;       -------------
                         ;
F101  6720474345203139   CRTKEY  DB     $67,' GCE 1982',$80
                         ;
                         ;       VECTREX LOGO
                         ;       ------------
                         ;
F10C  F16027CF           LOGO0   DW     $F160,$27CF
F110  5645435452455880           DB     'VECTREX',$80
                         ;
F118  F36026CF           LOGO1   DW     $F360,$26CF
F11C  5645435452455880           DB     'VECTREX',$80
                         ;
F124  FC60               LOGO2   DW     $FC60
F126  DFE9                       DW     $DFE9
F128  47434580                   DB     'GCE',$80
F12C  FC38                       DW     $FC38
F12E  CCD1                       DW     $CCD1
F130  454E544552544149           DB     'ENTERTAINING',$80
F13D  FC38                       DW     $FC38
F13F  BCDC                       DW     $BCDC
F141  4E45572049444541           DB     'NEW IDEAS',$80
F14B  00                         DB     0
```

```
                       ;  INITIALIZE PIA
                       ;  ==============
                       ;
                       ;        COMMENT(S)
                       ;        ----------
                       ;           PERFORMS 'DEFLOK'
                       ;           ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND ON RETURN TO USER
                       ;
                       ;        ENTRY VALUES
                       ;        ------------
                       ;           NONE REQUIRED
                       ;
                       ;        RETURN VALUES
                       ;        -------------
                       ;           D  = $0301
                       ;           X  = $F9F4 (#KEPALV + 4)
                       ;           DP = $D0
                       ;
F14C  8D5C      ^F1AA  INTPIA  BSR     DPIO            ;  SET DP = I/O
F14E  = 00D0                   SETDP   $D0             ;  .
                       ;
F14E  CC9FFF                   LDD     #$9FFF          ;  SET REGISTER DIRECTIONS
F151  DD02                     STD     DCNTRL          ;  .
                       ;
F153  CC0100                   LDD     #$0100          ;  INITIALIZE CONTROL REGISTERS
F156  DD00                     STD     CNTRL           ;  .
F158  CC987F                   LDD     #$987F          ;  .    SET-UP TIMERS
F15B  970B                     STA     ACNTRL          ;  .    .
F15D  D704                     STB     T1LOLC          ;  .    .    SET VECTOR LENGTH
F15F  BDF354                   JSR     ZERGND          ;  ZERO INTEGRATORS AND SET ACTIVE GROUND
F162  203E      ^F1A2          BRA     FRWT1           ;  SET FRAME BOUNDARY TIMER
```

```
                        ;  INITIALIZE MISC. PARAMETERS
                        ;  ==========================
                        ;
                        ;        ENTRY VALUES
                        ;        ------------
                        ;           NONE REQUIRED
                        ;
                        ;        RETURN VALUES
                        ;        -------------
                        ;           A  = $05
                        ;           B  = $07
                        ;           X  = $C800 (#REG0)
                        ;           DP = $C8
                        ;
F164  8D49     ^F1AF    INTMSC   BSR    DPRAM           ;  SET DP = RAM
F166  = 00C8                     SETDP  $C8             ;  .
                        ;
F166  C67A                       LDB    #OPTION-REG0    ;  CLEAR EXECUTIVE MEMORY
F168  8EC800                     LDX    #REG0           ;  .
F16B  BDF53F                     JSR    BCLR            ;  .
                        ;
F16E  CCC87D                     LDD    #RANCID         ;  SET INITIAL RANDOM NUMBER GENERATOR SEED
F171  DD7B                       STD    SEED            ;  .
F173  0C7D              IMSC0    INC    RANCID          ;  .
F175  27FC     ^F173             BEQ    IMSC0           ;  .
                        ;
F177  8605                       LDA    #$05            ;  SET DOT DWELL TIME
F179  9728                       STA    DWELL           ;  .
                        ;
F17B  CC3075                     LDD    #MSEC20         ;  SET FRAME RATE TO 20 MSEC
F17E  DD3D                       STD    FRMTIM          ;  .
                        ;
F180  CC0103                     LDD    #$0103          ;  ENABLE POT READS
F183  DD1F                       STD    EPOT0           ;  .     CONTROLLER #1
F185  CC0507                     LDD    #$0507          ;  .     CONTROLLER #2
F188  DD21                       STD    EPOT2           ;  .     .
                        ;
F18A  39                         RTS                    ;  RETURN TO CALLER
```

```
                         ;  COMPLETE INITIALIZATION
                         ;  ======================
                         ;
                         ;        COMMENT(S)
                         ;        ----------
                         ;
                         ;          PERFORMS 'DEFLOK'
                         ;          ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND ON RETURN TO USER
                         ;
                         ;        ENTRY VALUES
                         ;        ------------
                         ;
                         ;          NONE REQUIRED
                         ;
                         ;        RETURN VALUES
                         ;        -------------
                         ;
                         ;          A  = $3F
                         ;          B  = $FF
                         ;          X  = $C83F (#REG0)
                         ;          DP = $D0
                         ;
F18B  = 0000                      SETDP   $00
                         ;        =====   ===
                         ;
F18B  8DD7      ^F164    INTALL   BSR     INTMSC          ;  INITIALIZE MISC. VALUES
F18D  8DBD      ^F14C             BSR     INTPIA          ;  INITIALIZE PIA
F18F  = 00D0                      SETDP   $D0             ;  .    SET DP = I/O
F18F  7EF272                      JMP     INTPSC          ;  INITIALIZE SOUND GENERATOR
```

```
                        ;  WAIT FOR FRAME BOUNDARY
                        ;  ======================
                        ;
                        ;       COMMENT(S)
                        ;       ----------
                        ;          PERFORMS 'DEFLOK'
                        ;          ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND ON RETURN TO USER
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          NONE REQUIRED
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          D  = $0301
                        ;          X  = $F9F4 (REPALV + 4)
                        ;          DP = $D0
                        ;
F192  = 0000                    SETDP   $00
                        ;       =====   ===
                        ;
F192  BEC825            FRWAIT  LDX     FRAME-1         ;  INCREMENT FRAME COUNTER
F195  3001                      LEAX    1,X             ;  .
F197  BFC825                    STX     FRAME-1         ;  .
                        ;
F19A  8D0E       ^F1AA          BSR     DPIO            ;  SET DP = I/O
F19C  = 00D0                    SETDP   $D0             ;  .
                        ;
F19C  8620                      LDA     #$20            ;  WAIT FOR FRAME TIMER
F19E  950D              FRWT0   BITA    IFLAG           ;  .
F1A0  27FC       ^F19E          BEQ     FRWT0           ;  .
                        ;
F1A2  FCC83D            FRWT1   LDD     FRMTIM          ;  RESET FRAME TIMER
F1A5  DD08                      STD     T2LOLC          ;  .
                        ;
F1A7  7EF2E6                    JMP     DEFLOK          ;  PREVENT SCAN COLLAPSE
```

```
                        ;  SET DIRECT REGISTER FOR I/O ACCESS
                        ;  ================================
                        ;
                        ;        ENTRY VALUES
                        ;        ------------
                        ;
                        ;           NONE REQUIRED
                        ;
                        ;        RETURN VALUES
                        ;        -------------
                        ;
                        ;           A  = $D0
                        ;           DP = $D0
                        ;
F1AA  86D0              DPIO     LDA     #$D0
F1AC  1F8B                       TFR     A,DP
F1AE  39                         RTS
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;  SET DIRECT REGISTER FOR RAM ACCESS
                        ;  ================================
                        ;
                        ;        ENTRY VALUES
                        ;        ------------
                        ;
                        ;           NONE REQUIRED
                        ;
                        ;        RETURN VALUES
                        ;        -------------
                        ;
                        ;           A  = $C8
                        ;           DP = $C8
                        ;
F1AF  86C8              DPRAM    LDA     #$C8
F1B1  1F8B                       TFR     A,DP
F1B3  39                         RTS
```

```
                        ;  READ CONTROLLER BUTTONS WITH DE-BOUNCE MASK
                        ;  ===========================================
                        ;
                        ;        ENTRY VALUES
                        ;        ------------
                        ;
                        ;           A  = DIRECT RESPONSE SWITCH MASK
                        ;           DP = $D0
                        ;
                        ;        RETURN VALUES
                        ;        -------------
                        ;
                        ;           A  = CONTENTS OF 'EDGE'
                        ;           B  = $00
                        ;           X  = $C81A (#KEY7 + 1)
                        ;
F1B4  = 00D0                     SETDP   $D0
                        ;        =====   ===
                        ;
F1B4  B4C80F            DBNCE    ANDA    TRIGGR          ;  MASK-OFF DIRECT RESPONSE BITS
F1B7  B7C80F                     STA     TRIGGR          ;
                        ;
                        ;
                        ;  READ CONTROLLER BUTTONS
                        ;  =======================
                        ;
                        ;        ENTRY VALUES
                        ;        ------------
                        ;           DP = $D0
                        ;
                        ;        RETURN VALUES
                        ;        -------------
                        ;
                        ;           A  = CONTENTS OF 'EDGE'
                        ;           B  = $00
                        ;           X  = $C81A (#KEY7 + 1)
                        ;
F1BA  = 00D0                     SETDP   $D0
                        ;        =====   ===
                        ;
F1BA  8EC812            INPUT    LDX     #KEY0           ;  SAVE OLD SWITCH SETTING
F1BD  A61D                       LDA     -3,X            ;  .    FETCH FROM 'TRIGGR'
F1BF  A71E                       STA     -2,X            ;  .    IN 'TRIGGR+1'
                        ;
F1C1  860E                       LDA     #$0E            ;  READ FROM SOUND GENERATOR
F1C3  9701                       STA     DAC             ;  .    SET REGISTER ADDRESS
                        ;
F1C5  CC1901                     LDD     #$1901          ;  .    ZERO INTEGRATORS
F1C8  9700                       STA     CNTRL           ;  .    .
F1CA  12                         NOP                     ;  .    .    (( TIMING ))
F1CB  D700                       STB     CNTRL           ;  .    .
                        ;
F1CD  0F03                       CLR     DDAC            ;  .    SET PIA DIRECTION FOR READ
F1CF  CC0901                     LDD     #$0901          ;  .    .
F1D2  9700                       STA     CNTRL           ;  .    .
F1D4  12                         NOP                     ;  .    .    (( TIMING ))
```

```
                        ;
F1D5  9601                     LDA    DAC          ; .     READ FROM PSG
F1D7  43                       COMA                ; .     .   COMPLEMENT DATA
F1D8  A71D                     STA    -3,X         ; .     .   SAVE SWITCH SETTINGS IN 'TRIGGR'
                        ;
F1DA  D700                     STB    CNTRL        ; .     SET PIA DIRECTION FOR OUTPUT
F1DC  C6FF                     LDB    #$FF         ; .     .
F1DE  D703                     STB    DDAC         ; .     .
                        ;
F1E0  43                       COMA                ; DE-EDGE SWITCHES
F1E1  AA1E                     ORA    -2,X         ; .     FETCH FROM 'TRIGGR+1'
F1E3  43                       COMA                ; .
F1E4  A71F                     STA    -1,X         ; .     TO 'EDGE'
F1E6  3402                     PSHS   A            ; .     SAVE FOR EXIT
                        ;
F1E8  C601                     LDB    #1           ; FORM 'KEYX' TERMS
F1EA  1F98              INPUTO TFR    B,A          ; .
F1EC  A4E4                     ANDA   ,S           ; .
F1EE  A780                     STA    X+           ; .
F1F0  58                       ASLB                ; .
F1F1  26F7      ^F1EA          BNE    INPUTO       ; .
                        ;
F1F3  3582                     PULS   A,PC         ; RETURN TO CALLER
                        ;
                        ;
                        ; READ JOYSTICKS
                        ; ==============
                        ;
                        ;      COMMENT(S)
                        ;      ----------
                        ;        POTS TO BE READ MUST BE ENABLED WITH THE FOLLOWING CODES:
                        ;             'EPOT0' = $01 (POT0)
                        ;             'EPOT1' = $03 (POT1)
                        ;             'EPOT2' = $05 (POT2)
                        ;             'EPOT3' = $07 (POT3)
                        ;
                        ;        'INITMSG' ENABLES ALL POTS TO BE READ - SELECTED POTS MAYBE
                        ;             DISABLED BY WRITING THE RESPECTIVE 'EPOTX' WITH $00
                        ;
                        ;      ENTRY VALUES
                        ;      ------------
                        ;        DP = $D0
                        ;
                        ;      RETURN VALUES
                        ;      -------------
                        ;        A  = $01
                        ;        B  = CONTENTS OF 'POT3'
                        ;        X  = $C823 (#LIST)
                        ;
F1F5  = 00D0                   SETDP  $D0
                        ;      =====  ===
                        ;
F1F5  7AC823            JOYSTK DEC    LIST         ; SET-UP FOR SUCCESSIVE APPROXIMATION READ
```

```
                         ;
                         ;
                         ;  READ DIRECTION OF JOYSTICKS
                         ;  ===========================
                         ;
                         ;       COMMENT(S)
                         ;       ----------
                         ;          POTS TO BE READ MUST BE ENABLED WITH THE FOLLOWING CODES:
                         ;               'EPOT0' = $01 (POT0)
                         ;               'EPOT1' = $03 (POT1)
                         ;               'EPOT2' = $05 (POT2)
                         ;               'EPOT3' = $07 (POT3)
                         ;
                         ;          'INITMSC' ENABLES ALL POTS TO BE READ - SELECTED POTS MAYBE
                         ;               DISABLED BY WRITING THE RESPECTIVE 'EPOTX' WITH $00
                         ;
                         ;       ENTRY VALUES
                         ;       ------------
                         ;          DP = $D0
                         ;
                         ;       RETURN VALUES
                         ;       -------------
                         ;          A  = $01
                         ;          B  = CONTENTS OF 'POT3'
                         ;          X  = $C823 (#LIST)
                         ;
F1F8  = 00D0                     SETDP    $D0
                         ;       =====    ===
                         ;
F1F8  8EC81F             JOYBIT  LDX      #EPOT0           ;  SET-UP TO READ JOYSTICK POTS
                         ;
F1FB  A680               JBIT0   LDA      X+               ;  FETCH POT ENABLE
F1FD  260C      ^F20B            BNE      JBIT2            ;  .    IS POT DISABLED ?
                         ;
F1FF  8CC823             JBIT1   CMPX     #EPOT3+1         ;  ALL FOUR POTS READ ?
F202  26F7      ^F1FB            BNE      JBIT0            ;  .
                         ;
F204  6F84                       CLR      ,X               ;  RETURN TO CALLER
F206  8601                       LDA      #1               ;  .    RESET MUX ADDRESS
F208  9700                       STA      CNTRL            ;  .    .
F20A  39                         RTS                       ;  .
                         ;
                         ;
F20B  9700               JBIT2   STA      CNTRL            ;  SET MULTIPLEXER ADDRESS
F20D  0F01                       CLR      DAC              ;  .    SET DAC TO MID-RANGE
F20F  0A00                       DEC      CNTRL            ;  .
                         ;
F211  C660                       LDB      #$60             ;  DELAY BEFORE READING POT
F213  5C                 JBIT3   INCB                      ;  .
F214  2AFD      ^F213            BPL      JBIT3            ;  .
                         ;=====================================================================JJH
                         ;       LDA      LIST             ;  CODE DELETED - REV. B1 CHANGES   =====JJH
                         ;       BMI      JBIT8            ;  .                                =====JJH
```

```
                        ;       LDA     #$20            ; .                                     =====JJH
                        ;       INC     CNTRL           ; .                                     =====JJH
                        ;======================================================================JJH
                        ;
                        ;======================================================================JJH
F216 0C00                       INC     CNTRL           ; CODE ADDED - REV. B1 CHANGES          =====JJH
F218 B6C823                     LDA     LIST            ; .                                     =====JJH
F21B 2B23       ^F240           BMI     JBIT8           ; .                                     =====JJH
F21D 8620                       LDA     #$20            ; .                                     =====JJH
                        ;======================================================================JJH
F21F 9500                       BITA    CNTRL           ;; FIRST TEST ) OR ( 0
F221 270A       ^F22D           BEQ     JBIT4           ;; .
F223 C640                       LDB     #$40            ;; PLUS: CHECK MID-RANGE
F225 D701                       STB     DAC             ;; .
F227 9500                       BITA    CNTRL           ;; .
F229 260B       ^F236           BNE     JBIT6           ;; VALID IF ABOVE MIDDLE POS VALUE
F22B 2008       ^F235           BRA     JBIT5           ;; .
                        ;
F22D C6C0               JBIT4   LDB     #$C0            ;; MINUS
F22F D701                       STB     DAC             ;; .
F231 9500                       BITA    CNTRL           ;; .
F233 2701       ^F236           BEQ     JBIT6           ;; .
F235 5F                 JBIT5   CLRB                    ;; .
F236 E71B               JBIT6   STB     -5,X            ;; PUTS IN POTO-3
F238 20C5       ^F1FF           BRA     JBIT1           ;; END OF ROUTINE - - - - --
                        ;
                        ;
                        ;       SUCCESSIVE APPROXIMATION ROUTINE
                        ;       --------------------------------
                        ;
F23A 1F98               JBIT7   TFR     B,A             ; TRY NEW APPROXIMATION VALUE
F23C 9A01                       ORA     DAC             ; .
F23E 9701                       STA     DAC             ; .
                        ;
F240 8620               JBIT8   LDA     #$20            ; IS THIS BIT HIGHER OR LOWER ?
F242 9500                       BITA    CNTRL           ; .
F244 2606       ^F24C           BNE     JBIT9           ; .
                        ;
F246 1F98                       TFR     B,A             ; THIS BIT IS HIGHER, COMPLEMENT
F248 9801                       EORA    DAC             ; .
F24A 9701                       STA     DAC             ; .
                        ;
F24C 54                 JBIT9   LSRB                    ; SET-UP FOR NEXT BIT OF APPROXIMATION
F24D F1C81A                     CMPB    POTRES          ; .
F250 26E8       ^F23A           BNE     JBIT7           ; .
                        ;
F252 D601                       LDB     DAC             ; READ POT VALUE FROM 'DAC'
F254 20E0       ^F236           BRA     JBIT6           ; .
```

```
                        ;  WRITE TO PSG AND MIRROR (REGX)
                        ;  ==============================
                        ;
                        ;        ENTRY VALUES
                        ;        ------------
                        ;           A  = PSG ADDRESS
                        ;           B  = PSG DATA
                        ;           DP = $D0
                        ;
                        ;        RETURN VALUES
                        ;        -------------
                        ;           B  = $01
                        ;           X  = $C800 (#REG0)
                        ;
F256  = 00D0                     SETDP   $D0
                        ;        =====   ===
                        ;
F256  8EC800            WRREG    LDX     #REG0           ;  SET-UP FOR 'REGX' MIRROR
                        ;
                        ;
                        ;  WRITE TO PSG AND INDICATED MIRROR (POINTED TO BY 'X')
                        ;  =====================================================
                        ;
                        ;        ENTRY VALUES
                        ;        ------------
                        ;           A  = PSG ADDRESS
                        ;           B  = PSG DATA
                        ;           X  = POINTER TO MIRROR AREA
                        ;           DP = $D0
                        ;
                        ;        RETURN VALUES
                        ;        -------------
                        ;           B  = $01
                        ;
F259  = 00D0                     SETDP   $D0
                        ;        =====   ===
                        ;
F259  E786              WRPSG    STB     A,X             ;  SET PSG ADDRESS
F25B  9701                       STA     DAC             ;  .
F25D  8619                       LDA     #$19            ;;  CODE FOR ADDR LATCH
F25F  9700                       STA     CNTRL           ;;  .
F261  8601                       LDA     #1              ;;  .
F263  9700                       STA     CNTRL           ;;  LATCH ADDRESS TO PSG
F265  9601                       LDA     DAC             ;;  GET A REG BACK
F267  D701                       STB     DAC             ;;  .
F269  C611                       LDB     #$11            ;;  .
F26B  D700                       STB     CNTRL           ;;  LATCH DATA
F26D  C601                       LDB     #1              ;;  .
F26F  D700                       STB     CNTRL           ;;  .
F271  39                         RTS                     ;  RETURN TO CALLER
```

```
                             ;  INITIALIZE SOUND GENERATOR
                             ;  ==========================
                             ;
                             ;         ENTRY VALUES
                             ;         ------------
                             ;            DP = $D0
                             ;
                             ;         RETURN VALUES
                             ;         -------------
                             ;            A  = $3F
                             ;            B  = $FF
                             ;            X  = $C83F (#REQ0)
                             ;
F272  = 00D0                           SETDP   $D0
                             ;         =====   ===
                             ;
F272  CC0E00                 INTPSG    LDD     #$0E00          ;  CLEAR PSG REGISTERS ($00 - $0E)
F275  8DDF         ^F256     INPSG0    BSR     WRREG           ;  .
F277  4A                               DECA                    ;  .
F278  2AFB         ^F275               BPL     INPSG0          ;  .
F27A  7EF533                           JMP     INTREQ          ;  CLEAR 'REQX' AREA
                             ;
                             ;
                             ;
                             ;
                             ;
                             ;  SEND SOUND STRING TO PSG AND MIRROR
                             ;  ===================================
                             ;
                             ;         ENTRY VALUES
                             ;         ------------
                             ;            U  = POINTER TO SOUND STRING
                             ;            DP = $D0
                             ;
                             ;         RETURN VALUES
                             ;         -------------
                             ;            D  = SOUND STRING TERMINATOR
                             ;            X  = $C800 (REG0)
                             ;            U  = POINTER TO END OF SOUND STRING
                             ;
F27D  = 00D0                           SETDP   $D0
                             ;         =====   ===
                             ;
F27D  8EC800                 PSGLST    LDX     #REG0           ;  SET-UP FOR 'REGX' MIRROR
F280  2002         ^F284               BRA     PSGMIR          ;  .
```

```
                         ;  SEND SOUND STRING TO PSG AND INDICATED MIRROR
                         ;  ============================================
                         ;
                         ;       ENTRY VALUES
                         ;       ------------
                         ;          X  = POINTER TO PSG MIRROR
                         ;          U  = POINTER TO SOUND STRING
                         ;          DP = $D0
                         ;
                         ;       RETURN VALUES
                         ;       -------------
                         ;          D  = SOUND STRING TERMINATOR
                         ;          U  = POINTS TO END OF SOUND STRING
                         ;
F282  = 00D0                     SETDP   $D0
                         ;       =====   ===
                         ;
F282  8DD5        ^F259  PMIRO   BSR     WRPSG           ;  WRITE PSG ADDRESS & DATA
                         ;
F284  ECC1               PSGMIR  LDD     U++             ;  FETCH BYTE FROM SOUND STRING
F286  2AFA        ^F282          BPL     PMIRO           ;  .    END OF STRING ?
F288  39                         RTS                     ;  .    .    RETURN TO CALLER
                         ;
                         ;
                         ;  SEND 'REQX' TO PSG AND MIRROR
                         ;  =============================
                         ;
                         ;       ENTRY VALUES
                         ;       ------------
                         ;          DP = $D0
                         ;
                         ;       RETURN VALUES
                         ;       -------------
                         ;          A  = $FF
                         ;          B  = CONTENTS OF 'REQ0'
                         ;          X  = $C80D (#REGD)
                         ;          U  = $C84C (#REQD)
                         ;
F289  = 00D0                     SETDP   $D0
                         ;       =====   ===
                         ;
F289  8EC800             REQOUT  LDX     #REG0           ;  SET-UP TO SEND 'REQX'
F28C  CEC83F                     LDU     #REQ0           ;  .
F28F  860D                       LDA     #$0D            ;  .
                         ;
F291  E6C0               REQOT0  LDB     U+              ;  FETCH BYTE FROM 'REQX'
F293  E186                       CMPB    A,X             ;  .    COMPARE AGAINST 'REGX'
F295  2702        ^F299          BEQ     REQOT1          ;  .    .    SKIP PSG WRITE IF SAME
F297  8DC0        ^F259          BSR     WRPSG           ;  .    WRITE BYTE TO PSG AND MIRROR
F299  4A                 REQOT1  DECA                    ;  .    END OF 'REQX' ?
F29A  2AF5        ^F291          BPL     REQOT0          ;  .    .    DO NEXT ENTRY IF NOT
F29C  39                         RTS                     ;  .    .    RETURN TO CALLER
```

```
                          ;  SET BEAM INTENSITY TO 1/4 LEVEL
                          ;  ===============================
                          ;
                          ;       COMMENT(S)
                          ;       ----------
                          ;          SAVES NEW BEAM INTENSITY ($1F) IN 'TENSITY'
                          ;
                          ;       ENTRY VALUES
                          ;       ------------
                          ;          DP = $D0
                          ;
                          ;       RETURN VALUES
                          ;       -------------
                          ;          A  = $05
                          ;          B  = $01
                          ;
F29D  = 00D0                      SETDP   $D0
                          ;       =====   ===
                          ;
F29D  861F                INT1Q   LDA     #$1F            ;  SET INTENSITY
F29F  200A      ^F2AB             BRA     INTENS          ;  .
                          ;
                          ;
                          ;
                          ;
                          ;
                          ;  SET BEAM INTENSITY TO 1/2 LEVEL
                          ;  ===============================
                          ;
                          ;       COMMENT(S)
                          ;       ----------
                          ;          SAVES NEW BEAM INTENSITY ($3F) IN 'TENSITY'
                          ;
                          ;       ENTRY VALUES
                          ;       ------------
                          ;          DP = $D0
                          ;
                          ;       RETURN VALUES
                          ;       -------------
                          ;          A  = $05
                          ;          B  = $01
                          ;
F2A1  = 00D0                      SETDP   $D0
                          ;       =====   ===
                          ;
F2A1  863F                INT2Q   LDA     #$3F            ;  SET INTENSITY
F2A3  2006      ^F2AB             BRA     INTENS          ;  .
```

```
                          ;  SET BEAM INTENSITY TO 3/4 LEVEL
                          ;  ==============================
                          ;
                          ;        COMMENT(S)
                          ;        ----------
                          ;           SAVES NEW BEAM INTENSITY ($5F) IN 'TENSITY'
                          ;
                          ;        ENTRY VALUES
                          ;        ------------
                          ;           DP = $D0
                          ;
                          ;        RETURN VALUES
                          ;        -------------
                          ;           A  = $05
                          ;           B  = $01
                          ;
F2A5  = 00D0                       SETDP   $D0
                          ;        =====   ===
                          ;
F2A5  865F                INT3Q    LDA     #$5F            ;  SET INTENSITY
F2A7  2002      ^F2AB              BRA     INTENS          ;  .
                          ;
                          ;
                          ;
                          ;
                          ;
                          ;  SET BEAM INTENSITY TO MAXIMUM LEVEL
                          ;  =================================
                          ;
                          ;        COMMENT(S)
                          ;        ----------
                          ;           SAVES NEW BEAM INTENSITY ($7F) IN 'TENSITY'
                          ;
                          ;        ENTRY VALUES
                          ;        ------------
                          ;           DP = $D0
                          ;
                          ;        RETURN VALUES
                          ;        -------------
                          ;           A  = $05
                          ;           B  = $01
                          ;
F2A9  = 00D0                       SETDP   $D0
                          ;        =====   ===
                          ;
F2A9  867F                INTMAX   LDA     #$7F            ;  SET INTENSITY
```

```
                        ; SET BEAM INTENSITY TO INDICATED LEVEL
                        ; =====================================
                        ;
                        ;       COMMENT(S)
                        ;       ----------
                        ;
                        ;          SAVES NEW BEAM INTENSITY IN 'TENSITY'
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;
                        ;          A  = INTENSITY LEVEL ($00 - $7F)
                        ;          DP = $D0
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;
                        ;          A  = $05
                        ;          B  = $01
                        ;
F2AB  = 00D0                    SETDP   $D0
                        ;       =====   ===
                        ;
F2AB  9701              INTENS  STA     DAC             ; SET DAC FOR INTENSITY
F2AD  B7C827                    STA     TENSTY          ; .
                        ;
F2B0  CC0504                    LDD     #$0504          ; SELECT INTENSITY SAMPLE / HOLD
F2B3  9700                      STA     CNTRL           ; .     STROBE SAMPLE / HOLD
F2B5  D700                      STB     CNTRL           ; .     .
F2B7  D700                      STB     CNTRL           ; .     .     << TIMING >>
F2B9  C601                      LDB     #$01            ; .     DE-SELECT SAMPLE / HOLD
F2BB  D700                      STB     CNTRL           ; .     .
                        ;
F2BD  39                        RTS                     ; RETURN TO CALLER
                        ;
                        ;
                        ; DRAW ONE DOT FROM 'DIFFY' STYLE LIST (SETS DWELL TIME)
                        ; ======================================================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          B  = DOT DWELL TIME
                        ;          X  = POINTER TO 'DIFFY' STYLE LIST
                        ;          DP = $D0
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          A  = $FF
                        ;          B  = $00
                        ;          X  = ENTRY VALUE + 2
                        ;
F2BE  = 00D0                    SETDP   $D0
                        ;       =====   ===
                        ;
F2BE  F7C028            DOTTIM  STB     DWELL           ; SET 'DWELL' TIME
```

```
                        ;  DRAW ONE DOT FROM 'DIFFY' STYLE LIST
                        ;  ===================================
                        ;
                        ;        ENTRY VALUES
                        ;        ------------
                        ;           X  = POINTER TO 'DIFFY' STYLE LIST
                        ;           DP = $D0
                        ;
                        ;        RETURN VALUES
                        ;        -------------
                        ;           A  = $FF
                        ;           B  = $00
                        ;           X  = ENTRY VALUE + 2
                        ;
F2C1  = 00D0                     SETDP   $D0
                        ;        =====   ===
                        ;
F2C1  EC81              DOTX     LDD     X++               ;  FETCH DOT VECTOR
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;  DRAW ONE DOT FROM THE CONTENTS OF 'A' & 'B'
                        ;  ==========================================
                        ;
                        ;        ENTRY VALUES
                        ;        ------------
                        ;           A  = 'Y' VECTOR VALUE
                        ;           B  = 'X' VECTOR VALUE
                        ;           DP = $D0
                        ;
                        ;        RETURN VALUES
                        ;        -------------
                        ;           A  = $FF
                        ;           B  = $00
                        ;
F2C3  = 00D0                     SETDP   $D0
                        ;        =====   ===
                        ;
F2C3  8D4D       ^F312  DOTAB    BSR     POSITN            ;  POSITION FOR ONE DOT
```

