```
*****************************************************************
*                                                               *
*                          TI INVADERS  DISK VERSION            *
*                                                               *
*                PROGRAMMER:  GARTH DOLLAHITE                   *
*                             JIM DRAMIS                        *
*                                                               *
*                          AUGUST  1981                         *
*                          JANUARY 1982                         *
*                                                               *
*                FILE:      HC2.JEDI.SRC.INVADER               *
*                                                               *
*****************************************************************
*
       IDT    'TI'
       TITL   'INVADERS'
       DEF  SFIRST,SLAST,SLOAD                                  TEST
SFIRST EQU  $                                                   TEST
SLOAD  EQU  $
       B    @START
*****************************************************************
* SYSTEM EQUATES                                                *
*****************************************************************
SCAN   EQU  >000E          ADDRESS OF SCAN ROUTINE
SOUBLK EQU  >83CC          SOUND BLOCK ADDRESS
SOUDC  EQU  >83CE          SOUND DOWN COUNT ADDRESS
SCANW  EQU  >83D4          SCAN WINDOW TO KEEP VDP REG 1
GPLWS  EQU  >83E0          GPL WORKSPACE
VDPRD  EQU  >8800          VDP READ DATA WINDOW ADDRESS
VDPWD  EQU  >8C00          VDP WRITE DATA WINDOW ADDRESS
VDPADR EQU  >8C02
*****************************************************************
* CONSTANTS                                                     *
*****************************************************************
       EVEN
H0020  DATA >0020
H0030  DATA >0030
H0082  DATA >0082
H0182  DATA >0182
H0282  DATA >0282
H0302  DATA >0302
H0604  DATA >0604
H001C  DATA >001C
H9E7E  DATA >9E7E
HFFFF  DATA >FFFF
D1800  DATA 1800
D0600  DATA 600
H01    BYTE >01
H05    BYTE >05
H07    BYTE >07
H08    BYTE >08
H09    BYTE >09
H0A    BYTE >0A
H0B    BYTE >0B
H0C    BYTE >0C
H0D    BYTE >0D
H0E    BYTE >0E
H0F    BYTE >0F
H10    BYTE >10
H14    BYTE >14
H23    BYTE >23
```

```
H28     BYTE >28
H2A     BYTE >2A
D21     BYTE 21
H18     BYTE >18
H1E     BYTE >1E
H31     BYTE >31
H32     BYTE >32
D55     BYTE 55
H3C     BYTE >3C
H40     BYTE >40
H50     BYTE >50
H48     BYTE >48
H4E     BYTE >4E
H59     BYTE >59
H80     BYTE >80          ADDRESS OF DROPPINGS PTRN IN SDB
H81     BYTE >81
H82     BYTE >82
H83     BYTE >83
H84     BYTE >84
H85     BYTE >85
H86     BYTE >86
H87     BYTE >87
H88     BYTE >88
H8C     BYTE >8C
H90     BYTE >90
HCF     BYTE >CF          POINTER TO LOWER SCREEN BLANKS
HD0     BYTE >D0
HFD     BYTE >FD
H68     BYTE >68
HSHPTR  BYTE >A4          HORIZONTAL SHOT DB POINTER
HBC     BYTE >BC
HC0     BYTE >C0
HC8     BYTE >C8
HE2     BYTE >E2          DOUBLE SIZE SPRITES
HF1     BYTE >F1          WHITE TEXT, BLACK BACKDROP
NEGA    BYTE >F6          NEGATIVE 10
HF9     BYTE >F9          WHITE TEXT, LIGHT RED BACKDROP
NEG6    BYTE >FA          NEGATIVE 6
HFE     BYTE >FE
NEG2    EQU  HFE          NEGATIVE 2
H02     EQU  H0302+
H03     EQU  H0302
H04     EQU  H0604+1
H06     EQU  H0604
H00     EQU  H0020
H20     EQU  H0020+1
H30     EQU  H0030+1
TSHOT   EQU  H01          TIMER TO MOVE SHOT/DROPS
TSCAN   EQU  H01          TIMER TO SCAN
TSSHP   DATA 1200         TIMER TO START SHIP
SEC2    DATA 120          TWO SECOND DELAY
SBTR    DATA >0257        BOTTOM RIGHT CORNER OF SHIELDS PTRN
SHTOPR  DATA >0220        SHIELD TOP ROW POINTER
UPL     DATA >0062        UPPER LEFT CORNER OF INVADER SCREEN
BTR     DATA >0270        BOTTOM RIGHT CNR OF INVADER SCREEN
D20     EQU  H14
D60     EQU  H3C
SUPMGN  EQU  H10          SHOT UPPER MARGIN
GLMRGN  EQU  H20          GUN LEFT MARGIN
GRMRGN  BYTE 216          GUN RIGHT MARGIN
MSGPAU  BYTE >FF,>70,>61,>75,>73,>65,>FF
MSGBLK  BYTE >86,>86,>86,>86,>86,>86,>86
```

```
MSGTST BYTE >74,>65,>73,>74,>FF
MSGSPD BYTE >63,>5C,>5F,>67,>FF,>63,>60,>55
       BYTE >55,>54,>38,>69,>3F,>5E,>39
SHFREQ BYTE >14,>A0,>14  SHOT SOUND FREQUENCIES
BITMSK BYTE >80,>40,>20,>10,>08,>04,>02,>01    BIT MASK
GHPTRS BYTE >A0,>A0,>A0,>9C             GUN HIT PATTERNS
       BYTE >9C,>9C,>98,>98,>98
DEMOM  BYTE >FE,1,08,>FE,2,14         DEMO MOVES
       BYTE >FE,1,04,>FE,2,4
       BYTE >FE,1,12,>FE,2,6
       BYTE 1,1,2,2,1,1,2,2,1,1,2,2,1,1,2,2
       BYTE >FE,1,10,>FE,2,12
       BYTE >FE,1,04,>FE,2,6
       BYTE >FE,1,08,>FE,2,10
       BYTE >FE,1,12,>FE,2,10
       BYTE >FE,1,08
       BYTE 2,2,1,1,2,2,1,1
       BYTE >FE,2,6
       BYTE >FE,1,04,>FE,2,6
       BYTE >FF,>FD
MSGSCN BYTE >63,>53,>62,>55,>55,>5E,>38,>40    MSG FOR CHEAT
       BYTE >40,>3D,>44,>40,>09
MSGFF  BYTE >FF,>FF,>FF,>FF,>FF,>FF,>FF,>FF
       BYTE >FF,>FF,>FF,>FF,>FF,>FF,>FF,>FF
       BYTE >FF,>FF,>FF,>FF,>FF,>FF,>FF,>FF
       BYTE >FF,>FF,>FF,>FF
MSGRED BYTE >FF,>FF,>FF,>FF,>FF,>70,>72,>65
       BYTE >73,>73,>FF,>72,>65,>64,>6F,>FF
       BYTE >6F,>72,>FF,>62,>61,>63,>6B,>FF
       BYTE >FF,>FF,>FF,>FF
ZERO50 BYTE >50,>50,>50,>50,>50          HIGH SCORE VDP >1008 ZEROED
IROWS  BYTE >00,>00,>10,>10,>20,>20      INVADER ROWS
TSSAU  DATA >3EBC,>8C01             TITLE SCREEN SAUCER
       DATA >3EBC,>A80A
       BYTE >D0
IINVAD DATA >0001,>1415,>2627       INITIAL INVADERS FOR TS
IINVD  DATA >0202,>FDFE,>0202       INITIAL INV DIRECTIONS
INVADS BSS  6
INVDIR BSS  6
TIME   DATA >0000              TIME COUNT TO GO TO DEMO
WAIT   DATA >0000
WNDPOS DATA >0000
MAXPOS BSS  1
STORAG BSS  5                  - JEDI -
CHEAT  BSS  1                  - JEDI -
CHEATS BSS  1                  - JEDI -
SCRNUM BSS  1                  - JEDI -
SAVR11 BSS  1                  - JEDI -
SAVREG BSS  1                  - JEDI -
INDEX3 BSS  2                  - JEDI -
TITLFG BSS  1                  - JEDI -

*****************************************************************
* CHARACTERS TO GO TO PATTERN NAME TABLE                        *
*****************************************************************
SVALL  BYTE >73,>61,>75,>63,>65,>72      SAUCER VALUE MSG
       BYTE >FF,>76,>61,>6C,>75,>65
       BYTE >5A,>FF,>F0,>53,>50
COPYRT BSS  8             COPY RIGHT SYMBOL
CHAR   BSS  512           CHARACTER SET
***CHAR1  BSS  512           CHARACTER SET  - JEDI -
SCOREL TEXT 'SCORE 00000   HI SCORE '
```

```
HISC    TEXT '00000'                    CURRENT HIGH SCORE
        BYTE >FD
PSCORE  BSS  5
HSCORE  BSS  5
SHIELD  BYTE >30,>31,>32,>FE,>FF,3
        BYTE >36,>37,>38,>FE,>FF,4
        BYTE >3C,>3D,>3E,>FE,>FF,3
        BYTE >42,>43,>44,>FE,>FF,10
        BYTE >33,>34,>35,>FE,>FF,3
        BYTE >39,>3A,>3B,>FE,>FF,4
        BYTE >3F,>40,>41,>FE,>FF,3
        BYTE >45,>46,>47,>FD
TITLES  DATA >FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFE4,>D9B0,>D9DE
        DATA >E6D1,>D4D5,>E2E3,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFA0,>9F99
        DATA >9EA4,>A3FF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>FF22,>2385,>7F81,>80FF,>FFFF,>1011,>8180
        DATA >7F82,>80FF,>FFFF,>0607,>0105,>7F83,>80FF,>FFFF
        DATA >FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>9899,>A470,>A498,>9570,>A995,>9C9C,>9FA7
        DATA >70A3,>91A5,>9395,>A2FF,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>999E,>70A4,>9895,>7093,>959E,>A495,>A270
        DATA >969F,>A2FF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>9D91,>A899,>9DA5,>9D70,>A09F,>999E,>A4A3
        DATA >FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>8F8F,>8FFF,>FFFF
        DATA >FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>95A8,>A4A2,>9170,>9D99,>A3A3,>999C,>9570
        DATA >9291,>A395,>7091,>A791,>A294,>9594,>FFFF,>FFFF
        DATA >FFFF,>91A4,>7083,>8080,>8070,>A09F,>999E,>A4A3
        DATA >7E70,>709F,>9E95,>7092,>91A3,>95FF,>FFFF,>FFFF
        DATA >FFFF,>A295,>A091,>99A2,>9594,>7095,>A695,>A2A9
        DATA >7081,>807C,>8080,>8070,>A09F,>999E,>A4A3,>7EFF
        DATA >FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>FFFF,>FFFF,>FFFF,>FFD9,>DEE6,>D1D4,>D5E2
        DATA >B0DF,>E0E4,>D9DF,>DEE3,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>FFFF,>FFFF,>C1BE,>B0DD,>D5E2,>D5DC,>E9B0
        DATA >D1D7,>D7E2,>D5E3,>E3D9,>E6D5,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>FFFF,>FFFF,>C2BE,>B0D4,>DFE7,>DEE2,>D9D7
        DATA >D8E4,>B0DE,>D1E3,>E4E9,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>695F,>6562,>3053
        DATA >585F,>5953,>554F,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF,>FFFF
        DATA >FFFF,>FFFF,>FF90,>8189,>8881,>70A4,>95A8,>91A3
        DATA >7099,>9EA3,>A4A2,>A59D,>959E,>A4A3,>FFFF,>FFFF
BUNKER  BYTE >80,>FE,>81,10
        BYTE >FF,>FF,>FE,>81,10,>82
```

```
        BYTE >FE,>FF,8,>83
        BYTE >FE,>FF,22
        BYTE >84,>FE,>FF,8
        BYTE >85,>FE,>86,22,>87
        BYTE >FD
PAKTB   TEXT 'PRESS ANY KEY TO BEGIN'
        BYTE >FD
*****************************************************************
* INITIAL SPRITE ATTRIBUTE BLOCK                                *
*****************************************************************
ISAB    DATA >C000,>800D      DROPINGS
        DATA >C000,>800D
        DATA >C000,>800D
        DATA >C000,>800D
        DATA >C000,>840F      SHOT
        DATA >C000,>8C01      SAUCER WINDOWS
SAUBOD  DATA >C000,>A80A      SAUCER BODY
        DATA >C000,>9009      SPLAT
GUNIP   DATA >997C,>8807      GUN
        DATA >A978,>9404      LIFT
EGUNIP  DATA >AA48,>8804      EXTRA GUNS
        DATA >AA30,>8804
        DATA >AA30,>8800
        BYTE >D0
*****************************************************************
* SPRITE DESCRIPTOR BLOCK                                       *
*****************************************************************
DROPP   DATA >0000,>0000,>0000,>0000   DROPPINGS
        DATA >0000,>4080,>4080,>4080
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
SHOT    DATA >8080,>8080,>8000,>0000   SHOT
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
BGUN    DATA >0000,>0000,>0000,>0808   WHOLE GUN
        DATA >0808,>1C7F,>FFFF,>FF63
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>8080,>8000
SHIPW   DATA >0000,>0000,>8400,>0000   SAUCER WINDOWS
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>2000,>0000
        DATA >0000,>0000,>0000,>0000
SPLAT   DATA >0500,>2200,>10A5,>0245   SPLATTERED INVADER
        DATA >2052,>08A0,>1560,>0000
        DATA >0000,>2000,>4028,>0080
        DATA >2050,>8028,>4030,>0000
LIFT    DATA >FFFF,>FF00,>0000,>0000   LIFT
        DATA >0000,>0000,>0000,>0000
        DATA >FFFF,>FF00,>0000,>0000
        DATA >0000,>0000,>0000,>0000
GUNRR   DATA >0000,>0000,>0000,>0000   REMAINS (HIT ON R)
        DATA >0808,>1878,>FCFC,>FE62