```
                           ;  TURN-ON BEAM FOR DOT
                           ;  ===================
                           ;
                           ;       ENTRY VALUES
                           ;       ------------
                           ;          DP = $D0
                           ;
                           ;       RETURN VALUES
                           ;       -------------
                           ;          A  = $FF
                           ;          B  = $00
                           ;
F2C5  = 00D0                       SETDP   $D0
                           ;       =====   ===
                           ;
F2C5  86FF                 DOT     LDA     #$FF            ;  TURN-ON CRT BEAM
F2C7  970A                         STA     SHIFT           ;  .
                           ;
F2C9  F6C828                       LDB     DWELL           ;  DWELL FOR DOT
F2CC  5A                   DOT0    DECB                    ;  .
F2CD  26FD        ^F2CC            BNE     DOT0            ;  .
                           ;
F2CF  0F0A                         CLR     SHIFT           ;  TURN-OFF CRT BEAM
F2D1  39                           RTS                     ;  RETURN TO CALLER
                           ;
                           ;
                           ;  DRAW DOTS ACCORDING TO 'DIFFY' FORMAT
                           ;  =====================================
                           ;
                           ;       COMMENT(S)
                           ;       ----------
                           ;          ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND ON RETURN TO USER
                           ;
                           ;       ENTRY VALUES
                           ;       ------------
                           ;          X  = POINTER TO 'DIFFY' STYLE LIST
                           ;          DP = $D0
                           ;
                           ;          LIST     = NUMBER OF VECTORS - 1
                           ;
                           ;       RETURN VALUES
                           ;       -------------
                           ;          D  = $0301
                           ;          X  = END OF 'DIFFY' LIST + 2
                           ;
F2D2  = 00D0                       SETDP   $D0
                           ;       =====   ===
                           ;
F2D2  7AC823               DFDOT0  DEC     LIST             ;  REPEAT DIFFY STYLE LIST
                           ;
F2D5  8DEA        ^F2C1    DIFDOT  BSR     DOTX             ;  DRAW DOT ACCORDING TO DIFFY STYLE
F2D7  B6C823                       LDA     LIST             ;  .    END-OF-DIFFY LIST ?
F2DA  26F6        ^F2D2            BNE     DFDOT0           ;  .    .
```

```
F2DC  2076       ^F354             BRA     ZERGND          ;  ZERO INTEGRATORS AND SET ACTIVE GROUND
                          ;
                          ;
                          ;  DRAW DOTS ACCORDING TO PACKET FORMAT
                          ;  ====================================
                          ;
                          ;        COMMENT(S)
                          ;        ----------
                          ;           ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND ON RETURN TO USER
                          ;
                          ;        ENTRY VALUES
                          ;        ------------
                          ;           X  = POINTER TO 'PACKET' STYLE LIST
                          ;           DP = $D0
                          ;
                          ;        RETURN VALUES
                          ;        -------------
                          ;           D  = $0301
                          ;           X  = POINTS TO END OF PACKET LIST + 1
                          ;
F2DE  = 00D0                       SETDP   $D0
                          ;        =====   ===
                          ;
F2DE  A680                DOTPCK   LDA     ,X+             ;  FETCH CONTROL BYTE
F2E0  2E72       ^F354             BGT     ZERGND          ;  ZERO INTEGRATORS AND SET ACTIVE GROUND
F2E2  8DDD       ^F2C1             BSR     DOTX            ;  DRAW DOT FOR THIS ENTRY
F2E4  20F8       ^F2DE             BRA     DOTPCK          ;  .    SET-UP FOR NEXT BYTE
                          ;
                          ;
                          ;  OVER-COME SCAN COLLAPSE CIRCUITRY
                          ;  =================================
                          ;
                          ;        COMMENT(S)
                          ;        ----------
                          ;           ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND ON RETURN TO USER
                          ;
                          ;        ENTRY VALUES
                          ;        ------------
                          ;           DP = $D0
                          ;
                          ;        RETURN VALUES
                          ;        -------------
                          ;           D  = $0301
                          ;           X  = $F9F4 (#KEPALV + 4)
                          ;
F2E6  = 00D0                       SETDP   $D0
                          ;        =====   ===
                          ;
F2E6  8EF9F0              DEFLOK   LDX     #KEPALV         ;  DRAW 2 LONG VECTORS FOR KEEP-ALIVE
F2E9  8D1D       ^F308             BSR     POSIT2          ;  .    $7F,$7F
F2EB  BDF36B                       JSR     ZERO            ;  .    .    ZERO INTEGRATORS
F2EE  8D20       ^F310             BSR     POSITX          ;  .    $80,$80
F2F0  2062       ^F354             BRA     ZERGND          ;  ZERO INTEGRATORS AND SET ACTIVE GROUND
```

```
                        ;
                        ;
                        ;  POSITION RELATIVE VECTOR USING 32-BIT 'YX' FORMAT
                        ;  =============================================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;
                        ;          X  = POINTS TO 32-BIT WIDE RELATIVE 'Y:X'
                        ;          DP = $D0
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;
                        ;          D  = DESTROYED
                        ;
F2F2  = 00D0                   SETDP   $D0
                        ;      =====   ===
                        ;
F2F2  C67F              POSWID  LDB     #SCAL1X          ; SET 1X VECTOR LENGTH
F2F4  D704                      STB     T1LOLC           ; .
F2F6  A684                      LDA     ,X               ; FETCH 'Y' VALUE
F2F8  E602                      LDB     2,X              ; FETCH 'X' VALUE
F2FA  2016       ^F312          BRA     POSITN           ; POSITION TO GIVEN VALUES
                        ;
                        ;
                        ;  POSITION RELATIVE VECTOR
                        ;  ========================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;
                        ;          A  = RELATIVE 'Y' VECTOR
                        ;          B  = RELATIVE 'X' VECTOR
                        ;          DP = $D0
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;
                        ;          D  = DESTROYED
                        ;
F2FC  = 00D0                   SETDP   $D0
                        ;      =====   ===
                        ;
F2FC  9701              POSITD  STA     DAC              ; SET 'Y' AXIS VALUE
F2FE  3406                      PSHS    D                ; SAVE ENTRY VALUES
F300  867F                      LDA     #SCAL1X          ; SET 1X VECTOR LENGTH
F302  9704                      STA     T1LOLC           ; .
F304  0F00                      CLR     CNTRL            ;; .
F306  2010       ^F318          BRA     PSTNO            ;; .
                        ;
                        ;
                        ;  POSITION RELATIVE VECTOR FROM LIST (SCALE = 2X)
                        ;  ============================================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;           X  = POINTS TO RELATIVE Y:X VALUES
                        ;           DP = $D0
                        ;
                        ;        RETURN VALUES
                        ;        -------------
                        ;           D  = DESTROYED
                        ;           X  = ENTRY VALUE + 2
                        ;
F308  = 00D0                     SETDP   $D0
                        ;        =====   ===
                        ;
F308  C6FF              POSIT2   LDB     #SCAL2X           ;  SET 2X VECTOR LENGTH
F30A  2002       ^F30E           BRA     POSITB            ;  +
                        ;
                        ;
                        ;  POSITION RELATIVE VECTOR FROM LIST (SCALE = 1X)
                        ;  ===============================================
                        ;
                        ;        ENTRY VALUES
                        ;        ------------
                        ;           X  = POINTS TO RELATIVE Y:X VALUES
                        ;           DP = $D0
                        ;
                        ;        RETURN VALUES
                        ;        -------------
                        ;           D  = DESTROYED
                        ;           X  = ENTRY VALUE + 2
                        ;
F30C  = 00D0                     SETDP   $D0
                        ;        =====   ===
                        ;
F30C  C67F              POSIT1   LDB     #SCAL1X           ;  SET 1X VECTOR LENGTH
                        ;
                        ;
                        ;  POSITION RELATIVE VECTOR FROM LIST
                        ;  ==================================
                        ;
                        ;        ENTRY VALUES
                        ;        ------------
                        ;           B  = VECTOR LENGTH
                        ;           X  = POINT TO RELATIVE Y:X VALUES
                        ;           DP = $D0
                        ;
                        ;        RETURN VALUES
                        ;        -------------
                        ;           D  = DESTROYED
                        ;           X  = ENTRY VALUE + 2
                        ;
F30E  = 00D0                     SETDP   $D0
                        ;        =====   ===
                        ;
F30E  D704              POSITB   STB     T1LOLC            ;  SET USER TIMER VALUE
                        ;
```

```
                         ;
                         ; POSITION RELATIVE VECTOR FROM LIST ('TIMER' SET)
                         ; ================================================
                         ;
                         ;       ENTRY VALUES
                         ;       ------------
                         ;          X  = POINTS TO RELATIVE Y:X VALUES
                         ;          DP = $D0
                         ;
                         ;       RETURN VALUES
                         ;       -------------
                         ;          D  = DESTROYED
                         ;          X  = ENTRY VALUE + 2
                         ;
F310  = 00D0                     SETDP   $D0
                         ;       =====   ===
                         ;
F310  EC81               POSITX  LDD     X++            ; FETCH 'Y:X' VALUE FROM LIST
                         ;
                         ;
                         ; POSITION RELATIVE VECTOR ('TIMER' SET)
                         ; ======================================
                         ;
                         ;       ENTRY VALUES
                         ;       ------------
                         ;          A  = RELATIVE 'Y' VECTOR VALUE
                         ;          B  = RELATIVE 'X' VECTOR VALUE
                         ;          DP = $D0
                         ;
                         ;       RETURN VALUES
                         ;       -------------
                         ;          D  = DESTROYED
                         ;
F312  = 00D0                     SETDP   $D0
                         ;       =====   ===
                         ;
F312  9701               POSITN  STA     DAC            ; SET 'Y' AXIS VALUE
F314  0F00                       CLR     CNTRL          ; .     START SAMPLE / HOLD STROBE
F316  3406                       PSHS    D              ; .     SAVE ENTRY VALUES
F318  86CE               PSTNO   LDA     #PUNZRO        ; .     RELEASE INTEGRATORS
F31A  970C                       STA     PCNTRL         ; .     ,
F31C  0F0A                       CLR     SHIFT          ; .     TURN-OFF CRT BEAM
F31E  0C00                       INC     CNTRL          ; .     STOP SAMPLE / HOLD STROBE
F320  D701                       STB     DAC            ; SET 'X' AXIS VALUE
F322  0F05                       CLR     T1HOC          ; START VECTOR RAMP
F324  3506                       PULS    D              ; CALCULATE VECTOR LENGTH DELAY
F326  BDF584                     JSR     ABSAB          ; .     FIND ABSOLUTE VALUE OF ENTRY VALUES
F329  E77F                       STB     -1,S           ; .
F32B  AA7F                       ORA     -1,S           ; .
F32D  C640                       LDB     #$40           ; .
F32F  8140                       CMPA    #$40           ; .
F331  2312      ^F345            BLS     PSTN4          ;; IF VOLTAGES LOW SKIP DELAY
F333  8164                       CMPA    #$64           ; .
```

```
F335  2304       ^F33B          BLS     PSTN1            ; .
F337  8608                      LDA     #8               ; .
F339  2002       ^F33D          BRA     PSTN2            ; .
                        ;
F33B  8604              PSTN1   LDA     #4               ; .
F33D  D50D              PSTN2   BITB    IFLAG            ;; ENDING DELAY FOR VOLTAGES OVER $40
F33F  27FC       ^F33D          BEQ     PSTN2            ; .
                        ;
F341  4A                PSTN3   DECA                     ; .
F342  26FD       ^F341          BNE     PSTN3            ; .
F344  39                        RTS                      ; .
                        ;
F345  D50D              PSTN4   BITB    IFLAG            ; .
F347  27FC       ^F345          BEQ     PSTN4            ; .
F349  39                        RTS                      ;; IF NO EXTRA DELAY-
                        ;
                        ;
                        ;  SET DP = I/O, ZERO INTEGRATORS AND SET ACTIVE GROUND
                        ;  ====================================================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          NONE REQUIRED
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          D  = $0301
                        ;          DP = $D0
                        ;
                        ;
F34A  BDF1AA            DZERO   JSR     DPIO             ; SET DP = I/O
F34D  = 00D0                    SETDP   $D0              ; .
F34D  2005       ^F354          BRA     ZERGND           ; ZERO INTEGRATORS AND SET ACTIVE GROUND
                        ;
                        ;
                        ;  CONDITIONAL ZERO INTEGRATORS AND ACTIVE GROUND
                        ;  ==============================================
                        ;
                        ;       COMMENT(S)
                        ;       ----------
                        ;          SKIPS INTEGRATOR ZEROING AND ACTIVE GROUND IF 'ZSKIP' = $00
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          DP = $D0
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          A  = $03 (IF ZEROING OCCURRED)
                        ;               $00 (IF ZEROING SKIPPED)'
                        ;
                        ;          B  = $01 (IF ZEROING OCCURRED)
                        ;               ENTRY VALUE (IF ZEROING SKIPPED)
```

```
                          ;
F34F  = 00D0                    SETDP   $D0
                          ;     =====   ===
                          ;
F34F  B6C824              CZERO LDA     ZSKIP          ; IF 'ZSKIP' = $00,
F352  2716      ^F36A           BEQ     ACTV0          ; .    THEN SKIP INTEGRATOR ZEROING
                          ;
                          ;
                          ; ZERO INTEGRATORS AND SET ACTIVE GROUND
                          ; ======================================
                          ;
                          ;     ENTRY VALUES
                          ;     ------------
                          ;        DP = $D0
                          ;
                          ;     RETURN VALUES
                          ;     -------------
                          ;        D  = $0301
                          ;
F354  = 00D0                    SETDP   $D0
                          ;     =====   ===
                          ;
F354  CC00CC              ZERGND LDD    #PZERO         ; ZERO INTEGRATORS
F357  D70C                      STB     PCNTRL         ; .    SET ZEROING CONTROL BIT
F359  970A                      STA     SHIFT          ; .    TURN-OFF CRT BEAM
                          ;
                          ;
                          ; SET ACTIVE GROUND
                          ; =================
                          ;
                          ;     ENTRY VALUES
                          ;     ------------
                          ;        DP = $D0
                          ;
                          ;     RETURN VALUES
                          ;     -------------
                          ;        D  = $0301
                          ;
F35B  = 00D0                    SETDP   $D0
                          ;     =====   ===
                          ;
F35B  CC0302              ACTGND LDD    #$0302         ; SET ACTIVE GROUND SAMPLE / HOLD
F35E  0F01                      CLR     DAC            ; .    SET DAC TO ZERO VOLTS
F360  9700                      STA     CNTRL          ; .    STROBE SAMPLE / HOLD
F362  D700                      STB     CNTRL          ; .    .
F364  D700                      STB     CNTRL          ; .    .    << TIMING >>
F366  C601                      LDB     #$1            ; .    SET MUX ADDRESS FOR 'Y' S/H
F368  D700                      STB     CNTRL          ; .    .
F36A  39                  ACTV0 RTS                    ; .    RETURN TO CALLER
                          ;
                          ;
                          ; ZERO INTEGRATORS
                          ; ================
                        ;
                        ;        ENTRY VALUES
                        ;        ------------
                        ;           DP = $D0
                        ;
                        ;        RETURN VALUES
                        ;        -------------
                        ;           A  = $00
                        ;           B  = $CC
                        ;
F36B  = 00D0                     SETDP   $D0
                        ;        =====   ===
                        ;
F36B  CC00CC            ZERO     LDD     #PZERO           ;  ZERO INTEGRATORS
F36E  D70C                       STB     PCNTRL           ;  .    SET INTEGRATOR ZERO CONTROL BIT
F370  970A                       STA     SHIFT            ;  .    TURN-OFF CRT BEAM
F372  39                         RTS                      ;  .    RETURN TO CALLER
                        ;
                        ;
                        ;  FETCH SIZE, POSITION AND DISPLAY RASTER MESSAGE
                        ;  ==============================================
                        ;
                        ;        COMMENT(S)
                        ;        ----------
                        ;           ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND ON RETURN TO USER
                        ;
                        ;        ENTRY VALUES
                        ;        ------------
                        ;           U  = POINTS TO RASTER STRING
                        ;           DP = $D0
                        ;
                        ;        RETURN VALUES
                        ;        -------------
                        ;           D  = $0301
                        ;           X  = $FBB4
                        ;           U  = END OF MESSAGE STRING + 1
                        ;
F373  = 00D0                     SETDP   $D0
                        ;        =====   ===
                        ;
F373  ECC1              RSTSIZ   LDD     U++              ;  GET RASTER SIZE
F375  FDC82A                     STD     SIZRAS           ;  .
                        ;
                        ;
                        ;  FETCH POSITION AND DISPLAY RASTER MESSAGE
                        ;  ========================================
                        ;
                        ;        COMMENT(S)
                        ;        ----------
                        ;           ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND ON RETURN TO USER
                        ;
                        ;        ENTRY VALUES
                        ;        ------------
```

```
                    ;           U  = POINTS TO RASTER MESSAGE STRING
                    ;           DP = $D0
                    ;
                    ;         RETURN VALUES
                    ;         -------------
                    ;           D  = $0301
                    ;           X  = $FBB4
                    ;           U  = END OF MESSAGE STRING + 1
                    ;
F378  = 00D0                  SETDP   $D0
                    ;         =====   ===
                    ;
F378  ECC1          RSTPOS    LDD     U++               ; GET RASTER POSITION
                    ;
                    ;
                    ;  POSITION AND DISPLAY RASTER MESSAGE
                    ;  ===================================
                    ;
                    ;         COMMENT(S)
                    ;         ----------
                    ;           ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND ON RETURN TO USER
                    ;
                    ;         ENTRY VALUES
                    ;         ------------
                    ;           A  = RELATIVE 'Y' VECTOR VALUE
                    ;           B  = RELATIVE 'X' VECTOR VALUE
                    ;           U  = POINTS TO RASTER MESSAGE STRING
                    ;           DP = $D0
                    ;
                    ;         RETURN VALUES
                    ;         -------------
                    ;           D  = $0301
                    ;           X  = $FBB4
                    ;           U  = END OF MESSAGE STRING + 1
                    ;
F37A  = 00D0                  SETDP   $D0
                    ;         =====   ===
                    ;
F37A  BDF2FC        MSSPOS    JSR     POSITD            ; POSITION MESSAGE STRING
F37D  BDF575                  JSR     DEL28             ;  .
F380  7EF495                  JMP     RASTER            ; DISPLAY RASTER MESSAGE
                    ;
                    ;
                    ;  FETCH SIZE, POSITION AND DISPLAY MULTIPLE TEXT STRINGS
                    ;  ======================================================
                    ;
                    ;         COMMENT(S)
                    ;         ----------
                    ;           ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND ON RETURN TO USER
                    ;
                    ;         ENTRY VALUES
                    ;         ------------
                    ;           U  = POINTS TO MULTIPLE TEXT STRINGS
```

```
                       ;          DP = $D0
                       ;
                       ;        RETURN VALUES
                       ;        -------------
                       ;          D  = $0301
                       ;          X  = $FBB4
                       ;          U  = END OF MESSAGE STRING + 1
                       ;
F383  = 00D0                    SETDP   $D0
                       ;        =====   ===
                       ;
F383  8DEE      ^F373  TXSZ0    BSR     RSTSIZ          ; FETCH SIZE, POSITION AND DISLAY MESSAGE
                       ;
F385  A6C4             TXTSIZ   LDA     ,U              ; END OF TEXT STRINGS ?
F387  26FA      ^F383           BNE     TXSZ0           ; .    FETCH NEXT LINE
F389  39                        RTS                     ; .    RETURN TO CALLER
                       ;
                       ;
                       ; FETCH POSITION AND DISLAY MULTIPLE TEXT STRINGS
                       ; ==============================================
                       ;
                       ;        COMMENT(S)
                       ;        ----------
                       ;          ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND ON RETURN TO USER
                       ;
                       ;        ENTRY VALUES
                       ;        ------------
                       ;          U  = POINTS TO MULTIPLE TEXT STRINGS
                       ;          DP = $D0
                       ;
                       ;        RETURN VALUES
                       ;        -------------
                       ;          D  = $0301
                       ;          X  = $FBB4
                       ;          U  = END OF MESSAGE STRING + 1
                       ;
F38A  = 00D0                    SETDP   $D0
                       ;        =====   ===
                       ;
F38A  8DEC      ^F378  TXPS0    BSR     RSTPOS          ; FETCH POSITION AND DISPLAY MESSAGE
                       ;
F38C  A6C4             TXTPOS   LDA     ,U              ; END OF TEXT STRINGS ?
F38E  26FA      ^F38A           BNE     TXPS0           ; .    FETCH NEXT LINE
F390  39                        RTS                     ; .    RETURN TO CALLER
                       ;
                       ;
                       ; DISPLAY MARKERS (COUNT REMAINING) - INDEXED POSITIONING
                       ; =======================================================
                       ;
                       ;        COMMENT(S)
                       ;        ----------
                       ;          ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND ON RETURN TO USER
                       ;
```

```
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          A  = ASCII CODE OF SYMBOL
                        ;          B  = NUMBER OF MARKERS (0 - 9)
                        ;          X  = POINTER TO THE LOCATION ON THE SCREEN
                        ;          DP = $D0
                        ;
                        ;          T1LOLC   = .
                        ;          SIZRAS   = .
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          D  = $0301
                        ;          X  = $FBB4
                        ;          U  = DESTROYED
                        ;
F391  = 00D0                    SETDP   $D0
                        ;       =====   ===
                        ;
F391  AE84              SHIPX   LDX     ,X              ; FETCH POSITION
                        ;
                        ;
                        ; DISPLAY MARKERS (COUNT REMAINING)
                        ; ================================
                        ;
                        ;       COMMENT(S)
                        ;       ----------
                        ;          ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND ON RETURN TO USER
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          A  = ASCII CODE OF SYMBOL
                        ;          B  = NUMBER OF MARKERS (0 - 9)
                        ;          X  = POSITION ON SCREEN
                        ;          DP = $D0
                        ;
                        ;          SIZRAS   = .
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          D  = $0301
                        ;          X  = $FBB4
                        ;          U  = DESTROYED
                        ;
F393  = 00D0                    SETDP   $D0
                        ;       =====   ===
                        ;
F393  3404              DSHIP   PSHS    B               ;; IF X HOLDS POSITION
F395  C680                      LDB     #$80            ;;                :
F397  3378                      LEAU    -8,S            ;;
F399  3606                      PSHU    D               ;; SHIP SYMBOL,DELIMETER
F39B  3502                      PULS    A               ;;
F39D  8109                      CMPA    #9              ;; ,
F39F  2302       ^F3A3         BLS     SHIP0            ;; .
F3A1  863C                     LDA     #$3C             ;; .
F3A3  8B30              SHIP0  ADDA    #$30             ;; .
F3A5  C62D                     LDB     #$2D             ;; .
F3A7  3606                     PSHU    D                ;; NUMBER,DASH
F3A9  3610                     PSHU    X                ;; POSITION
F3AB  20CB       ^F378         BRA     RSTPOS           ;; EXIT FROM THERE - ALL REGISTERS ? ON EXIT
                                                        ;; USES 16 BYTES UNDER STACK - HOW LOW CAN YOU GET
                        ;
                        ;
                        ;  DRAW FROM 'DUFFY' STYLE LIST (VECTORS - 1 ARE PART OF LIST AND TIME IS SET)
                        ;  ========================================================================
                        ;
                        ;        COMMENT(S)
                        ;        ----------
                        ;           ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND IF 'ZSKIP' <> 0
                        ;
                        ;        ENTRY VALUES
                        ;        ------------
                        ;           X  = POINTER TO 'DUFFY' STYLE LIST
                        ;           DP = $D0
                        ;
                        ;        RETURN VALUES
                        ;        -------------
                        ;           A  = DESTROYED
                        ;           B  = DESTROYED
                        ;           X  = END OF LIST + 2
                        ;
F3AD  = 00D0                   SETDP   $D0
                        ;      =====   ===
                        ;
F3AD  A680              DUFFAX LDA     X+               ; FETCH NUMBER OF VECTORS - 1
F3AF  2008       ^F3B9         BRA     LDUFFY           ; .
                        ;
                        ;
                        ;  DRAW FROM 'DUFFY' STYLE LIST ('LIST' IS SET)
                        ;  ============================================
                        ;
                        ;        COMMENT(S)
                        ;        ----------
                        ;           ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND IF 'ZSKIP' <> 0
                        ;
                        ;        ENTRY VALUES
                        ;        ------------
                        ;           B  = VECTOR LENGTH
                        ;           X  = POINTER TO 'DUFFY' STYLE LIST
                        ;           DP = $D0
                        ;
                        ;        RETURN VALUES
                        ;        -------------
                        ;           A  = DESTROYED
                        ;           B  = DESTROYED
                        ;           X  = END OF LIST + 2
```

```
                          ;
F3B1  = 00D0                       SETDP   $D0
                          ;        =====   ===
                          ;
F3B1  D704                DUFTIM   STB     T1LOLC          ; SET USER VECTOR LENGTH
F3B3  2007      ^F3BC              BRA     DUFFY           ; >
                          ;
                          ;
                          ;  DRAW FROM 'DUFFY' STYLE LIST (TIME AND VECTORS - 1 ARE PART OF LIST)
                          ;  =====================================================================
                          ;
                          ;        COMMENT(S)
                          ;        ----------
                          ;           ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND IF 'ZSKIP' <> 0
                          ;
                          ;        ENTRY VALUES
                          ;        ------------
                          ;           X  = POINTER TO 'DUFFY' STYLE LIST
                          ;           DP = $D0
                          ;
                          ;        RETURN VALUES
                          ;        -------------
                          ;           A  = DESTROYED
                          ;           B  = DESTROYED
                          ;           X  = END OF LIST + 2
                          ;
F3B5  = 00D0                       SETDP   $D0
                          ;        =====   ===
                          ;
F3B5  EC81                DUFLST   LDD     X++             ; FETCH VECTOR COUNT AND VECTOR LENGTH
                          ;
                          ;
                          ;  DRAW FROM 'DUFFY' STYLE LIST
                          ;  ============================
                          ;
                          ;        COMMENT(S)
                          ;        ----------
                          ;           ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND IF 'ZSKIP' <> 0
                          ;
                          ;        ENTRY VALUES
                          ;        ------------
                          ;           A  = NUMBER OF VECTORS - 1
                          ;           B  = VECTOR LENGTH
                          ;           X  = POINTER TO 'DUFFY' STYLE LIST
                          ;           DP = $D0
                          ;
                          ;        RETURN VALUES
                          ;        -------------
                          ;           A  = DESTROYED
                          ;           B  = DESTROYED
                          ;           X  = END OF LIST + 2
                          ;
F3B7  = 00D0                       SETDP   $D0
```

```
                      ;       =====   ===
                      ;
F3B7  D704            TDUFFY  STB     T1LOLC          ; SET USER VECTOR LENGTH
                      ;
                      ;
                      ;  DRAW FROM 'DUFFY' STYLE LIST (TIME IS SET)
                      ;  =========================================
                      ;
                      ;       COMMENT(S)
                      ;       ----------
                      ;          ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND IF 'ZSKIP' <> 0
                      ;
                      ;       ENTRY VALUES
                      ;       ------------
                      ;          A  = NUMBER OF VECTORS - 1
                      ;          X  = POINTER TO 'DUFFY' STYLE LIST
                      ;          DP = $D0
                      ;
                      ;       RETURN VALUES
                      ;       -------------
                      ;          A  = DESTROYED
                      ;          B  = DESTROYED
                      ;          X  = END OF LIST + 2
                      ;
F3B9  = 00D0                  SETDP   $D0
                      ;       =====   ===
                      ;
F3B9  B7C823          LDUFFY  STA     LIST            ; SET VECTOR COUNT
                      ;
                      ;
                      ;  DRAW FROM 'DUFFY' STYLE LIST DEFINED BY 'X' ('LIST' AND TIME ARE SET)
                      ;  ======================================================================
                      ;
                      ;       COMMENT(S)
                      ;       ----------
                      ;          ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND IF 'ZSKIP' <> 0
                      ;
                      ;       ENTRY VALUES
                      ;       ------------
                      ;          X  = POINTER TO 'DUFFY' STYLE LIST
                      ;          DP = $D0
                      ;
                      ;       RETURN VALUES
                      ;       -------------
                      ;          A  = DESTROYED
                      ;          B  = DESTROYED
                      ;          X  = END OF LIST + 2
                      ;
F3BC  = 00D0                  SETDP   $D0
                      ;       =====   ===
                      ;
F3BC  EC84            DUFFY   LDD     ,X              ; FETCH RELATIVE 'Y:X'
                      ;
```

```
                            ;
                            ;  DRAW ONE VECTOR CONTAINED IN 'A' & 'B'
                            ;  ====================================
                            ;
                            ;
                            ;       COMMENT(S)
                            ;       ----------
                            ;
                            ;          ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND IF 'ZSKIP' <> 0
                            ;
                            ;       ENTRY VALUES
                            ;       ------------
                            ;
                            ;          A  = RELATIVE 'Y' VECTOR VALUE
                            ;          B  = RELATIVE 'X' VECTOR VALUE
                            ;          DP = $D0
                            ;
                            ;       RETURN VALUES
                            ;       -------------
                            ;
                            ;          A  = DESTROYED
                            ;          B  = DESTROYED
                            ;          X  = ENTRY VALUE + 2
                            ;
F3BE  = 00D0                        SETDP   $D0
                            ;       =====   ===
                            ;
F3BE  9701                  DUFFAB  STA     DAC             ;  SET 'Y' AXIS VALUE
F3C0  0F00                          CLR     CNTRL           ;  .    START SAMPLE / HOLD STROBE
F3C2  3002                          LEAX    2,X             ;  .    POSITION TO NEXT ENTRY
F3C4  12                            NOP                     ;  .    .    << TIMING >>
F3C5  0C00                          INC     CNTRL           ;  .    STOP SAMPLE / HOLD STROBE
                            ;
F3C7  D701                          STB     DAC             ;  SET 'X' AXIS VALUE
                            ;
F3C9  CC0000                        LDD     #0              ;  TURN-OFF BEAM & START VECTOR
F3CC  201F      ^F3ED               BRA     DIFA0           ;  .
                            ;
                            ;
                            ;  DRAW FROM 'DIFFY' STYLE LIST (VECTORS-1 ARE PART OF LIST AND TIME IS SET)
                            ;  ======================================================================
                            ;
                            ;       COMMENT(S)
                            ;       ----------
                            ;
                            ;          ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND IF 'ZSKIP' <> 0
                            ;
                            ;       ENTRY VALUES
                            ;       ------------
                            ;
                            ;          X  = POINTER TO 'DIFFY' STYLE LIST
                            ;          DP = $D0
                            ;
                            ;       RETURN VALUES
                            ;       -------------
                            ;
                            ;          A  = DESTROYED
                            ;          B  = DESTROYED
                            ;          X  = END OF LIST + 2
                            ;
```