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
GUNRM   DATA >0000,>0000,>0000,>0000   REMAINS (HIT IN M)
        DATA >0000,>0063,>E3F7,>FF63
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>8080,>8000
GUNRL   DATA >0000,>0000,>0000,>0000   REMAINS (HIT ON L)
        DATA >0808,>0C0F,>1F1F,>3F23
        DATA >0000,>0000,>0000,>0000
```

```
        DATA >0000,>0000,>8080,>8000
HDROPP  DATA >0000,>0000,>0000,>0000    INVADER HORIZ SHOT
        DATA >0000,>0000,>0000,>342A
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
SHIP    DATA >0103,>3FFF,>FFFF,>3F07    SAUCER 15
        DATA >0000,>0000,>0000,>0000
        DATA >0080,>F8FE,>FEFE,>F8C0
        DATA >0000,>0000,>0000,>0000
        DATA >0307,>3FFF,>FFFF,>3F0F    SAUCER 14
        DATA >0000,>0000,>0000,>0000
        DATA >0080,>F0FC,>FCFC,>F0C0
        DATA >0000,>0000,>0000,>0000
        DATA >0207,>3FFF,>FFFF,>3F0F    SAUCER 13
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>E0F8,>F8F8,>E080
        DATA >0000,>0000,>0000,>0000
        DATA >060F,>3FFF,>FFFF,>3F0F    SAUCER 12
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>C0F0,>F0F0,>C000
        DATA >0000,>0000,>0000,>0000
        DATA >040E,>3FFF,>FFFF,>3F0E    SAUCER 11
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>80E0,>E0E0,>8000
        DATA >0000,>0000,>0000,>0000
        DATA >0C3F,>FFFF,>FF3F,>0000    SAUCER 10
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>C0C0,>C000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >1C3E,>FFFF,>FF3E,>0000    SAUCER 9
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>8080,>8000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >183C,>FFFF,>FF3C,>0000    SAUCER 8
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >1038,>FEFE,>FE38,>0000    SAUCER 7
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >30FC,>FCFC,>7800,>0000    SAUCER 6
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >20F8,>F8F8,>7000,>0000    SAUCER 5
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >60F0,>F060,>0000,>0000    SAUCER 4
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >40E0,>E000,>0000,>0000    SAUCER 3
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >C0C0,>0000,>0000,>0000    SAUCER 2
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >8000,>0000,>0000,>0000    SAUCER 1
```

```
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
ZEROES  DATA >0000,>0000,>0000,>0000    SAUCER 0
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
*****************************************************************
* PATTERN GENERATOR SETS                                        *
*                                                               *
*    INVADER LABEL DIGITS:                                      *
*       3.   INVADER NUMBER - 1 THRU 8                          *
*       4.   POSITION WITHIN CHARACTERS - 0,2,4,6,8             *
*       5.   WHICH HALF OF INVADER - L,R                        *
*       6.   TOP OR BOTTOM HALF FOR POSITION 8                  *
*    THESE LABELS ARE FOR REFERENCE ONLY.   ONLY THE FIRST      *
*    LABEL FOR EACH INVADER IS USED IN THE PROGRAM.             *
*****************************************************************
*
*** EYES ***
*
IP80L   DATA >0012,>0000,>0000,>0000
IP80R   DATA >0000,>0000,>0000,>0000
IP82L   DATA >0004,>0000,>0000,>0000
IP82R   DATA >0080,>0000,>0000,>0000
IP84L   DATA >0001,>0000,>0000,>0000
IP84R   DATA >0020,>0000,>0000,>0000
IP86L   DATA >0000,>0000,>0000,>0000
IP86R   DATA >0048,>0000,>0000,>0000
IP88LT  DATA >0000,>0000,>0012,>0000
IP88RT  DATA >0000,>0000,>0000,>0000
IP88LB  DATA >0000,>0000,>0000,>0000
IP88RB  DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
*       PAGE
*** 00111 FLASHER ***
*
IPA0L   DATA >0000,>0000,>0000,>0000
IPA0R   DATA >0000,>0000,>0000,>0000
IPA2L   DATA >0000,>0000,>0000,>0000
IPA2R   DATA >0000,>0000,>0000,>0000
IPA4L   DATA >0303,>0007,>0705,>090A
IPA4R   DATA >F0F0,>C0F8,>F828,>2414
IPA6L   DATA >0000,>0001,>0101,>0100
IPA6R   DATA >FCFC,>30FE,>FE4A,>4AB4
IPA8LT  DATA >0000,>0000,>3F3F,>0C7F
IPA8RT  DATA >0000,>0000,>0000,>0080
IPA8LB  DATA >7F92,>522D,>0000,>0000
IPA8RB  DATA >8040,>8000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
*       PAGE
*** 11000 FLASHER ***
*
IP90L   DATA >3F3F,>0C7F,>7F52,>92A1
IP90R   DATA >0000,>0080,>8080,>4040
IP92L   DATA >0F0F,>031F,>1F14,>140B
```

```
IP92R   DATA >C0C0,>00E0,>E0A0,>A040
IP94L   DATA >0000,>0000,>0000,>0000
IP94R   DATA >0000,>0000,>0000,>0000
IP96L   DATA >0000,>0000,>0000,>0000
IP96R   DATA >0000,>0000,>0000,>0000
IP98LT  DATA >0000,>0000,>0000,>0000
IP98RT  DATA >0000,>0000,>0000,>0000
IP98LB  DATA >0000,>0000,>0000,>0000
IP98RB  DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
*       PAGE
*** BAT ***
*
IP80L   DATA >0018,>3C5A,>4242,>2400
IP80R   DATA >0000,>0000,>0000,>0000
IP82L   DATA >0000,>364F,>4600,>0000
IP82R   DATA >0000,>C020,>2000,>0000
IP84L   DATA >0000,>001D,>2301,>0000
IP84R   DATA >0000,>00B8,>C180,>0000
IP86L   DATA >0000,>0304,>0400,>0000
IP86R   DATA >0000,>6CF2,>6200,>0000
IP88LT  DATA >0000,>0000,>005A,>BD99
IP88RT  DATA >0000,>0000,>0000,>0000
IP88LB  DATA >4200,>0000,>0000,>0000
IP88RB  DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
*       PAGE
*** PULSAR ***
*
IP70L   DATA >3C7E,>FFDB,>FFFF,>7E3C
IP70R   DATA >0000,>0000,>0000,>0000
IP72L   DATA >000F,>1F16,>1F1F,>0F00
IP72R   DATA >0000,>8080,>8080,>0000
IP74L   DATA >0000,>0102,>0301,>0000
IP74R   DATA >0000,>8040,>C080,>0000
IP76L   DATA >0000,>0101,>0101,>0000
IP76R   DATA >00F0,>F868,>F8F8,>F000
IP78LT  DATA >0000,>0000,>003C,>7E5A
IP78RT  DATA >0000,>0000,>0000,>0000
IP78LB  DATA >7E7E,>3C00,>0000,>0000
IP78RB  DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
*       PAGE
*** HALF FLASHER WITH EYES ***
*
IP60L   DATA >3F3F,>0C7F,>7F52,>92A1
IP60R   DATA >0000,>0080,>8080,>4040
IP62L   DATA >0F0F,>031F,>1F14,>140B
IP62R   DATA >C0C0,>00E0,>E0A0,>A040
IP64L   DATA >0004,>0000,>0000,>0000
IP64R   DATA >0080,>0000,>0000,>0000
IP66L   DATA >0000,>0000,>0000,>0000
IP66R   DATA >0012,>0000,>0000,>0000
```

```
IP68LT  DATA >0000,>0000,>3F3F,>0C7F
IP68RT  DATA >0000,>0000,>0000,>0080
IP68LB  DATA >7F92,>522D,>0000,>0000
IP68RB  DATA >8040,>8000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
*       PAGE
*** TURNOVER ***
*
IP50L   DATA >0000,>001C,>3E36,>6363
IP50R   DATA >0000,>0000,>0000,>0000
IP52L   DATA >000C,>0E07,>0301,>0101
IP52R   DATA >0000,>0000,>8080,>8080
IP54L   DATA >0101,>0101,>0101,>0101
IP54R   DATA >8080,>8080,>8080,>8080
IP56L   DATA >0000,>0000,>0000,>0000
IP56R   DATA >0018,>3870,>E0C0,>C0C0
IP58LT  DATA >0000,>0000,>0000,>0000
IP58RT  DATA >0000,>0000,>0000,>0000
IP58LB  DATA >0000,>FFFF,>0000,>0000
IP58RB  DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
*       PAGE
*** SNAKE ***
*
IP40L   DATA >3E2A,>3E08,>0808,>1C1C
IP40R   DATA >0000,>0000,>0000,>0000
IP42L   DATA >0F0A,>0F02,>0408,>1C1C
IP42R   DATA >8080,>8000,>0000,>0000
IP44L   DATA >0F0A,>0F02,>0100,>0101
IP44R   DATA >8080,>8000,>0080,>C0C0
IP46L   DATA >0000,>0000,>0000,>0101
IP46R   DATA >F8A8,>F820,>4080,>C0C0
IP48LT  DATA >0000,>0000,>003E,>2A3E
IP48RT  DATA >0000,>0000,>0000,>0000
IP48LB  DATA >1C1C,>0000,>0000,>0000
IP48RB  DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000,>0000,>0000
*       PAGE
*** INVADER THREE ***
*
IP30L   DATA >081C,>3E6B,>7F14,>2241
IP30R   DATA >0000,>0000,>0000,>0000
IP32L   DATA >0207,>0F1A,>1F05,>0805
IP32R   DATA >0000,>80C0,>C000,>8000
IP34L   DATA >0001,>0306,>0701,>0204
IP34R   DATA >80C0,>E0B0,>F040,>2010
IP36L   DATA >0000,>0001,>0100,>0000
IP36R   DATA >2070,>F8AC,>FC50,>8850
IP38LT  DATA >0000,>0000,>081C,>3E6B
IP38RT  DATA >0000,>0000,>0000,>0000
IP38LB  DATA >7F14,>2214,>0000,>0000
IP38RB  DATA >0000,>0000,>0000,>0000
        DATA >0000,>0014,>0022,>1C00
```

```
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0014,>0000,>1C22
        DATA >0000,>0000,>0000,>0000
*       PAGE
*** INVADER TWO ***
*
IP20L   DATA >6322,>3E6B,>FFBE,>A236
IP20R   DATA >0000,>0000,>8080,>8000
IP22L   DATA >1808,>2F2A,>3F0F,>0830
IP22R   DATA >C080,>A0A0,>E080,>8060
IP24L   DATA >0602,>0306,>0F0B,>0A03
IP24R   DATA >3020,>E0B0,>F8E8,>2860
IP26L   DATA >0100,>0202,>0300,>0003
IP26R   DATA >8C88,>FAAA,>FEF8,>8806
IP28LT  DATA >0000,>0000,>6322,>BEAA
IP28RT  DATA >0000,>0000,>0000,>8080
IP28LB  DATA >FF3E,>22C1,>0000,>0000
IP28RB  DATA >8000,>0080,>0000,>0000
        DATA >0000,>0014,>0022,>1C00
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0014,>0000,>1C22
        DATA >0000,>0000,>0000,>0000
*       PAGE
*** INVADER ONE ***
*
IP10L   DATA >1EFF,>CCFF,>FF12,>21C0
IP10R   DATA >00C0,>C0C0,>C000,>00C0
IP12L   DATA >073F,>333F,>3F04,>080C
IP12R   DATA >80F0,>30F0,>F080,>40C0
IP14L   DATA >010F,>0C0F,>0F01,>020C
IP14R   DATA >E0FC,>CCFC,>FC20,>100C
IP16L   DATA >0003,>0303,>0300,>0000
IP16R   DATA >78FF,>33FF,>FF48,>84CC
IP18LT  DATA >0000,>0000,>1EFF,>CCFF
IP18RT  DATA >0000,>0000,>00C0,>C0C0
IP18LB  DATA >FF12,>2133,>0000,>0000
IP18RB  DATA >C000,>0000,>0000,>0000
        DATA >0000,>3300,>0021,>1E00
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>3300,>0000,>1E21
        DATA >0000,>0000,>0000,>0000
*       PAGE
* BOTTOM BORDER
*
BBORDR  DATA >0000,>7F7F,>7F70,>7070
        DATA >0000,>FFFF,>FF00,>0000
        DATA >0000,>FFFF,>FF07,>0707
        DATA >7070,>7070,>7070,>7070
        DATA >0707,>0707,>0707,>0707
        DATA >7070,>7070,>707F,>7F7F
        DATA >0000,>0000,>00FF,>FFFF
        DATA >0707,>0707,>07FF,>FFFF
*                                   YELLOW SAUCER VALUES
YSVAL   DATA >FFF0,>5255,>FFF0,>5550
        DATA >FFF0,>5755,>FF51,>5050
        DATA >FF51,>5255,>FF51,>5550
        DATA >FF52,>5050,>FF53,>5050
*                                   RED SAUCER VALUES
RSVAL   DATA >FFF0,>5350,>FFF0,>5353
        DATA >FEF0,>5356,>FDF0,>5358
        DATA >FCF0,>5452,>FBF0,>5455
        DATA >FAF0,>5550,>F9F0,>5556
```

```
        DATA >F8F0,>5653,>F7F0,>5751
        DATA >F6F0,>5853,>F551,>5050
        DATA >F451,>5255,>F351,>5657
        DATA >F252,>5550,>F155,>5050
        DATA >F078,>7878
SHLDP   DATA >0003,>070F,>1F3F,>3F3F
        DATA >00FF,>FFFF,>FFFF,>FFFF
        DATA >00C0,>E0F0,>F8FC,>FCFC
        DATA >3F3F,>3F3F,>3F3F,>3F3F
        DATA >FFFF,>FFFF,>C381,>8181
        DATA >FCFC,>FCFC,>FCFC,>FCFC
*       PAGE
*****************************************************************
* PATTERN COLOR TABLE                                           *
*****************************************************************
PCTBLB  DATA >9090              EYES
PCTBLA  DATA >6060              00111 FLASHER
PCTBL9  DATA >4040              11000 FLASHER
PCTBL8  DATA >2020              BAT
PCTBL7  DATA >D0D0              PULSAR
PCTBL6  DATA >C0C0              HALF FLASHER WITH EYES
PCTBL5  DATA >7070              TURNOVER
PCTBL4  DATA >A0A0              SNAKE
*
* INITIAL PATTERN COLOR TABLE
*
PCTBL3  DATA >3030              INVADER 3
        DATA >5050              INVADER 2
        DATA >6060              INVADER 1
        DATA >F0F0,>F000        SHIELDS
        DATA >F0F0              NUMBERS
        DATA >F0F0,>F0F0        LETTERS
        DATA >4000              BOTTOM SCREEN
        DATA >0000,>0000,>0000,>0000
        DATA >0000,>0000        TOP SCREEN
        DATA >0000              TOP SCREEN
*
* TITLE SCREEN PATTERN COLOR TABLE
*
TSPCTB  DATA >3030                          INVADER 3
        DATA >5050                          INVADER 2
        DATA >6060                          INVADER 1
        DATA >F0F0,>F0F0,>F0F0,>F0F0        CHARACTER SET 1
        DATA >D0D0,>D0D0,>D0D0,>D0D0        CHARACTER SET 2
        DATA >7070,>7070,>7070,>7070        CHARACTER SET 3
        DATA >0000                          BLANKS
*       PAGE
*****************************************************************
* SOUND LIST                                                    *
*****************************************************************
IMSL    BYTE 6,>9F,>BF,>D0,>FF,>CF,>3F,1    INVADER MOVE SOUND
        BYTE 1,>DF,0
GOSL    BYTE 4,>9F,>BF,>DF,>FF,0            ALL GENERATORS OFF
GSSL    BYTE 8,>9B,>BB,>DF,>FF              GUN SHOT
        BYTE >98,>14,>A0,>14,0
IHSL    BYTE 7,>9F,>BF,>DF,>F2,>C0,>01,>E7,1 INVADER HIT
        BYTE 2,>C0,>03,1
        BYTE 2,>C0,>05,1
        BYTE 1,>FF,0
SHSL    BYTE 6,>9F,>B2,>DF,>FF,>A0,>08,6    SAUCER HIT
        BYTE 1,>B4,4
        BYTE 1,>B6,3
```

```
        BYTE 1,>B8,2
        BYTE 1,>BF,0
GHSL    BYTE 5,>9F,>BF,>DF,>F6,>E4,3      GUN HIT
        BYTE 1,>F4,15
        BYTE 1,>F5,13
        BYTE 1,>F6,11
        BYTE 1,>F7,9
        BYTE 1,>F8,7
        BYTE 2,>F9,>E5,5
        BYTE 1,>FA,4
        BYTE 1,>FB,4
        BYTE 2,>FC,>E6,3
        BYTE 1,>FD,2
        BYTE 1,>FF,0
S2HSL   BYTE 6,>9F,>B2,>DF,>FF,>A0,>28,6      SAUCER 2 HIT
        BYTE 1,>B4,4
        BYTE 1,>B6,3
        BYTE 1,>B8,2
S2MSL   BYTE 7,>9F,>BF,>DF,>FF,>C0,>02,>E3,1   SAUCER 2 MOVING
        BYTE 1,>FE,2
        BYTE 1,>FC,2
        BYTE 1,>FA,2
        BYTE 1,>F8,2
        BYTE 1,>F6,2
        BYTE 1,>F4,2
        BYTE 1,>F2,2
        BYTE 1,>F0,2
        BYTE 0,>12,>11
SLS2SL  BYTE 7,>9F,>BF,>DF,>FF,>C0,>02,>E3,2   SLOW SAUCER 2
        BYTE 1,>FE,4
        BYTE 1,>FC,4
        BYTE 1,>FA,4
        BYTE 1,>F8,4
        BYTE 1,>F6,4
        BYTE 1,>F4,4
        BYTE 1,>F2,4
        BYTE 1,>F0,4
        BYTE 0,>12,>50
S1MSL   BYTE 7,>9F,>BF,>DF,>FF,>C0,>06,>E3,1   SAUCER 1 MOVING
        BYTE 1,>FE,2
        BYTE 1,>FC,2
        BYTE 1,>FA,2
        BYTE 1,>F8,2
        BYTE 1,>F6,2
        BYTE 1,>F4,2
        BYTE 1,>F2,2
        BYTE 1,>F0,2
        BYTE 0,>13,>00
***
CHAR1   DATA >0000,>0000,>0000,>0000,>2020,>2020,>2020,>0020
        DATA >4848,>4800
        DATA >0000,>0000,>0048,>FC48,>48FC,>4800,>103C,>5038
        DATA >1478,>1000,>C0C4,>0810,>2040
        DATA >8C0C,>6090,>9060,>6094,>8874,>0810,>2000,>0000
        DATA >0000,>0810,>2020,>2020,>1008
        DATA >4020,>1010,>1010,>2040,>0000,>4830,>CC30,>4800
        DATA >0000,>1010,>7C10,>1000,>0000
        DATA >0000,>0030,>1020,>0000,>0000,>7C00,>0000,>0000
        DATA >0000,>0000,>3030,>0004,>0810
        DATA >2040,>8000,>3844,>4444,>4444,>4438,>1030,>5010
        DATA >1010,>107C,>7884,>0408,>1020
        DATA >40FC,>7884,>0438,>0404,>8478,>0C14,>2444,>84FC
```

```
        DATA >0404,>F880,>80F8,>0404,>8478
        DATA >7880,>80F8,>8484,>8478,>FC04,>0408,>1020,>4040
        DATA >7884,>8478,>8484,>8478,>7884
        DATA >8484,>7C04,>0478,>0030,>3000,>0030,>3000,>0030
        DATA >3000,>0030,>1020,>0008,>1020
        DATA >4020,>1008,>0000,>007C,>007C,>0000,>0040,>2010
        DATA >0810,>2040,>3844,>0408,>1010
        DATA >0010,>0078,>849C,>A498,>807C,>7884,>8484,>FC84
        DATA >8484,>F844,>4478,>4444,>44F8
        DATA >7884,>8080,>8080,>8478,>F844,>4444,>4444,>44F8
        DATA >FC80,>80F0,>8080,>80FC,>FC80
        DATA >80F0,>8080,>8080,>7884,>8080,>9C84,>8478,>8484
        DATA >84FC,>8484,>8484,>7C10,>1010
        DATA >1010,>107C,>0404,>0404,>0484,>8478,>8890,>A0C0
        DATA >A090,>8884,>4040,>4040,>4040
        DATA >407C,>84CC,>B484,>8484,>8484,>84C4,>A494,>8C84
        DATA >8484,>FC84,>8484,>8484,>84FC
        DATA >F884,>8484,>F880,>8080,>7884,>8484,>8494,>8874
        DATA >F884,>8484,>F890,>8884,>7884
        DATA >8078,>0404,>8478,>7C10,>1010,>1010,>1010,>8484
        DATA >8484,>8484,>8478,>4444,>4444
        DATA >2828,>1010,>8484,>8484,>84B4,>CC84,>8484,>4830
        DATA >3048,>8484,>4444,>4428,>1010
        DATA >1010,>FC04,>0810,>2040,>80FC,>3820,>2020,>2020
        DATA >2038,>0080,>4020,>1008,>0400
        DATA >7010,>1010,>1010,>1070,>1028,>4482,>0000,>0000
        DATA >0000,>0000,>0000,>00FC,>0000,>0000,>0000,>0000

*****************************************************************
* SOUND LIST
*****************************************************************
IMOVES  EQU  >1100                  INVADER MOVE
GENOFF  EQU  >1110                  ALL GENERATORS OFF
GNSHOT  EQU  >1120                  GUN SHOT
INVHIT  EQU  >1130                  INVADER HIT
SHPHIT  EQU  >1180                  SHIP HIT, SHOT OFF
GUNHIT  EQU  >11A0                  GUN HIT (EXPLOSION)
S2HIT   EQU  >1200                  SAUCER 2 (RED) HIT
SAUCR2  EQU  >1211                  SAUCER 2 MOVING SOUND
SLOWS2  EQU  >1250                  SLOW SAUCER 2
SAUCR1  EQU  >1300                  SAUCER 1 MOVING SOUND
*       PAGE
*****************************************************************
* CONSTANTS                                                     *
*****************************************************************
O1VALB  DATA >0055          OPTION 1 INVADER POINTS
O1VALA  DATA >0050
O1VAL9  DATA >0045
O1VAL8  DATA >0040
O1VAL7  DATA >0035
O1VAL6  DATA >0030
O1VAL5  DATA >0025
O1VAL4  DATA >0020
O1VAL3  DATA >0015
        DATA >0010
        DATA >0005
O2VALB  DATA >0110          OPTION 2 INVADER POINTS
O2VALA  DATA >0100
O2VAL9  DATA >0090
O2VAL8  DATA >0080
O2VAL7  DATA >0070
O2VAL6  DATA >0060
```

```
O2VAL5 DATA >0050
O2VAL4 DATA >0040
O2VAL3 DATA >0030
       DATA >0020
       DATA >0010
GAMEOV BYTE >67,>61,>6D,>65,>FF,>6F     'GAME OVER' LETTERING
       BYTE >76,>65,>72
REGLD  BYTE >00,>E2,>00,>0E,>01,>06,>00,>11
*****************************************************************
* KEY CODE EQUATES                                              *
*****************************************************************
REDOKY EQU  H06            'REDO' KEY
K8     BYTE >38            '8' - REDO ON 4A WITHOUT FCTN
KR     BYTE >52            'R' - REDO ON 4  WITHOUT SHIFT
BACKKY EQU  H0F            'BACK' KEY
FIREK1 BYTE >12           FIRE KEY 1 - Q, Y, JOYSTICKS' FIRE
FIREK2 EQU  H0D           FIRE KEY 2 - ENTER ON 4,PERIOD ON 4A
LEFTK  EQU  H02           LEFT ARROW KEYS - S, J
RIGHTK EQU  H03           RIGHT ARROW KEYS - D, K
HFF    EQU  HFFFF         'NO KEY RETURNED' CODE, NEGATIVE 1
*      PAGE
*****************************************************************
* COLOR TABLE                                                   *
*****************************************************************
CLEAR  EQU  H00
BLACK  EQU  H01
MGREEN EQU  H02
LGREEN EQU  H03
DBLUE  EQU  H04
LBLUE  EQU  H05
DRED   EQU  H06
CYAN   EQU  H07
MRED   EQU  H08
LRED   EQU  H09
DYELL  EQU  H0A
LYELL  EQU  H0B
DGREEN EQU  H0C
MAGENT EQU  H0D
GRAY   EQU  H0E
WHITE  EQU  H0F
*
SHP1CL EQU  DYELL
SHP2CL EQU  LRED
*      PAGE
*****************************************************************
* CPU RAM                                                       *
*****************************************************************
       EVEN
SAB    EQU  >8300
CPURAM EQU  >8300
DROP   EQU  SAB            INVADER DROPPINGS
DROP4  EQU  DROP+12
SHOTPY EQU  DROP4+4        SHOT POSITION
SHOTPX EQU  SHOTPY+1
SHOTDB EQU  SHOTPY+2
SHOTCL EQU  SHOTPY+3
SHPWPY EQU  SHOTPY+4       SHIP WINDOWS POSITION
SHPWPX EQU  SHPWPY+1
SHPWDB EQU  SHPWPY+2
SHPWCL EQU  SHPWPY+3
SHIPPY EQU  SHPWPY+4       SHIP POSITION
SHIPPX EQU  SHIPPY+1
```

```
SHIPDB  EQU  SHIPPY+2
SHIPCL  EQU  SHIPPY+3
SPLATP  EQU  SHIPPY+4      SPLAT POSITION
SPLATC  EQU  SPLATP+3
GUNPY   EQU  SPLATP+4      GUN POSITION
GUNPX   EQU  GUNPY+1
GUNDB   EQU  GUNPY+2
GUNCL   EQU  GUNPY+3       GUN COLOR - ALSO FIRE ENABLE/DISABLE
LIFTPY  EQU  GUNPY+4       LIFT POSITION
EGUN1   EQU  LIFTPY+4      EXTRA GUNS MODIFIED ONLY WHEN NEEDED
EGUN1C  EQU  EGUN1+3
EGUN2   EQU  EGUN1+4
EGUN2C  EQU  EGUN2+3
EGUN3   EQU  EGUN2+4
VLINE   EQU  CPURAM+>25    VDP LINE
ORLINE  EQU  CPURAM+>42    ORIGINAL LINE
RBYTE   EQU  ORLINE
SCREEN  EQU  CPURAM+>5F    SCREEN COUNTER (0-40 IN MULT OF 5)
SCRTCH  EQU  CPURAM+>60    SCRATCH AREA
ADDCH   EQU  SCRTCH+2      AMOUNT TO ADD TO CHANGE CHAR BYTE
GUNDIR  EQU  SCRTCH+4      GUN AUTO MOVE DIRECTION
YREM    EQU  SCRTCH+4
FLAG    EQU  SCRTCH+6
LPDROP  EQU  CPURAM+>68    HORIZ POSITION OF LAST DROP
DEMO    EQU  CPURAM+>69    POINTER TO DEMO MOVES (0=DEMO OFF)
BTL     EQU  CPURAM+>6A    BOTTOM LEFT CORNER OF INV MOVE AREA
LDIGIT  EQU  CPURAM+>6C    LEFT SCORE DIGIT-FIX GUN WHEN CHANGE
HITCNT  EQU  CPURAM+>6D    HIT COUNT
STOPSC  EQU  CPURAM+>6E    FLAG TO STOP SCREEN (GUN HIT/NOINVL)
DRPFLG  EQU  CPURAM+>6F    DROP FLAG
KEYBRD  EQU  CPURAM+>74    KEY BOARD TO SCAN FROM
KEY     EQU  CPURAM+>75    KEY CODE RETURNED
JOYY    EQU  CPURAM+>76
JOYX    EQU  CPURAM+>77
TIMER   EQU  CPURAM+>79
MOTION  EQU  CPURAM+>7A
STATUS  EQU  CPURAM+>7C
RA      EQU  CPURAM+>80    TEN RETURN ADDRESSES
SHPDIR  EQU  CPURAM+>8E    SHIP DIRECTION (-1,+1)
TCSHOT  EQU  CPURAM+>90    TIME COUNT TO MOVE SHOT/DROPS
TCMSHP  EQU  CPURAM+>91    TIME COUNT TO MOVE SHIP
TCSSHP  EQU  CPURAM+>92    TIME COUNT TO START SHIP
TCSCAN  EQU  CPURAM+>94    TIME COUNT TO SCAN
PREKEY  EQU  CPURAM+>95    PREVIOUS KEY CODE RETURNED
IDIR    EQU  CPURAM+>96    INVADER DIRECTION (-2,+2)
CIPN    EQU  CPURAM+>97    CURRENT INVADER PTRN NUM (0,2,4,6,8)
TROWI   EQU  CPURAM+>98    CURRENT TOP ROW OF INVADERS
BROWI   EQU  CPURAM+>9A    CURRENT BOTTOM ROW OF INVADERS
EROWI   EQU  TROWI         END ROW - INDEXED TO TROWI OR BROWI
TMOVEI  EQU  CPURAM+>9C    TIMER TO MOVE INVADERS
TCYCLE  EQU  TMOVEI        TIMING CYCLE
TMSHP   EQU  CPURAM+>9D    TIMER TO MOVE SHIP
CYSSLF  EQU  CPURAM+>9E    CLEAR YELLOW SAUCER SCORE LINE FLAG
OPTION  EQU  CPURAM+>9F
CPUD6   EQU  CPURAM+>D6    SCREEN TIME-OUT COUNTER
*****************************************************************
* WORKING SPACE REGISTERS ( START AT >83A0 )                    *
*****************************************************************
MYWS    EQU  CPURAM+>A0
VDPADD  EQU  R0            VDP ADDRESS TO READ/WRITE TO
VADDLB  EQU  MYWS+1        VDP ADDRESS LOWER BYTE
Y       EQU  VDPADD
```