```
                        ;         LIST     = $00
                        ;
F3CE  = 00D0                      SETDP   $D0
                        ;         =====   ===
                        ;
F3CE  A680              DIFFAX    LDA     X+              ; FETCH NUMBER OF VECTORS - 1
F3D0  2008      ^F3DA             BRA     LDIFFY          ; .
                        ;
                        ;
                        ; DRAW FROM 'DIFFY' STYLE LIST ('LIST' IS SET)
                        ; ============================================
                        ;
                        ;         COMMENT(S)
                        ;         ----------
                        ;            ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND IF 'ZSKIP' <> 0
                        ;
                        ;         ENTRY VALUES
                        ;         ------------
                        ;            B  = VECTOR LENGTH
                        ;            X  = POINTER TO 'DIFFY' STYLE LIST
                        ;            DP = $D0
                        ;
                        ;         RETURN VALUES
                        ;         -------------
                        ;            A  = DESTROYED
                        ;            B  = DESTROYED
                        ;            X  = END OF LIST + 2
                        ;
                        ;            LIST     = $00
                        ;
F3D2  = 00D0                      SETDP   $D0
                        ;         =====   ===
                        ;
F3D2  D704              DIFTIM    STB     T1LOLC          ; SET USER VECTOR LENGTH
F3D4  2007      ^F3DD             BRA     DIFFY           ; .
                        ;
                        ;
                        ; DRAW FROM 'DIFFY' STYLE LIST (TIME AND VECTORS-1 ARE PART OF LIST)
                        ; ==================================================================
                        ;
                        ;         COMMENT(S)
                        ;         ----------
                        ;            ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND IF 'ZSKIP' <> 0
                        ;
                        ;         ENTRY VALUES
                        ;         ------------
                        ;            X  = POINTER TO 'DIFFY' STYLE LIST
                        ;            DP = $D0
                        ;
                        ;         RETURN VALUES
                        ;         -------------
                        ;            A  = DESTROYED
                        ;            B  = DESTROYED
                        ;          X  = END OF LIST + 2
                        ;
                        ;          LIST     = $00
                        ;
F3D6  = 00D0                    SETDP   $D0
                        ;       =====   ===
                        ;
F3D6  EC81              DIFLST  LDD     X++             ; FETCH VECTOR COUNT AND VECTOR LENGTH
                        ;
                        ;
                        ;  DRAW FROM 'DIFFY' STYLE LIST
                        ;  ============================
                        ;
                        ;       COMMENT(S)
                        ;       ----------
                        ;          ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND IF 'ZSKIP' <> 0
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          A  = NUMBER OF VECTORS - 1
                        ;          B  = VECTOR LENGTH
                        ;          X  = POINTER TO 'DIFFY' STYLE LIST
                        ;          DP = $D0
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          A  = DESTROYED
                        ;          B  = DESTROYED
                        ;          X  = END OF LIST + 2
                        ;
                        ;          LIST     = $00
                        ;
F3D8  = 00D0                    SETDP   $D0
                        ;       =====   ===
                        ;
F3D8  D704              TDIFFY  STB     T1LOLC          ; SET USER VECTOR LENGTH
                        ;
                        ;
                        ;  DRAW FROM 'DIFFY' SYTLE LIST (TIME IS SET)
                        ;  ==========================================
                        ;
                        ;       COMMENT(S)
                        ;       ----------
                        ;          ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND IF 'ZSKIP' <> 0
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          A  = NUMBER OF VECTORS - 1
                        ;          X  = POINTER TO 'DIFFY' STYLE LIST
                        ;          DP = $D0
                        ;
                        ;       RETURN VALUES
                        ;       -------------
```

```
                         ;          A  = DESTROYED
                         ;          B  = DESTROYED
                         ;          X  = END OF LIST + 2
                         ;
                         ;          LIST     = $00
                         ;
F3DA  = 00D0                    SETDP   $D0
                         ;      =====   ===
                         ;
F3DA  B7C823            LDIFFY  STA     LIST              ; SET VECTOR COUNT
                         ;
                         ;
                         ;  DRAW FROM 'DIFFY' STYLE LIST DEFINED BY 'X' ('LIST' AND TIME ARE SET)
                         ;  ===================================================================
                         ;
                         ;      COMMENT(S)
                         ;      ----------
                         ;          ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND IF 'ZSKIP' <> 0
                         ;
                         ;      ENTRY VALUES
                         ;      ------------
                         ;          X  = POINTER TO 'DIFFY' STYLE LIST
                         ;          DP = $D0
                         ;
                         ;      RETURN VALUES
                         ;      -------------
                         ;          A  = DESTROYED
                         ;          B  = DESTROYED
                         ;          X  = END OF LIST + 2
                         ;
                         ;          LIST     = $00
                         ;
F3DD  = 00D0                    SETDP   $D0
                         ;      =====   ===
                         ;
F3DD  EC84              DIFFY   LDD     ,X                ; FETCH RELATIVE 'Y:X'
                         ;
                         ;
                         ;  DRAW ONE VECTOR CONTAINED IN 'A' & 'B'
                         ;  ======================================
                         ;
                         ;      COMMENT(S)
                         ;      ----------
                         ;          ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND IF 'ZSKIP' <> 0
                         ;
                         ;      ENTRY VALUES
                         ;      ------------
                         ;          A  = RELATIVE 'Y' VECTOR VALUE
                         ;          B  = RELATIVE 'X' VECTOR VALUE
                         ;          DP = $D0
                         ;
                         ;          LIST     = $00
                         ;
```

```
                       ;       RETURN VALUES
                       ;       -------------
                       ;          A  = DESTROYED
                       ;          B  = DESTROYED
                       ;          X  = ENTRY VALUE + 2
                       ;
F3DF  = 00D0                   SETDP   $D0
                       ;       =====   ===
                       ;
F3DF  9701             DIFFAB  STA     DAC             ;  SET 'Y' AXIS VALUE
F3E1  0F00                     CLR     CNTRL           ;  .    START SAMPLE / HOLD STROBE
F3E3  3002                     LEAX    2,X             ;  .    POSITION FOR NEXT ENTRY
F3E5  12                       NOP                     ;  .    .    << TIMING >>
F3E6  0C00                     INC     CNTRL           ;  .    STOP SAMPLE / HOLD STROBE
F3E8  D701                     STB     DAC             ;  SET 'X' AXIS VALUE
                       ;
F3EA  CCFF00                   LDD     #$FF00          ;  TURN-ON CRT BEAM
F3ED  970A             DIFA0   STA     SHIFT           ;  .
F3EF  D705                     STB     T1HOC           ;  START VECTOR RAMP
                       ;
F3F1  CC0040                   LDD     #$0040          ;  WAIT FOR VECTOR COMPLETION
F3F4  D50D             DIFA1   BITB    IFLAG           ;  .
F3F6  27FC       ^F3F4         BEQ     DIFA1           ;  .
F3F8  12                       NOP                     ;  .    << TIMING >>
                       ;
F3F9  970A                     STA     SHIFT           ;  TURN-OFF CRT BEAM
                       ;
F3FB  B6C823                   LDA     LIST            ;  END OF LIST ?
F3FE  4A                       DECA                    ;  .
F3FF  2AD9       ^F3DA         BPL     LDIFFY          ;  .
F401  7EF34F                   JMP     CZERO           ;  .    CONDITIONAL INTEGRATOR ZEROING
                       ;
                       ;
                       ;  DRAW FROM 'PACKET' STYLE LIST (SCALE = 2X)
                       ;  =========================================
                       ;
                       ;       COMMENT(S)
                       ;       ----------
                       ;          ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND IF 'ZSKIP' <> 0
                       ;
                       ;       ENTRY VALUES
                       ;       ------------
                       ;          X  = POINTS TO 'PACKET' STYLE LIST
                       ;          DP = $D0
                       ;
                       ;       RETURN VALUES
                       ;       -------------
                       ;          D  = DESTROYED
                       ;          X  = POINTS TO END OF LIST
                       ;
F404  = 00D0                   SETDP   $D0
                       ;       =====   ===
                       ;
```

```
F404  C6FF             PACK2X   LDB    #SCAL2X         ;  SET 2X VECTOR LENGTH
F406  2006       ^F40E          BRA    TPACK           ;  .
                       ;
                       ;
                       ;  DRAW FROM 'PACKET STYLE LIST (SCALE = 1X)
                       ;  ========================================
                       ;
                       ;        COMMENT(S)
                       ;        ----------
                       ;           ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND IF 'ZSKIP' <> 0
                       ;
                       ;        ENTRY VALUES
                       ;        ------------
                       ;           X  = POINTS TO 'PACKET' STYLE LIST
                       ;           DP = $D0
                       ;
                       ;        RETURN VALUES
                       ;        -------------
                       ;           D  = DESTROYED
                       ;           X  = POINTS TO END OF LIST
                       ;
F408   = 00D0                   SETDP  $D0
                       ;        =====  ===
                       ;
F408  C67F             PACK1X   LDB    #SCAL1X         ;  SET 1X VECTOR LENGTH
F40A  2002       ^F40E          BRA    TPACK           ;  .
                       ;
                       ;
                       ;  DRAW FROM 'PACKET' STYLE LIST (TIME IS PART OF LIST)
                       ;  =====================================================
                       ;
                       ;        COMMENT(S)
                       ;        ----------
                       ;           ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND IF 'ZSKIP' <> 0
                       ;
                       ;        ENTRY VALUES
                       ;        ------------
                       ;           X  = POINTS TO 'PACKET' STYLE LIST
                       ;           DP = $D0
                       ;
                       ;        RETURN VALUES
                       ;        -------------
                       ;           D  = DESTROYED
                       ;           X  = POINTS TO END OF LIST
                       ;
F40C   = 00D0                   SETDP  $D0
                       ;        =====  ===
                       ;
F40C  E680             TPACK    LDB    X+              ;  FETCH TIMER VALUE FROM LIST
                       ;
                       ;
                       ;  DRAW FROM 'PACKET' STYLE LIST
                       ;  ============================
                        ;
                        ;       COMMENT(S)
                        ;       ----------
                        ;          ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND IF 'ZSKIP' <> 0
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          B  = VECTOR LENGTH
                        ;          X  = POINTS TO 'PACKET' STYLE LIST
                        ;          DP = $D0
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          D  = DESTROYED
                        ;          X  = POINTS TO END OF LIST
                        ;
F40E  = 00D0                    SETDP   $D0
                        ;       =====   ===
                        ;
F40E  D704              TPACK   STB     T1LOLC          ; SET USER VECTOR LENGTH
                        ;
                        ;
                        ;  DRAW FROM 'PACKET' STYLE LIST (TIMER IS SET)
                        ;  ==========================================
                        ;
                        ;       COMMENT(S)
                        ;       ----------
                        ;          ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND IF 'ZSKIP' <> 0
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          X  = POINTS TO 'PACKET' STYLE LIST
                        ;          DP = $D0
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          D  = DESTROYED
                        ;          X  = POINTS TO END OF LIST
                        ;
F410  = 00D0                    SETDP   $D0
                        ;       =====   ===
                        ;
F410  EC01              PACKET  LDD     1,X             ; FETCH 'Y:X' VALUES
                        ;
F412  9701                      STA     DAC             ; SET 'Y' AXIS VALUE
F414  0F00                      CLR     CNTRL           ;  .    START SAMPLE / HOLD STROBE
F416  A684                      LDA     ,X              ;  .    FETCH CRT BEAM ENABLE
F418  3003                      LEAX    3,X             ;  .    POSITION FOR NEXT ENTRY
F41A  0C00                      INC     CNTRL           ;  .    STOP SAMPLE / HOLD STROBE
                        ;
F41C  D701                      STB     DAC             ; SET 'X' AXIS VALUE
F41E  970A                      STA     SHIFT           ; SET CRT BEAM ON / OFF ACCORDING ENTRY
F420  0F05                      CLR     T1HOC           ; START VECTOR RAMP
```

```
                        ;
F422  CC0040                    LDD     #$0040          ;  WAIT FOR VECTOR COMLETION
F425  D500              PCKO    BITB    IFLAG           ;  .
F427  27FC      ^F425           BEQ     PCKO            ;  .
F429  12                        NOP                     ;  .    (( TIMING ))
                        ;
F42A  970A                      STA     SHIFT           ;  TURN-OFF CRT BEAM
                        ;
F42C  A684                      LDA     ,X              ;  PACKET TERMINATOR ?
F42E  2FE0      ^F410           BLE     PACKET          ;  .
F430  7EF34F                    JMP     CZERO           ;  .    CONDITIONAL INTEGRATOR ZEROING
                        ;
                        ;
                        ;  DRAW DASHED LINES ACCORDING TO 'DIFFY' FORMAT (VECTOR COUNT)
                        ;  ===============================================================
                        ;
                        ;       COMMENT(S)
                        ;       ----------
                        ;
                        ;          ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND IF 'ZSKIP' <> 0
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;
                        ;          A  = NUMBER OF VECTORS
                        ;          X  = POINTS TO 'DIFFY' STYLE LIST
                        ;          DP = $D0
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;
                        ;          D  = DESTROYED
                        ;          X  = END OF LIST + 2
                        ;
F433  = 00D0                    SETDP   $D0
                        ;       =====   ===
                        ;
F433  4A                DSHDF1  DECA                    ;  DECREMENT VECTOR COUNT
                        ;
                        ;
                        ;  DRAW DASHED LINES ACCORDING TO 'DIFFY' FORMAT (VECTOR COUNT - 1)
                        ;  ===================================================================
                        ;
                        ;       COMMENT(S)
                        ;       ----------
                        ;
                        ;          ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND IF 'ZSKIP' <> 0
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;
                        ;          A  = NUMBER OF VECTORS - 1
                        ;          X  = POINTS TO 'DIFFY' STYLE LIST
                        ;          DP = $D0
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;
                        ;          D  = DESTROYED
```

```
                          ;       X  = END OF LIST + 2
                          ;
F434  = 00D0                      SETDP  $D0
                          ;       =====  ===
                          ;
F434  B7C823              DSHDF   STA    LIST            ; SAVE VECTOR COUNT
                          ;
                          ;
                          ; DRAW DASHED LINES ACCORDING TO 'DIFFY' FORMAT ('LIST' IS SET)
                          ; =============================================================
                          ;
                          ;       COMMENT(S)
                          ;       ----------
                          ;          ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND IF 'ZSKIP' <> 0
                          ;
                          ;       ENTRY VALUES
                          ;       ------------
                          ;          X  = POINTS TO 'DIFFY' STYLE LIST
                          ;          DP = $D0
                          ;
                          ;       RETURN VALUES
                          ;       -------------
                          ;          D  = DESTROYED
                          ;          X  = END OF LIST + 2
                          ;
F437  = 00D0                      SETDP  $D0
                          ;       =====  ===
                          ;
F437  EC84                DASHDF  LDD    ,X              ; FETCH 'Y:X' VALUES
                          ;
F439  9701                        STA    DAC             ; SET 'Y' AXIS VALUE
F43B  0F00                        CLR    CNTRL           ; .    START SAMPLE / HOLD STROBE
F43D  3002                        LEAX   2,X             ; .    POSITION TO NEXT ENTRY
F43F  0C00                        INC    CNTRL           ; .    STOP SAMPLE / HOLD STROBE
                          ;
F441  D701                        STB    DAC             ; SET 'X' AXIS VALUE
                          ;
F443  B6C829                      LDA    DASH            ; DRAW DASHED LINE DURING VECTOR
F446  C640                        LDB    #$40            ; .
F448  970A                        STA    SHIFT           ; .   SET DASH PATTERN IN SHIFT REGISTER
F44A  0F05                        CLR    T1HOC           ; .   START VECTOR RAMP
F44C  = 0000                      SETDP  $00             ; .   IS VECTOR COMPLETE ?
F44C  F5D000                      BITB   IFLAG           ; .   .
F44F  = 00D0                      SETDP  $D0             ; .   .
F44F  270B        ^F45C           BEQ    DSHD1           ; .   .    LONG DASHED LINE ?
F451  0F0A                        CLR    SHIFT           ; .   SHORT DASH LINE, CLEAR DASHING
                          ;
F453  B6C823                      LDA    LIST            ; END OF LIST ?
F456  26DB        ^F433           BNE    DSHDF1          ; .
F458  39                          RTS                    ; .    RETURN TO CALLER
                          ;
                          ;
F459  B6C829              DSHDF0  LDA    DASH            ; LONG DASHED LINE
```

```
F45C  970A            DSHD1    STA    SHIFT          ;  .    REPEAT DASH PATTERN
F45E  12                       NOP                   ;  .    .    (( TIMING ))
F45F  D500                     BITB   IFLAG          ;  .    WAIT FOR VECTOR COMPLETION
F461  27F6     ^F459           BEQ    DSHDF0         ;  .    .
F463  B6C823                   LDA    LIST           ;  .    VECTOR COMPLETE
F466  0F0A                     CLR    SHIFT          ;  .    .    CLEAR DASHING
F468  4D                       TSTA                  ;  .    .    END OF DASHING LIST ?
F469  26C8     ^F433           BNE    DSHDF1         ;  .    .    .
F46B  7EF34F                   JMP    CZERO          ;  .    .    CONDITIONAL INTEGRATOR ZEROING
                      ;
                      ;
                      ;  DRAW DASHED LINES ACCORDING TO 'PACKET' FORMAT
                      ;  ==============================================
                      ;
                      ;        COMMENT(S)
                      ;        ----------
                      ;           ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND IF 'ZSKIP' <> 0
                      ;
                      ;        ENTRY VALUES
                      ;        ------------
                      ;           X  = POINTS TO 'PACKET' STYLE LIST
                      ;           DP = $D0
                      ;
                      ;        RETURN VALUES
                      ;        -------------
                      ;           D  = DESTROYED
                      ;           X  = END OF LIST + 2
                      ;
F46E  = 00D0                   SETDP  $D0
                      ;        =====  ===
                      ;
F46E  B6C824          DASHPK   LDA    ZSKIP          ;;  DASHED LINES- 3 BYTE PER VECTOR <----
F471  3402                     PSHS   A              ;;  .
F473  7FC824                   CLR    ZSKIP          ;;  .
F476  A680            DSHPK0   LDA    X+             ;;  .
F478  2A04     ^F47E           BPL    DSHPK1         ;;  .
F47A  8DBB     ^F437           BSR    DASHDF         ;;  $FF ENABLES DOTTED LINE
F47C  20F8     ^F476           BRA    DSHPK0         ;;  .
                      ;
F47E  2605     ^F485  DSHPK1   BNE    DSHPK2         ;;  .
F480  BDF3BC                   JSR    DUFFY          ;;  $00 REQUESTS BLANK LINE
F483  20F1     ^F476           BRA    DSHPK0         ;;  .
                      ;
F485  4A              DSHPK2   DECA                  ;;  .
F486  2705     ^F48D           BEQ    DSHPK3         ;;  .
F488  BDF3DD                   JSR    DIFFY          ;;  $02 IS SOLID LINE
F48B  20E9     ^F476           BRA    DSHPK0         ;;  .
                      ;
F48D  3502            DSHPK3   PULS   A              ;;  $01 DELIMIT
F48F  B7C824                   STA    ZSKIP          ;;  .
F492  7EF34F                   JMP    CZERO          ;  CONDITIONAL INTEGRATOR ZEROING
                      ;
                      ;
```

```
                        ; DISPLAY RASTER STRING AS INDICATED BY 'U'
                        ; ========================================
                        ;
                        ;       COMMENT(S)
                        ;       ----------
                        ;          ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND ON RETURN TO USER
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          U  = POINTS TO RASTER MESSAGE STRING
                        ;          DP = $D0
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          D  = $0301
                        ;          X  = $FBB4
                        ;          U  = END OF MESSAGE STRING + 1
                        ;
F495  = 00D0                    SETDP   $D0
                        ;       =====   ===
                        ;
F495  FFC82C            RASTER  STU     MESAGE          ; SAVE MESSAGE POINTER IN 'MESAGE'
                        ;
                        ;
                        ; DISPLAY RASTER STRING INDICATED BY 'MESAGE'
                        ; ===========================================
                        ;
                        ;       COMMENT(S)
                        ;       ----------
                        ;          ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND ON RETURN TO USER
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          DP = $D0
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          D  = $0301
                        ;          X  = $FBB4
                        ;          U  = END OF MESSAGE STRING + 1
                        ;
F498  = 00D0                    SETDP   $D0
                        ;       =====   ===
                        ;
F498  8EF9D4            MRASTR  LDX     #ASCII-$20      ;; FOR STANDARD ASCII <----
F49B  CC1883                    LDD     #$1883          ;; ALPHA-NUM USING ASCII CODES <----
F49E  0F01                      CLR     DAC             ;; .
                                                        ;; SUPPORTS CODES $20-$6F. $20-$5F ARE STANDARD.$80=DELIM
F4A0  970B                      STA     ACNTRL          ;; MINIMUM 3 CHARS. CODES $60-$6F=GRAPHIC CHARS, SEE DATA
F4A2  8EF9D4                    LDX     #ASCII-$20      ;; DECODE TABLE 5X7
F4A5  D700              RSTRO   STB     CNTRL           ;; START ZEREF UPDATE
F4A7  0A00                      DEC     CNTRL           ;; .
F4A9  CC8081                    LDD     #$8081          ;; .
```

```
F4AC  12                      NOP                       ; .    << TIMING >>
F4AD  0C00                    INC     CNTRL             ;; .
F4AF  D700                    STB     CNTRL             ;; .
F4B1  9700                    STA     CNTRL             ;; START Y=0 SAMPLE
                      ;=================================================================JJH
                      ;       TST     RECO              ; CODE DELETED - REV. C CHANGES    =====JJH
                      ;=================================================================JJH
                      ;
                      ;=================================================================JJH
F4B3  BDF57D                  JSR     DEL13             ; CODE ADDED - REV. C CHANGES      =====JJH
                      ;=================================================================JJH
F4B6  0C00                    INC     CNTRL             ;; Y S/H FINISHED
F4B8  B6C82B          RSTR1   LDA     SIZRAS+1          ;; .
F4BB  9701                    STA     DAC               ;; FOR HORIZ SPEED
F4BD  CC0100                  LDD     #$0100            ;; .
F4C0  FEC82C                  LDU     MESAGE            ;; .
F4C3  9700                    STA     CNTRL             ;; START RAMP
F4C5  2004      ^F4CB         BRA     RSTR3             ;; 3
                      ;
F4C7  A686            RSTR2   LDA     A,X               ;; 5- CYCLE COUNT
F4C9  970A                    STA     SHIFT             ;; 4
F4CB  A6C0            RSTR3   LDA     U+                ;; 6
F4CD  2AF8      ^F4C7         BPL     RSTR2             ;; 3- TOTAL 18 FOR LOOP
F4CF  8681                    LDA     #$81              ;; 2
F4D1  9700                    STA     CNTRL             ;; 4. STOP RAMP, TIME=18 CYCLES+18 PER CHAR
F4D3  0001                    NEG     DAC               ;; .
F4D5  8601                    LDA     #$01              ;; .
F4D7  9700                    STA     CNTRL             ;; START RETRACE 2X FWD SPEED
F4D9  8CFBB4                  CMPX    #ASCII+$50*6-$20 ; 4
F4DC  272C      ^F50A         BEQ     RSTR5             ;; 3
F4DE  308850                  LEAX    $50,X             ;; 5
F4E1  1F30                    TFR     U,D               ;; 6 CYCLES - CALCULATE #CHARS SHOWN
F4E3  B3C82C                  SUBD    MESAGE            ;; 7 - RESULT IN B IS 1+#CHARS
F4E6  C002                    SUBB    #2                ;; 2
F4E8  58                      ASLB                      ;; 2
F4E9  2100      ^F4EB         BRN     RSTR4             ;; 3
F4EB  8681            RSTR4   LDA     #$81              ;; 2 DECR LOOP = 9 PER
F4ED  12                      NOP                       ;; .    << TIMING >>
F4EE  5A                      DECB                      ;; 2
F4EF  26FA      ^F4EB         BNE     RSTR4             ;; 3
F4F1  9700                    STA     CNTRL             ;; 4  WHOLE RETRACE= 45+ 9 PER B DELAY LOOP
                                                        ;; 39-12 FOR FWD OVERHEAD=27
                                                        ;; EQUIV OF 3 CHARS
F4F3  F6C82A                  LDB     SIZRAS            ;; .
                      ;=================================================================JJH
                      ;       STB     DAC               ; CODE DELETED - REV. B1 CHANGES   =====JJH
                      ;       DEC     CNTRL             ; .                                =====JJH
                      ;       LDD     #$8101            ; .                                =====JJH
                      ;       NOP                       ; .                                =====JJH
                      ;       STA     CNTRL             ; .                                =====JJH
                      ;       CLR     DAC               ; .                                =====JJH
                      ;       STB     CNTRL             ; .                                =====JJH
                      ;       STA     CNTRL             ; .                                =====JJH
```

```
                        ;       LDB     #$03            ; .                             =====JJH
                        ;==================================================================JJH
                        ;
                        ;==================================================================JJH
F4F6  58                        ASLB                    ; CODE ADDED - REV. B1 CHANGES  =====JJH
F4F7  58                        ASLB                    ; .                             =====JJH
F4F8  D701                      STB     DAC             ; .                             =====JJH
F4FA  0A00                      DEC     CNTRL           ; .                             =====JJH
F4FC  C601                      LDB     #$01            ; .                             =====JJH
F4FE  0C00                      INC     CNTRL           ; .                             =====JJH
F500  0F01                      CLR     DAC             ; .                             =====JJH
F502  D700                      STB     CNTRL           ; .                             =====JJH
F504  C683                      LDB     #$83            ; .                             =====JJH
F506  9700                      STA     CNTRL           ; .                             =====JJH
                        ;==================================================================JJH
F508  209B      ^F4A5           BRA     RSTR0           ;; END 0 RASTER
                        ;
F50A  8698              RSTR5   LDA     #$98            ;; .
F50C  970B                      STA     ACNTRL          ;; .
F50E  7EF354                    JMP     ZERGND          ; ZERO INTEGRATOR AND SET ACTIVE GROUND
                        ;
                        ;
                        ;  CALCULATE THREE NEW RANDOM NUMBERS
                        ;  ==================================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          NONE REQUIRED
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          A  = RANDOM NUMBER
                        ;
F511  = 0000                    SETDP   $00
                        ;       =====   ===
                        ;
F511  3414              RAND3   PSHS    B,X             ;; RANDOM, 3 NEW BITS <----
F513  C602                      LDB     #2              ;; .
F515  2003      ^F51A           BRA     RANDO           ;; ANSWER RETURNS IA 'A'
                        ;
                        ;
                        ;  CALCULATE NEW RANDOM NUMBER
                        ;  ===========================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          NONE REQUIRED
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          A  = RANDOM NUMBER
                        ;
F517  = 0000                    SETDP   $00
```

```
                        ;       =====   ===
                        ;
F517  3414              RANDOM  PSHS    B,X                 ;; RANDOM, 1 NEW BIT (----
F519  5F                        CLRB                        ;; .
F51A  BEC87B            RAND0   LDX     SEED                ;; 'SEED' IS INDIRECT POINTER
F51D  A601              RAND1   LDA     1,X                 ;; ABOUT 50 USEC
F51F  49                        ROLA                        ;; ALL REGS INTACT BESIDES 'A'
F520  49                        ROLA                        ;; .
F521  49                        ROLA                        ;; .
F522  49                        ROLA                        ;; .
F523  A802                      EORA    2,X                 ;; .
F525  46                        RORA                        ;; INTO CARRY
F526  6984                      ROL     ,X                  ;; .
F528  6901                      ROL     1,X                 ;; .
F52A  6902                      ROL     2,X                 ;; .
F52C  5A                        DECB                        ;; .
F52D  2AEE       ^F51D          BPL     RAND1               ;; .
F52F  A684                      LDA     ,X                  ;; .
F531  3594                      PULS    B,X,PC              ;; END OF RANDOM
                        ;
                        ;
                        ; INITIALIZE THE 'REQX' AREA
                        ; ==========================
                        ;
                        ;       COMMENT(S)
                        ;       ----------
                        ;          SETS REQ0 - REQ5, REQ7 - REQD = $00
                        ;               REQ6 = $3F
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          NONE REQUIRED
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          A  = $3F
                        ;          B  = $FF
                        ;          X  = $C83F (#REQ0)
                        ;
F533  = 0000                    SETDP   $00
                        ;       =====   ===
                        ;
F533  C60D              INTREQ  LDB     #$0D               ; CLEAR 'REQX' AREA
F535  8EC83F                    LDX     #REQ0              ; .
F53A  8D05       ^F53F          BSR     BCLR               ; .
F53A  863F                      LDA     #$3F               ; SET 'REQ6' = $3F
F53C  A706                      STA     6,X                ; .
F53E  39                        RTS                        ; RETURN TO CALLER
                        ;
                        ;
                        ; CLEAR 'B' BYTES STARTING AT 'X'
                        ; ===============================
                        ;
```

```
                          ;       ENTRY VALUES
                          ;       ------------
                          ;          B  = NUMBER OF BYTES TO BE CLEARED
                          ;          X  = POINTER TO BUFFER TO BE CLEARED
                          ;
                          ;       RETURN VALUES
                          ;       -------------
                          ;          D  = $FFFF
                          ;
F53F  = 0000                      SETDP   $00
                          ;       =====   ===
                          ;
F53F  4F                  BCLR    CLRA                    ; EXTEND VALUE IN REGISTER 'B'
F540  2006       ^F548            BRA     CLRBLK          ; CLEAR INDICATED BLOCK
                          ;
                          ;
                          ; CLEAR EXECUTIVE AREA OF MEMORY ($C800 - $C8FF)
                          ; ==============================================
                          ;
                          ;       ENTRY VALUES
                          ;       ------------
                          ;          NONE REQUIRED
                          ;
                          ;       RETURN VALUES
                          ;       -------------
                          ;          D  = $FFFF
                          ;
F542  = 0000                      SETDP   $00
                          ;       =====   ===
                          ;
F542  8EC800              CLREX   LDX     #$C800          ; CLEAR 256 BYTES STARTING AT $C800
                          ;
                          ;
                          ; CLEARS 256 BYTES STARTING AT 'X'
                          ; ================================
                          ;
                          ;       ENTRY VALUES
                          ;       ------------
                          ;          X  = POINTER TO BUFFER TO BE CLEARED
                          ;
                          ;       RETURN VALUES
                          ;       -------------
                          ;          D  = $FFFF
                          ;
F545  = 0000                      SETDP   $00
                          ;       =====   ===
                          ;
F545  CC00FF              CLR256  LDD     #$00FF          ; SET-UP TO CLEAR 256 BYTES
                          ;
                          ;
                          ; CLEARS A BLOCK OF MEMORY
                          ; ========================
                          ;
```