```
YLB     EQU  VADDLB        Y LOW BYTE
RLOC    EQU  R1            POINTER TO BUFFER TO BE READ INTO
RCOUNT  EQU  R2            # OF BYTES IN BUFFER TO WRITE
WCOUNT  EQU  R2            # OF BYTES IN BUFFER TO WRITE
WLOC    EQU  R3            POINTER TO BUFFER TO BE WRITTEN
INDEX   EQU  R4
INDXLB  EQU  MYWS+9        INDEX LOW BYTE
COUNT   EQU  R5            COUNTER FOR RED/WRITE LOOP
TEMP    EQU  R6            TEMPORARY VDP ADDRESS
X       EQU  R7
VDIGIT  EQU  X             VDP SCORE DIGIT
VDOTLB  EQU  MYWS+15       VDP SCORE DIGIT LOW BYTE
PTRNNO  EQU  R8            PATTERN NUMBER
REG8    EQU  MYWS+16
PTNOLB  EQU  MYWS+17       PATTERN NUMBER LOW BYTE
XREM    EQU  R9            X REMAINDER
REG10   EQU  MYWS+20
YXPOS   EQU  R12           Y,X POSITION
YPOS    EQU  MYWS+24       YXPOS HIGH BYTE
XPOS    EQU  MYWS+25       YXPOS LOW BYTE
BIAS    EQU  R12
CARRY   EQU  R13
MASK    EQU  R13
*       PAGE
*****************************************************************
*                        START OF MAINLINE                      *
*****************************************************************
START   LWPI MYWS                       SET WORKSPACE TO MY AREA
        LI   R1,REGLD                   SET BYTE ADDRESS
        LI   R2,>7F00                   REG WRITE CONSTANT -1
LOOPIV  SWPB R2                         MOVE HIGH TO LOW/LOW TO HIGH
        INC  R2                         SET REGISTER NUMBER
        MOVB *R1+,R2                    SET VALUE FOR OUTPUT
        MOVB R2,@>8C02                  WRITE VALUE
        SWPB R2                         MOVE HIGH TO LOW/LOW TO HIGH
        MOVB R2,@>8C02                  WRITE REGISTER NUMBER
        CI   R2,>8700                   CHECK FOR LAST REGISTER
        JL   LOOPIV                     GO BACK FOR NEXT REGISTER & VALUE
***
***  DATA STACK AND SUBRTN STACK ARE NOW INIT
***  TO >9E,>7E, RESPECTFULLY IN ORDER FOR THE
***  SUBRTN STACK TO POINT TO FREE CPURAM IN THE
***  PROGRAM(>8380,>8381) FOR STACKING DURING
***  CALLS TO GPL THROUGH THE "SCAN" ROUTINE.
***  PREVIOUSLY THE STACK WAS POINTING TO LOCATION
***  >8390,>8391 CPURAM WHICH ADVERSELY AFFECTED
***  "TCSHOT AND TCMSHP" - JEDI ...
***
       MOV  H9E7E,@>8372        ZAP DATA AND SUBROUTINE STACK VALUES
       BL   @RDVCL              SAVE SMALL CHAR SET - JEDI -
       DATA >900,512,CHAR
       BL   @WRTVCL             CLEAR SCREEN
       DATA >FF8,8,ZEROES
       BL   @CLRSCR
       BL   @RDVCL              SAVE COPY RIGHT SYMBOL
       DATA >850,8,COPYRT
       LI   VDPADD,>1008
       LI   WCOUNT,5
       LI   WLOC,ZER050
       BL   @WRITE
       MOVB @H00,@MOTION          NO AUTO SPRITE MOTION
       BL   @WRTVCL             YELLOW SAUCER VALUES
```

```
        DATA >1010,32,YSVAL
        BL   @WRTVCL              RED SAUCER VALUES
        DATA >1040,68,RSVAL
        BL   @WRTVCL              SOUND LISTS
        DATA IMOVES,16,IMSL
        BL   @WRTVCL
        DATA GENOFF,16,GOSL
        BL   @WRTVCL
        DATA GNSHOT,16,GSSL
        BL   @WRTVCL
        DATA INVHIT,20,IHSL
        BL   @WRTVCL
        DATA SHPHIT,26,SHSL
        BL   @WRTVCL
        DATA GUNHIT,60,GHSL
        BL   @WRTVCL
        DATA S2HIT,80,S2HSL
        BL   @WRTVCL
        DATA SLOWS2,60,SLS2SL
        BL   @WRTVCL
        DATA SAUCR1,60,S1MSL
        BL   @WRTVCL              SPRITE DESCRIPTOR BLOCKS
        DATA >400,832,DROPP
        BL   @WTEXT               DEMO MOVES
        DATA >1501,DEMOM,0
        MOVB @H00,@DEMO
        JMP  BLCS
*****************************************************************
* SET UP OPTION SCREEN      'BACK'                              *
*****************************************************************
BACK    BL   @SCANKY
        CB   KEY,HOF
        JEQ  BACK
        BL   @HISCOR
BLCS    MOVB H00,CHEAT            INIT TO NO CHEATING - SCREEN
        MOVB H00,CHEATS           INIT TO NO CHEATING - SPEED
BACKCH  BL   @CLRSCR              BACK FROM THE "CHEAT"
        MOVB H01,TITLFG           JEDI - SET TITLE SCREEN UP FLAG
        BL   @WRTVCL              INVADER PATTERNS
        DATA >800,384,IP3OL
        BL   @WRTVCL              FIRST CHARACTER SET
        DATA >980,512,CHAR
        AI   VDPADD,>200
        BL   @WRITE               SECOND CHARACTER SET
        BL   @WRTVCL
        DATA >0D80,512,CHAR1      THIRD CHAR SET(LARGE CHAR SET) - JEDI
        BL   @WRTVCL              COPY RIGHT SYMBOL
        DATA >C80,8,COPYRT
        BL   @WRTVCL              TITLE SCREEN PATTERN COLOR TABLE
        DATA >380,32,TSPCTB
        CLR  VDPADD               JEDI - TITLE SCREEN TEXT
        LI   WCOUNT,768           JEDI
        LI   WLOC,TITLES          JEDI
        BL   @WRITE               JEDI
        BL   @WRTVCL              DISPLAY TITLE SCREEN SAUCER
        DATA >300,9,TSSAU
        CLR  INDEX
SETII   MOVB @IIINVAD(INDEX),@INVADS(INDEX)  INITIAL INV FOR TS
        INC  INDEX
        CI   INDEX,12
        JL   SETII
        CLR  @TIME                SET TIMERS FOR  TITLE SCREEN
```

```
         CLR   @TIMER
         MOVB  @D1800,@WAIT
         CB    @DEMO,@H00
         JEQ   RESETW
         MOVB  @D0600,@WAIT          SHORTER WAIT ON DEMO
RESETW   MOV   @TSSAU,@WNDPOS        RESET SAUCER WINDOWS
CKWIND   CB    @WNDPOS+1,@HC0
         JEQ   RESETW
         BL    @WRTVCL
         DATA  >300,2,WNDPOS
         INC   @WNDPOS
         CLR   INDEX
MOVEIV   MOVB  @INVADS(INDEX),MASK
         MOVB  @H06,@MAXPOS              FIND INV MAXIMUM POSITION
         CB    @INVDIR(INDEX),@H02
         JEQ   ANDMSK
         MOVB  @H00,@MAXPOS
ANDMSK   ANDI  MASK,>0F00
         CB    MASK,@MAXPOS
         JNE   INCINV
         NEG   @INVDIR(INDEX)
         JMP   IINDX
INCINV   A     @INVDIR(INDEX),@INVADS(INDEX)     MOVE INVADER
IINDX    INCT  INDEX                        NEXT INVADER
         CI    INDEX,4
         JLE   MOVEIV
         BL    @WRTVCL              WRITE FIRST INVADER TO PNT
         DATA  >96,2,INVADS
         BL    @WRTVCL
         DATA  >8C,2,INVADS+2       WRITE SECOND INVADER TO PNT
         BL    @WRTVCL
         DATA  >83,2,INVADS+4       WRITE THIRD INVADER TO PNT
         MOVB  @H00,@OPTION
CTDEMO   CB    @TIMER,@H02            MOVE INVADS EVERY 2 TIMER BYTES
         JL    SCNKEY
         MOVB  @H00,@TIMER-1          ZERO HIGH BYTE OF TIMER
         A     @TIMER,@TIME
         CLR   @TIMER
         C     @TIME,@WAIT
         JL    CKWIND
         CB    @DEMO,@H00             TIME IS UP
         JNE   JUMP05
         MOVB  @H01,@DEMO
         B     @ST2OPT
SCNKEY   CLR   @KEYBRD
CSCAN    BL    @SCANI                   SCAN
         CB    KEY,H2A                "*"
         JEQ   CHK23
         B     @GOOGAM                CONTINUE
JUMP05   B     @ST2OPT
CHK23    BL    @SCANKY
         CB    KEY,HFF
         JEQ   CHK23
         CB    KEY,H2A
         JEQ   CHK23
         CB    KEY,H23                "#"
         JNE   CSCAN                  START AGAIN
CHK2A    BL    @SCANKY
         CB    KEY,HFF
         JEQ   CHK2A
         CB    KEY,H23
         JEQ   CHK2A
```

```
        CB    KEY,H2A           "*"
        JNE   CSCAN             START AGAIN
***
***   "***" WAS ENTERED
***
        LI    VDPADD,0
        LI    WCOUNT,1
SOME20  LI    WLOC,HFF          GET BLANKS(>FF)
        BL    @WRITE
        INC   VDPADD
        CI    VDPADD,769
        JNE   SOME20
***
***   DELETE ALL SPRITES
***
        LI    VDPADD,>0300
        LI    WCOUNT,1
        LI    WLOC,HD0
        BL    @WRITE
***
***   PUT OUT "CHEAT" MESSAGES
***
        LI    VDPADD,165
        LI    WCOUNT,15
        LI    WLOC,MSGSPD       "SLOW SPEED(Y/N)" MESSAGE
        BL    @WRITE
CHKRND  BL    @SCANKY
        CB    KEY,H2A
        JEQ   CHKRND
        CB    KEY,H59           Y OR N ?
        JNE   NOTYES
        MOVB  H01,CHEATS        SET SLOWER SPEED CHEAT FLAG
        JMP   SCRNOW
NOTYES  CB    KEY,H4E
        JNE   CHKRND
SCRNOW  LI    VDPADD,229
        LI    WCOUNT,13
        LI    WLOC,MSGSCN       "SCREEN(00-40)" MESSAGE
        BL    @WRITE
CHKSCN  BL    @SCANKY
        MOVB  STATUS,STATUS
        JEQ   CHKSCN
        SB    H30,KEY
        CB    KEY,H04           0 - 4 ??
        JH    CHKSCN
        MOVB  KEY,SCRNUM        GOT SCREEN LEVEL
CHKNXT  BL    @SCANKY
        MOVB  STATUS,STATUS
        JEQ   CHKNXT
        SB    H30,KEY
        CB    KEY,H09           0 - 9 ??
        JH    CHKNXT
        MOVB  SCRNUM,SCRNUM
        JEQ   DIGZER
        CB    SCRNUM,H01
        JEQ   DIGONE
        CB    SCRNUM,H02
        JEQ   DIGTWO
        CB    SCRNUM,H03
        JEQ   DIGTHR
        MOVB  H28,SCRNUM
        JMP   DIG40
```

```
DIGTHR  MOVB  H1E,SCRNUM
        JMP   DIGZER
DIGTWO  MOVB  H14,SCRNUM
        JMP   DIGZER
DIGONE  MOVB  H0A,SCRNUM
DIGZER  AB    KEY,SCRNUM
DIG40   MOVB  H01,CHEAT          SET FOR CHEAT ON SCREEN NUMBER
        B     @BACKCH            GO BACK TO TITLE SCREEN SELECTION
GOOGAM  CB    @KEY,@H31
        JEQ   OPTN1
        CB    @KEY,@H32
        JEQ   OPTN2
        CB    @KEYBRD,@H00
        JEQ   IKB
        CB    @JOYX,@H00
        JLT   OPTN1
        JGT   OPTN2
IKB     AB    @H01,@KEYBRD
        CB    @KEYBRD,@H02
        JLE   JUMP10
        B     @CTDEMO
JUMP10  B     @CSCAN
OPTN1   MOVB  @H00,@DEMO
        JMP   BPD0
OPTN2   MOVB  @H00,@DEMO
ST2OPT  MOVB  @H02,@OPTION
BPD0    LI    VDPADD,>E80        BLANK PATTERNS >D0+
ZD0     BL    @WRITCL
        DATA  32,ZEROES
        AI    VDPADD,32
        CI    VDPADD,>F80
        JL    ZD0
        BL    @WRTVCL
        DATA  >A00,512,CHAR
        JMP   BLCLRS
*****************************************************************
* SET UP GAME SCREEN      'REDO'                                *
*****************************************************************
REDO    BL    @HISCOR
BLCLRS  BL    @CLRSCR
        MOVB  H00,TITLFG         RESET TITLE SCREEN FLAG
        BL    @WRTVCL            INITIAL SPRITE ATTRIBUTE BLOCK
        DATA  >300,53,ISAB
        LI    VDPADD,>0C00       WRITE OUT THE DATA FOR BOTTOM BORDER
        LI    WCOUNT,64          JEDI
        LI    WLOC,BBORDR        JEDI
        BL    @WRITE             WRITE THE BORDER NOW
        LI    VDPADD,>380        PATTERN COLOR TABLE
        LI    WLOC,PCTBL3        JEDI
        LI    WCOUNT,32          JEDI
        BL    @WRITE             JEDI
        BL    @WTEXT
        DATA  >02,SCOREL,>20
        MOVB  @H50,@LDIGIT
        CB    CHEAT,H00          JEDI
        JEQ   NOCH00             JEDI
        JMP   TESTIT             JEDI
NOCH00  CB    CHEATS,H00         JEDI
        JEQ   NOTEST             JEDI
TESTIT  LI    VDPADD,2           JEDI
        LI    WCOUNT,5           JEDI
        LI    WLOC,MSGTST        JEDI
```

```
        BL    @WRITE              JEDI
***
***  GET HIGH SCORE DIGITS
***
NOTEST LI     VDPADD,>1008
       LI     RCOUNT,5
       LI     RLOC,STORAG
       LI     WLOC,STORAG
       BL     @READ
       LI     VDPADD,25
       BL     @WRITE
       BL     @WTEXT              WRITE BOTTOM SCREEN AREA
       DATA   >2A4,BUNKER,0
       CB     @DEMO,@H00
       JEQ    FSCR
       BL     @WTEXT              'PRESS ANY KEY TO BEGIN'
       DATA   >2C5,PAKTB,>20
       MOVB   @CLEAR,@EGUN1C        DON'T DISPLAY EXTRA GUNS
       MOVB   @CLEAR,@EGUN2C
***
***  DON'T DISPLAY EXTRA GUNS
***
       LI     WLOC,IROWS          (>00)
       LI     WCOUNT,1
       LI     VDPADD,>032B
       BL     @WRITE
       LI     VDPADD,>032F
       BL     @WRITE
FSCR   MOVB   @H00,@SCREEN          INITIALIZE SCREEN COUNT
       CB     CHEAT,H00            JEDI -
       JEQ    NOSET1               JEDI -
       MOVB   SCRNUM,SCREEN        JEDI -
       MOVB   H00,CHEAT            JEDI -
NOSET1 MOVB   @H00,@STOPSC          CLEAR STOP SCREEN FLAG
*      PAGE
*****************************************************************
* SET UP NEXT ROUND - DISPLAY INVADERS, SHIELDS                 *
*****************************************************************
NXTSCR LI     VDPADD,>40          BLANK ROWS 2-20
BLKROW BL     @WRITCL
       DATA   2,HFFFF
       INCT   VDPADD
       CI     VDPADD,>280
       JLE    BLKROW
       LI     VDPADD,>980         REBUILD SHIELD PATTERNS
RSP    BL     @WRITCL
       DATA   48,SHLDP
       AI     VDPADD,48
       CI     VDPADD,>A40
       JLE    RSP
       BL     @WTEXT              SHIELDS TO SCREEN
       DATA   >225,SHIELD,0
       MOVB   @SCREEN,@INDXLB
       ANDI   INDEX,>001E
       CI     INDEX,16            JEDI ( NEW LINE OF CODE HERE)
       JLE    01IV
       LI     INDEX,16
01IV   LI     VDPADD,>10A0        OPTION 1 INVADER VALUES
       LI     WCOUNT,6
       LI     WLOC,01VAL3
       S      INDEX,WLOC
       BL     @WRITE
```

```
        LI    VDPADD,>10A6      OPTION 2 INVADER VALUES
        LI    WLOC,O2VAL3
        S     INDEX,WLOC
        BL    @WRITE
        LI    VDPADD,>380       PATTERN COLOR TABLE
        LI    WLOC,PCTBL3
        S     INDEX,WLOC
        BL    @WRITE
        LI    VDPADD,>800       INVADER PATTERNS
        LI    WCOUNT,384
        LI    WLOC,IP30L
        SLA   INDEX,6              MULTIPLY BY 64
        S     INDEX,WLOC
        BL    @WRITE
        MOVB  @SCREEN,@INDXLB
        ANDI  INDEX,1
        CI    INDEX,1           JEDI
        JNE   ADDONE            JEDI
        CLR   INDEX             JEDI
        JMP   NOADDS            JEDI
ADDONE  AI    INDEX,1           JEDI
NOADDS  MOVB  @HO2,@IDIR
        LI    VDPADD,>82        START INV ON LEFT IF SAU OFF LEFT
        CB    @SHPDIR,@HOO
        JLT   ROWS
        MOVB  @NEG2,@IDIR
        AI    VDPADD,6          START INV ON RIGHT IF SAU OFF RIGHT
ROWS    LI    R10,5             ROWS
NEWROW  LI    R8,11             COLUMNS
        MOVB  @IROWS(INDEX),@SCRTCH
        MOVB  @IROWS(INDEX),@SCRTCH+1
        AB    @HO1,@SCRTCH+1
        LI    WLOC,SCRTCH
        LI    WCOUNT,2
NXTCOL  BL    @WRITE
        INCT  VDPADD            NEXT COLUMN
        DEC   R8
        JGT   NXTCOL
        AI    VDPADD,42         NEXT ROW
        INC   INDEX
        DEC   R10
        JGT   NEWROW
        AB    @HO1,@SCREEN        INCREMENT SCREEN COUNT
        MOVB  @HOO,@CIPN          INVADERS START AT PATTERN O
        MOV   @HOO82,@TROWI       SET ROW PATTERNS
        MOV   @HO182,@BROWI
        MOV   @HO282,@BTL
        LI    VDPADD,>0318        JEDI - RE-INIT SAUCER TO >A8 CHAR
        LI    WCOUNT,4            JEDI -
        LI    WLOC,SAUBOD         JEDI -
        BL    @WRITE              JEDI -
        BL    @RDVCL
        DATA  >300,53,SAB         JEDI - CHANGE TO 53 FROM 52 BYTES READ IN
        MOVB  @HOO,@DRPFLG
        CB    CHEATS,HOO          JEDI
        JEQ   GETSPD              JEDI
        MOVB  H3C,TMOVEI          JEDI
        JMP   HC55                JEDI
GETSPD  MOVB  @D21,@TMOVEI
        CB    @SCREEN,@H10
        JH    HIGHSP
        SB    @SCREEN,@TMOVEI
```

```
        JMP   HC55
HIGHSP  MOVB  @H05,@TMOVEI
HC55    MOVB  @D55,@HITCNT
        MOVB  @SHP1CL,@SHIPCL
        CLR   @TCSHOT
        CLR   @TCSSHP
        CLR   @TIMER
*****************************************************************
* MAIN PROGRAM LOOP                                             *
*****************************************************************
SCANLP  BL    @SCANK                SCAN KEYBOARD, MOVE GUN / FIRE
CKTIME  MOVB  H00,CPUD6             CLEAR SCREEN TIME-OUT COUNTER
        CB    @TIMER,@TCYCLE        CHECK TIME TO MOVE INVADERS
        JL    CKSHOT
        MOVB  @HC0,@SPLATP
        CB    @STOPSC,@H00
        JEQ   BLMI
        CB    @CIPN,@H00
        JEQ   STOPI                  DON'T MOVE WHEN GUN HIT
        BL    @MOVEI
        JMP   RESETT
BLMI    BL    @MOVEI
        CB    @SHIPCL,@SHP2CL
        JEQ   ADJT
        MOVB  @CIPN,INDEX
        ANDI  INDEX,>FB00          WILL BE 0 WHEN CIPN = 0 OR 4
        JNE   BLDRP                MOVE SOUND ON PATTERNS 0,4 ONLY
MSOUND  BL    @SOUND
        DATA  IMOVES
BLDRP   BL    @DROPSP
ADJT    A     @TIMER,@TCSSHP        ADJUST TIME TO START SHIP
RESETT  CLR   @TIMER               RESET TIMERS
        CLR   @TCSHOT
        MOVB  @H00,@TCSCAN
CKSHOT  CB    @TIMER,@TCSHOT        CHECK TIME TO MOVE SHOT/DROPS
        JL    CKMSHP
        BL    @UPSHOT
        BL    @MOVED
        BL    @UPSHOT
        BL    @MOVED
        BL    @UPSHOT
        AB    @TSHOT,@TCSHOT
CKMSHP  CB    @STOPSC,@H00          CHECK TIME TO MOVE SHIP
        JEQ   CTTMS
        CB    @CIPN,@H00
        JNE   CTTMS
        CB    @GUNDB,@GUNIP+2
        JNE   EML                    DON'T MOVE WHEN GUN HIT
CTTMS   CB    @TIMER,@TCMSHP
        JL    CTTCSL
        BL    @MOVESP
        AB    @TMSHP,@TCMSHP
CTTCSL  CB    @CYSSLF,@H00          HAS LINE ALREADY BEEN CLEARED?
        JEQ   CKSSHP
        C     @TCSSHP,@SEC2         CHECK TIME TO CLR SAUCER LINE
        JL    CKSSHP
        LI    VDPADD,>40
CLRRW2  BL    @WRITCL              CLR ROW 2 (YELLOW SAUCER SCORE)
        DATA  2,HFFFF
        INCT  VDPADD
        CI    VDPADD,>60
        JL    CLRRW2
```

```
        MOVB  @H00,@CYSSLF
CKSSHP  CB    @STOPSC,@H00          CHECK TIME TO START SHIP
        JNE   CLRTCS               DON'T START WHEN GUN HIT
        CB    @SHIPPY,@HC0
        JNE   EML                  DON'T WHEN ALREADY MOVING
        C     @TCSSHP,@TSSHP        CHECK TIME
        JL    EML
BLSTRT  BL    @STARTS
CLRTCS  CLR   @TCSSHP
EML     JMP   SCANLP
        PAGE
*****************************************************************
* GUN HAS BEEN HIT                                              *
*****************************************************************
HITGUN  MOVB  @MAGENT,@GUNCL        CHANGE GUN COLOR
        MOVB  @GHPTRS(YXPOS),@GUNDB
        BL    @WPSAB
        BL    @SOUND1
        DATA  GUNHIT
        AB    @H02,@STOPSC
        B     @CKTIME              UPDATE SPRITES
STOPI   CB    @STOPSC,@H01          STOP SCREEM MOTION
        JNE   FSPRIT
        CB    @SHIPPY,@HC0
        JNE   FSPRIT
        MOVB  @HC0,@SHOTPY
FSPRIT  CLR   INDEX                FIRST SPRITE
ANYSL   CB    @DROP(INDEX),@HC0     ANY SPRITES LEFT?
        JNE   RESETT
        AI    INDEX,4
        CI    INDEX,16
        JLE   ANYSL
        MOVB  @HC0,@SPLATP
        BL    @WPSAB
        CB    @STOPSC,@H01          NO SPRITES LEFT
        JEQ   SCLEAR               JMP TO NO INVADERS
        CB    @DROP+10,@HSHPTR      REMOVE GUN
        JEQ   BULS
        BL    @RRSAB               SEE IF ANY GUNS LEFT
        CB    @EGUN2+2,@GUNIP+2
        JEQ   MOREG
        CB    @EGUN3+3,@EGUNIP+3
        JEQ   MOREG
BULS    B     @ULOSE
MOREG   MOV   @GUNIP,R10            MOVE GUN TO CENTER, DOWN
        AI    R10,>1100
        BL    @AMGUN
        BL    @RRSAB
        CLR   INDEX                FIND SPOT TO DUMP
        LI    R10,>AAC8
CEGDB   CB    @EGUN1(INDEX)+2,@GUNIP+2
        JEQ   BLMG
        AI    INDEX,4
        AI    R10,-24
        JMP   CEGDB
BLMG    BL    @AMGUN                MOVE GUN TO SCRAP PILE
        MOV   @EGUN1(INDEX),R10      SWAP SPRITES
        MOV   @GUNPY,@EGUN1(INDEX)
        MOV   @GUNDB,@EGUN1(INDEX)+2
        MOV   R10,@GUNPY
        MOV   @EGUNIP+2,@GUNDB
        CB    @DEMO,@H00
```

```
        JEQ   WTGUNS
        MOVB  @CLEAR,@EGUN1(INDEX)+3 CLEAR EXTRA GUNS DURING DEMO
WTGUNS  BL    @WESAB
        MOV   @GUNIP,R10          MOVE NEW GUN TO LIFT, UP
        BL    @AMGUN
        MOVB  @GUNIP+3,@GUNCL      RESTORE COLOR
        BL    @WPSAB
        SB    @H02,@STOPSC         HIT AND NOINVL AT SAME TIME?