```
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          D  = NUMBER OF BYTES TO BE CLEARED
                        ;          X  = POINTER TO BUFFER TO BE CLEARED
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          D  = $FFFF
                        ;
F548  = 0000                    SETDP   $00
                        ;       =====   ===
                        ;
F548  6F8B              CLRBLK  CLR     D,X             ;  CLEAR BUFFER BYTE
F54A  830001                    SUBD    #1              ;  DECREMENT BYTE COUNTER
F54D  2AF9      ^F548           BPL     CLRBLK          ;  .    IF >0, THEN CLEAR NEXT BYTE
F54F  39                        RTS                     ;  .    RETURN TO CALLER
                        ;
                        ;
                        ;  SET A BLOCK OF MEMORY STARTING AT 'X' TO $80
                        ;  ============================================
                        ;
                        ;       COMMENT(S)
                        ;       ----------
                        ;          TYPICAL USE FOR THIS ROUTINE WOULD BE FOR CLEARING THE SCORE
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          B  = NUMBER OF BYTES TO SET ($01 - $7F)
                        ;          X  = POINTER TO BUFFER TO BE SET
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          A  = $80
                        ;          B  = $00
                        ;
F550  = 0000                    SETDP   $00
                        ;       =====   ===
                        ;
F550  8680              CLR80   LDA     #$80            ;  SET 'BLKFILL' PATTERN
                        ;
                        ;
                        ;  SET A BLOCK OF MEMORY STARTING AT 'X'
                        ;  =====================================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          A  = FILL PATTERN
                        ;          B  = NUMBER OF BYTES TO BE SET ($01 - $7F)
                        ;          X  = POINTER TO BUFFER TO BE FILLED
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          B  = $00
                        ;
F552 = 0000                     SETDP   $00
                        ;       =====   ===
                        ;
F552 A785               BLKFIL  STA     B,X             ; STORE BYTE OF DATA
F554 5A                         DECB                    ; DECREMENT BYTE COUNTER
F555 26FB      ^F552            BNE     BLKFIL          ; .    IF >0, THEN STORE NEXT BYTE
F557 A784                       STA     ,X              ; .    SET LAST BYTE TO PATTERN
F559 39                         RTS                     ; .    RETURN TO CALLER
                        ;
                        ;
                        ; DECREMENT 3 INTERVAL TIMERS (XTMR0 - XTMR2)
                        ; ==========================================
                        ;
                        ;       COMMENT(S)
                        ;       ----------
                        ;          NEEDS TO BE CALLED ONCE PER FRAME (IF USED)
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          NONE REQUIRED
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          B  = $FF
                        ;          X  = #XTMR2
                        ;
F55A = 0000                     SETDP   $00
                        ;       =====   ===
                        ;
F55A C602               D3TMR   LDB     #2              ; SET-UP FOR FIRST THREE TIMERS ONLY
F55C 2002      ^F560            BRA     DTMR0           ; .
                        ;
                        ;
                        ; DECREMENT ALL INTERVAL TIMERS (XTMR0 - XTMR5)
                        ; ============================================
                        ;
                        ;       COMMENT(S)
                        ;       ----------
                        ;          NEEDS TO BE CALLED ONCE PER FRAME (IF USED)
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          NONE REQUIRED
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          B  = $FF
                        ;          X  = #XTMR5
                        ;
F55E = 0000                     SETDP   $00
                        ;       =====   ===
                        ;
```

```
F55E C605            DECTMR  LDB    #5           ; SET-UP FOR ALL INTERVAL TIMERS
                     ;
F560 8EC82E          DTMR0   LDX    #XTMR0       ; DECREMENT TIMERS
F563 6D85            DTMR1   TST    B,X          ; .    IS TIMER ALREADY ZERO ?
F565 2702    ^F569           BEQ    DTMR2        ; .    .    SKIP TO NEXT TIMER
F567 6A85                    DEC    B,X          ; .    DECREMENT TIMER
F569 5A              DTMR2   DECB                ; .    DECREMENT TIMER COUNTER
F56A 2AF7    ^F563           BPL    DTMR1        ; .    .    ALL TIMERS DECREMENTED ?
F56C 39                      RTS                 ; .    .    RETURN TO CALLER
```

```
                              ;  38 CYCLE DELAY (X.XXXX US)
                              ;  ==========================
                              ;
                              ;        ENTRY VALUES
                              ;        ------------
                              ;           NONE REQUIRED
                              ;
                              ;        RETURN VALUES
                              ;        -------------
                              ;           B  = $FF
                              ;
F56D  = 0000                           SETDP   $00
                              ;        =====   ===
                              ;
F56D  C603                    DEL38    LDB     #3
F56F  2009         ^F57A               BRA     DEL
                              ;
                              ;
                              ;  33 CYCLE DELAY (X.XXXX US)
                              ;  ==========================
                              ;
                              ;        ENTRY VALUES
                              ;        ------------
                              ;           NONE REQUIRED
                              ;
                              ;        RETURN VALUES
                              ;        -------------
                              ;           B  = $FF
                              ;
F571  = 0000                           SETDP   $00
                              ;        =====   ===
                              ;
F571  C602                    DEL33    LDB     #2
F573  2005         ^F57A               BRA     DEL
                              ;
                              ;
                              ;  28 CYCLE DELAY (X.XXXX US)
                              ;  ==========================
                              ;
                              ;        ENTRY VALUES
                              ;        ------------
                              ;           NONE REQUIRED
                              ;
                              ;        RETURN VALUES
                              ;        -------------
                              ;           B  = $FF
                              ;
F575  = 0000                           SETDP   $00
                              ;        =====   ===
                              ;
F575  C601                    DEL28    LDB     #1
F577  2001         ^F57A               BRA     DEL
```

```
                        ;  20 CYCLE DELAY (X.XXXX US)
                        ;  ==========================
                        ;
                        ;        ENTRY VALUES
                        ;        ------------
                        ;           NONE REQUIRED
                        ;
                        ;        RETURN VALUES
                        ;        -------------
                        ;           B  = $FF
                        ;
F579  = 0000                     SETDP   $00
                        ;        =====   ===
                        ;
F579  5F                DEL20    CLRB
                        ;
                        ;
                        ;  GENERAL PURPOSE DELAY
                        ;  =====================
                        ;
                        ;        COMMENT(S)
                        ;        ----------
                        ;           MINIMUM DELAY = 20 CYCLES (X.XXXX US)
                        ;
                        ;        ENTRY VALUES
                        ;        ------------
                        ;           B  = DELAY PERIOD (SEE COMMENT(S))
                        ;
                        ;        RETURN VALUES
                        ;        -------------
                        ;           B  = $FF
                        ;
F57A  = 0000                     SETDP   $00
                        ;        =====   ===
                        ;
F57A  5A                DEL      DECB
F57B  2AFD       ^F57A           BPL     DEL
                        ;
                        ;
                        ;  13 CYCLE DELAY (X.XXXX US)
                        ;  ==========================
                        ;
                        ;        ENTRY VALUES
                        ;        ------------
                        ;           NONE REQUIRED
                        ;
F57D  = 0000                     SETDP   $00
                        ;        =====   ===
                        ;
F57D  39                DEL13    RTS
```

```
                    ;  DECODE BIT POSITION
                    ;  ===================
                    ;
                    ;       COMMENT(S)
                    ;       ----------
                    ;         NO ERROR CHECKING IS PERFORMED BY THIS ROUTINE, THEREFORE
                    ;            THE VALUE IN 'A' MUST BE BETWEEN $00 - $07
                    ;
                    ;         'A'
                    ;         REG    VALUE RETURNED IN 'A'
                    ;         ---    ---------------------
                    ;         $00    $01
                    ;         $01    $02
                    ;         $02    $04
                    ;         $03    $08
                    ;
                    ;         $04    $10
                    ;         $05    $20
                    ;         $06    $40
                    ;         $07    $80
                    ;
                    ;       ENTRY VALUES
                    ;       ------------
                    ;         A  = BIT NUMBER ($00 - $07)
                    ;
                    ;       RETURN VALUES
                    ;       -------------
                    ;         A  = SEE TABLE IN COMMENT(S) ABOVE
                    ;         X  = $F9DC (#DECTBL)
                    ;
F57E  = 0000                SETDP   $00
                    ;       =====   ===
                    ;
F57E  8EF9DC        DECBIT  LDX     #DECTBL         ; LOOK-UP BIT FROM DECODE TABLE
F581  A686                  LDA     A,X             ; .
F583  39                    RTS                     ; .
```

```
                            ; CALCULATE ABSOLUTE VALUE OF 'A' & 'B' REGISTERS
                            ; ===============================================
                            ;
                            ;       COMMENT(S)
                            ;       ----------
                            ;
                            ;          $80 WILL NOT EVALUATE CORRECTLY
                            ;
                            ;       ENTRY VALUES
                            ;       ------------
                            ;
                            ;          A  = VALUE TO BE EVALUATED
                            ;          B  = VALUE TO BE EVALUATED
                            ;
                            ;       RETURN VALUES
                            ;       -------------
                            ;
                            ;          A  = ABSOLUTE VALUE OF ENTRY 'A' VALUE
                            ;          B  = ABSOLUTE VALUE OF ENTRY 'B' VALUE
                            ;
F584  = 0000                        SETDP  $00
                            ;       =====  ===
                            ;
F584  4D                    ABSAB   TSTA                     ; FORM ABSOLUTE VALUE FOR 'A' REGISTER
F585  2A04        ^F58B             BPL    ABSB              ; .
F587  40                            NEGA                     ; .
F588  2801        ^F58B             BVC    ABSB              ; .
F58A  4A                            DECA                     ; .
                            ;
                            ;
                            ;
                            ; CALCULATE ABSOLUTE VALUE OF 'B' REGISTER
                            ; ========================================
                            ;
                            ;       COMMENT(S)
                            ;       ----------
                            ;
                            ;          $80 WILL NOT EVALUATE CORRECTLY
                            ;
                            ;       ENTRY VALUES
                            ;       ------------
                            ;
                            ;          B  = VALUE TO BE EVALUATED
                            ;
                            ;       RETURN VALUES
                            ;       -------------
                            ;
                            ;          B  = ABSOLUTE VALUE OF ENTRY 'B' VALUE
                            ;
F58B  = 0000                        SETDP  $00
                            ;       =====  ===
                            ;
F58B  5D                    ABSB    TSTB                     ; FORM ABSOLUTE VALUE FOR 'B' REGISTER
F58C  2A04        ^F592             BPL    ABSBO             ; .
F58E  50                            NEGB                     ; .
F58F  2801        ^F592             BVC    ABSBO             ; .
F591  5A                            DECB                     ; .
F592  39                    ABSBO   RTS                      ; RETURN TO CALLER
```

```
                      ;  RETURN ANGLE FOR GIVEN DELTA 'Y:X'
                      ;  ================================
                      ;
                      ;        ENTRY VALUES
                      ;        ------------
                      ;           A  = DELTA 'Y'
                      ;           B  = DELTA 'X'
                      ;           DP = $C8
                      ;
                      ;        RETURN VALUES
                      ;        -------------
                      ;           A  = ANGLE FOR DELTA 'Y:X'
                      ;           B  = ANGLE FOR DELTA 'Y:X'
                      ;
                      ;           ANGLE    = ANGLE FOR DELTA 'Y:X'
                      ;
                      ;                   :  $XX
                      ;                   :
                      ;           $XX     :
                      ;           ------   ------
                      ;                   :    $XX
                      ;                   :
                      ;               $XX :
                      ;
F593  = 00C8                   SETDP   $C8
                      ;        =====   ===
                      ;
F593  3410            CMPASS   PSHS    X                ;  SAVE ENTRY VALUES
F595  DD34                     STD     ABSY             ;  .
                      ;
F597  59                       ROLB                     ;;  SHIFT SIGNS INTO 'ANGLE'
F598  C600                     LDB     #$00             ;;  .
F59A  59                       ROLB                     ;;  .
F59B  49                       ROLA                     ;;  .
F59C  59                       ROLB                     ;;  .
F59D  58                       ASLB                     ;;  .
F59E  D736                     STB     ANGLE            ;;  .
                      ;
F5A0  DC34                     LDD     ABSY             ;;  .
F5A2  8DE0      ^F584          BSR     ABSAB            ;;  MAKE POS
F5A4  9734                     STA     ABSY             ;;  SET FLAG IF Y<X
F5A6  D134                     CMPB    ABSY             ;;  .
F5A8  2308      ^F5B2          BLS     CMPS1            ;;  .
F5AA  0C36                     INC     ANGLE            ;;  BITS 2,1,0=X SIGN,Y SIGN, AND X>Y BIT
F5AC  1E89                     EXG     A,B              ;;  SWAP X,Y ID Y<X
F5AE  2002      ^F5B2          BRA     CMPS1            ;;  SQUASH TO <10
                      ;
F5B0  44              CMPS0    LSRA                     ;;  .
F5B1  54                       LSRB                     ;;  .
                      ;
F5B2  8109            CMPS1    CMPA    #9               ;;  .
F5B4  22FA      ^F5B0          BHI     CMPS0            ;;  .
F5B6  DD34                     STD     ABSY             ;;  NOW IN RANGE
```

```
                              ;
F5B8  D636                            LDB     ANGLE           ;;  INDEXT TRANSLATOR WEIRD TABLE
F5BA  8EFC24                          LDX     #WEIRD          ;;  .
F5BD  E685                            LDB     B,X             ;;  .
                              ;
F5BF  8EFC2C                  CMPS2   LDX     #ROWTRI         ;;  DECODES ROWS FOR TRIANGLE LOOKUP
F5C2  A686                            LDA     A,X             ;;  (A HAD LARGER VALUE) GET ROW LOCATION
F5C4  9B35                            ADDA    ABSX            ;;  ADD IN SMALLER VALUE FOR COLUMN
F5C6  8B0A                            ADDA    #10             ;;  GO TO START OF FIBTBL
F5C8  C501                            BITB    #1              ;;  B HAS 'ANGLE'
F5CA  2604       ^F5D0                BNE     CMPS3           ;;  BIT WAZ SET IF ADJUST NEEDED
F5CC  EB86                            ADDB    A,X             ;;  .
F5CE  2003       ^F5D3                BRA     CMPS4           ;;  .
                              ;
F5D0  5A                      CMPS3   DECB                    ;;  IF Y<X
F5D1  E086                            SUBB    A,X             ;;  .
                              ;
F5D3  D736                    CMPS4   STB     ANGLE           ;  RETURN TO CALLER
F5D5  9636                            LDA     ANGLE           ;  .    PLACE ANSWER IN 'A', 'B' AND 'ANGLE'
F5D7  3590                            PULS    X,PC            ;  .    .
                              ;
                              ;
                              ;  CALCULATE THE COSINE OF 'A'
                              ;  ==========================
                              ;
                              ;       ENTRY VALUES
                              ;       ------------
                              ;          A  = ANGLE TO BE EVALUATED
                              ;
                              ;                  !  $XX
                              ;                  !
                              ;          $XX     !
                              ;          ------   ------
                              ;                  !    $XX
                              ;                  !
                              ;             $XX  !
                              ;
                              ;       RETURN VALUES
                              ;       -------------
                              ;          D  = COSINE OF ANGLE
                              ;                  A = VALUE
                              ;                  B = SIGN / OVERFLOW
                              ;          X  = $FC6D (#RTRICS)
                              ;
F5D9  = 0000                          SETDP   $00
                              ;       =====   ===
                              ;
F5D9  8B10                    COSINE  ADDA    #$10            ;;  GETS COSINE OF 'A' <----
                              ;
                              ;
                              ;  CALCULATE THE SINE OF 'A'
                              ;  ========================
                              ;
```

```
                       ;       ENTRY VALUES
                       ;       ------------
                       ;          A  = ANGLE TO BE EVALUATED
                       ;
                       ;                 |  $XX
                       ;                 |
                       ;          $XX    |
                       ;          ------    ------
                       ;                 |    $XX
                       ;                 |
                       ;             $XX |
                       ;
                       ;       RETURN VALUES
                       ;       -------------
                       ;          D  = SINE OF ANGLE
                       ;                  A = VALUE
                       ;                  B = SIGN / OVERFLOW
                       ;          X  = $FC6D (#RTRIGS)
                       ;
F5DB  = 0000                   SETDP   $00
                       ;       =====   ===
                       ;
F5DB  8EFC6D           SINE    LDX     #RTRIGS         ;;  GETS SINE OF 'A' <---- 26-30 CYCLES
F5DE  5F                       CLRB
F5DF  8520                     BITA    #$20            ;;  TEST FOR NEG
F5E1  2702      ^F5E5          BEQ     SIN0            ;;  .
F5E3  C680                     LDB     #$80            ;;  .
F5E5  841F             SIN0    ANDA    #$1F            ;;  .
F5E7  8110                     CMPA    #$10            ;;  TEST FOR 90 DEG
F5E9  2601      ^F5EC          BNE     SIN1            ;;  .
F5EB  5C                       INCB                    ;;  INDICATE OVERFLOW
F5EC  A686             SIN1    LDA     A,X             ;;  .
F5EE  39                       RTS                     ;;  ANSWER IN D REG, A=VALUE,B=SIGN/OVERFLOW
                       ;
                       ;
                       ; CALCULATE THE SINE AND COSINE OF 'ANGLE'
                       ; ========================================
                       ;
                       ;       ENTRY VALUES
                       ;       ------------
                       ;          A  = .
                       ;          B  = .
                       ;          DP = $C8
                       ;
                       ;                 |  $XX
                       ;                 |
                       ;          $XX    |
                       ;          ------    ------
                       ;                 |    $XX
                       ;                 |
                       ;             $XX |
                       ;
                       ;       RETURN VALUES
```

```
                        ;       ------------
                        ;          A  = .
                        ;          B  = ,
                        ;
F5EF  = 00C8                    SETDP   $C8
                        ;       =====   ===
                        ;
F5EF  3410              SINCOS  PSHS    X               ;; GETS SINE,COSINE OF 'ANGLE' <----
F5F1  9636                      LDA     ANGLE           ;; .
F5F3  8DE6      ^F5DB           BSR     SINE            ;; .
F5F5  DD37                      STD     WSINE           ;; .
F5F7  9636                      LDA     ANGLE           ;; .
F5F9  8DDE      ^F5D9           BSR     COSINE          ;; .
F5FB  DD39                      STD     WCSINE          ;; .
F5FD  3590                      PULS    X,PC            ;; .
                        ;
                        ;
                        ;  ROTATE A SINGLE LINE
                        ;  ====================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          A  = INITIAL 'X' VALUE
                        ;          B  = ANGLE OF ROTATION
                        ;          DP = $C8
                        ;
                        ;               !  $XX
                        ;               !
                        ;          $XX  !
                        ;          ------  ------
                        ;               !    $XX
                        ;               !
                        ;            $XX !
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          A  = ROTATED 'Y' VECTOR VALUE
                        ;          B  = ROTATED 'X' VECTOR VALUE
                        ;
F5FF  = 00C8                    SETDP   $C8
                        ;       =====   ===
                        ;
F5FF  C010              LROT90  SUBB    #$10            ; SUBTRACT 90 DEGREES
                        ;
                        ;
                        ;  ROTATE A SINGLE LINE
                        ;  ====================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          A  = INITIAL 'Y' VALUE
                        ;          B  = ANGLE OF ROTATION
                        ;          DP = $C8
                        ;
                        ;                 ¦  $XX
                        ;                 ¦
                        ;          $XX    ¦
                        ;          ------    ------
                        ;                 ¦    $XX
                        ;                 ¦
                        ;             $XX ¦
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          A  = ROTATED 'Y' VECTOR VALUE
                        ;          B  = ROTATED 'X' VECTOR VALUE
                        ;
F601  = 00C8                    SETDP   $C8
                        ;       =====   ===
                        ;
F601  D736              LNROT   STB     ANGLE           ; SET-UP FOR ROTATION
                        ;
                        ;
                        ;  ROTATE A SINGLE LINE ('ANGLE' IS SET)
                        ;  =====================================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          A  = INITIAL 'Y' VALUE
                        ;          DP = $C8
                        ;
                        ;                 ¦  $XX
                        ;                 ¦
                        ;          $XX    ¦
                        ;          ------    ------
                        ;                 ¦    $XX
                        ;                 ¦
                        ;             $XX ¦
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          A  = ROTATED 'Y' VECTOR VALUE
                        ;          B  = ROTATED 'X' VECTOR VALUE
                        ;
F603  = 00C8                    SETDP   $C8
                        ;       =====   ===
                        ;
F603  973B              ALNROT  STA     LEG             ;; A=Y VALUE, X ASSUMED=0 (----
F605  8DE8      ^F5EF           BSR     SINCOS          ;;  .
F607  8D54      ^F65D           BSR     LSINE           ;;  .
F609  40                        NEGA                    ;;  .
F60A  3402                      PSHS    A               ;;  .
F60C  8D55      ^F663           BSR     LCSINE          ;;  .
F60E  3584                      PULS    B,PC            ;; Y,X COORDS IN A,B REGS AT EXIT
                        ;
                        ;
```

```
                         ;  'DIFFY' STYLE LIST ROTATION (VALUES IN REGISTERS)
                         ;  ==================================================
                         ;
                         ;        ENTRY VALUES
                         ;        ------------
                         ;           A  = ROTATION ANGLE
                         ;           B  = NUMBER OF VECTORS - 1
                         ;           X  = POINTER TO 'DIFFY' LIST
                         ;           U  = POINTER TO DESTINATION BUFFER
                         ;
                         ;                   !  $XX
                         ;                   !
                         ;           $XX     !
                         ;           ------   ------
                         ;                   !    $XX
                         ;                   !
                         ;              $XX  !
                         ;
                         ;        RETURN VALUES
                         ;        -------------
                         ;           A  = .
                         ;           B  = .
                         ;           X  = .
                         ;           U  = .
                         ;
F610  = 0000                      SETDP    $00
                         ;        =====    ===
                         ;
F610  B7C836             DROT     STA      ANGLE           ; SAVE 'ANGLE'
                         ;
                         ;
                         ;  'DIFFY' STYLE LIST ROTATION ('ANGLE' IS SET)
                         ;  =============================================
                         ;
                         ;        ENTRY VALUES
                         ;        ------------
                         ;           B  = NUMBER OF VECTORS - 1
                         ;           X  = POINTER TO 'DIFFY' LIST
                         ;           U  = POINTER TO DESTINATION BUFFER
                         ;
                         ;                   !  $XX
                         ;                   !
                         ;           $XX     !
                         ;           ------   ------
                         ;                   !    $XX
                         ;                   !
                         ;              $XX  !
                         ;
                         ;        RETURN VALUES
                         ;        -------------
                         ;           A  = .
                         ;           B  = .
                         ;           X  = .
```

```
                        ;          U  = .
                        ;
F613  = 0000                    SETDP   $00
                        ;       =====   ===
                        ;
F613  F7C823            BDROT   STB     LIST            ; SET 'LIST'
                        ;
                        ;
                        ;  'DIFFY' STYLE LIST ROTATION ('ANGLE' AND 'LIST' SET)
                        ;  ====================================================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          X  = POINTER TO 'DIFFY' LIST
                        ;          U  = POINTER TO DESTINATION BUFFER
                        ;
                        ;                 !  $XX
                        ;                 !
                        ;          $XX    !
                        ;          ------   ------
                        ;                 !    $XX
                        ;                 !
                        ;             $XX !
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          A  = .
                        ;          B  = .
                        ;          X  = .
                        ;          U  = .
                        ;
F616  = 0000                    SETDP   $00
                        ;       =====   ===
                        ;
F616  3408              ADROT   PSHS    DP               ; SET-UP FOR 'DIFFY' ROTATION
F618  BDF1AF                    JSR     DPRAM            ; .    SET DP = RAM
F61B  = 00C8                    SETDP   $C8              ; .    .
F61B  8DD2      ^F5EF           BSR     SINCOS           ; .
F61D  2018      ^F637           BRA     APRT2            ; .
                        ;
                        ;
                        ;  'PACKET' STYLE LIST ROTATION
                        ;  ============================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          A  = ROTATION ANGLE
                        ;          X  = POINTER TO 'PACKET' LIST
                        ;          U  = POINTER TO DESTINATION BUFFER
                        ;
                        ;                 !  $XX
                        ;                 !
                        ;          $XX    !
                        ;           ------   ------
                        ;                  ¦    $XX
                        ;                  ¦
                        ;              $XX ¦
                        ;
                        ;        RETURN VALUES
                        ;        -------------
                        ;           A  = .
                        ;           B  = .
                        ;           X  = .
                        ;           U  = .
                        ;
F61F  = 0000                     SETDP   $00
                        ;        =====   ===
                        ;
F61F  B7C836            PROT     STA     ANGLE            ; SET 'ANGLE'
                        ;
                        ;
                        ;  'PACKET' STYLE LIST ROTATION ('ANGLE' SET)
                        ;  ==========================================
                        ;
                        ;        ENTRY VALUES
                        ;        ------------
                        ;           X  = POINTER TO 'PACKET' LIST
                        ;           U  = POINTER TO DESTINATION BUFFER
                        ;
                        ;                  ¦  $XX
                        ;                  ¦
                        ;           $XX    ¦
                        ;           ------   ------
                        ;                  ¦    $XX
                        ;                  ¦
                        ;              $XX ¦
                        ;
                        ;        RETURN VALUES
                        ;        -------------
                        ;           A  = .
                        ;           B  = .
                        ;           X  = .
                        ;           U  = .
                        ;
F622  = 0000                     SETDP   $00
                        ;        =====   ===
                        ;
F622  3408              APROT    PSHS    DP               ; SET DP = RAM
F624  BDF1AF                     JSR     DPRAM            ; .
F627  = 00C8                     SETDP   $C8              ; .
F627  9723                       STA     LIST             ; .    SET 'LIST' TO MINUS FOR 'PACKETS'
                        ;
F629  8DC4       ^F5EF           BSR     SINCOS           ; CALCULATE SINE / COSINE FOR GIVEN ANGLE
                        ;
F62B  A680              APRTO    LDA     X+               ; FETCH PACKET CONTROL BYTES
F62D  A7C0                       STA     U+               ; .    SAVE IN DESTINATION BUFFER
```

```
F62F  2F06      ^F637           BLE     APRT2           ; .    PACKET TERMINATOR ?
                        ;
F631  0F23                      CLR     LIST            ; END OF PACKET - RETURN TO CALLER
F633  3588                      PULS    DP,PC           ; .
                        ;
                        ;
                        ;
F635  0A23              APRT1   DEC     LIST            ; DECREMENT 'LIST' COUNTER
                        ;
F637  A680              APRT2   LDA     X+              ;; NEXT LINE SPEC Y,X
F639  8D26      ^F661           BSR     MCSINE          ;; .
F63B  A7C4                      STA     ,U              ;; 1 BYTE FOR INTERMEDIATE RESULTS
                        ;
F63D  A684                      LDA     ,X              ;; .
F63F  8D1A      ^F65B           BSR     MSINE           ;; .
F641  ABC4                      ADDA    ,U              ;; RESULT IS NEW Y VAL
F643  A7C0                      STA     U+              ;; NEW Y=YCOS(ANGLE) + XSIN(ANGLE)
                        ;
F645  A61F                      LDA     -1,X            ;; .
F647  8D12      ^F65B           BSR     MSINE           ;; .
F649  A7C4                      STA     ,U              ;; .
                        ;
F64B  A680                      LDA     X+              ;; LEAVES X AND U POINT TO NEXT VECTOR
F64D  8D12      ^F661           BSR     MCSINE          ;; .
F64F  A0C4                      SUBA    ,U              ;; .
F651  A7C0                      STA     U+              ;; NEW X=XCOS(ANGLE) - YSIN(ANGLE)
                        ;
F653  9623                      LDA     LIST            ; 'PACKET' OR 'DIFFY' / 'DUFFY' FORMATS ?
F655  2BD4      ^F62B           BMI     APRT0           ; .    'PACKET' FORMAT ?
F657  26DC      ^F635           BNE     APRT1           ; .    DONE WITH 'DIFFY' / 'DUFFY' ?
F659  3588                      PULS    DP,PC           ; .    .    END OF 'DIFFY' - RETURN TO CALLER
```

```
                          ;  MULTIPLY 'A' BY PREVIOUS SINE VALUE
                          ;  ==================================
                          ;
                          ;       ENTRY VALUES
                          ;       ------------
                          ;          A  = .
                          ;          B  = .
                          ;          DP = $C8
                          ;
                          ;       RETURN VALUES
                          ;       -------------
                          ;          A  = .
                          ;          B  = .
                          ;
F65B  = 00C8                      SETDP   $C8
                          ;       =====   ===
                          ;
F65B  973B                MSINE   STA     LEG              ;  SAVE 'A' REGISTER
                          ;
                          ;
                          ;
                          ;
                          ;
                          ;  MULTIPLY 'LEG' BY PREVIOUS SINE VALUE
                          ;  ====================================
                          ;
                          ;       ENTRY VALUES
                          ;       ------------
                          ;          A  = .
                          ;          B  = .
                          ;          DP = $C8
                          ;
                          ;       RETURN VALUES
                          ;       -------------
                          ;          A  = .
                          ;          B  = .
                          ;
F65D  = 00C8                      SETDP   $C8
                          ;       =====   ===
                          ;
F65D  DC37                LSINE   LDD     WSINE            ;  FETCH PREVIOUS SINE VALUE
F65F  2004       ^F665            BRA     LCOSO            ;  .
```

```
                        ;  MULTIPLY 'A' BY PREVIOUS COSINE
                        ;  =============================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;         A  = .
                        ;         B  = .
                        ;         DP = $C8
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;         A  = .
                        ;         B  = .
                        ;
F661  = 00C8                    SETDP   $C8
                        ;       =====   ===
                        ;
F661  973B              MCSINE  STA     LEG             ;  SAVE 'A' REGISTER
                        ;
                        ;
                        ;  MULTIPLY 'LEG' BY PREVIOUS COSINE
                        ;  ================================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;         A  = .
                        ;         B  = .
                        ;         DP = $C8
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;         A  = .
                        ;         B  = .
                        ;
F663  = 00C8                    SETDP   $C8
                        ;       =====   ===
                        ;
F663  DC39              LCSINE  LDD     WCSINE          ;  FETCH PREVIOUS COSINE VALUE
F665  D73C              LCOS0   STB     LAC             ;;  HOLDS SIGN,OVERFLOW BITS 7,0
                                                        ;;  WARNING NO INPUT=$80 ALLOWED
F667  C501                      BITB    #1              ;;  SHOULD MULT?
F669  2704      ^F66F           BEQ     LCOS1           ;;  NO, FLIP ONLY
F66B  963B                      LDA     LEG             ;;  .
F66D  200A      ^F679           BRA     LCOS3           ;;  .
                        ;
F66F  D63B              LCOS1   LDB     LEG             ;;  LEG IS INPUT VECTOR
F671  2A03      ^F676           BPL     LCOS2           ;;  .
F673  033C                      COM     LAC             ;;  TWO WRONGS MAKE A RITE
F675  50                        NEGB                    ;;  .
F676  3D                LCOS2   MUL                     ;;  .
F677  8900                      ADCA    #0              ;;  .
F679  D63C              LCOS3   LDB     LAC             ;;  CHECK SIGN FLIP
F67B  2A01      ^F67E           BPL     LCOS4           ;;  .
```