        JNE   SCLEAR
        CLR   @STOPSC              CLR STOPSC AND DRPFLG
        CB    @SHIPPY,@HC0
        JEQ   RMPL
        BL    @SOUND1              TURN SAUCER SOUND BACK ON
        DATA  SAUCR1
RMPL    B     @SCANLP              RETURN TO MAIN PROGRAM LOOP
        PAGE
*****************************************************************
* NO INVADERS LEFT ON SCREEN                                    *
*****************************************************************
NOINVS  MOVB  @HC0,@SHOTPY          ERASE SHOT
        BL    @WPSAB
        AB    @H01,@STOPSC         SET STOPSC AND NOINVL FLAGS
        B     @CKTIME              UPDATE SPRITES
******
SCLEAR  CB    @SHIPPY,@HC0          MOVE YELLOW SHIP OFF SCREEN
        JEQ   LRS
        B     @RESETT
LRS     MOVB  @HF9,@VDPADR          LIGHT RED SCREEN
        LI    VDPADD,>60           CLEAR SHIELD AREA
        MOVB  @H87,@VDPADR
CLRSA   BL    @WRITCL
        DATA  2,HFFFF
        INC   VDPADD
        CI    VDPADD,>280
        JL    CLRSA
        MOVB  @HF1,@VDPADR          RESET TO BLACK SCREEN
        CLR   STOPSC
        MOVB  @H87,@VDPADR
        MOVB  @SHP2CL,@SHIPCL
        B     @BLSTRT              GO TO INTERMISSION
        PAGE
*****************************************************************
* PLAYER LOST ALL GUNS                                          *
*****************************************************************
***
*** THIS WAS IN THE ORIGINAL GPL CODE  - JEDI -
***
ULOSE   LI    VDPADD,>0300          JEDI -
        LI    WCOUNT,36             JEDI -
        LI    WLOC,SAB              JEDI -
        BL    @WRITE                JEDI -
        CB    @STOPSC,@H03          CHECK TO SEE IF ANY INV LEFT
        JNE   FAITCB
        MOVB  @H02,@STOPSC
        B     @NXTSCR
FAITCB  BL    @HISCOR               JEDI -
        MOV   TROWI,INDEX3          JEDI - STORE TOP ROW POINTER(GPL)
        MOV   @BROWI,VDPADD         FIND AN INV TO CLEAR BTTM AREA
        CLR   PTRNNO
RAI     BL    @READCL
        DATA  1,PTNOLB
        CI    PTRNNO,>30
```

```
        JL    CKBTMR
        INC   VDPADD
        JMP   RAI
CKBTMR  C     @BROWI,@BTL        CHECK FOR INV ON BTTM ROW
        JNE   MTID
CKPO    CB    @CIPN,@H00         CHECK FOR PATTERN ZERO
        JEQ   CKIAL
        BL    @DELAY
        BL    @READCL
        DATA  2,REG8
        AI    R8,>FDFE           MOVE ONE INVADER LEFT
        BL    @WRITR             JEDI - WRITE OUT NEW INVADER PATTERN
        SB    @H02,@CIPN
        JMP   CKPO
CKIAL   BL    @READCL
        DATA  2,REG8
        C     VDPADD,@BROWI       CHECK FOR INVADER AT LEFT
        JEQ   MLID1              JEDI - WAS "JEQ   MLID
        DEC   VDPADD             SHIFT THIS INVADER LEFT
        AI    R8,>0606           SET TO PATTERN 6
        MOVB  @H06,@CIPN
        BL    @DELAY
        BL    @WRITR
        INCT  VDPADD             BLANK RIGHT CHAR WHERE USED TO BE
        BL    @WRITCL
        DATA  1,HFF
        DECT  VDPADD
        JMP   CKPO
MLID1   MOV   R8,VLINE+1         JEDI - NEW CODE HERE
        JMP   MLID               JEDI - NEW CODE HERE
MTID    MOV   PTRNNO,R10         MOVE THIS INVADER DOWN
        MOVB  @PTNOLB,R10
        INC   R10
        MOVB  @H01,@TMOVEI
        BL    @MOVE1D
        A     @H20,@BROWI
SBLL    CB    @CIPN,@H00          MOVE INVADER TO LEFT MARGIN
        JNE   MBRTR
        MOV   @BROWI,VDPADD
        BL    @READCL
        DATA  2,VLINE+1
        CB    @VLINE+1,@H30
        JL    MLID
MBRTR   MOV   @BROWI,@TROWI
        MOVB  @NEG2,@IDIR
        MOVB  @H01,@TMOVEI
        BL    @DELAY
        BL    @MOVEI
        JMP   SBLL
MLID    MOV   @BROWI,VDPADD       MOVE LEFTMOST INV DOWN
        MOV   @VLINE+1,R10
CIV2A2  CI    VDPADD,>2A2
        JHE   SRPBA
        BL    @MOVE1D
        JMP   CIV2A2
SRPBA   MOV   VDPADD,@TROWI       SET ROW POINTER FOR BOTTOM AREA
        MOV   VDPADD,@BROWI
        MOV   @H0302,@BTL
        MOVB  @H02,@IDIR
        LI    R13,9
        CLR   @KEYBRD
MOVE1I  BL    @MOVEI              MOVE ONE INVADER TO CLEAR BTTM
```

```
        MOVB  H00,TIMER          JEDI -
MOVDEL  CB    TIMER,H01          JEDI -
        JNE   MOVDEL             JEDI -
        CLR   @KEYBRD
        BL    @SCANI
        LI    INDEX,28
        BL    @RRSAB
CKSPC   CB    @DROP(INDEX),@HC0   CHECK SPRITE COINCIDENCE
        JHE   NXTSPR
        AB    @H10,@GUNPY
        MOV   @DROP(INDEX),YXPOS
        SB    @H10,@GUNPY
        AI    YXPOS,>0600
        BL    @FPTTRN
        CI    PTRNNO,>30
        JHE   NXTSPR
        MOVB  @HC0,@DROP(INDEX)   CLEAR SPRITE
NXTSPR  AI    INDEX,4
        CI    INDEX,48
        JLE   CKSPC
        BL    @WESAB
        CI    R13,9              PLACE TO WRITE 'GAME OVER'?
        JLT   CKR13
        LI    YXPOS,>B088
        BL    @FPTTRN
        CI    PTRNNO,>30
        JHE   MOVE1I               NOT YET
CKR13   CI    R13,0                YES,CHECK FOR DONE WRITING
        JEQ   CKIOFF
        CB    @CIPN,@H00
        JNE   MOVE1I
        DEC   R13                  WRITE A LETTER
        LI    VDPADD,>2CB
        A     R13,VDPADD
        LI    WLOC,GAMEOV
        A     R13,WLOC
        LI    WCOUNT,1
        BL    @WRITE
        JMP   MOVE1I
CKIOFF  LI    YXPOS,>B8E0          DONE WRITING, SEE IF AT END
        BL    @FPTTRN
        CI    PTRNNO,>30
        JHE   MOVE1I
*****************************************************************
* ALL INVADERS TO JUMP UP AND DOWN                              *
*****************************************************************
        MOV   INDEX3,TROWI   JEDI  - TOP ROW POINTER PREVIOUSLY SAVED
        MOV   H0302,BTL      JEDI  - ORIGINAL GPL CODE
        CB    DEMO,H00       JEDI
        JNE   JI             JEDI
        LI    VDPADD,226     JEDI  - REDO AND BACK MESSAGE
        LI    WCOUNT,28      JEDI
        LI    WLOC,MSGRED    JEDI
        BL    @WRITE         JEDI
        LI    VDPADD,258     JEDI
        LI    WLOC,MSGFF     JEDI
        BL    @WRITE         JEDI
JI      LWPI  MYWS
        LI    R13,200            JUMP COUNT
JUMPUP  MOVB  @H0A,@ADDCH          JUMP UP
        MOV   TROWI,VDPADD
CBH     LI    RLOC,@VLINE+1        CHANGE BOTTOM HALF
```

```
        BL    @CHNGVL
        AI    VDPADD,-32          PUT IN TOP HALF
        BL    @READL              READ ORIGINAL LINE
        DATA  ORLINE+1
        CLR   INDEX
NEXTCH  INC   INDEX               NEXT CHARACTER
        CB    @VLINE(INDEX),@H30
        JHE   RNI
        SB    @H02,@VLINE(INDEX)
        JMP   INCCH
RNI     MOVB  @ORLINE(INDEX),@VLINE(INDEX) RESET NON-INVADER
INCCH   CI    INDEX,27
        JL    NEXTCH
        BL    @WRITE
        AI    VDPADD,64
        CI    VDPADD,>300
        JL    CBH
        BL    @SCANE
        LI    VDPADD,>282         MOVE INVADERS DOWN
        MOV   VDPADD,@BROWI
        MOVB  @H08,@CIPN
        S     @H0020,@TROWI
        BL    @MOVEI
        LI    VDPADD,>2C2         MOVE BOTTOM INVADER DOWN
        MOV   VDPADD,@BROWI
        MOV   @TROWI,@RA+8
        MOV   VDPADD,@TROWI
        MOVB  @H08,@CIPN
        BL    @MOVEI
        MOV   @RA+8,@TROWI
        BL    @SCANE
        CB    @DEMO,@H00
        JEQ   DECJC               'BACK' IF ON DEMO
        CI    R13,175
        JLE   BBACK               'BACK' IF ON DEMO
DECJC   DECT  R13                 DECREMENT JUMP COUNT
        JGT   JUMPUP
BBACK   B     @BACK               TIME IS UP, 'BACK'
        PAGE
*****************************************************************
* SCAN AT END OF GAME                                           *
*****************************************************************
SCANE   MOV   R11,R15
        CLR   @TIMER
CLRKB   CLR   @KEYBRD
SI      BL    @SCANI
        MOVB  @KEY,INDEX
        CB    @JOYY,@HFF            DOWN IS 'REDO'
        JGT   CBKFF
BREDO   B     @REDO
CBKFF   CB    @KEYBRD,@H00
        JNE   CFIREK
        CB    INDEX,@K8             '8' IS 'REDO'
        JEQ   BREDO
        CB    INDEX,@KR             'R' IS 'REDO'
        JEQ   BREDO
        CB    INDEX,@HFF
        JNE   BBACK          ANY KEY OTHER THAN 'REDO' DEFAULTS
CFIREK  CB    INDEX,@FIREK1         FIRE KEYS ARE 'BACK'
        JEQ   BBACK
        AB    @H01,@KEYBRD
        CB    @KEYBRD,@H02
```

```
        JLE   SI                    SCAN FOR NEXT KEYBOARD
        CB    @TIMER,@H06             DELAY
        JL    CLRKB
        B     *R15
*       PAGE
*===============================================================
* CLEAR SCREEN                                                 *
*===============================================================
CLRSCR  MOV   R11,R15
        BL    @WRTVCL               NO SPRITES
        DATA  >300,1,H00
        CLR   VDPADD                CLEAR PATTERN NAME TABLE
CS      BL    @WRITCL
        DATA  2,HFFFF
        INCT  VDPADD
        CI    VDPADD,>300
        JL    CS
        BL    @SOUND1               ALL SOUND OFF
        DATA  GENOFF
        B     *R15
        PAGE
*===============================================================
* SCAN KEYBOARD DURING MAIN PLAY                               *
*===============================================================
SCANK   MOV   R11,R14
        CB    @GUNDB,@GUNIP+2         DON'T SCAN IF GUN HIT
        JEQ   CFLGS
        B     @DSCAN
CFLGS   CLR   @SCRTCH                 CLEAR KEY CODE FLAGS
        CLR   @KEYBRD                 BOARDS 1 AND 2 + JOYSTICKS
INCKB   AB    @H01,@KEYBRD              INCREMENT KEYBOARD
        BL    @SCANI                SCAN FOR INPUT
        CB    @KEY,@FIREK1
        JEQ   SETFF
        CB    @KEY,@FIREK2
        JNE   CKGUNM
SETFF   MOVB  @H01,@SCRTCH            SET FIRE FLAG
CKGUNM  CB    @JOYX,@H00              CHECK FOR GUN MOVE
        JLT   SETML
        CB    @KEY,@LEFTK
        JNE   CKMGR
SETML   MOVB  @H01,@SCRTCH+1          SET MOVE FLAG TO LEFT
CKMGR   CB    @JOYX,@H00
        JGT   SETMR
        CB    @KEY,@RIGHTK
        JNE   RKB2
SETMR   MOVB  @H02,@SCRTCH+1          SET MOVE FLAG TO RIGHT
RKB2    CB    @KEYBRD,@H02            REPEAT FOR KEYBOARD 2
        JLT   INCKB
        CB    @SCRTCH+1,@PREKEY       DON'T IGNORE IF MOVE CHANGED
        JNE   ABTSC
        CB    @SHIPCL,@SHP2CL         FASTER GUN ON INTERMISSION
        JEQ   CKMOVL
        CB    @TIMER,@TCSCAN          CHECK TIMER
        JL    ENDSC
ABTSC   AB    @TSCAN,@TCSCAN
        CB    @DEMO,@H00
        JEQ   CKMOVL                SKIP DURING GAME PLAY
        CLR   @KEYBRD
        BL    @SCANI
        MOVB  @DEMO,VDPADD            GET MOVES FOR DEMO
        SRL   VDPADD,8              MOVE OFFSET TO LOW BYTE
```

```
        AI    VDPADD,>1500       DEMO MOVES START AT VDP >1501
        BL    @READCL            READ DEMO MOVE
        DATA  2,SCRTCH
        CB    @TCSCAN,@TSCAN
        JNE   CKMOVL             REPEAT SAME MOVE 20 TIMES
        AB    @HO2,@DEMO
        CB    @SCRTCH,@HFF
        JNE   CKMOVL
        MOVB  @HO1,@DEMO           RECYCLE THROUGH DEMO MOVES
CKMOVL  CB    @SCRTCH+1,@HO1       CHECK FOR MOVE GUN LEFT
        JNE   CKMOVR
        CB    @GUNPX,@GLMRGN
        JLE   CHECKF
        DEC   @GUNPX               MOVE GUN LEFT
        B     @MOVEG
CKMOVR  CB    @SCRTCH+1,@HO2       CHECK FOR MOVE GUN RIGHT
        JNE   CHECKF
        CB    @GUNPX,@GRMRGN
        JHE   CHECKF
        INC   @GUNPX               MOVE GUN RIGHT
MOVEG   BL    @WPSAB               REWRITE GUN POSITION
        MOVB  @HO1,@DRPFLG
CHECKF  CB    @SCRTCH,@HOO         CHECK FIRE FLAG
        JEQ   ENDSC
        CB    @GUNCL,@GUNIP+3      ONLY FIRE IF INITIAL COLOR
        JNE   ENDSC
        CB    @SHOTPY,@HCO         DON'T FIRE IF ALREADY EXITS
        JNE   ENDSC
        MOV   @GUNPY,@SHOTPY       FIRE SHOT
        A     @HO604,@SHOTPX
        BL    @WPSAB
        MOVB  @HO1,@DRPFLG
        LI    VDPADD,@GNSHOT+6     SET STARTING FREQ FOR SHOT
        BL    @WRITCL
        DATA  3,SHFREQ
ENDSC   MOVB  @SCRTCH+1,@PREKEY    SET PREVIOUS MOVE KEY
DSCAN   B     *R14
        PAGE
*==============================================================
* SCAN FOR INPUT                                               *
*==============================================================
SCANI   LIMI  0                  DISABLE INTERRUPTS
        LWPI  GPLWS              SET TO GPL WORKSPACE
        MOVB  @HE2,@SCANW          KEEP DOUBLE SIZE SPRITES
        BL    @SCAN
        LWPI  MYWS               RESET TO MY WORKSPACE
        CB    @DEMO,@HOO
        JEQ   CKKBO
        CB    @KEY,@HFF
        JEQ   CKKBO
        MOVB  @HOO,@DEMO           'BACK' IF KEY HIT DURING DEMO
        B     @BACK
CKKBO   CB    @KEYBRD,@HOO
        JNE   ENINT
        CB    @KEY,@REDOKY         CHECK FOR 'REDO'
        JNE   CKBACK
        CB    TITLFG,HO1         JEDI - NO REDO IF ON TITLE SCREEN
        JEQ   CKBACK             JEDI
        B     @REDO
CKBACK  CB    @KEY,@BACKKY           CHECK FOR 'BACK'
        JNE   ENINT
        B     @BACK
```

```
ENINT  LIMI 2
*******************************************
*  AT THIS POINT, CHECK FOR PAUSE KEY    *
*******************************************
       CB    KEYBRD,H00
       JEQ   NOTP01
       CB    TITLFG,H01          JEDI - NO PAUSE FEATURE DURING TITLE SCREEN
       JEQ   NOTP01              JEDI
       CB    KEY,H0B             "P" KEY HIT TO STOP GAME?