```
F67D  40                     NEGA                       ;;  .
F67E  39             LC064   RTS                        ;;  ANSWER IN A REG
```

```
                        ;  TRANSFER 'A'+1 BYTES FROM SOURCE TO DESTINATION BUFFER
                        ;  ==================================================
                        ;
                        ;       COMMENT(S)
                        ;       ----------
                        ;          NO MORE THAN $80 BYTES MAY BE MOVED WITH THIS ROUTINE
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          A  = (NUMBER OF BYTES -1) TO BE TRANSFERED
                        ;          X  = DESTINATION BUFFER POINTER
                        ;          U  = SOURCE BUFFER POINTER
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          A  = $FF
                        ;          B  = CONTENTS OF LAST BYTE TRANSFERED
                        ;
F67F  = 0000                    SETDP   $00
                        ;       =====   ===
                        ;
F67F  E4C6              BLKMV1  LDB     A,U             ;  PICK-UP SOURCE BUFFER BYTE
F681  E786                      STB     A,X             ;  PLACE IN DESTINATION BUFFER
                        ;
                        ;
                        ;  TRANSFER 'A' BYTES FROM SOURCE TO DESTINATION BUFFER
                        ;  ================================================
                        ;
                        ;       COMMENT(S)
                        ;       ----------
                        ;          NO MORE THAN $7F BYTES MAY BE MOVED WITH THIS ROUTINE
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          A  = NUMBER OF BYTES TO BE TRANSFERED
                        ;          X  = DESTINATION BUFFER POINTER
                        ;          U  = SOURCE BUFFER POINTER
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          A  = $FF
                        ;          B  = CONTENTS OF LAST BYTE TRANSFERED
                        ;
F683  = 0000                    SETDP   $00
                        ;       =====   ===
                        ;
F683  4A                BLKMOV  DECA                    ;  DECREMENT BYTE COUNTER
F684  2AF9      ^F67F           BPL     BLKMV1          ;  .    IF >0, THEN MOVE ANOTHER BYTE
F686  39                BLKMO   RTS                     ;  .    RETURN TO CALLER
```

```
                         ;  SET 'REQX' FOR GIVEN TUNE
                         ;  =========================
                         ;
                         ;        ENTRY VALUES
                         ;        ------------
                         ;
                         ;           X  = .
                         ;           U  = POINTER TO TUNE LIST
                         ;           DP = $C8
                         ;
                         ;        RETURN VALUES
                         ;        -------------
                         ;
                         ;           A  = .
                         ;           B  = .
                         ;           X  = .
                         ;           U  = .
                         ;
F687  = 00C8                      SETDP   $C8
                         ;        =====   ===
                         ;
F687  9656               REPLAY   LDA     TSTAT           ;;  TO START, PUT $01 IN TSTAT
F689  2B28       ^F6B3            BMI     TPLY0           ;;  TSTAT= $80 DURING TUNE, =0 AT IDLE
F68B  27F9       ^F686            BEQ     BLKMO           ;;  EXIT IF NOT GOING
                         ;
                         ;
                         ;  SET TUNE SEQUENCE
                         ;  =================
                         ;
                         ;        ENTRY VALUES
                         ;        ------------
                         ;
                         ;           A  = .
                         ;           B  = .
                         ;           X  = .
                         ;           U  = POINTER TO TUNE LIST
                         ;           DP = $C8
                         ;
                         ;        RETURN VALUES
                         ;        -------------
                         ;
                         ;           A  = .
                         ;           B  = .
                         ;           X  = .
                         ;           U  = .
                         ;
F68D  = 00C8                      SETDP   $C8
                         ;        =====   ===
                         ;
F68D  8EFC8D             SPLAY    LDX     #NOTES          ;  SET-UP FOR STANDARD NOTES
```

```
                      ;  SET TUNE SEQUENCE (ALTERNATE NOTES)
                      ;  =================================
                      ;
                      ;        ENTRY VALUES
                      ;        ------------
                      ;           A  = .
                      ;           B  = .
                      ;           X  = POINTER TO ALTERNATE NOTE TABLE
                      ;           U  = POINTER TO TUNE LIST
                      ;           DP = $C8
                      ;
                      ;        RETURN VALUES
                      ;        -------------
                      ;           A  = .
                      ;           B  = .
                      ;           X  = .
                      ;           U  = .
                      ;
F690  = 00C8                   SETDP   $C8
                      ;        =====   ===
                      ;
F690  9F4D            ASPLAY   STX     DOREMI          ;  SAVE DESIRED NOTE TABLE POINTER
                      ;
                      ;
                      ;  INITIALIZE TUNE SEQUENCE
                      ;  =======================
                      ;
                      ;        ENTRY VALUES
                      ;        ------------
                      ;           A  = .
                      ;           B  = .
                      ;           X  = .
                      ;           U  = POINTER TO TUNE LIST
                      ;           DP = $C8
                      ;
                      ;        RETURN VALUES
                      ;        -------------
                      ;           A  = .
                      ;           B  = .
                      ;           X  = .
                      ;           U  = .
                      ;
F692  = 00C8                   SETDP   $C8
                      ;        =====   ===
                      ;
F692  8680            TPLAY    LDA     #$80            ;  INDICATED THAT TUNE IS STILL PENDING
F694  9756                     STA     TSTAT           ;  .
```

```
F696  ECC1                       LDD    U++            ;;  IF NEW, U SHOULD POINT TO TUNE
F698  DD4F                       STD    FADE           ;;  .
                         ;
F69A  ECC1                       LDD    U++            ;;  .
F69C  DD51                       STD    VIBE           ;;  .
                         ;
F69E  DF53                       STU    TUNE           ;;  .
                         ;
F6A0  BDF533                     JSR    INTREQ         ;;  CLEAR 13 REGS, SET REG6=$3F
                         ;
F6A3  CC1F1F                     LDD    #$1F1F         ;;  .
F6A6  DD5F                       STD    FADEB          ;;  .
                         ;
F6A8  CC0000                     LDD    #$0000         ;;  .
F6AB  DD63                       STD    TONEB          ;;  .
F6AD  DD65                       STD    TONEC          ;;  .
F6AF  9755                       STA    NEWGEN         ;;  .
F6B1  2039         ^F6EC         BRA    TPLY7          ;;  .
                         ;
                         ;
F6B3  CEC85E             TPLY0   LDU    #FADEA         ;;  REFRESH TONE AMPLITUDES
F6B6  C602                       LDB    #2             ;;  .
F6B8  A6C5               TPLY1   LDA    B,U            ;;  .
F6BA  811F                       CMPA   #$1F           ;;  LIMIT AT $1F
F6BC  2702         ^F6C0         BEQ    TPLY2          ;;  .
F6BE  6CC5                       INC    B,U            ;;  .
F6C0  5A                 TPLY2   DECB                  ;;  .
F6C1  2AF5         ^F6B8         BPL    TPLY1          ;;  .
                         ;
F6C3  9E51                       LDX    VIBE           ;;  PLOGLAMMABLE VIBLATO
F6C5  CEC858                     LDU    #RATEA         ;;  TABLE INDIRECT 'VIBE'
F6C8  8607                       LDA    #7             ;;  VARIED RATES
F6CA  6CC4               TPLY3   INC    ,U             ;;  .
F6CC  A1C4                       CMPA   ,U             ;;  MOD COUNTERS
F6CE  2C02         ^F6D2         BGE    TPLY4          ;;  .
F6D0  6FC4                       CLR    ,U             ;;  .
F6D2  E6C0               TPLY4   LDB    U+             ;;  DECODE TABLE BY TUNE
F6D4  C407                       ANDB   #7             ;;  .
F6D6  E685                       LDB    B,X            ;;  .
F6D8  E7C0                       STB    U+             ;;  .
F6DA  4C                         INCA                  ;;  .
F6DB  8109                       CMPA   #9             ;;  .
F6DD  23EB         ^F6CA         BLS    TPLY3          ;;  .
F6DF  0A57                       DEC    RESTC          ;;  CHECK REST
F6E1  266B         ^F74E         BNE    XPLY1          ;;  BRANCH IF NOT TIMED OUT
                         ;
F6E3  9655               TPLY5   LDA    NEWGEN         ;;  CYCLE TO NEXT TONE GEN
F6E5  4A                         DECA                  ;;  .
F6E6  2A02         ^F6EA         BPL    TPLY6          ;;  .
F6E8  8602                       LDA    #2             ;;  .
F6EA  9755               TPLY6   STA    NEWGEN         ;;  .
F6EC  E69FC853           TPLY7   LDB    [TUNE]         ;;  PICK UP NEW NOTE
F6F0  CEC85E                     LDU    #FADEA         ;;  .
```

```
F6F3  6FC6                        CLR    A,U            ;;  .
F6F5  C540                        BITB   #$40           ;;  .
F6F7  2719      ^F712             BEQ    TPLY8          ;;  .
                           ;
F6F9  8EF9E4                      LDX    #CRT           ;;  NOISE IS ON FOR THIS NOTE
F6FC  A686                        LDA    A,X            ;;  .
F6FE  9445                        ANDA   REQ6           ;;  .
F700  9745                        STA    REQ6           ;;  .
                           ;
F702  9655                        LDA    NEWGEN         ;;  .
F704  8B03                        ADDA   #3             ;;  .
F706  A686                        LDA    A,X            ;;  .
F708  9A45                        ORA    REQ6           ;;  .
F70A  9745                        STA    REQ6           ;;  .
                           ;
F70C  C41F                        ANDB   #$1F           ;;  .
F70E  D746                        STB    REQ7           ;;  .
F710  2023      ^F735             BRA    TPLY9          ;;  .
                           ;
                           ;
F712  8EF9EA               TPLY8  LDX    #CRATE         ;;  .
F715  A686                        LDA    A,X            ;;  .
F717  9445                        ANDA   REQ6           ;;  .
F719  9745                        STA    REQ6           ;;  .
                           ;
F71B  9655                        LDA    NEWGEN         ;;  .
F71D  8B03                        ADDA   #3             ;;  .
F71F  A686                        LDA    A,X            ;;  .
F721  9A45                        ORA    REQ6           ;;  .
F723  9745                        STA    REQ6           ;;  .
                           ;
F725  9655                        LDA    NEWGEN         ;;  .
F727  48                          ASLA                  ;;  INDEX FOR NOTE LOC
F728  8B03                        ADDA   #3             ;;  POINT TO NOTES
F72A  33C6                        LEAU   A,U            ;;  U NOW HAS PLACE TO DUMP NOTE
F72C  C43F                        ANDB   #$3F           ;;  STRIP OFF FLAG
F72E  58                          ASLB                  ;;  .
                           ;
F72F  9E4D                        LDX    DOREMI         ;;  .
F731  EC85                        LDD    B,X            ;;  .
F733  EDC4                        STD    ,U             ;;  STORE NOTE VALUE
                           ;
F735  9E53                 TPLY9  LDX    TUNE           ;;  ANOTHER NOTE?
F737  E680                        LDB    X+             ;;  .
F739  9F53                        STX    TUNE           ;;  .
F73B  5D                          TSTB                  ;;  .
F73C  2BA5      ^F6E3             BMI    TPLY5          ;;  YES
F73E  E680                        LDB    X+             ;;  .
F740  2A06      ^F748             BPL    XPLY0          ;;  .
                           ;
                           ;
                           ;  TERMINATE CURRENT TUNE
                           ;  ======================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          A  = .
                        ;          B  = .
                        ;          X  = .
                        ;          Y  = .
                        ;          U  = .
                        ;          DP = $C8
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          A  = .
                        ;          B  = .
                        ;          X  = .
                        ;          Y  = .
                        ;          U  = .
                        ;
F742  = 00C8                    SETDP   $C8
                        ;       =====   ===
                        ;
F742  BDF533            XPLAY   JSR     INTREQ          ;;  TO STOP TUNE BEFORE END <----
F745  0F56                      CLR     TSTAT           ;;  PREPARE TO END
F747  39                        RTS                     ;;  CC SET FOR QUICK TEST
                        ;
                        ;
                        ;
F748  9F53              XPLY0   STX     TUNE            ;;  SET UP WAIT
F74A  C43F                      ANDB    #$3F            ;;  .
F74C  D757                      STB     RESTC           ;;  .
F74E  109E4F            XPLY1   LDY     FADE            ;;  GETS ADDR OF FADEOUT TABLE
F751  CEC85E                    LDU     #FADEA          ;;  FADE SPEC TONE GEN A
F754  8EC842                    LDX     #REQ3           ;;  .
F757  8602                      LDA     #2              ;;  .
F759  E6C0              XPLY2   LDB     U+              ;;  .
F75B  C501                      BITB    #1              ;;  ODD OR EVEN?
F75D  2707        ^F766         BEQ     XPLY3           ;;  .
F75F  54                        LSRB                    ;;  DIV/2
F760  E6A5                      LDB     B,Y             ;;  .
F762  C40F                      ANDB    #$F             ;;  .
F764  2007        ^F76D         BRA     XPLY4           ;;  .
                        ;
F766  54                XPLY3   LSRB                    ;;  .
F767  E6A5                      LDB     B,Y             ;;  GET IGH BYTE
F769  54                        LSRB                    ;;  .
F76A  54                        LSRB                    ;;  .
F76B  54                        LSRB                    ;;  .
F76C  54                        LSRB                    ;;  .
F76D  E786              XPLY4   STB     A,X             ;;  .
F76F  4A                        DECA                    ;;  .
F770  2AE7        ^F759         BPL     XPLY2           ;;  .
F772  CEC867                    LDU     #TONEC+2        ;;  VIBRATO SECTION HERE
F775  8EC847                    LDX     #REQ8           ;;  .
```

```
F778  ECC3            XPLY5    LDD     --U                ;;  .
F77A  6D58                     TST     -8,U               ;;  USE ONLY VIRC FOR FIRST
F77C  2A0A      ^F788          BPL     XPLY6              ;;  .
F77E  6058                     NEG     -8,U               ;;  NEG, SHOULD SUBTRACT
F780  E058                     SUBB    -8,U               ;;  .
F782  8200                     SBCA    #0                 ;;  .
F784  6058                     NEG     -8,U               ;;  .
F786  2004      ^F78C          BRA     XPLY7              ;;  .
                        ;
F788  EB58            XPLY6    ADDB    -8,U               ;;  .
F78A  8900                     ADCA    #0                 ;;  .
F78C  ED81            XPLY7    STD     X++                ;;  .
F78E  8CC84D                   CMPX    #REQC+2            ;;  .
F791  26E5      ^F778          BNE     XPLY5              ;;  .
F793  39              XPLY8    RTS                        ;;  .
                        ;
                        ;
                        ;  FETCH GAME OPTIONS
                        ;  ==================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;
                        ;          A  = .
                        ;          B  = .
                        ;          X  = .
                        ;          Y  = .
                        ;          U  = .
                        ;          DP = $C8
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;
                        ;          A  = .
                        ;          B  = .
                        ;          X  = .
                        ;          Y  = .
                        ;          U  = .
                        ;          DP = .
                        ;
F794  20C040C0504C4159  MPLAY    DB      $20,$C0,$40,$C0,'PLAYER',$80       ;  PLAYER OPTION MESSAGE
                        ;
F79F  E0C001C02047414D  MGAME    DB      $E0,$C0,$01,$C0,' GAME',$80        ;  GAME OPTION MESSAGE
                        ;
                        ;
F7A9  = 0000                   SETDP   $00
                        ;      =====   ===
                        ;
F7A9  FDC84F          SELOPT   STD     FADE               ;;  DISPLAYS AND SETS PLAYER/GAME OPTIONS <----
F7AC  4D                       TSTA                       ;;  .
F7AD  2702      ^F7B1          BEQ     OPTN0              ;;  .
F7AF  8601                     LDA     #1                 ;;  .
F7B1  5D              OPTN0    TSTB                       ;;  .
F7B2  2702      ^F7B6          BEQ     OPTN1              ;;  .
F7B4  C601                     LDB     #1                 ;;  .
```

```
F7B6  FDC879           OPTN1    STD     PLAYRS          ;;  .
F7B9  BDF1AF                    JSR     DPRAM           ;  SET DP = RAM
F7BC  = 00C8                    SETDP   $C8             ;  .
F7BC  CCF850                    LDD     #$F850          ;;  .
F7BF  DD2A                      STD     SIZRAS          ;;  .
F7C1  973C                      STA     LAG             ;;  TO NOT GLITCH ON EDGE
F7C3  2067        ^F82C         BRA     OPTN8           ;;  .
                       ;
F7C5  BDF192           OPTN2    JSR     FRWAIT          ;;  ENTRY VALUES: A=MAX PLAYERS, 0=DON'T SHOW
F7C8  = 00D0                    SETDP   $D0             ;  .    SET DP = I/O
F7C8  4F                        CLRA                    ;;  .
F7C9  BDF1B4                    JSR     DRNCE           ;;  B=MAX GAMES,  "  "      "
F7CC  BDF55A                    JSR     D3TMR           ;;  .
F7CF  BDF2A9                    JSR     INTMAX          ;;  .
F7D2  = 00D0                    SETDP   $D0             ;;  .
F7D2  B6C879                    LDA     PLAYRS          ;;  .
F7D5  108EF794                  LDY     #MPLAY          ;;  PLAYER MESSAGE
F7D9  8D5A        ^F835         BSR     OPTN9           ;;  .
F7DB  B6C87A                    LDA     OPTION          ;;  .
F7DE  108EF79F                  LDY     #MGAME          ;;  .
F7E2  8D51        ^F835         BSR     OPTN9           ;;  .
F7E4  = 00C8                    SETDP   $C8             ;;  .
F7E4  BDF1AF                    JSR     DPRAM           ;  SET DP= RAM
F7E7  = 00C8                    SETDP   $C8             ;  .
F7E7  963C                      LDA     LAG             ;;  WAIT FOR KEY RELEASE
F7E9  2706        ^F7F1         BEQ     OPTN3           ;;  .
F7EB  960F                      LDA     TRIGGR          ;;  .
F7ED  263D        ^F82C         BNE     OPTN8           ;;  .
F7EF  0F3C                      CLR     LAG             ;;  .
F7F1  962F             OPTN3    LDA     XTMR1           ;;  .
F7F3  279E        ^F793         BEQ     XPLY8           ;;  TIMEOUT
F7F5  962E                      LDA     XTMR0           ;;  KEY0 PLAYER SEL
F7F7  26CC        ^F7C5         BNE     OPTN2           ;;  .
F7F9  9615                      LDA     KEY3            ;;  .
F7FB  2696        ^F793         BNE     XPLY8           ;;  START KEY
F7FD  9612                      LDA     KEY0            ;;  .
F7FF  270F        ^F810         BEQ     OPTN5           ;;  .
F801  9679                      LDA     PLAYRS          ;;  .
F803  270B        ^F810         BEQ     OPTN5           ;;  .
F805  4C                        INCA                    ;;  .
F806  914F                      CMPA    FADE            ;;  LIMIT
F808  2302        ^F80C         BLS     OPTN4           ;;  .
F80A  8601                      LDA     #1              ;;  .
F80C  9779             OPTN4    STA     PLAYRS          ;;  .
F80E  201C        ^F82C         BRA     OPTN8           ;;  .
                       ;
F810  967A             OPTN5    LDA     OPTION          ;;  .
F812  27B1        ^F7C5         BEQ     OPTN2           ;;  .
F814  D613                      LDB     KEY1            ;;  KEY1 GAME UP
F816  2709        ^F821         BEQ     OPTN6           ;;  .
F818  4C                        INCA                    ;;  .
F819  9150                      CMPA    FADE+1          ;;  .
F81B  230D        ^F82A         BLS     OPTN7           ;;  .
```

```
F81D  8601                          LDA     #1              ;;  .
F81F  2009       ^F82A              BRA     OPTN7           ;;  .
                          ;
F821  D614                OPTN6     LDB     KEY2            ;;  KEY2 GAME DOWN
F823  27A0       ^F7C5              BEQ     OPTN2           ;;  .
F825  4A                            DECA                    ;;  .
F826  2602       ^F82A              BNE     OPTN7           ;;  .
F828  9650                          LDA     FADE+1          ;;  WRAP MINUS
F82A  977A                OPTN7     STA     OPTION          ;;  .
F82C  86F3                OPTN8     LDA     #$F3            ;;  .
F82E  972F                          STA     XTMR1           ;;  TIMEOUT PARAM
F830  43                            COMA                    ;;  .
F831  972E                          STA     XTMR0           ;;  .
F833  2090       ^F7C5              BRA     OPTN2           ;;  .
                          ;
F835  8EC85E              OPTN9     LDX     #FADEA          ;;  SHOWS NUMBER AND MESSAGE FOR 'OPTION' ABOVE <----
F838  3402                          PSHS    A               ;;  .
F83A  8D13       ^F84F              BSR     SCLR            ;;  .
F83C  A6E0                          LDA     S+              ;;  TO SET CC
F83E  270E       ^F84E              BEQ     OPTNA           ;;  .
F840  8D1C       ^F85E              BSR     BYTADD          ;;  SHOW NUMBER
F842  1F13                          TFR     X,U             ;;  .
F844  ECA1                          LDD     Y++             ;;  .
F846  BDF37A                        JSR     MSSPOS          ;;  .
F849  1F23                          TFR     Y,U             ;;  Y HELD NUMBER,MESSAGE LOC, AND MESSAGE
F84B  BDF378                        JSR     RSTPOS          ;;  .
F84E  39                  OPTNA     RTS                     ;;  .
                          ;
                          ;
                          ;  CLEAR INDICATED SCORE
                          ;  =====================
                          ;
                          ;         ENTRY VALUES
                          ;         ------------
                          ;
                          ;            X  = POINTS TO SCORE FIELD TO BE CLEARED
                          ;
                          ;         RETURN VALUES
                          ;         -------------
                          ;
                          ;            D  = $3080
                          ;
F84F  = 0000                        SETDP   $00
                          ;         =====   ===
                          ;
F84F  CC2020              SCLR      LDD     #$2020          ;  CLEAR FIRST 5 BYTES
F852  ED84                          STD     ,X              ;  .    SET TO SPACES
F854  ED02                          STD     2,X             ;  .    .
F856  A704                          STA     4,X             ;  .    .
F858  CC3080                        LDD     #$3080          ;  SET BYTE #5 = $30 (ASCII '0')
F85B  ED05                          STD     5,X             ;  SET BYTE #6 = $80 (TERMINATOR)
F85D  39                            RTS                     ;  RETURN TO CALLER
                          ;
                          ;
                          ;  ADD CONTENTS OF 'A' TO INDICATED SCORE
```

```
                        ; =====================================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          A  = 2 DIGIT BCD NUMBER
                        ;          X  = POINTS TO SCORE FIELD
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          A  = .
                        ;          B  = .
                        ;          X  = .
                        ;          U  = .
                        ;
F85E  = 0000                    SETDP   $00
                        ;       =====   ===
                        ;
F85E  CE0000            BYTADD  LDU     #0              ; CONVERT 'A' TO BCD FORMAT
F861  8163              BYTAD0  CMPA    #99             ; .
F863  2308      ^F86D           BLS     BYTAD1          ; .
F865  8064                      SUBA    #100            ; .
F867  33C90100                  LEAU    $0100,U         ; .
F86B  20F4      ^F861           BRA     BYTAD0          ; .
                        ;
F86D  8109              BYTAD1  CMPA    #9              ; .
F86F  2307      ^F878           BLS     BYTAD2          ; .
F871  800A                      SUBA    #10             ; .
F873  33C810                    LEAU    $0010,U         ; .
F876  20F5      ^F86D           BRA     BYTAD1          ; .
                        ;
F878  33C6              BYTAD2  LEAU    A,U             ; .
F87A  1F30                      TFR     U,D             ; SET-UP FOR 'SCRADD'
                        ;
                        ;
                        ; ADD CONTENTS OF 'D' TO INDICATED SCORE
                        ; =====================================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          D  = 4 DIGIT BCD NUMBER
                        ;          X  = POINTS TO SCORE FIELD
                        ;          U  = .
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          A  = .
                        ;          B  = .
                        ;          X  = .
                        ;          U  = .
                        ;
F87C  = 0000                    SETDP   $00
                        ;       =====   ===
                        ;
```

```
F87C  3402             SCRADD   PSHS   A                ;  SET-UP BCD SCORE FOR 'STKADD'
F87E  3404                      PSHS   B                ;  .
                       ;
                       ;
                       ;  ADD CONTENTS OF STACK TO INDICATED SCORE
                       ;  ========================================
                       ;
                       ;        ENTRY VALUES
                       ;        ------------
                       ;
                       ;           X  = POINTS TO SCORE FIELD
                       ;           U  = .
                       ;           S  = .
                       ;
                       ;        RETURN VALUES
                       ;        -------------
                       ;
                       ;           A  = .
                       ;           B  = .
                       ;           X  = .
                       ;           U  = .
                       ;
F880   = 0000                   SETDP  $00
                       ;        =====  ===
                       ;
F880  C605             STKADD   LDB    #5               ;;  HERE IF DATA ON STACK (----
F882  4F               STKAD0   CLRA                    ;;  X POINTS TO MSD OF 6 DIG(BYTE) SCORE
F883  C101                      CMPB   #1               ;;  ADDS 4 PACKED BCD DIGS(2 BYTES)
F885  2310     ^F897            BLS    STKAD3           ;;  DP REG NO CARE
F887  C501                      BITB   #1               ;;  DOES NOT CHANGE X REG
F889  2704     ^F88F            BEQ    STKAD1           ;;  .
F88B  A6E4                      LDA    ,S               ;;  .
F88D  2006     ^F895            BRA    STKAD2           ;;  .
                       ;
F88F  A6E0             STKAD1   LDA    S+               ;;  1ST AND 3RD INPUT DIGS
F891  44                        LSRA                    ;;  .
F892  44                        LSRA                    ;;  .
F893  44                        LSRA                    ;;  .
F894  44                        LSRA                    ;;  .
F895  840F             STKAD2   ANDA   #$0F             ;;  MASK FROM UNPACKING
F897  BBC823           STKAD3   ADDA   LIST             ;;  SERVES AS CARRY IND
F89A  7FC823                    CLR    LIST             ;;  .
F89D  AB85                      ADDA   B,X              ;;  .
F89F  812F                      CMPA   #$2F             ;;  .
F8A1  2E02     ^F8A5            BGT    STKAD4           ;;  ADJUST IF JUST ZERO
F8A3  8B10                      ADDA   #$10             ;;  .
F8A5  8139             STKAD4   CMPA   #$39             ;;  TEST FOR CARRY
F8A7  2305     ^F8AE            BLS    STKAD5           ;;  .
F8A9  800A                      SUBA   #10              ;;  .
F8AB  7CC823                    INC    LIST             ;;  .
F8AE  A785             STKAD5   STA    B,X              ;;  STORE RESULT
F8B0  5A                        DECB                    ;;  B TICKER
F8B1  2ACF     ^F882            BPL    STKAD0           ;;  .
F8B3  7FC823                    CLR    LIST             ;;  .
F8B6  5F                        CLRB                    ;;  .
```

```
                              ;
F8B7  A685                    STKAD6  LDA    B,X           ; HANDLE ZERO SUPPRESSION
F8B9  8130                            CMPA   #$30          ; .
F8BB  2609         ^F8C6              BNE    STKAD7        ; .
F8BD  8620                            LDA    #$20          ; .
F8BF  A785                            STA    B,X           ; .
F8C1  5C                              INCB                 ; .
F8C2  C105                            CMPB   #5            ; .
F8C4  2DF1         ^F8B7              BLT    STKAD6        ; .
F8C6  39                      STKAD7  RTS                  ; .
                              ;
                              ;
                              ;  DETERMINE THE GREATER OF TWO ASCII SCORES
                              ;  ======================================
                              ;
                              ;       ENTRY VALUES
                              ;       ------------
                              ;
                              ;          X  = POINTS TO SCORE FIELD #1
                              ;          U  = POINTS TO SCORE FIELD #2
                              ;
                              ;       RETURN VALUES
                              ;       -------------
                              ;
                              ;          A  = 0 - SCORE #1 = SCORE #2
                              ;               1 - SCORE #1 ) SCORE #2
                              ;               2 - SCORE #1 ( SCORE #2
                              ;          B  = .
                              ;
F8C7  = 0000                          SETDP  $00
                              ;       =====  ===
                              ;
F8C7  3450                    WINNER  PSHS   X,U           ; SAVE USER ENTRY VALUES
                              ;
F8C9  4F                              CLRA                 ; COMPARE SCORE #1 VS. SCORE #2
F8CA  E680                    WIN1    LDB    X+            ; .     FETCH BYTE FOR SCORE #1
F8CC  2B08         ^F8D6              BMI    WIN3          ; .     .     END OF SCORE FIELD DETECTED ?
F8CE  E1C0                            CMPB   U+            ; .     COMPARE AGAINST SCORE #2
F8D0  27F8         ^F8CA              BEQ    WIN1          ; .     .     IF EQUAL, TRY NEXT BYTE
F8D2  2201         ^F8D5              BHI    WIN2          ; .     .     SCORE #1 ) SCORE #2
                              ;
F8D4  4C                              INCA                 ; RETURN - SCORE #1 ( SCORE #2
                              ;
F8D5  4C                      WIN2    INCA                 ; RETURN - SCORE #1 ) SCORE #2
                              ;
F8D6  35D0                    WIN3    PULS   X,U,PC        ; RETURN - SCORE #1 = SCORE #2
                              ;
                              ;
                              ;
                              ;  CALCULATE HIGH SCORE AND SAVE FOR LOGO
                              ;  ======================================
                              ;
                              ;       ENTRY VALUES
                              ;       ------------
                              ;
                              ;          X  = POINTS TO SCORE FIELD
```

```
                        ;         U  = POINTS TO HIGH SCORE FIELD
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;         A  = .
                        ;         B  = .
                        ;         X  = .
                        ;         U  = .
                        ;
F8D8  = 0000                    SETDP   $00
                        ;       =====   ===
                        ;
F8DA  8DED      ^F8C7   HISCR   BSR     WINNER          ; NEW HIGH SCORE ?
F8DA  8101                      CMPA    #1              ; .
F8DC  2606      ^F8E4           BNE     HISCR1          ; .
                        ;
F8DE  A680              HISCR0  LDA     X+              ; SAVE NEW HIGH SCORE
F8E0  A7C0                      STA     U+              ; .
F8E2  2AFA      ^F8DE           BPL     HISCR0          ; .
                        ;
F8E4  39                HISCR1  RTS                     ; RETURN TO CALLER
                        ;
                        ;
                        ; SYMMETRIC COLLISION TEST (INDEXED OFF-SET)
                        ; ========================================
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;         A  = BOX 'Y' DIMENSION (DELTA 'Y')
                        ;         B  = BOX 'X' DIMENSION (DELTA 'X')
                        ;         X  = 'Y:X' COORDINATES OF POINT TO BE TESTED
                        ;         Y  = 'Y:X' COORDINATES OF CENTER OF BOX
                        ;         U  = POINTER TO OFF-SET VALUE
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;         C  = 1 - COLLISION DETECTED
                        ;
F8E5  = 0000                    SETDP   $00
                        ;       =====   ===
                        ;
F8E5  3420              OFF1BX  PSHS    Y               ; SAVE ENTRY VALUES
F8E7  3436                      PSHS    A,B,X,Y         ; .
                        ;
F8E9  EC64                      LDD     4,S             ;; .
F8EB  ABC4                      ADDA    ,U              ;; .
F8ED  EB41                      ADDB    1,U             ;; .
F8EF  ED64              OFF10   STD     4,S             ;; .
F8F1  2010      ^F903           BRA     BOX0            ;; .
                        ;
                        ;
                        ; SYMMETRIC COLLISION TEST (OFF-SET)
                        ; =================================
                            ;
                            ;       ENTRY VALUES
                            ;       ------------
                            ;          A  = BOX 'Y' DIMENSION (DELTA 'Y')
                            ;          B  = BOX 'X' DIMENSION (DELTA 'X')
                            ;          X  = 'Y:X' COORDINATES OF POINT TO BE TESTED
                            ;          Y  = 'Y:X' COORDINATES OF CENTER OF BOX
                            ;          U  = OFF-SET VALUE
                            ;
                            ;       RETURN VALUES
                            ;       -------------
                            ;          C  = 1 - COLLISION DETECTED
                            ;
F8F3  = 0000                        SETDP   $00
                            ;       =====   ===
                            ;
F8F3  3420                  OFF2BX  PSHS    Y                ;  SAVE ENTRY VALUES
F8F5  3436                          PSHS    A,B,X,Y          ;  .
                            ;
F8F7  1F30                          TFR     U,D              ;;  .
F8F9  AB64                          ADDA    4,S              ;;  .
F8FB  EB65                          ADDB    5,S              ;;  .
F8FD  20F0        ^F8EF             BRA     OFF10            ;;  .
```

```
                        ;  SYMMETRIC COLLISION TEST
                        ;  ========================
                        ;
                        ;        ENTRY VALUES
                        ;        ------------
                        ;
                        ;           A  = BOX 'Y' DIMENSION (DELTA 'Y')
                        ;           B  = BOX 'X' DIMENSION (DELTA 'X')
                        ;           X  = 'Y:X' COORDINATES OF POINT TO BE TESTED
                        ;           Y  = 'Y:X' COORDINATES OF CENTER OF BOX
                        ;
                        ;        RETURN VALUES
                        ;        -------------
                        ;
                        ;           C  = 1 - COLLISION DETECTED
                        ;
F8FF  = 0000                     SETDP   $00
                        ;        =====   ===
                        ;
F8FF  3420              BXTEST   PSHS    Y                 ;  SAVE ENTRY VALUE
F901  3436                       PSHS    A,B,X,Y           ;  .
                        ;
F903  1F41              BOX0     TFR     S,X               ;;  .
F905  5F                         CLRB                      ;;  .
F906  3A                BOX1     ABX                       ;;  .
F907  A604                       LDA     4,X               ;;  .
F909  AB84                       ADDA    ,X                ;;  .
F90B  2802       ^F90F           BVC     BOX2              ;;  .
F90D  867F                       LDA     #$7F              ;;  .
F90F  A102              BOX2     CMPA    2,X               ;;  .
F911  2D15       ^F928           BLT     BOX4              ;;  .
F913  A604                       LDA     4,X               ;;  .
F915  A084                       SUBA    ,X                ;;  .
F917  2802       ^F91B           BVC     BOX3              ;;  .
F919  8680                       LDA     #$80              ;;  .
F91B  A102              BOX3     CMPA    2,X               ;;  .
F91D  2E09       ^F928           BGT     BOX4              ;;  .
F91F  5C                         INCB                      ;;  .
F920  C102                       CMPB    #2                ;;  .
F922  25E2       ^F906           BLO     BOX1              ;;  .
                        ;
F924  1A01                       ORCC    #$01              ;  SET CARRY FOR COLLISION DETECTED
F926  2002       ^F92A           BRA     BOX5              ;  .
                        ;
F928  1CFE              BOX4     ANDCC   #$FE              ;  SET CARRY FOR NO COLLISION DETECTED
                        ;
F92A  3536              BOX5     PULS    A,B,X,Y           ;  RETURN TO CALLER
F92C  35A0                       PULS    Y,PC              ;  .    RECOVER ENTRY VALUES
```

```
; COMPLEX EXPLOSION SOUND-EFFECT
; ==============================
;
;       COMMENT(S)
;       ----------
;          'REQOUT' MUST BE CALLED ONCE EACH FRAME
;
;          RATEA   LOCAL PSG ENABLES. ROUTINE STEALS REGS FROM TUNE PLAYER
;                  LOG ENABLES AS IN PSG BUT USES POS LOG. UP TO 3 TONES, 3 NOISE
;                  THE 'AXE' ROUTINE WILL NOT SCREW WITH UNCALLED REGISTERS
;
;          VIBA    NOISE FREQ MOVE DIR, 3 CONDITIONS:
;                  1) IF BIT7 SET, THEN LOWER 5 BITS= STEADY NOISE FREQ
;                  2) IF BYTE NON-ZERO NOISE FREQ GOING DOWN
;                  3) IF BYTE=0 NOISE GOING UP
;
;          RATER   LOUDNESS MOVE DIR, 3 CONDITIONS:
;                  1) IF BIT 7 SET, LOWER 4 BITS=STEADY LOUDNESS
;                  2) IF BYTE NON=ZERO VOLUME GOING UP
;                  3) IF BYTE=0 VOL GOING DOWN
;
;          VIBB    COUNTDOWN RATE FROM $7F. RANGE 1(LONGEST) TO $80(SHORTEST)
;
;          SAMPLE EXPLOSION TABLES
;
;              TYPE1   DB      $19,$3F,$00,$02
;
;              TYPE2   DB      $3F,$00,$00,$01
;
;
;       ENTRY VALUES
;       ------------
;          U  = POINTER TO EXPLOSION TABLE
;                 BYTE 0 = TONE AND NOISE CHANNEL ENABLES
;                               BIT 0 = TONE CHANNEL #
;                                   1 = TONE CHANNEL #
;                                   2 = TONE CHANNEL #
;                                   3 = NOISE SOURCE #
;                                   4 = NOISE SOURCE #
;                                   5 = NOISE SOURCE #
;                 BYTE 1 = NOISE SOURCE SWEEP
;                               = 0    SWEEP FREQUENCY UP
;                               ) 0    SWEEP FREQUENCY DOWN
;                               ( 0    NO SWEEP (FIXED NOISE FREQUENCY)
;                 BYTE 2 = VOLUME SWEEP
;                               = 0    SWEEP VOLUME DOWN
;                               ) 0    SWEEP VOLUME UP
;                               ( 0    NO SWEEP (FIXED VOLUME)
;                 BYTE 3 = EXPLOSION DURATION
;                               $01 = LONGEST
;                               $80 = SHORTEST
;          DP = $C8
;
```

```
                        ;       RETURN VALUES
                        ;       -------------
                        ;          A  = DESTROYED
                        ;          B  = DESTROYED
                        ;          X  = DESTROYED
                        ;
                        ;          XACON    = $00 (WHEN EXPLOSION IS COMPLETED)
                        ;
F92E  = 00C8                    SETDP   $C8
                        ;       =====   ===
                        ;
F92E  9667              EXPLOD  LDA     SATUS           ;;  EXPLOSION SEQUENCER FOR GI-PSG (----
F930  2A29      ^F95B           BPL     EXPLO           ;;  EITHER GOING, NOT GOING, OR REQUEST
F932  847F                      ANDA    #$7F            ;;  REQUEST TO START EXPLOSION
F934  9767                      STA     SATUS           ;;  .
F936  8EC858                    LDX     #RATEA          ;;  LOAD IN 4 PARAMETER BYTES
F939  8604                      LDA     #4              ;;  .
F93B  BDF683                    JSR     BLKMOV          ;;  B HAS 'XREQ6' DATA AT END
F93E  54                        LSRB                    ;;  .
F93F  54                        LSRB                    ;;  .
F940  54                        LSRB                    ;;  .
F941  DA58                      ORB     RATEA           ;;  .
F943  C407                      ANDB    #$07            ;;  .
F945  D754                      STB     TUNE+1          ;;  USE FOR VOL ENABL POS LOG
F947  D658                      LDB     RATEA           ;;  .
F949  C438                      ANDB    #$38            ;;  .
F94B  D753                      STB     TUNE            ;;  NONZERO= NOISE SELECTED
F94D  D658                      LDB     RATEA           ;;  .
F94F  C407                      ANDB    #$07            ;;  .
F951  D75D                      STB     VIBC            ;;  BIT POS FOR EACH TONE CHANNEL
F953  C602                      LDB     #2              ;;  .
F955  D75C                      STB     RATEC           ;;  BIT FOR WHICH RUMBLE
F957  867F                      LDA     #$7F            ;;  .
F959  200D      ^F968           BRA     EXPL2           ;;  .
                        ;
F95B  9677              EXPLO   LDA     XACON           ;;  .
F95D  276A      ^F9C9           BEQ     EXPLC           ;;  NOT GOING
F95F  905B              EXPL1   SUBA    VIBB            ;;  SEQUENCER
F961  2A05      ^F968           BPL     EXPL2           ;;  .
F963  5F                        CLRB                    ;;  .
F964  D777                      STB     XACON           ;;  PREPARE TO EXIT
F966  2062      ^F9CA           BRA     SETAMP          ;;  .
                        ;
F968  9777              EXPL2   STA     XACON           ;;  .
F96A  44                        LSRA                    ;;  SCALE TO $1F RANGE
F96B  44                        LSRA                    ;;  SUITABLE FOR NOISE SPEC
F96C  D653                      LDB     TUNE            ;;  IF=0, NO NOISE ENABLED
F96E  270D      ^F97D           BEQ     EXPL4           ;;  SKIP THIS, JUST PUT LOUDNESS IN
F970  9746                      STA     REQ7            ;;  .
F972  D659                      LDB     VIBA            ;;  WANT UP, DOWN, OR STEADY?
F974  2B05      ^F97B           BMI     EXPL3           ;;  BIT7 SET= STEADY NOISE FREQ
F976  2705      ^F97D           BEQ     EXPL4           ;;  ZERO BYTE= NOISE FREQ GOING UP
F978  1F89                      TFR     A,B             ;;  NOISE GOING DOWN
```

```
F97A  53                          COMB                    ;;  .
F97B  D746              EXPL3     STB      REQ7           ;;  .
F97D  44                EXPL4     LSRA                    ;;  SETUP LOUDNESS ENVELOPE
F97E  8107                        CMPA     #7             ;;  .
F980  2305       ^F987            BLS      EXPL5          ;;  .
F982  810F                        CMPA     #$0F           ;;  .
F984  2701       ^F987            BEQ      EXPL5          ;;  .
F986  4C                          INCA                    ;;  STIFFEN ROLLOFF
F987  D65A              EXPL5     LDB      RATEB          ;;  LOUDNESS UP, DOWN, OR STEADY?
F989  2B06       ^F991            BMI      EXPL7          ;;  IF BIT 7 SET, STEADY LOUDNESS VAL=L.DIR
F98B  2702       ^F98F            BEQ      EXPL6          ;;  IF L.DIR=0, LOUDNESS GOING DOWN
F98D  880F                        EORA     #$0F           ;;  .
F98F  1F89              EXPL6     TFR      A,B            ;;  .
F991  8D37       ^F9CA  EXPL7     BSR      SETAMP         ;;  PUTS IN 1-3 REGS CHA,B,CV
                                                          ;;  RUMBLING TONES
F993  D65D                        LDB      VIBC           ;;  .
F995  272B       ^F9C2            BEQ      EXPLB          ;;  SKIP IF NO TONES
F997  965C              EXPL8     LDA      RATEC          ;;  .
F999  4A                          DECA                    ;;  .
F99A  2A02       ^F99E            BPL      EXPL9          ;;  .
F99C  8602                        LDA      #2             ;;  .
F99E  975C              EXPL9     STA      RATEC          ;;  CYCLE THRU CHANS
F9A0  BDF57E                      JSR      DECBIT         ;;  TEST BIT POS
F9A3  955D                        BITA     VIBC           ;;  .
F9A5  27F0       ^F997            BEQ      EXPL8          ;;  FIND IT
F9A7  D65C                        LDB      RATEC          ;;  .
F9A9  58                          ASLB                    ;;  .
F9AA  50                          NEGB                    ;;  .
F9AB  8EC84B                      LDX      #REGC          ;;  .
F9AE  3085                        LEAX     B,X            ;;  (CAN'T USE ABX)
F9B0  BDF517                      JSR      RANDOM         ;;  .
F9B3  840F                        ANDA     #$0F           ;;  .
F9B5  8105                        CMPA     #5             ;;  FILTER HI TONES
F9B7  2203       ^F9BC            BHI      EXPLA          ;;  .
F9B9  48                          ASLA                    ;;  .
F9BA  8B05                        ADDA     #5             ;;  .
F9BC  A784              EXPLA     STA      ,X             ;;  .
F9BE  967E                        LDA      RANCID+1       ;;  .
F9C0  A701                        STA      1,X            ;;  LO BYTE TOO
F9C2  9658              EXPLB     LDA      RATEA          ;;  .
F9C4  43                          COMA                    ;;  .
F9C5  9445                        ANDA     REQ6           ;;  .
F9C7  9745                        STA      REQ6           ;;  KEEP ENABLED TO AVOID BUG
F9C9  39                EXPLC     RTS                     ;;  .
                        ;
                        ;
                        ;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
                        ;
                        ;         ENTRY VALUES
                        ;         ------------
                        ;            A  = .
                        ;            B  = .
                        ;            X  = .
```

```
                             ;          DP = $C8
                             ;
                             ;        RETURN VALUES
                             ;        -------------
                             ;          A  = .
                             ;          B  = .
                             ;          X  = .
                             ;
F9CA  = 00C8                          SETDP   $C8
                             ;        =====   ===
                             ;
F9CA  9654                   SETAMP   LDA     TUNE+1          ;;  STUFF A,B,AND/OR C VOLS
F9CC  8EC845                          LDX     #REG5+1         ;;  PUT LOUD VALUE IN UP TO 3 CHANNELS
F9CF  4D                     SAMP0    TSTA                    ;;  .
F9D0  2709         ^F9DB              BEQ     SAMP1           ;;  .
F9D2  301F                            LEAX    -1,X            ;;  .
F9D4  44                              LSRA                    ;;  .
F9D5  24F8         ^F9CF              BCC     SAMP0           ;;  .
F9D7  E784                            STB     ,X              ;;  .
F9D9  20F4         ^F9CF              BRA     SAMP0           ;;  .
F9DB  39                     SAMP1    RTS                     ;;  .
                             ;
                             ;
                             ;
                             ;  BIT DECODE TABLE
                             ;  ================
                             ;
F9DC  01                     DECTBL   DB      $01
F9DD  02                              DB      $02
F9DE  04                              DB      $04
F9DF  08                              DB      $08
F9E0  10                              DB      $10
F9E1  20                              DB      $20
F9E2  40                              DB      $40
F9E3  80                              DB      $80
                             ;
                             ;
                             ;
F9E4  F7EFDF010204           CRT      DB      $F7,$EF,$DF,$01,$02,$04  ;;  FOR 'REPLAY'
F9EA  FEFDFB081020           CRATE    DB      $FE,$FD,$FB,$08,$10,$20
                             ;
                             ;
                             ;  KEEP-ALIVE VECTORS
                             ;  ==================
                             ;
F9F0  7F7F8080               KEPALV  DB     $7F,$7F,$80,$80
                             ;
                             ;
                             ;  ASCII RASTER TABLE
                             ;  ==================
                             ;
                             ;;  5X7 CHARACTER DECODE $20 THRU $6F
                             ;;  CHAR SET GROUPED BY ROW, TOP TO BOTTOM
```

```
                                 ;;  $20-$2F= ' !"#$%&'()*+,-./'
                                 ;;  $30-$3F= '0123456789:; <=>?'
                                 ;;  $40-$4F= '@ARCDEFGHIJKLMNO'
                                 ;;  $50-$5F= 'PQRSTUVWXYZ[\]^_'
                                 ;;  $60-$6F= W.T. CHAR SET. SEE DOC FOR PICTURES
                                 ;;  $60 = CAR
                                 ;;  $61 = UP ARROW . FOR RIGHT ARROW USE -> 2 CHARS
                                 ;;  $62 = MUSICAL NOTE
                                 ;;  $63 = DOWN ARROW. FOR LEFT ARROW USE <- 2 CHARS
                                 ;;  $64 = CIRCLE: LARGE, HOLLOW
                                 ;;  $65 = CIRCLE: LARGE, SOLID
                                 ;;  $66 = CIRCLE: SMALL, SOLID
                                 ;;  $67 = COPYRIGHT CIRCLE
                                 ;;  $68 = SPACE SHIP
                                 ;;  $69 = SPACEPERSON
                                 ;;  $6A = HAVE A NICE DAY
                                 ;;  $6B = HAVE A BAD DAYW
                                 ;;  $6C = INFINITY
                                 ;;  $6D = HOLLOW SQUARE
                                 ;;  $6E = 7X8 FILL CHAR .DO NOT USE AT END OF LINE
                                 ;;  $6F = 7X7 FILL CHAR. OK AT END OF LINE
                                 ;
F9F4  0020505020C82010  ASCII   DW     $0020,$5050,$20C8,$2010,$1040,$2000,$0000,$0008      ;;  ROW7
FA04  3020707010F830F8          DW     $3020,$7070,$10F8,$30F8,$7070,$0060,$0000,$0070
FA14  7020F070F0F8F878          DW     $7020,$F070,$F0F8,$F878,$8870,$0888,$8088,$88F8
FA24  F070F070F8888888          DW     $F070,$F070,$F888,$8888,$8888,$F870,$8070,$2000
FA34  0020082000000038          DW     $0020,$0820,$0000,$0038,$1020,$4444,$00FE,$FFFE
                         ;
FA44  007050507 8C85020          DW     $0070,$5050,$78C8,$5020,$2020,$A820,$0000,$0008      ;;  ROW6
FA54  4860888830804008          DW     $4860,$8888,$3080,$4008,$8888,$6060,$1000,$4088
FA64  8850488848808080          DW     $8850,$4888,$4880,$8080,$8820,$0890,$80D8,$C888
FA74  88888888A8888888          DW     $8888,$8888,$A888,$8888,$8888,$0840,$8008,$5000
FA84  00700C2070700044          DW     $0070,$0C20,$7070,$0044,$1070,$0000,$6C82,$FFFE
                         ;
FA94  007050F8A0105040          DW     $0070,$50F8,$A010,$5040,$4010,$7020,$0000,$0010      ;;  ROW5
FAA4  4820080850F08010          DW     $4820,$0808,$50F0,$8010,$8888,$6000,$2078,$2008
FAB4  A888488048808080          DW     $A888,$4880,$4880,$8080,$8820,$08A0,$80A8,$A888
FAC4  8888884020888888          DW     $8888,$8840,$2088,$8888,$5050,$1040,$4008,$8800
FAD4  70A80A2088F860BA          DW     $70A8,$0A20,$88F8,$60BA,$3820,$0000,$9282,$FFFE
                         ;
FAE4  0020005070206000          DW     $0020,$0050,$7020,$6000,$4010,$A8F8,$0070,$0020      ;;  ROW4
FAF4  482070309008F020          DW     $4820,$7030,$9008,$F020,$7078,$0060,$4000,$1010
FB04  R888708048E0E098          DW     $B888,$7080,$48E0,$E098,$F820,$08C0,$80A8,$9888
FB14  F088F020208850A8          DW     $F088,$F020,$2088,$50A8,$2020,$2040,$2008,$0000
FR24  FE20082088F8F0A2          DW     $FE20,$0820,$88F8,$F0A2,$38F8,$8238,$9282,$FFFE
                         ;
FB34  000000F87040A800          DW     $0000,$00F8,$7040,$A800,$4010,$A820,$4000,$0040      ;;  ROW3
FB44  48208008F8088840          DW     $4820,$8008,$F808,$8840,$8808,$6060,$2078,$2020
FB54  B0F8488048808088          DW     $B0F8,$4880,$4880,$8088,$8820,$08A0,$8088,$8888
FB64  80A8A010208850A8          DW     $80A8,$A010,$2088,$50A8,$5020,$4040,$1008,$0000
FR74  FE2078A888F8F0BA          DW     $FE20,$78A8,$88F8,$F0BA,$7C20,$4444,$6C82,$FFFE
                         ;
FB84  0000005028989000          DW     $0000,$0050,$2898,$9000,$2020,$0020,$4000,$0080      ;;  ROW2
```

```
FB94  4820808810888880          DW      $4820,$8088,$1088,$8880,$8810,$6020,$1000,$4000
FBA4  8088488848808088          DW      $8088,$4888,$4880,$8088,$8820,$8890,$8888,$8888
FBB4  8090908820882 0A8         DW      $8090,$9088,$2088,$20A8,$8820,$8040,$0808,$0000
FBC4  4820F07070706044          DW      $4820,$F070,$7070,$6044,$6C50,$3882,$0082,$FFFE
                        ;
FBD4  00200050F8986800          DW      $0020,$0050,$F898,$6800,$1040,$0000,$8000,$8080      ;;  ROW1
FBE4  3070F87010707080          DW      $3070,$F870,$1070,$7080,$7060,$0040,$0000,$0020
FBF4  7888F070F0F88078          DW      $7888,$F070,$F0F8,$8078,$8870,$7088,$F888,$88F8
FC04  8068887020702050          DW      $8068,$8870,$2070,$2050,$8820,$F870,$0870,$00F8
FC14  0020602000000038          DW      $0020,$6020,$0000,$0038,$8288,$0000,$00FE,$FFFE
                        ;
FC24  0011413021102031  WEIRD   DW      $0011,$4130,$2110,$2031  ;;  TRANSLATE 8 WAYS
                        ;
FC2C  000103060A0F151C  ROWTRI  DW      $0001,$0306,$0A0F,$151C,$242D  ;;  FOR COMPAS ROW DECODE
                        ;
                        ;
                        ;  QUICK ARC-TAN LOOKUP
                        ;  ====================
                        ;
FC36  08                FIBTBL  DB      $08
FC37  1008                      DB      $10,$08
FC39  100B08                    DB      $10,$0B,$08
FC3C  100D0A08                  DB      $10,$0D,$0A,$08
FC40  100E0B0908                DB      $10,$0E,$0B,$09,$08
FC45  100E0C0A0908              DB      $10,$0E,$0C,$0A,$09,$08
FC4B  100E0D0B0A0908            DB      $10,$0E,$0D,$0B,$0A,$09,$08
FC52  100F0D0C0B0A0908          DB      $10,$0F,$0D,$0C,$0B,$0A,$09,$08
FC5A  100F0E0C0B0A0909          DB      $10,$0F,$0E,$0C,$0B,$0A,$09,$09,$08
FC63  100F0E0D0C0B0A09          DB      $10,$0F,$0E,$0D,$0C,$0B,$0A,$09,$09,$08
                        ;
                        ;
                        ;  SINE LOOK-UP
                        ;  ============
                        ;
FC6D  0019324A          RTRIGS  DB      0,25,50,74
FC71  62798EA2                  DB      98,121,142,162
FC75  B5C6D5E2                  DB      181,198,213,226
FC79  EDF5FBFF                  DB      237,245,251,255
FC7D  FFFFFBF5                  DB      255,255,251,245
FC81  EDE2D5C6                  DB      237,226,213,198
FC85  B5A28E79                  DB      181,162,142,121
FC89  624A3219                  DB      98,74,50,25
                        ;
                        ;
                        ;  NOTE TABLE (FOR GAME MELODIES)
                        ;  ==============================
                        ;
      = 0000            G2      EQU     0                    ;;  G = 15 8VES BELOW MIDDLE C
FC8D  03BD              NOTES   DW      957                  ;;  BEGINNING FREQUENCY
                        ;
      = 0001            GS2     EQU     1                    ;;  G SHARP (SECOND 8VE) , ETC.
FC8F  0387                      DW      903                  ;;  .
                        ;
```

```
     = 0002       A2      EQU    2             ;;  .
FC91  0354                DW     852           ;;  .
                  ;
     = 0003       AS2     EQU    3             ;;  .
FC93  0324                DW     804           ;;  .
                  ;
     = 0004       B2      EQU    4             ;;  .
FC95  02F7                DW     759           ;;  .
                  ;
     = 0005       C3      EQU    5             ;;  .
FC97  02CD                DW     717           ;;  .
                  ;
     = 0006       CS3     EQU    6             ;;  .
FC99  02A4                DW     676           ;;  .
                  ;
     = 0007       D3      EQU    7             ;;  .
FC9B  027E                DW     638           ;;  .
                  ;
     = 0008       DS3     EQU    8             ;;  .
FC9D  025B                DW     603           ;;  .
                  ;
     = 0009       E3      EQU    9             ;;  .
FC9F  0239                DW     569           ;;  .
                  ;
     = 000A       F3      EQU    $0A           ;;  .
FCA1  0219                DW     537           ;;  .
                  ;
     = 000B       FS3     EQU    $0B           ;;  .
FCA3  01FB                DW     507           ;;  .
                  ;
     = 000C       G3      EQU    $0C           ;;  .
FCA5  01DE                DW     478           ;;  .
                  ;
     = 000D       GS3     EQU    $0D           ;;  .
FCA7  01C3                DW     451           ;;  .
                  ;
     = 000E       A3      EQU    $0E           ;;  .
FCA9  01AA                DW     426           ;;  .
                  ;
     = 000F       AS3     EQU    $0F           ;;  .
FCAB  0192                DW     402           ;;  .
                  ;
     = 0010       B3      EQU    $10           ;;  .
FCAD  017C                DW     380           ;;  .
                  ;
     = 0011       C4      EQU    $11           ;;  .
FCAF  0166                DW     358           ;;  .
                  ;
     = 0012       CS4     EQU    $12           ;;  .
FCB1  0152                DW     338           ;;  .
                  ;
     = 0013       D4      EQU    $13           ;;  .
FCB3  013F                DW     319           ;;  .
```

```
                       ;
      = 0014           DS4     EQU     $14             ;; .
FCB5  012D                     DW      301             ;; .
                       ;
      = 0015           E4      EQU     $15             ;; .
FCB7  011C                     DW      284             ;; .
                       ;
      = 0016           F4      EQU     $16             ;; .
FCB9  010C                     DW      268             ;; .
                       ;
      = 0017           FS4     EQU     $17             ;; .
FCBB  00FD                     DW      253             ;; .
                       ;
      = 0018           G4      EQU     $18             ;; .
FCBD  00EF                     DW      239             ;; .
                       ;
      = 0019           GS4     EQU     $19             ;; .
FCBF  00E2                     DW      226             ;; .
                       ;
      = 001A           A4      EQU     $1A             ;; .
FCC1  00D5                     DW      213             ;; .
                       ;
      = 001B           AS4     EQU     $1B             ;; .
FCC3  00C9                     DW      201             ;; .
                       ;
      = 001C           B4      EQU     $1C             ;; .
FCC5  00BE                     DW      190             ;; .
                       ;
      = 001D           C5      EQU     $1D             ;; .
FCC7  00B3                     DW      179             ;; .
                       ;
      = 001E           CS5     EQU     $1E             ;; .
FCC9  00A9                     DW      169             ;; .
                       ;
      = 001F           D5      EQU     $1F             ;; .
FCCB  00A0                     DW      160             ;; .
                       ;
      = 0020           DS5     EQU     $20             ;; .
FCCD  0097                     DW      151             ;; .
                       ;
      = 0021           E5      EQU     $21             ;; .
FCCF  008E                     DW      142             ;; .
                       ;
      = 0022           F5      EQU     $22             ;; .
FCD1  0086                     DW      134             ;; .
                       ;
      = 0023           FS5     EQU     $23             ;; .
FCD3  007F                     DW      127             ;; .
                       ;
      = 0024           G5      EQU     $24             ;; .
FCD5  0078                     DW      120             ;; .
                       ;
      = 0025           GS5     EQU     $25             ;; .
```