       JNE   NOTP01              NO,  KEEP ON GOING
       MOV   R11,SAVR11
KEYO   BL    @SCANKY
       MOVB  STATUS,STATUS
       JNE   KEYO
       LI    VDPADD,>02ED
       LI    WCOUNT,7
MOREMG LI    WLOC,MSGPAU
       BL    @WRITE
       MOVB  H00,TIMER
       MOVB  H00,CPUD6
DELYP1 BL    @SCANKY
       MOVB  STATUS,STATUS
       JNE   BLKMSG
       CB    TIMER,H30
       JNE   DELYP1
       LI    WLOC,MSGBLK
       BL    @WRITE
       MOVB  H00,TIMER
DELYP2 BL    @SCANKY
       MOVB  STATUS,STATUS
       JNE   NOTPOA
       CB    TIMER,H10
       JNE   DELYP2
       JMP   MOREMG
BLKMSG LI    WLOC,MSGBLK
       BL    @WRITE
NOTPOA MOV   SAVR11,R11
NOTP01 LIMI 2
       B     *R11
***
***  SCAN KEYBOARD
***
SCANKY LIMI 0
       LWPI  GPLWS
       BL    @SCAN
       LWPI  MYWS
       LIMI 2
       B     *R11

*==============================================================
* MOVE INVADERS                                               *
*--------------------------------------------------------------
MOVEI  MOV   R11,R14
       CB    @CIPN,@H08
       JNE   CEOR
       B     @COMPLD
CEOR   CB    @CIPN,@H00          CHECK FOR ANY AT END OF ROW
       JNE   NOTDWN
       CLR   PTRNNO
       MOV   @TROWI,VDPADD
       CB    @IDIR,@NEG2
       JEQ   RDCOL
```

```
        AI    VDPADD,27
RDCOL   BL    @READCL              LOOP DOWN COLUMN
        DATA  1,PTNOLB
        CI    PTRNNO,>30
        JL    SHIFTD
        AI    VDPADD,64            DOWN TWO ROWS
        CI    VDPADD,>300
        JL    RDCOL
NOTDWN  MOV   @TROWI,VDPADD        SHIFT OF BYTES NEEDED?
        CB    @IDIR,@NEG2
        JEQ   GOINGL
        CB    @CIPN,@H06
        JEQ   SHIFTR
        JMP   CHNGCH
GOINGL  CB    @CIPN,@H00
        JEQ   SHIFTL
CHNGCH  MOVB  @IDIR,@ADDCH         CHANGE CHARACTERS (BYTES)
        LI    RLOC,@VLINE+1
        AB    @IDIR,@CIPN
        JMP   SHIFT
*****************************************************************
* SHIFT RIGHT                                                   *
*****************************************************************
* MOVE CHARS ONE TO THE RIGHT AND CHANGE FROM PATTERN 6 TO 0
SHIFTR  MOVB  @HFF,@VLINE+1          BLANK LEFTMOST COLUMN
        MOVB  @NEG6,@ADDCH           INITIALIZE FOR CHGNVL SUBROUT
        LI    RLOC,@VLINE+2
        MOVB  @H00,@CIPN
        JMP   SHIFT
*****************************************************************
* SHIFT LEFT                                                    *
*****************************************************************
* MOVE CHARS ONE TO THE LEFT AND CHANGE FROM PATTERN 0 TO 6
SHIFTL  MOVB  @HFF,@VLINE+28         BLANK RIGHTMOST COLUMN
        MOVB  @H06,@ADDCH            INITIALIZE FOR CHGNVL SUBROUT
        LI    RLOC,@VLINE
        MOVB  @H06,@CIPN
SHIFT   BL    @CHNGVL
        AI    VDPADD,>0040           DOWN TWO ROWS
        C     VDPADD,@BROWI
        JLE   SHIFT
        B     *R14
*****************************************************************
* SHIFT DOWN                                                    *
*****************************************************************
* CHANGE FROM PATTERN 0 TO 8, WRITE PATTERNS A,B BELOW
SHIFTD  MOVB  @H08,@ADDCH
        MOVB  @H08,@CIPN
        MOV   @TROWI,VDPADD
SHD     LI    RLOC,@VLINE+1
        BL    @CHNGVL
        AI    VDPADD,>0020           DOWN ONE ROW
        BL    @READL                 READ ORIGINAL BOTTOM LINE
        DATA  @ORLINE+1
        CLR   INDEX
INCIDX  INC   INDEX
        CB    @VLINE(INDEX),@H30
        JHE   RNONI
        AB    @H02,@VLINE(INDEX)     SET TO BOTTOM HALF OF INVADER
        JMP   TEOL
RNONI   MOVB  @ORLINE(INDEX),@VLINE(INDEX)  RESET NON-INVADER
TEOL    CI    INDEX,29               TEST FOR END-OF-LINE
```

```
        JL    INCIDX
        BL    @WRITE
        AI    VDPADD,>0020
        C     VDPADD,@BROWI
        JLE   SHD
        B     *R14
*****************************************************************
* COMPLETE DOWNWARD MOVEMENT                                    *
*****************************************************************
* CHANGE FROM PATTERN 8,9 TO >DO AND BEYOND
* CHANGE FROM PATTERN A TO 0
COMPLD  MOVB  @HC8,@ADDCH
        LI    RLOC,@VLINE+1
        MOV   @TROWI,VDPADD
ERASEI  BL    @CHNGVL
        AI    VDPADD,>0040
        C     VDPADD,@BROWI
        JLE   ERASEI
        MOVB  @NEGA,@ADDCH
        A     @H0020,@TROWI        INC TOP ROW PTR ONE LINE
        A     @H0020,@BROWI
        MOV   @TROWI,VDPADD
SHDC    BL    @CHNGVL
        AI    VDPADD,>0040
        C     VDPADD,@BROWI
        JLE   SHDC
        MOVB  @H00,@CIPN
        C     @BROWI,@BTL          CHECK IF ON BOTTOM LINE
        JEQ   IHRBR
        CB    @IDIR,@NEG2
        JNE   MOVL
        MOVB  @H02,@IDIR
        JMP   DCOMPD
MOVL    MOVB  @NEG2,@IDIR
DCOMPD  B     *R14
*****************************************************************
* INVADERS HAVE REACHED BOTTOM ROW                              *
*****************************************************************
IHRBR   MOV   @GUNPY,YXPOS
        AI    YXPOS,>0800
        BL    @FPTTRN
        MOV   VDPADD,X
        MOVB  @H01,@IDIR
FAIIBR  BL    @READ              FIND AN INVADER IN BOTTOM ROW
        CI    PTRNNO,>30
        JL    IIBR
        DEC   VDPADD               LOOK LEFT OF GUN
        CI    VDPADD,>282
        JHE   FAIIBR
        MOV   X,VDPADD             LOOK RIGHT OF GUN
        MOVB  @HFF,@IDIR
FAIOR   BL    @READ
        CI    PTRNNO,>30
        JL    IIBR
        INC   VDPADD
        CI    VDPADD,>29D
        JLE   FAIOR
IIBR    MOVB  @HSHPTR,@DROP+10     SET 3RD DROP TO HORIZ SHOT
        BL    @FYXPOS
        AI    YXPOS,>F600        DEC Y BY 10
        MOV   YXPOS,@DROP+8
        BL    @WPSAB
```

```
        MOVB @H04,@STOPSC
        B    *R14
        PAGE
*==============================================================
* CHANGE VDP LINE                                              *
*==============================================================
* ADD THE VALUE OF ADDCH TO EACH BYTE IN VDP LINE
CHNGVL  MOV  R11,R15
        LI   RCOUNT,28
        BL   @READ              READ VDP LINE (SHIFT)
        MOV  RLOC,RA+6          SAVE READ LOCATION
        BL   @READL             READ ORIGINAL VDP LINE
        DATA ORLINE+1
        CLR  INDEX
INCPT   INC  INDEX
        CB   VLINE(INDEX),H30       AN INVADER HERE?
        JHE  NONINV
        AB   @ADDCH,@VLINE(INDEX)     SHIFT INVADER
        JMP  CINDX
NONINV  CB   @ORLINE(INDEX),@H30      NON-INVADER HERE
        JHE  RSETNI
        MOVB @HFF,VLINE(INDEX)  BLANK WHERE INVADER USED TO BE
        JMP  CINDX
RSETNI  MOVB @ORLINE(INDEX),@VLINE(INDEX)     RESET NON-INVADERS
CINDX   CI   INDEX,29
        JL   INCPT
        MOV  @RA+6,RLOC          RESTORE READ LOCATION
        BL   @WRITL             WRITE VDP LINE
        DATA VLINE+1
        B    *R15
        PAGE
*==============================================================
* DROP ANOTHER SPRITE                                          *
*==============================================================
* OFFSET RANGES FROM 2 TO 25 TO DROP ONLY OVER BUNKER
DROPSP  MOV  R11,R14
        CLR  INDEX
        MOVB @LPDROP,@INDXLB
        AI   INDEX,5            NEXT COLUMN TO DROP FROM
        CI   INDEX,25           DON'T ALLOW OFFSET OVER 25
        JLE  STOREI             DROP ONLY OVER BUNKER
        AI   INDEX,-24
STOREI  MOVB @INDXLB,@LPDROP
        CB   @DRPFLG,@H00         GIVE PLAYR CHANCE TO MOVE/FIRE
        JEQ  NODROP
        MOV  @BROWI,VDPADD       FIND AN INVADER IN THIS COLUMN
        A    INDEX,VDPADD       SET TO OVER BUNKER
        CLR  PTRNNO
RDPTRN  BL   @READCL
        DATA 1,PTNOLB
        CI   PTRNNO,>30
        JL   FAI
        AI   VDPADD,-64         SEE IF ONE ABOVE
        C    VDPADD,@TROWI
        JHE  RDPTRN
        CB   @OPTION,@H02         NONE IN THIS COLUMN
        JNE  NODROP
        CB   @DROP4,@HC0
        JNE  NODROP
        LI   INDEX,12           SET DROP 4 OVER GUN
        MOV  @GUNPY,YXPOS
        BL   @FPTTRN
```

```
FAIOG   BL    @READ               FIND AN INV OVER GUN
        CI    PTRNNO,>30
        JL    CKROW
        AI    VDPADD,-32
        C     VDPADD,@TROWI
        JHE   FAIOG
        JMP   NODROP
FAI     CLR   INDEX               FOUND AN INVADER
FAAS    CB    @DROP(INDEX),@HCO   FIND AN AVAILABLE SPRITE
        JEQ   CKROW
        AI    INDEX,4
        CI    INDEX,12
        JL    FAAS
        JMP   NODROP
CKROW   BL    @FYXPOS             CHECK ROW FOR ANY OTHER DROPS
        MOV   INDEX,@SCRTCH
        CLR   INDEX
        AI    YXPOS,>F800         DEC Y BY 8
        MOV   YXPOS,@SCRTCH+2
        AI    YXPOS,>1000         INC Y BY 16
COMROW  CB    @DROP(INDEX),@SCRTCH+2
        JL    NEXTDR
        CB    @DROP(INDEX),YXPOS
        JL    NODROP
NEXTDR  AI    INDEX,4
        CI    INDEX,12
        JL    COMROW
        AI    YXPOS,>F100         POSITION NEW SPRITE
        MOV   @SCRTCH,INDEX
        AI    YXPOS,4
        CB    @CIPN,@H08
        JEQ   SIRH
        AB    @CIPN,XPOS           MOVE DROP TO CENTER OF INVADER
SIRH    ANDI  PTRNNO,1            PATTERN NUMBER EVEN OR ODD?
        JEQ   MXLB                SKIP IF LEFT CHAR (EVEN)
        AI    YXPOS,-8            SHIFT DROP TO LEFT CHARACTER
MXLB    MOV   YXPOS,@DROP(INDEX)
        BL    @WPSAB
NODROP  B     *R14
        PAGE
*==============================================================
* UPDATE PLAYER'S SHOT                                        *
*==============================================================
UPSHOT  MOV   R11,@RA+6
        CB    @SHOTPY,@HCO
        JNE   DECSHY
        B     @WSHOT
DECSHY  SB    @H01,@SHOTPY
        MOV   @SHOTPY,YXPOS
        BL    @TSTBIT           *** CHECK FOR A HIT ***
        CB    @FLAG,@H00              IS BIT OFF?
        JEQ   BCKOFF
        CI    PTRNNO,>48          INVADER OR SHIELD?