```
FCD7  0071                     DW      113             ;;  .
                        ;
      = 0026            A5     EQU     $26             ;;  .
FCD9  006B                     DW      107             ;;  .
                        ;
      = 0027            AS5    EQU     $27             ;;  .
FCDB  0065                     DW      101             ;;  .
                        ;
      = 0028            B5     EQU     $28             ;;  .
FCDD  005F                     DW      95              ;;  .
                        ;
      = 0029            C6     EQU     $29             ;;  .
FCDF  005A                     DW      90              ;;  .
                        ;
      = 002A            CS6    EQU     $2A             ;;  .
FCE1  0055                     DW      85              ;;  .
                        ;
      = 002B            D6     EQU     $2B             ;;  .
FCE3  0050                     DW      80              ;;  .
                        ;
      = 002C            DS6    EQU     $2C             ;;  .
FCE5  004B                     DW      75              ;;  .
                        ;
      = 002D            E6     EQU     $2D             ;;  .
FCE7  0047                     DW      71              ;;  .
                        ;
      = 002E            F6     EQU     $2E             ;;  .
FCE9  0043                     DW      67              ;;  .
                        ;
      = 002F            FS6    EQU     $2F             ;;  .
FCEB  003F                     DW      63              ;;  .
                        ;
      = 0030            G6     EQU     $30             ;;  .
FCED  003C                     DW      60              ;;  .
                        ;
      = 0031            GS6    EQU     $31             ;;  .
FCEF  0038                     DW      56              ;;  .
                        ;
      = 0032            A6     EQU     $32             ;;  .
FCF1  0035                     DW      53              ;;  .
                        ;
      = 0033            AS6    EQU     $33             ;;  .
FCF3  0032                     DW      50              ;;  .
                        ;
      = 0034            B6     EQU     $34             ;;  .
FCF5  002F                     DW      47              ;;  .
                        ;
      = 0035            C7     EQU     $35             ;;  .
FCF7  002D                     DW      45              ;;  .
                        ;
      = 0036            CS7    EQU     $36             ;;  .
FCF9  002A                     DW      42              ;;  .
                        ;
      = 0037          D7       EQU     $37             ;; .
FCFB  0028                     DW      40              ;; .
                      ;
      = 0038          DS7      EQU     $38             ;; .
FCFD  0026                     DW      38              ;; .
                      ;
      = 0039          E7       EQU     $39             ;; .
FCFF  0024                     DW      36              ;; .
                      ;
      = 003A          F7       EQU     $3A             ;; .
FD01  0022                     DW      34              ;; .
                      ;
      = 003B          FS7      EQU     $3B             ;; .
FD03  0020                     DW      32              ;; .
                      ;
      = 003C          G7       EQU     $3C             ;; .
FD05  001E                     DW      30              ;; .
                      ;
      = 003D          GS7      EQU     $3D             ;; .
FD07  001C                     DW      28              ;; .
                      ;
      = 003E          A7       EQU     $3E             ;; .
FD09  001B                     DW      27              ;; .
                      ;
      = 003F          AS7      EQU     $3F             ;; ZERO FREQUENCY FOR REST
FD0B  0000                     DW      0               ;; .
                      ;
                      ;
                      ;  OPENING TUNE FOR VECTREX
                      ;  =======================
                      ;
      = 001E          VH       EQU     30
      = 0012          VEQ      EQU     18
      = 0006          TRV      EQU     06
                      ;
FD0D  FEE8            VCTRX    DW      FADE4
FD0F  FEB6                     DW      VIBEHL
FD11  93                       DB      D4 OR $80
FD12  1F0C                     DB      D5,TRV+TRV
FD14  93                       DB      D4 OR $80
FD15  1F06                     DB      D5,TRV
FD17  98                       DB      G4 OR $80
FD18  9F                       DB      D5 OR $80
FD19  243C                     DB      G5,VH*2
FD1B  1180                     DB      C4,$80
                      ;
                      ;
                      ;  OPENING TUNE FOR BERZERK
                      ;  =======================
                      ;
      = 0007          TR8      EQU     7
      = 000E          TRQTR    EQU     14
      = 0002          ARPEG1   EQU     02
      = 0028          HA      EQU     40
                      ;
FD1D  FD69            BZERK   DW      FADE0
FD1F  FD79                    DW      VIBE0
FD21  2107                    DB      E5,TR8
FD23  2107                    DB      E5,TR8
FD25  2107                    DB      E5,TR8
FD27  2107                    DB      E5,TR8
FD29  2107                    DB      E5,TR8
FD2B  2107                    DB      E5,TR8
FD2D  210E                    DB      E5,TRQTR
FD2F  99                      DB      GS4 OR $80
FD30  9F                      DB      D5 OR $80
FD31  240E                    DB      G5,TRQTR
FD33  95                      DB      E4 OR $80
FD34  9B                      DB      AS4 OR $80
FD35  200E                    DB      DS5,TRQTR
FD37  2107                    DB      E5,TR8
FD39  2107                    DB      E5,TR8
FD3B  2107                    DB      E5,TR8
FD3D  2107                    DB      E5,TR8
FD3F  2107                    DB      E5,TR8
FD41  2107                    DB      E5,TR8
                      ;
                      ;
                      ;
                      ;
                      ;************************************************************************
                      ; THIS AREA DELETED
                      ;************************************************************************
                      ;
FD43  9D                      DB      C5 OR $80
FD44  A3                      DB      FS5 OR $80
FD45  280E                    DB      B5,TRQTR
FD47  A0                      DB      DS5 OR $80
FD48  A6                      DB      A5 OR $80
FD49  2B0E                    DB      D6,TRQTR
FD4B  2202                    DB      F5,ARPEG1
FD4D  2802                    DB      B5,ARPEG1
FD4F  2D02                    DB      E6,ARPEG1
FD51  2802                    DB      B5,ARPEG1
FD53  2202                    DB      F5,ARPEG1
FD55  2802                    DB      B5,ARPEG1
FD57  2D02                    DB      E6,ARPEG1
FD59  2802                    DB      B5,ARPEG1
FD5B  2202                    DB      F5,ARPEG1
FD5D  2802                    DB      B5,ARPEG1
FD5F  2D02                    DB      E6,ARPEG1
FD61  2802                    DB      B5,ARPEG1
FD63  2E02                    DB      F6,ARPEG1
FD65  2D28                    DB      E6,HA
FD67  2180            ENDZ    DB      E5,$80
FD69  EFFFFEDCBA000000 FADE0  DW      $EFFF,$FEDC,$BA00,$0000
```

```
FD71  0000000000000000            DW      0,0,0,0
FD79  0001020100FFFEFF  VIBEO     DB      0,1,2,1,0,-1,-2,-1
                        ;
                        ;
                        ;  OPENING TUNE FOR ARMOR ATTACK
                        ;  ============================
                        ;
      = 0018            QUO       EQU     24
      = 000C            EGG       EQU     12
      = 0006            SIP       EQU     06
      = 0004            SIT       EQU     04
      = 0010            SNAR      EQU     $10
                        ;
FD81  FDC3              ARMOR     DW      FADE12
FD83  FEB6                        DW      VIBEHL
FD85  5124                        DB      $40 OR SNAR+1,QUO+EGG
FD87  5006                        DB      $40 OR SNAR,SIP
FD89  5006                        DB      $40 OR SNAR,SIP
FD8B  500C                        DB      $40 OR SNAR,EGG
FD8D  5006                        DB      $40 OR SNAR,SIP
FD8F  5006                        DB      $40 OR SNAR,SIP
FD91  5004                        DB      $40 OR SNAR,SIT
FD93  5004                        DB      $40 OR SNAR,SIT
FD95  5004                        DB      $40 OR SNAR,SIT
FD97  5018                        DB      $40 OR SNAR,QUO
FD99  5004                        DB      $40 OR SNAR,SIT
FD9B  5004                        DB      $40 OR SNAR,SIT
FD9D  5004                        DB      $40 OR SNAR,SIT
FD9F  500C                        DB      $40 OR SNAR,EGG
FDA1  500C                        DB      $40 OR SNAR,EGG
FDA3  5024                        DB      $40 OR SNAR,QUO+EGG
FDA5  5006                        DB      $40 OR SNAR,SIP
FDA7  5006                        DB      $40 OR SNAR,SIP
FDA9  500C                        DB      $40 OR SNAR,EGG
FDAB  5006                        DB      $40 OR SNAR,SIP
FDAD  5006                        DB      $40 OR SNAR,SIP
FDAF  5004                        DB      $40 OR SNAR,SIT
FDB1  5004                        DB      $40 OR SNAR,SIT
FDB3  5004                        DB      $40 OR SNAR,SIT
FDB5  5018                        DB      $40 OR SNAR,QUO
FDB7  5004                        DB      $40 OR SNAR,SIT
FDB9  5004                        DB      $40 OR SNAR,SIT
FDBB  5004                        DB      $40 OR SNAR,SIT
FDBD  500C                        DB      $40 OR SNAR,EGG
FDBF  5018                        DB      $40 OR SNAR,QUO
FDC1  2680                        DB      A5,$80
                        ;
FDC3  FDBA987655443322  FADE12    DW      $FDBA,$9876,$5544,$3322,$1100,0,0,0
                        ;
                        ;
                        ;  OPENING TUNE FOR SCRAMBLE
                        ;  ========================
                        ;
```

```
          = 0010        QU      EQU     16
          = 0008        EIT     EQU     08
          = 0004        SINTH   EQU     04
          = 0030        HAT     EQU     48
                        ;
FDD3  FE28              SCRMBL  DW      FADE1
FDD5  FD79                      DW      VIBEO
FDD7  98                        DB      G4 OR $80
FDD8  1C10                      DB      B4,QU
FDDA  3F08                      DB      AS7,EIT
FDDC  98                        DB      G4 OR $80
FDDD  1C04                      DB      B4,SINTH
FDDF  98                        DB      G4 OR $80
FDE0  1C04                      DB      B4,SINTH
FDE2  98                        DB      G4 OR $80
FDE3  1C10                      DB      B4,QU
FDE5  3F08                      DB      AS7,EIT
FDE7  98                        DB      G4 OR $80
FDE8  1C04                      DB      B4,SINTH
FDEA  98                        DB      G4 OR $80
FDEB  1C04                      DB      B4,SINTH
FDED  98                        DB      G4 OR $80
FDEE  1C08                      DB      B4,EIT
FDF0  93                        DB      D4 OR $80
FDF1  1808                      DB      G4,EIT
FDF3  98                        DB      G4 OR $80
FDF4  1C08                      DB      B4,EIT
FDF6  9C                        DB      B4 OR $80
FDF7  1F08                      DB      D5,EIT
FDF9  98                        DB      G4 OR $80
FDFA  1C08                      DB      B4,EIT
FDFC  93                        DB      D4 OR $80
FDFD  1808                      DB      G4,EIT
FDFF  98                        DB      G4 OR $80
FE00  1C08                      DB      B4,EIT
FE02  93                        DB      D4 OR $80
FE03  1808                      DB      G4,EIT
FE05  98                        DB      G4 OR $80
FE06  1C08                      DB      B4,EIT
FE08  9C                        DB      B4 OR $80
FE09  1F08                      DB      D5,EIT
FE0B  98                        DB      G4 OR $80
FE0C  1C08                      DB      B4,EIT
FE0E  93                        DB      D4 OR $80
FE0F  1808                      DB      G4,EIT
FE11  98                        DB      G4 OR $80
FE12  1C08                      DB      B4,EIT
FE14  93                        DB      D4 OR $80
FE15  1808                      DB      G4,EIT
FE17  98                        DB      G4 OR $80
FE18  1C08                      DB      B4,EIT
FE1A  9C                        DB      B4 OR $80
FE1B  1F08                      DB      D5,EIT
```

```
FE1D  98                        DB      G4 OR $80
FE1E  1C08                      DB      B4,EIT
FE20  93                        DB      D4 OR $80
FE21  1808                      DB      G4,EIT
FE23  9C                        DB      B4 OR $80
FE24  1F30                      DB      D5,HAT
FE26  1A80                      DB      A4,$80
                       ;
FE28  FFFEDCBA98765432 FADE1    DB      $FF,$FE,$DC,$BA,$98,$76,$54,$32,$10
FE31  00000000000000            DB      0,0,0,0,0,0,0
                       ;
                       ;
                       ;  OPENING TUNE FOR SOLAR QUEST
                       ;  ============================
                       ;
      = 0018           QR       EQU     24
      = 0012           DOT8TH   EQU     18
      = 0006           SXTNTH   EQU     06
      = 003C           HALF     EQU     60
                       ;
FE3A  FE66             SOLAR    DW      FADE2
FE3A  FEB6                      DW      VIBENL
FE3C  0C18                      DB      G3,QR
FE3E  1118                      DB      C4,QR
FE40  0C18                      DB      G3,QR
FE42  1118                      DB      C4,QR
FE44  0C18                      DB      G3,QR
FE46  1118                      DB      C4,QR
FE48  0C12                      DB      G3,DOT8TH
FE4A  0C06                      DB      G3,SXTNTH
FE4C  1118                      DB      C4,QR
FE4E  9D                        DB      C5 OR $80
FE4F  2118                      DB      E5,QR
FE51  9F                        DB      D5 OR $80
FE52  2318                      DB      FS5,QR
FE54  A1                        DB      E5 OR $80
FE55  2418                      DB      G5,QR
FE57  A3                        DB      FS5 OR $80
FE58  2618                      DB      A5,QR
FE5A  9F                        DB      D5 OR $80
FE5B  A4                        DB      G5 OR $80
FE5C  2818                      DB      B5,QR
FE5E  0712                      DB      D3,DOT8TH
FE60  0706                      DB      D3,SXTNTH
FE62  003C                      DB      G2,HALF
FE64  1880                      DB      G4,$A0
                       ;
FE66  DEEFFEDCBA000000 FADE2    DB      $DE,$EF,$FE,$DC,$BA,$00,$00,$00
FE6E  0000000000000000          DB      0,0,0,0,0,0,0,0
                       ;
                       ;
                       ;  OPENING TUNE FOR CLEAN-SWEEP
                       ;  ============================
                            ;
      = 0006                SNTH    EQU     06
      = 000C                EIGHTH  EQU     12
      = 0018                QUR     EQU     24
      = 0024                DQTR    EQU     36
      = 0004                ARPEG   EQU     04
      = 0032                BIG     EQU     50
                            ;
FE76  FEB2                  SWEEP   DW      FADE3
FE78  FEB6                          DW      VIBENL
FE7A  1806                          DB      G4,SNTH
FE7C  1A06                          DB      A4,SNTH
FE7E  1C0C                          DB      B4,EIGHTH
FE80  180C                          DB      G4,EIGHTH
FE82  1A24                          DB      A4,EIGHTH+QUR
FE84  2318                          DB      FS5,QUR
FE86  1706                          DB      FS4,SNTH
FE88  1806                          DB      G4,SNTH
FE8A  1A0C                          DB      A4,EIGHTH
FE8C  170C                          DB      FS4,EIGHTH
FE8E  1824                          DB      G4,EIGHTH+QUR
FE90  2418                          DB      G5,QUR
FE92  A4                            DB      G5 OR $80
FE93  280C                          DB      B5,EIGHTH
FE95  A3                            DB      FS5 OR $80
FE96  260C                          DB      A5,EIGHTH
FE98  A1                            DB      E5 OR $80
FE99  240C                          DB      G5,EIGHTH
FE9B  9F                            DB      D5 OR $80
FE9C  230C                          DB      FS5,EIGHTH
FE9E  9D                            DB      C5 OR $80
FE9F  2118                          DB      E5,QUR
FEA1  9A                            DB      A4 OR $80
FEA2  1F18                          DB      D5,QUR
FEA4  1706                          DB      FS4,SNTH
FEA6  1806                          DB      G4,SNTH
FEA8  1A0C                          DB      A4,EIGHTH
FEAA  170C                          DB      FS4,EIGHTH
FEAC  1824                          DB      G4,DQTR
FEAE  2424                          DB      G5,DQTR
FEB0  1880                          DB      G4,$80
                            ;
FEB2  FFEEDDCC              FADE3   DB      $FF,$EE,$DD,$CC
FEB6  0000000000000000      VIBENL  DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                            ;
                            ;
                            ;  OPENING TUNE FOR STAR-TREEK
                            ;  ==========================
                            ;
      = 0032                HAQ     EQU     50
      = 001E                TRIQ    EQU     30
      = 0014                ITA     EQU     20
      = 000A                ITB     EQU     10
```

```
                         ;
FEC6  FEE8               TREK    DW      FADE4
FEC8  FEB6                       DW      VIBENL
FECA  96                         DB      F4 OR $80
FECB  9A                         DB      A4 OR $80
FECC  1D1E                       DB      C5,TRIQ
FECE  91                         DB      C4 OR $80
FECF  95                         DB      E4 OR $80
FED0  181E                       DB      G4,TRIQ
FED2  94                         DB      DS4 OR $80
FED3  98                         DB      G4 OR $80
FED4  1B1E                       DB      AS4,TRIQ
FED6  8F                         DB      AS3 OR $80
FED7  94                         DB      DS4 OR $80
FED8  1814                       DB      G4,ITA
FEDA  160A                       DB      F4,ITB
FEDC  8C                         DB      G3 OR $80
FEDD  91                         DB      C4 OR $80
FEDE  1514                       DB      E4,ITA
FEE0  160A                       DB      F4,ITB
FEE2  91                         DB      C4 OR $80
FEE3  95                         DB      E4 OR $80
FEE4  1832                       DB      G4,HAQ
FEE6  1880                       DB      G4,$80
                         ;
FEE8  EEFFFFEEEEDDCCBB   FADE4   DW      $EEFF,$FFEE,$EEDD,$CCBB,$AA99,$8888,$8888,$8888
                         ;
                         ;
                         ; FOLLOWING IS A SERIES OF SIX FANFARES SET IN VARIOUS KEYS, TEMPI
                         ; AND STYLES. THEY ARE 2-4 MEASURES LONG AND MAY BE USED TO HIGHLIGHT
                         ; PARTICULAR ACTIONS IN THE GAMES OR INTERRUPT AND CELEBRATE (OR WHAT-
                         ; EVER) A PARTICULAR SCORING MILESTONE.
                         ;
      = 0006             QRTR2   EQU     06
      = 001E             HAFF2   EQU     30
                         ;
FEF8  FF16               FFARE1  DW      FADE8
FEFA  FEB6                       DW      VIBENL
FEFC  1C06                       DB      B4,QRTR2
FEFE  1F06                       DB      D5,QRTR2
FF00  1C06                       DB      B4,QRTR2
FF02  1806                       DB      G4,QRTR2
FF04  1A06                       DB      A4,QRTR2
FF06  1806                       DB      G4,QRTR2
FF08  1506                       DB      E4,QRTR2
FF0A  1306                       DB      D4,QRTR2
FF0C  1806                       DB      G4,QRTR2
FF0E  1306                       DB      D4,QRTR2
FF10  1706                       DB      FS4,QRTR2
FF12  181E                       DB      G4,HAFF2
FF14  1880                       DB      G4,$80
                         ;
FF16  FFFFEEEEDDDDCCCC   FADE8   DW      $FFFF,$EEEE,$DDDD,$CCCC
```

```
FF1E  0000000000000000           DW      0,0,0,0
                          ;
                          ;
                          ; #2 IS A TRADITIONAL TRUMPET STYLE FANFARE
                          ;
      = 000F              QRTR3    EQU     15
      = 0005              TRATH3   EQU     05
      = 0032              QHELD    EQU     50
                          ;
FF26  FE28                FFARE2   DW      FADE1
FF28  FEB6                         DW      VIBENL
FF2A  160F                         DB      F4,QRTR3
FF2C  1605                         DB      F4,TRATH3
FF2E  1605                         DB      F4,TRATH3
FF30  1605                         DB      F4,TRATH3
FF32  1A0F                         DB      A4,QRTR3
FF34  160F                         DB      F4,QRTR3
FF36  1D0F                         DB      C5,QRTR3
FF38  1D05                         DB      C5,TRATH3
FF3A  1D05                         DB      C5,TRATH3
FF3C  1D05                         DB      C5,TRATH3
FF3E  210F                         DB      E5,QRTR3
FF40  1D32                         DB      C5,QHELD
FF42  1D80                         DB      C5,$80
                          ;
                          ;
                          ; FF03 IS SPECIFICALLY FOR BERZERK.
                          ;
      = 0006              Q3R      EQU     06
      = 0002              TH3R     EQU     02
      = 0032              QHOLD    EQU     50
                          ;
FF44  FE28                FF03     DW      FADE1
FF46  FEB6                         DW      VIBENL
FF48  1606                         DB      F4,Q3R
FF4A  1602                         DB      F4,TH3R
FF4C  1602                         DB      F4,TH3R
FF4E  1602                         DB      F4,TH3R
FF50  1A06                         DB      A4,Q3R
FF52  1606                         DB      F4,Q3R
FF54  1D06                         DB      C5,Q3R
FF56  1D02                         DB      C5,TH3R
FF58  1D02                         DB      C5,TH3R
FF5A  1D02                         DB      C5,TH3R
FF5C  2106                         DB      E5,Q3R
FF5E  1D32                         DB      C5,QHOLD
FF60  1180                         DB      C4,$80
                          ;
                          ;
                          ; #3 IS A COMIC VERSION OF #3
                          ;
      = 000F              Q4       EQU     15
      = 0005              TR4      EQU     05
      = 0030          H4      EQU     48
      = 000A          HR4     EQU     10
                      ;
FF62  FE28            FFARE3  DW      FADE1
FF64  FEB6                    DW      VIBENL
FF66  1B0F                    DB      AS4,Q4
FF68  1605                    DB      F4,TR4
FF6A  1605                    DB      F4,TR4
FF6C  1605                    DB      F4,TR4
FF6E  1730                    DB      FS4,H4
FF70  1605                    DB      F4,TR4
FF72  1605                    DB      F4,TR4
FF74  1605                    DB      F4,TR4
FF76  1730                    DB      FS4,H4
FF78  1680                    DB      F4,$80
                      ;
                      ;
                      ; #4 IS A CUTESY LITTLE RIFF IN THIRDS
                      ;
      = 0012          Q9      EQU     18
      = 000C          TR9     EQU     12
      = 0006          TR9A    EQU     06
      = 0032          HELD9   EQU     50
                      ;
FF7A  FD69            FFARE4  DW      FADE0
FF7C  FEB6                    DW      VIBENL
FF7E  A0                      DB      DS5 OR $80
FF7F  2312                    DB      FS5,Q9
FF81  A0                      DB      DS5 OR $80
FF82  230C                    DB      FS5,TR9
FF84  9C                      DB      B4 OR $80
FF85  2006                    DB      DS5,TR9A
FF87  9E                      DB      CS5 OR $80
FF88  2112                    DB      E5,Q9
FF8A  9C                      DB      B4 OR $80
FF8B  2032                    DB      DS5,HELD9
FF8D  1380                    DB      D4,$80
                      ;
                      ;
                      ; DIMINISHED PATTERN
                      ;
      = 0010          Q16     EQU     16
      = 0004          S16     EQU     04
      = 0008          EI16    EQU     08
                      ;
FF8F  FDC3            FFARE5  DW      FADE12
FF91  FEB6                    DW      VIBENL
FF93  1604                    DB      F4,S16
FF95  1604                    DB      F4,S16
FF97  1604                    DB      F4,S16
FF99  1604                    DB      F4,S16
FF9B  1A08                    DB      A4,EI16
FF9D  1C80                    DB      B4,$80
```

```
                        ;
                        ;
                        ;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
                        ;
                        ; FORMAT OF CONTROL BYTE:
                        ; 0-$F = # VECTORS OF GIVEN TYPE (ON OR OFF)
                        ; OR WITH $20 = ZERO AFTER AND QUIT
                        ; OR WITH $80 = BEAM ON FOR STRING
                        ; OR WITH $10 = DO POSITION MOVE
                        ;
                        ;       COMMENT(S)
                        ;       ----------
                        ;          ZEROES THE INTEGRATORS AND SETS ACTIVE GROUND ON RETURN TO USER
                        ;
                        ;       ENTRY VALUES
                        ;       ------------
                        ;          A  = .
                        ;          B  = .
                        ;          X  = .
                        ;          Y  = .
                        ;          U  = .
                        ;          DP = $D0
                        ;
                        ;       RETURN VALUES
                        ;       -------------
                        ;          D  = $0301
                        ;          X  = .
                        ;          Y  = .
                        ;          U  = .
                        ;
FF9F  = 00D0                    SETDP   $D0             ;;  1 BYTE/VECTOR, U INDEX <----
                        ;       =====   ===
                        ;
FF9F  A6A0              NIBBY   LDA     Y+              ;;  GET CONTROL BYTE
FFA1  B7C880                    STA     LASRAM          ;;  .
FFA4  8510                      BITA    #$10            ;;  .
FFA6  2707      ^FFAF           BEQ     NIBB0           ;;  .
FFA8  8D30      ^FFDA           BSR     NIBB4           ;;  .
FFAA  ECA1                      LDD     Y++             ;;  .
FFAC  BDF312                    JSR     POSITN          ;;  NEXT POSITION VECTOR
FFAF  A6A4              NIBB0   LDA     ,Y              ;;  START NEXT ACTUAL VECTOR
FFB1  47                        ASRA                    ;;  CAREFUL TO SIGN EXTEND
FFB2  84F8                      ANDA    #$F8            ;;  MASK DEPEND ON TIMES SHIFTED
FFB4  E6A0                      LDB     Y+              ;;  .
FFB6  58                        ASLB                    ;;  .
FFB7  58                        ASLB                    ;;  .
FFB8  58                        ASLB                    ;;  .
FFB9  58                        ASLB                    ;;  .
FFBA  57                        ASRB                    ;;  .
FFBB  C4F8                      ANDB    #$F8            ;;  .
FFBD  7DC880                    TST     LASRAM          ;;  .
FFC0  2605      ^FFC7           BNE     NIBB1           ;;  .
FFC2  BDF3BE                    JSR     DUFFAB          ;;  .
```

```
FFC5  2003      ^FFCA               BRA     NIBB2            ;;  TO GET NEXT ITEM
                           ;
FFC7  BDF3DF               NIBB1    JSR     DIFFAB           ;;  .
FFCA  B6C880               NIBB2    LDA     LASRAM           ;;  .
FFCD  850F                          BITA    #$0F             ;;  .
FFCF  2705      ^FFD6               BEQ     NIBB3            ;;  .
FFD1  7AC880                        DEC     LASRAM           ;;  .
FFD4  20D9      ^FFAF               BRA     NIBB0            ;;  IF ANOTHER VECTOR
                           ;
FFD6  8520                 NIBB3    BITA    #$20             ;;  .
FFD8  27C5      ^FF9F               BEQ     NIBBY            ;;  GET NEXT CONTROL BYTE
                           ;
FFDA  7EF354               NIBB4    JMP     ZERGND           ;  ZERO INTEGRATORS AND SET ACTIVE GROUND
```

```
                      ; *********************************************************
                      ; *********************************************************
                      ; ***                                                   ***
                      ; ***          I N T E R R U P T   V E C T O R S        ***
                      ; ***                                                   ***
                      ; *********************************************************
                      ; *********************************************************
                      ;
                      ;
FFDD  = FFF2                 ORG     $FFF2
                      ;      ===     =====
                      ;
FFF2  CBF2                   DW      VSWI2           ; SOFTWARE INTERRUPT #3
                      ;
FFF4  CBF2                   DW      VSWI2           ; SOFTWARE INTERRUPT #2
                      ;
FFF6  CBF5                   DW      VFIRQ           ; FAST INTERRUPT
                      ;
FFF8  CBF8                   DW      VIRQ            ; MASKABLE INTERRUPT
                      ;
FFFA  CBFB                   DW      VSWI            ; SOFTWARE INTERRUPT #1
                      ;
FFFC  CBFB                   DW      VSWI            ; NON-MASKABLE INTERRUPT
                      ;
FFFE  F000                   DW      PWRUP           ; POWER-UP RESTART
                      ;
                      ;
0000                         END

no ERRORs,  650 Labels, 7857h bytes not used.Program LWA = FFF2h.



 n A2     0002    111#54
 n A3     000E    112#37
   A4     001A    113#20  119/ 7  120/12  120/15  120/19  120/33  120/37
                  121/ 5  121/43  122/16  122/39  123/52
   A5     0026    114# 3  116/36  117/46  119/39  120/26
 n A6     0032    114#39
 n A7     003E    115#22
   ABSAB  F584     47/47   78#21   79/40
   ABSB   F58B     78/22   78/24   78#47
   ABSBO  F592     78/48   78/50   78#52
   ABSX   C835      5# 7   80/ 8
   ABSY   C834      4#43   79/29   79/39   79/41   79/42   79/53
   ACNTRL D00B      9#34   27/28   67/49   69/19
 n ACTGND F35B     49#42
   ACTPLY C89B      7#46   19/15
   ACTVO  F36A     49/ 6   49#49
 n ADOT   EA5D     15#47
 n ADROT  F616     85#35
 n ALNROT F603     83#45
```

```
  ANGLE  C836     5# 3   79/37   79/44   80/ 2   80/18   80/19   82/ 9
                 82/12   83/18   84/29   86/16
n APACK  EA7F    17# 2
n APROT  F622    86#45
  APRT0  F62B    86#52   87/29
  APRT1  F635    87# 8   87/30
  APRT2  F637    85/39   86/54   87#10
n ARMOR  FD81   117#14
n ARPEG  0004   120# 6
  ARPEG1 0002   115#53  116/38  116/39  116/40  116/41  116/42  116/43
                116/44  116/45  116/46  116/47  116/48  116/49  116/50
n AS2    0003   112# 4
  AS3    000F   112#40  121/13
  AS4    001B   113#23  116/16  121/12  123/ 6
n AS5    0027   114# 6
n AS6    0033   114#42
  AS7    003F   115#25  118/10  118/17
  ASCII  F9F4    67/45   67/50   68/29  110#23
  ASMESS EAA8    18#33   19/21   19/54   20/ 7
n ASPLAY F690    93#22
n B1FREQ C86B     6# 5
n B2     0004   112# 7
n B2FREQ C86D     6# 6
n B3     0010   112#43
  B4     001C   113#26  118/ 9  118/12  118/14  118/16  118/19  118/21
                118/23  118/27  118/28  118/31  118/35  118/39  118/40
                118/43  118/47  118/51  118/52  119/ 2  119/ 5  120/13
                121/39  121/41  123/31  123/35  123/53
  B5     0028   114# 9  116/34  116/39  116/41  116/43  116/45  116/47
                116/49  119/42  120/24
n B6     0034   114#45
n BACON  C876     6#12
  BCLR   F53F    28/20   70/45   71#13
n BDROT  F613    85# 6
n BIG    0032   120# 7
  BLKFIL F552    73# 5   73/ 7
  BLKMO  F686    91#49   92/22
  BLKMOV F683    91#47  107/18
  BLKMV1 F67F    91#22   91/48
  BOTH9  EAEF    20/ 4   20# 9
  BOX0   F903   103/49  105#21
  BOX1   F906   105#23  105/38
  BOX2   F90F   105/26  105#28
  BOX3   F91B   105/32  105#34
  BOX4   F928   105/29  105/35  105#43
  BOX5   F92A   105/41  105#45
n BXTEST F8FF   105#18
  BYTAD0 F861   100#19  100/23
  BYTAD1 F86D   100/20  100#25  100/29
  BYTAD2 F878   100/26  100#31
  BYTADD F85E    99/21  100#18
n BZERK  FD1D   116# 3
n C3     0005   112#10
```

```
  C4     0011    112*46  115/45  119/24  119/26  119/28  119/31  121/ 7
                 121/18  121/21  122/47
  C5     001D    113*29  116/32  119/32  120/31  121/ 6  122/18  122/19
                 122/20  122/21  122/23  122/24  122/41  122/42  122/43
                 122/44  122/46
n C6     0029    114*12
n C7     0035    114*48
n CLR256 F545     71*48
n CLR80  F550     72*39
  CLRBLK F548     71/14   72*13   72/15
n CLREX  F542     71*31
n CMPASS F593     79*28
  CMPS0  F5B0     79*48   79/52
  CMPS1  F5B2     79/43   79/46   79*51
n CMPS2  F5BF     80* 6
  CMPS3  F5D0     80/11   80*15
  CMPS4  F5D3     80/13   80*18
  CNTRL  D000      9*10   27/26   32/46   32/48   32/52   33/ 6   34/40
                  34/44   34/46   35/ 6   35/11   35/15   35/21   35/36
                  36/41   36/43   36/47   36/49   41/25   41/26   41/27
                  41/29   45/45   47/38   47/43   49/44   49/45   49/46
                  49/48   57/25   57/28   61/11   61/14   63/46   63/49
                  65/32   65/34   67/51   67/52   68/ 2   68/ 3   68/ 4
                  68/12   68/17   68/25   68/28   68/41   69/ 8   69/10
                  69/12   69/14
n COLD   F010     23*43
  COLD0  F01C     23*52   24/33
  COLD1  F029     24/ 4   24* 6
  COLD2  F052     24/22   24*24
  COLD3  F058     24*26   24/30
n CONE   EA3E     15*10
  CONE1  EA51     15/19   15*22
  CONE2  EA57     15/23   15*26
  COSINE F5D9     80*48   82/13
  CRATE  F9EA     95/21  109*40
  CRT    F9E4     95/ 5  109*39
  CRTKEY F101     24/40   24/45   26*17
n CS3    0006    112*13
n CS4    0012    112*49
  CS5    001E    113*32  123/33
n CS6    002A    114*15
n CS7    0036    114*51
  CZERO  F34F     49* 5   61/31   64/11   66/ 9   66/51
  D3     0007    112*16  119/43  119/44
  D3TMR  F55A     73*31   98/13
  D4     0013    112*52  115/38  115/40  118/24  118/32  118/36  118/44
                 118/48  119/ 3  121/46  121/48  123/37
  D5     001F    113*35  115/39  115/41  115/43  116/13  118/29  118/41
                 118/53  119/ 6  119/34  119/40  120/29  120/34  121/40
  D6     002B    114*18  116/37
n D7     0037    114*54
  DAC    D001      9*20   32/43   33/ 2   34/45   35/14   35/20   35/32
                  35/33   35/40   35/41   35/47   36/39   36/44   36/45
                  41/21   45/41   47/37   47/44   49/43   57/24   57/30
                  61/10   61/15   63/45   63/51   65/31   65/36   67/47
                  68/14   68/26   69/ 7   69/11
  DASH   C829      4*30   24/10   24/39   65/38   65/53
  DASHDF F437     65*29   66/37
n DASHPK F46E     66*32
  DBNCE  F1B4     20/41   32*18   98/12
  DCNTRL D002      9*22   27/23
  DCT2   EB1D     20/52   20/54   21* 4
  DCT3   EB29     21/ 5   21/ 7   21*10
  DCT4   EB35     21/11   21/13   21*16
  DDAC   D003      9*23   32/50   33/ 8
n DDOT   EA6D     16*23
  DECBIT F57E     77*34  108/23
  DECTBL F9DC     77/34  109*28
n DECTMR F55E     73*54
  DEFLOK F2E6     20/37   30/36   44*49
  DEL    F57A     75/16   75/34   75/52   76*36   76/37
  DEL13  F57D     68/10   76*50
n DEL20  F579     76*15
  DEL28  F575     51/40   75*51
n DEL33  F571     75*33
n DEL38  F56D     75*15
  DFDOT0 F2D2     43*49   43/53
  DIFA0  F3ED     57/33   61*18
  DIFA1  F3F4     61*22   61/23
  DIFDOT F2D5     14/12   43*51
  DIFFAB F3DF     61*10  125/ 3
n DIFFAX F3CE     58* 6
  DIFFY  F3DD     58/35   60*36   66/46
n DIFLST F3D6     59* 8
n DIFTIM F3D2     58*34
  DOREMI C84D      5*30   93/22   95/39
n DOT    F2C5     43*16
  DOT0   F2CC     43*20   43/21
  DOT8TH 0012    119*17  119/29  119/43
  DOTAB  F2C3     15/52   16/30   42*41
  DOTPCK F2DE     44*24   44/27
n DOTTIM F2BE     41*52
  DOTX   F2C1     42*18   43/51   44/26
n DPACK  EA8D     17*31
  DPIO   F1AA     27/19   30/26   31*13   48/31
  DPRAM  F1AF     23/18   24/36   25/18   28/15   31*34   85/36   86/46
                  98/ 2   98/23
  DQTR   0024    120* 5  120/39  120/40
n DROT   F610     84*29
n DS3    0008    112*19
  DS4    0014    113* 2  121/10  121/14
  DS5    0020    113*38  116/17  116/35  123/27  123/29  123/32  123/36
n DS6    002C    114*21
n DS7    0038    115* 4
  DSHD1  F45C     65/45   65*54
  DSHDF  F434     24/28   65* 6
  DSHDF0 F459    65*53   66/ 4
  DSHDF1 F433    64*35   65/49   66/ 8
n DSHIP  F393    53*48
  DSHPK0 F476    66*35   66/38   66/42   66/47
  DSHPK1 F47E    66/36   66*40
  DSHPK2 F485    66/40   66*44
  DSHPK3 F48D    66/45   66*49
n DSTARS E90B    13*26
  DSTRS1 E90D    13/ 6   13*28
  DTMR0  F560    73/32   74* 3
  DTMR1  F563    74* 4   74/ 8
  DTMR2  F569    74/ 5   74* 7
  DUFFAB F3BE    57*24  124/53
n DUFFAX F3AD    54*32
  DUFFY  F3BC    55/ 6   56*52   66/41
n DUFLST F3B5    55*30
n DUFTIM F3B1    55* 5
  DWELL  C828     4*29   28/28   41/52   43/19
n DZERO  F34A    48*31
n E3     0009   112*22
  E4     0015   113* 5  116/15  121/ 8  121/19  121/22  121/45
  E5     0021   113*41  116/ 5  116/ 6  116/ 7  116/ 8  116/ 9  116/10
                116/11  116/18  116/19  116/20  116/21  116/22  116/23
                116/52  119/33  119/36  120/27  120/32  122/22  122/45
                123/34
  E6     002D   114*24  116/40  116/44  116/48  116/51
n E7     0039   115* 7
n EDGE   C811     3*39
  EGG    000C   117* 9  117/16  117/19  117/29  117/30  117/31  117/34
                117/44
  EI16   0008   123*44  123/52
  EIGHTH 000C   120* 3  120/13  120/14  120/15  120/19  120/20  120/21
                120/24  120/26  120/28  120/30  120/37  120/38
  EIT    0008   118* 2  118/10  118/17  118/23  118/25  118/27  118/29
                118/31  118/33  118/35  118/37  118/39  118/41  118/43
                118/45  118/47  118/49  118/51  118/53  119/ 2  119/ 4
n ENDZ   FD67   116*52
  EPOT0  C81F     4*16   20/43   28/34   34/30
n EPOT1  C820     4*17
  EPOT2  C821     4*18   20/44   28/36
  EPOT3  C822     4*19   34/35
n ETMP1  C883     7*21
n ETMP10 C88C     7*30
n ETMP2  C884     7*22
n ETMP3  C885     7*23
n ETMP4  C886     7*24
n ETMP5  C887     7*25
n ETMP6  C888     7*26
n ETMP7  C889     7*27
n ETMP8  C88A     7*28
n ETMP9  C88B     7*29
  EXPL0  F95B   107/13  107*36
n EXPL1  F95F   107*38
```

```
  EXPL2  F968    107/34  107/39  107*44
  EXPL3  F97B    107/51  108* 2
  EXPL4  F97D    107/48  107/52  108* 3
  EXPL5  F987    108/ 5  108/ 7  108* 9
  EXPL6  F98F    108/11  108*13
  EXPL7  F991    108/10  108*14
  EXPL8  F997    108*18  108/25
  EXPL9  F99E    108/20  108*22
  EXPLA  F9BC    108/34  108*37
  EXPLB  F9C2    108/17  108*40
  EXPLC  F9C9    107/37  108*44
n EXPLOD F92E    107*12
n F1FREQ C86F      6* 7
n F3     000A    112*25
  F4     0016    113* 8  121/ 4  121/16  121/20  122/12  122/13  122/14
                 122/15  122/17  122/35  122/36  122/37  122/38  122/40
                 123/ 7  123/ 8  123/ 9  123/11  123/12  123/13  123/15
                 123/48  123/49  123/50  123/51
  F5     0022    113*44  116/38  116/42  116/46
  F6     002E    114*27  116/50
n F7     003A    115*10
  FADE   C84F      5*31   94/ 2   96/31   97/47   98/41   98/52   99/ 8
  FADE0  FD69    116/ 3  116*53  123/25
  FADE1  FE28    118/ 6  119* 9  122/10  122/33  123/ 4
  FADE12 FDC3    117/14  117*48  123/46
  FADE2  FE66    119/21  119*48
  FADE3  FEB2    120/ 9  120*43
  FADE4  FEE8    115/36  121/ 2  121*26
  FADE8  FF16    121/37  121*53
  FADEA  C85E      5*43   94/21   94/53   96/32   99/16
  FADEB  C85F      5*44   94/12
n FADEC  C860      5*45
n FEAST  C871      6* 8
n FF03   FF44    122*33
n FFARE1 FEF8    121*37
n FFARE2 FF26    122*10
n FFARE3 FF62    123* 4
n FFARE4 FF7A    123*25
n FFARE5 FF8F    123*46
n FIBTBL FC36    111*20
  FRAME  C826      4*26   24/ 2   24/19   24/31   24/42   24/43   25/42
                  30/22   30/24
  FRMTIM C83D      5* 8   28/31   30/33
  FRWAIT F192     20/33   24/15   25/23   30*22   98/ 9
  FRWT0  F19E     30*30   30/31
  FRWT1  F1A2     27/31   30*33
n FS3    000B    112*28
  FS4    0017    113*11  120/17  120/20  120/35  120/38  121/49  123/10
                 123/14
  FS5    0023    113*47  116/33  119/35  119/38  120/16  120/25  120/30
                 123/28  123/30
n FS6    002F    114*30
n FS7    003B    115*13
```

```
n FSTARS E8FD    12*51
  FSTR   CB71     8*21   12/19   14/ 7
  G2     0000   111*48  119/45
  G3     000C   112*31  119/23  119/25  119/27  119/29  119/30  121/17
  G4     0018   113*14  115/42  118/ 8  118/11  118/13  118/15  118/18
                118/20  118/22  118/25  118/26  118/30  118/33  118/34
                118/37  118/38  118/42  118/45  118/46  118/49  118/50
                118/54  119/ 4  119/46  120/11  120/14  120/18  120/21
                120/36  120/39  120/41  121/ 9  121/11  121/15  121/23
                121/24  121/42  121/44  121/47  121/50  121/51
  G5     0024   113*50  115/44  116/14  119/37  119/41  120/22  120/23
                120/28  120/40
n G6     0030   114*33
n G7     003C   115*16
  GAMCRT 0000     2*32   24/44
n GAP    C86A     6* 4
n GS2    0001   111*51
n GS3    000D   112*34
  GS4    0019   113*17  116/12
n GS5    0025   113*53
n GS6    0031   114*36
n GS7    003D   115*19
  H4     0030   122*54  123/10  123/14
  HA     0028   115*54  116/51
  HAFF2  001E   121*35  121/50
  HALF   003C   119*19  119/45
  HAQ    0032   120*50  121/23
  HAT    0030   118* 4  119/ 6
  HELD9  0032   123*23  123/36
  HISCOR CBEB     6*31   23/45   75/33
n HISCR  F8D8   103*13
  HISCR0 F8DE   103*17  103/19
  HISCR1 F8E4   103/15  103*21
n HR4    000A   123* 2
n IENABL D00E     9*38
  IFLAG  D00D     9*37   30/30   48/ 6   48/13   61/22   64/ 3   65/43
                 66/ 3
  IMSC0  F173    28*24   28/25
  INPSG0 F275    37*18   37/20
  INPUT  F1BA    23/17   32*38
  INPUT0 F1EA    33*17   33/21
n INT1Q  F29D    39*20
n INT2Q  F2A1    39*46
n INT3Q  F2A5    40*20
  INTALL F18B    23/36   29*23
  INTENS F2AB    39/21   39/47   40/21   41*21
  INTMAX F2A9    14/11   19/10   19/47   24/18   25/26   40*46   98/14
  INTMSC F164    28*15   29/23
  INTPIA F14C    27*19   29/24
  INTPSG F272    29/26   37*17
  INTREQ F533    37/21   70*43   94/ 9   96/22
n ISTARS E8E3    12*18
  ITA    0014   120*52  121/15  121/19
  ITB    000A    120*53  121/16  121/20
  JBIT0  F1FB     34*32   34/36
  JBIT1  F1FF     34*35   35/25
  JBIT2  F20B     34/33   34*44
  JBIT3  F213     34*49   34/50
  JBIT4  F22D     35/12   35*19
  JBIT5  F235     35/17   35*23
  JBIT6  F236     35/16   35/22   35*24   35/48
  JBIT7  F23A     35*31   35/45
  JBIT8  F240     35/ 8   35*35
  JBIT9  F24C     35/37   35*43
  JOYBIT F1F8     20/45   34*30
n JOYSTK F1F5     33*53
n K1FREQ C874      6*11
  KEPALV F9F0     44/49  109*46
  KEY0   C812      3*41   32/38   98/36
  KEY1   C813      3*42   98/49
  KEY2   C814      3*43   99/ 4
  KEY3   C815      3*44   98/34
n KEY4   C816      3*46
n KEY5   C817      3*47
n KEY6   C818      3*48
n KEY7   C819      3*49
  LAG    C83C      5* 7   89/39   89/48   89/52   98/ 6   98/25   98/29
  LASRAM C880      6*23  124/35  124/51  125/ 4  125/ 7
n LATUS  C868      6* 2
  LCOS0  F665     88/42   89*39
  LCOS1  F66F     89/42   89*46
  LCOS2  F676     89/47   89*50
  LCOS3  F679     89/44   89*52
  LCOS4  F67E     89/53   90* 2
  LCSINE F663     83/50   89*38
  LDIFFY F3DA     58/ 7   60*10   61/30
  LDUFFY F3B9     54/33   56*28
  LEG    C83B      5* 6   23/44   25/30   83/45   88/18   89/18   89/43
                  89/46
  LIST   C823      4*22   13/46   33/53   35/ 7   43/49   43/52   56/28
                  60/10   61/28   65/ 6   65/48   66/ 5   85/ 6   86/48
                  87/ 3   87/ 8   87/28  101/39  101/40  101/48  101/52
  LNROT  F601     10/44   83*18
  LOGO0  F10C     24/20   26*22
n LOGO1  F118     26*25
n LOGO2  F124     26*28
n LPACK  F40C     62*49
n LROT90 F5FF     82*43
  LSINE  F65D     83/47   88*41
n M.END  EE2F     22*36
  MARK0  F0E9     24/25   26* 4
  MARK1  F0FD     24/ 8   26*12
  MCSINE F661     87/11   87/24   89*18
  MESAGE C82C      4*34   67/22   68/16   68/33
  MGAME  F79F     97*41   98/20
n MLTY16 E7D2     11*34
```

```
  MLTY8  E7B5    10*42   11/36
  MPLAY  F794    97*39   98/17
n MRASTR F498    67*45
  MSEC20 3075     2*15   28/30
  MSINE  F65B    87/15   87/20   88*18
  MSSPOS F37A    25/29   25/36   51*39   99/24
n NEDGE  C873     6*10
  NEWGEN C855     5*34   94/17   94/47   94/51   95/10   95/26   95/32
  NIBB0  FFAF   124/37  124*41  125/ 8
  NIBB1  FFC7   124/52  125* 3
  NIBB2  FFCA   124/54  125* 4
  NIBB3  FFD6   125/ 6  125*10
  NIBB4  FFDA   124/38  125*13
  NIBBY  FF9F   124*34  125/11
  NOTES  FC8D    92/46  111*49
  OFF10  F8EF   103*48  104/23
n OFF1BX F8E5   103*42
n OFF2BX F8F3   104*17
  OPTION C87A     6*18   28/18   98/19   98/47   99/ 9
  OPTN0  F7B1    97/49   97*51
  OPTN1  F7B6    97/52   97*54
  OPTN2  F7C5    98* 9   98/33   98/48   99/ 5   99/14
  OPTN3  F7F1    98/26   98*30
  OPTN4  F80C    98/42   98*44
  OPTN5  F810    98/37   98/39   98*47
  OPTN6  F821    98/50   99* 4
  OPTN7  F82A    98/53   99/ 2   99/ 7   99* 9
  OPTN8  F82C    98/ 7   98/28   98/45   99*10
  OPTN9  F835    98/18   98/21   99*16
  OPTNA  F84E    99/20   99*27
n PACK1X F408    62*25
n PACK2X F404    61*54
  PACKET F410    63*43   64/10
  PCK0   F425    64* 3   64/ 4
  PCNTRL D00C     9*35   47/41   49/24   50/15
n PEDGE  C872     6* 9
  PLAYRS C879     6*17   20/ 3   97/54   98/16   98/38   98/44
  PMIR0  F282    38*18   38/21
n POSIT1 F30C    46*32
  POSIT2 F308    24/26   44/50   46*12
  POSITB F30E    46/13   46*52
  POSITD F2FC    17/ 5   18/36   45*41   51/39
  POSITN F312    42/41   45/22   47*37  124/40
  POSITX F310    44/52   47*18
  POSWID F2F2    17/34   45*18
n POT0   C81B     4*10
n POT1   C81C     4*11
n POT2   C81D     4*13
n POT3   C81E     4*14
  POTRES C81A     4* 4   35/44
n PROT   F61F    86*16
  PSCOR1 7FA0     2*17   19/52   21/33
  PSCOR2 7F10     2*18   20/ 5   21/34
  PSCRPT EDA3    19/16   21*33
n PSGLST F27D    37*44
  PSGMIR F284    37/45   38*20
  PSTN0  F318    45/46   47*40
  PSTN1  F33B    47/54   48* 5
  PSTN2  F33D    48/ 3   48* 6   48/ 7
  PSTN3  F341    48* 9   48/10
  PSTN4  F345    47/52   48*13   48/14
  PUNZRO 00CE     2*11   47/40
  PWRUP  F000    23*35  126/24
  PZERO  00CC     2*10   49/23   50/14
n Q16    0010   123*42
  Q3R    0006   122*29  122/35  122/39  122/40  122/41  122/45
  Q4     000F   122*52  123/ 6
  Q9     0012   123*20  123/28  123/34
  QHELD  0032   122* 8  122/23
  QHOLD  0032   122*31  122/46
  QR     0018   119*16  119/23  119/24  119/25  119/26  119/27  119/28
                119/31  119/33  119/35  119/37  119/39  119/42
  QRTR2  0006   121*34  121/39  121/40  121/41  121/42  121/43  121/44
                121/45  121/46  121/47  121/48  121/49
  QRTR3  000F   122* 6  122/12  122/16  122/17  122/18  122/22
  QU     0010   117*54  118/ 9  118/16
  QUO    0018   117* 8  117/16  117/25  117/31  117/40  117/45
  QUR    0018   120* 4  120/15  120/16  120/21  120/22  120/32  120/34
n RAMMES CA00     6*24
  RANCID C87D     6*21   28/22   28/24  108/38
  RAND0  F51A    69/39   70* 5
  RAND1  F51D    70* 6   70/17
n RAND3  F511    69*37
  RANDOM F517    14/34   14/37   15/12   70* 3  108/31
n RANPOS E98A    14*32
  RANPS1 E991    14*37   14/39   14/41
  RASTER F495    18/38   51/41   67*22
  RATEA  C858     5*37   94/31  107/16  107/22  107/25  107/28  108/40
  RATEB  C85A     5*39  108/ 9
  RATEC  C85C     5*41  107/32  108/18  108/22  108/26
  REG0   C800     3*16   28/18   28/19   36/18   37/44   38/42
n REG1   C801     3*17
n REG2   C802     3*18
n REG3   C803     3*19
n REG4   C804     3*20
n REG5   C805     3*21
n REG6   C806     3*22
n REG7   C807     3*23
n REG8   C808     3*24
n REG9   C809     3*25
n REGA   C80A     3*26
n REGB   C80B     3*27
n REGC   C80C     3*28
n REGD   C80D     3*29
n REGE   C80E     3*30
  REPLAY F687    24/14   25/22   92*20
  REQ0   C83F     5#14   38/43   70/44
n REQ1   C840     5#15
n REQ2   C841     5#16
  REQ3   C842     5#17   96/33
n REQ4   C843     5#18
  REQ5   C844     5#19  109/13
  REQ6   C845     5#20   95/ 7   95/ 8   95/13   95/14   95/23   95/24
                 95/29   95/30  108/42  108/43
  REQ7   C846     5#21   95/17  107/49  108/ 2
  REQ8   C847     5#22   96/53
n REQ9   C848     5#23
n REQA   C849     5#24
n REQB   C84A     5#25
  REQC   C84B     5#26   97/13  108/29
n REQD   C84C     5#27
  REQOT0 F291    38#46   38/51
  REQOT1 F299    38/48   38#50
  REQOUT F289    24/17   25/25   38#42
  RESTC  C857     5#36   94/44   96/30
  RESTFL CBFE     6#38   23/40   23/43
  REVB.0 EFE7    23#17   23/52
  ROWTRI FC2C    80/ 6  111#14
  RRVB01 F01F    23/22   24# 2
n RRVB11 F0A7    25#20
  RSTPOS F378    51#13   52/40   54/ 7   99/26
  RSTR0  F4A5    67#51   69/16
n RSTR1  F4B8    68#13
  RSTR2  F4C7    68#20   68/23
  RSTR3  F4CB    68/18   68#22
  RSTR4  F4EB    68/36   68#37   68/40
  RSTR5  F50A    68/30   69#18
  RSTSIZ F373    18/11   50#41   52/12
  RTRIGS FC6D    81/23  111#35
  S16    0004   123#43  123/48  123/49  123/50  123/51
  SAMP0  F9CF   109#14  109/18  109/20
  SAMP1  F9DB   109/15  109#21
  SATUS  C867     5#49  107/12  107/15
  SBTN   C8A0     7#17   20/40
  SCAL1X 007F     2#12   45/18   45/43   46/32   62/25
  SCAL2X 00FF     2#13   46/12   61/54
  SCLR   F84F    23/46   99/18   99#44
  SCOR1  C8A8     8#17   19/53   21/30
  SCOR2  C8AF     8#18   20/ 6   21/31
n SCRADD F87C   100#54
n SCRBTH EACF    19#47
n SCRMBL FDD3   118# 6
  SCRMES EAB4    19#10   20/38
  SCRPTR ED9F    19/19   21#30
  SEED   C87B     6#20   28/23   70/ 5
n SELOPT F7A9    97#47
  SETAMP F9CA   107/42  108/14  109#12
  SHIFT  D00A     9#33   43/17   43/23   47/42   49/25   50/16   61/18
                 61/26   63/52   64/ 7   65/40   65/46   65/54   66/ 6
```

```
                68/21
  SHIPO  F3A3   53/54   54* 3
n SHIPX  F391   53*20
  SIN0   F5E5   81/26   81*28
  SIN1   F5FC   81/30   81*32
  SINCOS F5EF   82* 8   83/46   85/38   86/50
  SINE   F5DB   81*23   82/10
  SINTH  0004  118* 3  118/12  118/14  118/19  118/21
  SIP    0006  117*10  117/17  117/18  117/20  117/21  117/32  117/33
               117/35  117/36
  SIT    0004  117*11  117/22  117/23  117/24  117/26  117/27  117/28
               117/37  117/38  117/39  117/41  117/42  117/43
  SIZRAS C82A    4*31   19/13   19/50   25/21   50/42   68/13   68/44
                98/ 5
  SJOY   C881    7*18   20/42
n SMESS  EA9D   18* 6
  SNAR   0010  117*12  117/16  117/17  117/18  117/19  117/20  117/21
               117/22  117/23  117/24  117/25  117/26  117/27  117/28
               117/29  117/30  117/31  117/32  117/33  117/34  117/35
               117/36  117/37  117/38  117/39  117/40  117/41  117/42
               117/43  117/44  117/45
  SNTH   0006  120* 2  120/11  120/12  120/17  120/18  120/35  120/36
n SOLAR  FE38  119*21
n SPEKT  C878    6*14
n SPLAY  F68D   92*46
  ST000  E915   13*34   14/ 2   14/ 4   14/13
  ST010  E920   13/35   13*43
  ST101  E8F1   12*25   12/30
  ST201  E904   13* 2   13/ 4
  STACK  CBEA    6*30   23/35
  STAR1  EDE0   12/18   21*44
n STAR2  EDE8   21*49
n STAR3  EDF0   21*54
n STAR4  EDF8   22* 6
n STAR5  EE00   22*11
n STAR6  EE08   22*16
n STAR7  EE10   22*21
n STAR8  EE18   22*26
  STKAD0 F882  101*25  101/51
  STKAD1 F88F  101/29  101*33
  STKAD2 F895  101/31  101*38
  STKAD3 F897  101/27  101*39
  STKAD4 F8A5  101/43  101*45
  STKAD5 F8AE  101/46  101*49
  STKAD6 F8B7  102* 2  102/ 9
  STKAD7 F8C6  102/ 4  102*10
n STKADD F880  101*24
  STORM  E000   10*10   25/ 8
n SWEEP  FE76  120* 9
  SXTNTH 0006  119*18  119/30  119/44
  T1HOC  D005    9*26   47/45   61/19   63/53   65/41
n T1HOL  D007    9*28
n T1LOL  D006    9*27
```

```
   T1LOLC D004      9#25   14/ 5   15/50   16/26   18/ 9   27/29   45/19
                   45/44   46/52   55/ 5   56/ 3   58/34   59/36   63/20
 n T2HOC  D009      9#31
   T2LOLC D008      9#30   30/34
 n TDIFFY F3D8     59#36
 n TDUFFY F3B7     56# 3
 n TEMP1  C88F      7#34
 n TEMP10 C898      7#43
 n TEMP2  C890      7#35
 n TEMP3  C891      7#36
 n TEMP4  C892      7#37
 n TEMP5  C893      7#38
 n TEMP6  C894      7#39
 n TEMP7  C895      7#40
 n TEMP8  C896      7#41
 n TEMP9  C897      7#42
   TENSTY C827      4#28   41/22
   TH3R   0002    122#30  122/36  122/37  122/38  122/42  122/43  122/44
 n TIMER  EB11     20#51
   TMR1   C89C      8# 4   20/51   20/53   21/ 2
   TMR2   C89F      8# 7   21/ 4   21/ 6   21/ 8
   TMR3   C8A2      8#10   21/10   21/12   21/14
   TMR4   C8A5      8#13   21/16   21/18   21/20
 n TONEA  C861      5#46
   TONEB  C863      5#47   94/15
   TONEC  C865      5#48   94/16   96/52
   TPACK  F40E     17/ 8   17/38   62/ 2   62/26   63#20
 n TPLAY  F692     93#46
   TPLY0  F6B3     92/21   94#21
   TPLY1  F6B8     94#23   94/28
   TPLY2  F6C0     94/25   94#27
   TPLY3  F6CA     94#33   94/43
   TPLY4  F6D2     94/35   94#37
   TPLY5  F6E3     94#47   95/47
   TPLY6  F6EA     94/49   94#51
   TPLY7  F6EC     94/18   94#52
   TPLY8  F712     95/ 3   95#21
   TPLY9  F735     95/18   95#43
   TR4    0005    122#53  123/ 7  123/ 8  123/ 9  123/11  123/12  123/13
   TR8    0007    115#51  116/ 5  116/ 6  116/ 7  116/ 8  116/ 9  116/10
                  116/18  116/19  116/20  116/21  116/22  116/23
   TR9    000C    123#21  123/30
   TR9A   0006    123#22  123/32
   TRATH3 0005    122# 7  122/13  122/14  122/15  122/19  122/20  122/21
 n TREK   FEC6    121# 2
   TRIGGR C80F      3#38   23/20   32/18   32/19   98/27
   TRIQ   001E    120#51  121/ 6  121/ 9  121/12
   TRQTR  000E    115#52  116/11  116/14  116/17  116/34  116/37
   TRV    0006    115#34  115/39  115/39  115/41
   TSTAT  C856      5#35   24/ 5   25/15   25/40   92/20   93/47   96/23
   TUNE   C853      5#33   94/ 7   94/52   95/43   95/45   96/28  107/24
                  107/27  107/47  109/12
   TXPS0  F38A     52#40   52/43
```

```
  TXSZO  F3A3    52*12   52/15
n TXTPOS F38C    52*42
  TXTSIZ F385    24/24   25/39   52*14
  VCTRX  FD0D    24/13  115*36
n VEQ    0012   115*33
  VFIRQ  CBF5     6*34  126/16
  VH     001E   115*32  115/44
  VIBA   C859     5*38  107/50
  VIBB   C85B     5*40  107/38
  VIBC   C85D     5*42  107/30  108/16  108/24
  VIBE   C851     5*32   94/ 5   94/30
  VIBEO  FD79   116/ 4  117* 2  118/ 7
  VIBENL FEB6   115/37  117/15  119/22  120/10  120*44  121/ 3  121/38
                122/11  122/34  123/ 5  123/26  123/47
  VIRQ   CBF8     6*35  126/18
  VSWI   CBFB     6*36  126/20  126/22
  VSWI2  CBF2     6*33  126/12  126/14
n WAIT   EAF0    20*33
  WAIT9  EB41    21/17   21/19   21*22
  WARM   F06C    23/21   23/41   24*36
  WARM0  F084    24*47   25/12
  WARM1  F092    25/ 5   25* 8
  WARM2  F097    25/ 3   25/ 7   25*11
  WARM3  F09E    25/ 9   25*15
  WARM4  F0A4    25*18   25/41   25/44
  WARM5  F0D2    25/31   25*37
n WARM6  F0E7    25*45
  WCSINE C839     5* 5   24/41   25/13   25/14   25/28   82/14   89/38
  WEIRD  FC24    80/ 3  111*12
  WIN1   F8CA   102*34  102/37
  WIN2   F8D5   102/38  102*42
  WIN3   F8D6   102/35  102*44
  WINNER F8C7   102*31  103/13
  WRPSG  F259    36*38   38/18   38/49
  WRREG  F256    36*18   37/18
  WSINE  C837     5* 4   25/16   25/37   82/11   88/41
  XACON  C877     6*13  107/36  107/41  107/44
n XATUS  C869     6* 3
n XPLAY  F742    96*22
  XPLY0  F748    95/49   96*28
  XPLY1  F74E    94/45   96*31
  XPLY2  F759    96*35   96/51
  XPLY3  F766    96/37   96*43
  XPLY4  F76D    96/41   96*49
  XPLY5  F778    96*54   97/14
  XPLY6  F788    97/ 3   97*10
  XPLY7  F78C    97/ 8   97*12
  XPLY8  F793    97*15   98/31   98/35
  XTMR0  C82E     4*37   74/ 3   98/32   99/13
  XTMR1  C82F     4*38   98/30   99/11
n XTMR2  C830     4*39
n XTMR3  C831     4*40
n XTMR4  C832     4*41
```

```
 n XTMR5  C833      4*42
   ZERGND F354     13/37   13/43   15/54   16/32   27/30   43/54   44/25
                   44/53   48/33   49*23   69/20  125/13
   ZERO   F36B     44/51   50*14
   ZSKIP  C824      4*23   24/12   24/29   49/ 5   66/32   66/34   66/50
   ZSTR   CB81      8*22   12/20   12/53   13/50
```