        JL    HITWHT
BCKOFF  B     @CKSOFF           *** MISSED ***
HITWHT  CI    PTRNNO,>30
        JL    HANINV
        B     @HSHLD
*                               *** HIT AN INVADER ***
* READ COLOR FROM PATTERN COLOR TABLE (>380)
HANINV  SRL   PTRNNO,4                MOVE HIGH NIBBLE OF LOW BYTE
        MOV   PTRNNO,VDPADD           GET SPLAT COLOR
```

```
        SLA   VDPADD,1              MULTIP PTRN NO BY 2
        AI    VDPADD,>380           OFFSET OF PATTERN COLOR TBL
        BL    @READCL
        DATA  2,INDXLB-1
        SRL   INDEX,4               GET LEFT MOST NIBBLE
        MOVB  INDEX,@SPLATC
* POINT TABLE AT >10A0
* >380 + >D20 = >10A0
        AI    VDPADD,>D20           GET POINTS FOR THIS INVADER
        CB    @OPTION,@H02
        JNE   GETPTS
        AI    VDPADD,6
GETPTS  BL    @READL
        DATA  REG10
        BL    @SCORE                ADD TO SCORE
        BL    @FPTTRN               PLACE SPLAT SPRITE
        BL    @FYXPOS
        ANDI  PTRNNO,1              CHECK FOR RIGHT HALF
        JEQ   POSS
        AI    YXPOS,-8
POSS    AI    YXPOS,>FBFE           POSITION SPRITE
        CB    @CIPN,@H08
        JEQ   MSPLTP
        AB    @CIPN,XPOS
MSPLTP  MOV   YXPOS,@SPLATP
        BL    @SOUND
        DATA  INVHIT
        MOV   @SHOTPY,YXPOS
        BL    @FPTTRN               GET VDPADD
        ANDI  PTRNNO,>000F          SEE IF HIT TOP OF INV POS 8
        CI    PTRNNO,>0008
        JL    BLANKC
        CI    PTRNNO,>0009
        JH    BLANKC
        AI    VDPADD,>0020
BLANKC  BL    @WRITCL               BLANK CHARACTER
        DATA  1,HFF
        CB    @CIPN,@H08              BLANK TOP CHAR IF MOVING DWN
        JNE   TRB
        S     @H0020,VDPADD
        BL    @WRITE
TRB     ANDI  PTRNNO,1              TEST RIGHT BIT
        JNE   DECV
        INC   VDPADD                  EVEN PATTERN (LEFT HALF)
        JMP   BLNK
DECV    DEC   VDPADD                  ODD PATTERN (RIGHT HALF)
BLNK    BL    @WRITE                BLANK OTHER HALF OF INVADER
        CB    @CIPN,@H08              BLANK BTTM CHAR IF MOVING DOWN
        JNE   IRID
        AI    VDPADD,>0020
        BL    @WRITE
IRID    LI    R13,64                ROW INCREMENT/DECREMENT
        CLR   R10                   INDEX TO TOP/BOTTOM ROW
MBTRP   MOV   @EROWI(R10),VDPADD    MOVE BOTTOM/TOP ROW POINTER
        BL    @READCL
        DATA  28,@VLINE+1
        LI    INDEX,28
CKFAI   CB    @VLINE(INDEX),@H30    CHECK FOR AN INVADER
        JL    NXTRWP                SKIP IF ONE FOUND
        DEC   INDEX
        JGT   CKFAI
        A     R13,@EROWI(R10)       NO INVADERS IN THIS LINE
```

```
        C     @BROWI,@TROWI
        JHE   MBTRP                 CHECK NEXT ROW
        B     @NOINVS               BRANCH IF NO INVADERS LEFT
NXTRWP  NEG   R13                   NEXT ROW POINTER (BROWI)
        INCT  R10
        CI    R10,4
        JL    MBTRP
        SB    @H01,@HITCNT            DECREMENT HIT COUNT
        CB    @HITCNT,@D20            TIME TO SPEED UP?
        JHE   BTSOFF                NO
        CB    @TMOVEI,@H03
        JLE   NFT3
        SB    @H01,@TMOVEI            SPEED UP INVADERS
        JMP   BTSOFF
NFT3    CB    @HITCNT,@H03            GO NO FASTER THAN 3 IF >2 INVS
        JHE   BTSOFF
        MOVB  @HITCNT,@TMOVEI
BTSOFF  B     @TSOFF
HSHLD   DEC   YXPOS             *** HIT A SHIELD ***
        BL    @BLAST2               BLANK OUT A PATTERN
        AI    YXPOS,>FF01
        BL    @BLAST2
        AI    YXPOS,>FF00
        BL    @BLKBIT
BTSSO   B     @TSSOFF
CKSOFF  CB    YXPOS,SUPMGN    *** CHECK FOR SHOT OFF TOP OF SCRN
        JL    BTSSO
        CB    @SHIPPY,@HC0      *** CHECK FOR HIT ON SAUCER ***
        JEQ   SMISS
        SB    @SHIPPY,YXPOS       CHECK Y OF SHOT TO SAUCER
        CB    YXPOS,@H08
        JHE   SMISS
        SB    @SHIPPX,@XPOS         CHECK X OF SHOT TO SAUCER
        CB    @XPOS,@H0F
        JL    HITSAU
SMISS   ANDI  YXPOS,>0700     *** SHOT MISSED EVERYTHING ***
        JGT   JWSHOT                RAISE FREQ EVERY 8TH PIXEL
        LI    VDPADD,@GNSHOT+6
        BL    @READCL
        DATA  3,YPOS
        AI    R12,>FF00
        AI    R13,>FF00
        BL    @WRITR
        BL    @SOUND
        DATA  GNSHOT
JWSHOT  JMP   WSHOT
HITSAU  CB    @SHIPCL,@SHP1CL *** HIT A SAUCER ***
        JEQ   HITYS
* RED SAUCER SCORES MORE AFTER EACH HIT
* SCORES THE SAME REGARDLESS OF COLUMN HIT
* LEFT EDGE OF RED SAUCER IS RIGHT NIBBLE OF BLANK
*     CHARACTER BEFORE THE FIRST ZERO
        LI    VDPADD,>B5            READ LEFT EDGE OF RED SAUCER
        BL    @READCL
        DATA  1,PTNOLB
        ANDI  PTRNNO,>000F          MASK OFF CHARACTER BIAS
        CB    YXPOS,@PTNOLB
        JHE   SMISS
        CB    @XPOS,@PTNOLB
        JHE   SMISS
*                               *** HIT RED SAUCER ***
        NEG   SHPDIR                REVERSE DIRECTION
```

```
        SB    @SCREEN,@SHIPPY      RAISE SAUCER
        CB    @SHIPPY,@H40         PREVENT WRAP AROUND
        JH    MINSY
        CB    @SHIPPY,@H10         NO FARTHER UP THAN >10
        JHE   SETWND
MINSY   MOVB  @H10,@SHIPPY         MINIMUM SAUCER Y POS
SETWND  MOVB  @SHIPPY,@SHPWPY      SET SAUCER WINDOWS
* USE THE SAUCER DESCRIPTOR BLOCK POINTER TO FIND THE
* VDP ADDRESS OF THE CURRENT VALUE OF THE RED SAUCER.
* RED SAUCER VALUES START AT VDP >1040
* FIRST SAUCER DB PTR = >A8
*
        LI    VDPADD,>F98          >F98 = >1040 - >A8
        CB    @TMSHP,@H01
        JH    SPEEDU
        AB    @H04,@SHIPDB           SMALLER RED SAUCER
        JMP   AR10V
SPEEDU  SB    @H01,@TMSHP            SPEED UP RED SAUCER
AR10V   CLR   R10                  GET SAUCER VALUE
        MOVB  @SHIPDB,@REG10+1       ADD SAUCER DB PTR TO VDPADD
        A     R10,VDPADD
        BL    @READCL              READ CURRENT SAUCER VALUE
        DATA  8,@ORLINE
        BL    @CONVSC              CONVERT, ADD TO SCORE
        LI    VDPADD,>B5           BEGINNING SCREEN ADDRESS
        BL    @WRITCL              NEXT SAUCER VALUE TO SCREEN
        DATA  4,ORLINE+4
        BL    @SOUND1
        DATA  S2HIT
        CB    @SHIPDB,@HBC    TURN WINDOWS OFF WHEN SAU TOO SMALL
        JL    TSOFF
        MOVB  @HC0,@SHPWPY
        JMP   TSOFF
* YELLOW SAUCER SCORES DIFFERENT ACCORDING TO THE
* COLUMN THAT IS HIT
HITYS   ANDI  YXPOS,>00FF    *** HIT YELLOW SAUCER ***
        CI    YXPOS,7
        JLE   GETSCA
        NEG   YXPOS                CHANGE 8-14 TO 6-0
        AI    YXPOS,14
GETSCA  LI    VDPADD,>1010         GET SCORE VDP ADDRESS
        SLA   YXPOS,2                MULT XPOS BY 4
        A     YXPOS,VDPADD
        BL    @READCL              READ POINTS FOR HITTING THIS X
        DATA  4,ORLINE
        MOV   @SHIPPY,YXPOS
        BL    @FPTTRN              GET VDPADD OF SAUCER
        DEC   VDPADD
        BL    @WRITCL              DISPLAY POINTS GIVEN
        DATA  4,ORLINE
        MOVB  @H01,@CYSSLF           SET FLAG TO CLEAR POINTS
        BL    @CONVSC              CONVERT, ADD TO SCORE
        MOVB  @HC0,@SHIPPY           REMOVE SAUCER
        MOVB  @HC0,@SHPWPY
        CLR   TCSSHP
        BL    @SOUND
        DATA  SHPHIT
        JMP   TSOFF
TSSOFF  BL    @SOUND               TURN SHOT SOUND OFF
        DATA  GENOFF
TSOFF   MOVB  @HC0,@SHOTPY           TURN SHOT OFF
WSHOT   BL    @WPSAB
```

```
        MOV   @RA+6,R11
        B     *R11
        PAGE
*==============================================================
* MOVE INVADERS' DROPPINGS                                     *
*==============================================================
MOVED   MOV   R11,@RA+6
        CLR   INDEX
CKNXTD  CB    @DROP(INDEX),@HCO    IS THIS ONE ON THE SCREEN?
        JEQ   NXTSP
        CB    @DROP(INDEX)+2,@HSHPTR CHECK FOR HORIZ SHOT
        JEQ   VERDRP
        AB    @HO1,@DROP(INDEX)
        JMP   SAVPOS
VERDRP  AB    @IDIR,@DROP(INDEX)+1
SAVPOS  MOV   @DROP(INDEX),YXPOS
        AI    YXPOS,>0400
        CB    YXPOS,@GUNPY         CHECK FOR A HIT ON THE GUN
        JL    RESPOS
        AI    YXPOS,>FA00
        CB    YXPOS,GUNPY
        JHE   ERASED               DROP HIT BUNKER
        ANDI  YXPOS,>00FF
        INC   YXPOS
        SB    @GUNPX,@XPOS          CHECK SHOT X TO GUN X
        CI    YXPOS,8
        JH    RESPOS
        CB    @DROP(INDEX)+2,@HSHPTR  ALWAYS HIT IF HORIZ SHOT
        JEQ   BHITG
        CB    @GUNDB,@GUNIP+2
        JNE   RESPOS               JUMP IF GUN ALREADY HIT
BHITG   MOVB  @HCO,@DROP(INDEX)
        B     @HITGUN
RESPOS  MOV   @DROP(INDEX),YXPOS RESET POSITION
        AI    YXPOS,>0E00
        BL    @FPTTRN
        CI    PTRNNO,>48           IS IT OVER ANYTHING?
        JHE   CSPRTC
        CI    PTRNNO,>30
        JL    CSPRTC
        BL    @TSTBIT              OVER A SHIELD, IS BIT ON?
        CB    @FLAG,@HOO
        JEQ   CSPRTC
        LI    R10,2
HASBA   BL    @BLAST3              HIT A SHIELD, BLAST AWAY
        AI    YXPOS,>00FF
        BL    @BLAST3
        AI    YXPOS,>0101
        DEC   R10
        JGT   HASBA
        BL    @BLAST2
ERASED  MOVB  @HCO,@DROP(INDEX)
        JMP   NXTSP
CSPRTC  CB    @XPOS,@SHOTPX          CHECK SPRITE COINCIDENCE
        JNE   NXTSP
        CB    YXPOS,@SHOTPY
        JL    NXTSP
        AI    YXPOS,>FC00
        CB    YXPOS,@SHOTPY
        JH    NXTSP
        MOVB  @HCO,@SHOTPY         HIT GUN SHOT
        BL    @SOUND
```

```
        DATA GENOFF
        LI   R10,1
        BL   @SCORE
        JMP  ERASED
NXTSP   AI   INDEX,4
        CI   INDEX,12
        JLE  CKNXTD
        BL   @WPSAB
        MOV  @RA+6,R11
        B    *R11
        PAGE
*==============================================================
* START SAUCER MOVING                                         *
*==============================================================
STARTS  MOV  R11,R14
        BL   @SOUND              START SOUND LIST
        DATA SAUCR1
        MOVB @H01,@SHPDIR         SET DIRECTION OPPOSITE INVADERS
        MOVB @H00,@SHIPPX
        CB   @IDIR,@H02
        JNE  SETW
        MOVB @HFF,@SHPDIR
        MOVB @HFF,@SHIPPX
SETW    MOVB @H10,@SHIPPY
        MOVB @H02,@TMSHP          SET MOVE TIMER
        CB   @SHIPCL,@SHP1CL
        JEQ  SWIND
        BL   @SOUND2             INTERMISSION SAUCER (RED)
        DATA SLOWS2
        MOVB @TMSHP,@TCYCLE
        MOVB @H40,@SHIPPY
        LI   VDPADD,>A8          WRITE SAUCER VALUE LINE
        LI   WLOC,SVALL
        LI   WCOUNT,17
        BL   @WRITE
SWIND   MOV  @SHIPPY,@SHPWPY      SET WINDOWS
        BL   @WPSAB
        B    *R14
        PAGE
*==============================================================
* MOVE SAUCER                                                 *
*==============================================================
MOVESP  MOV  R11,R14
        CB   @SHIPPY,@HC0
        JEQ  WSPRTS
        AB   @SHPDIR,@SHIPPX      MOVE SHIP
        MOVB @SHPDIR,@SCRTCH
        NEG  SCRTCH
        CB   @SHPDIR,@H00
        JLT  MVINGL
        CB   @SHIPPX,@SCRTCH
        JHE  TSPOFF
CNMWR   CB   @SHPWPX,@SHIPPX      CHECK IF NEED TO MOVE WINDOWS
        JHE  WSPRTS
        AB   @H05,@SHPWPX
        CB   @SHPWPX,@H05
        JL   TSWOFF              DON'T LET WINDOWS WRAP AROUND
        JMP  CNMWR
MVINGL  CB   @SHIPPX,@SCRTCH
        JLE  TSPOFF
CNMWL   MOVB @SHPWPX,@SCRTCH
        SB   @H05,@SCRTCH
```

```
        CB    @SHIPPX,@SCRTCH
        JHE   WSPRTS
        CB    @SCRTCH,@H05
        JL    TSWOFF              DON'T LET WINDOWS WRAP AROUND
        SB    @H05,@SHPWPX
        JMP   CNMWL
TSPOFF  MOVB  @HC0,@SHIPPY          TURN SHIP OFF
        BL    @SOUND
        DATA  GENOFF
        CB    @SHIPCL,@SHP2CL
        JNE   TSWOFF
        MOVB  @HC0,@SHPWPY
        BL    @WPSAB
        CLR   @STOPSC               INTERMISSION OVER
        LI    R0,>0202              NEXT SCREEN
        B     @NXTSCR
TSWOFF  MOVB  @HC0,@SHPWPY          TURN OFF DURING REG PLAY
WSPRTS  BL    @WPSAB
        B     *R14
        PAGE
*==============================================================
* BLAST A ROW OF PIXELS                                       *
*==============================================================
BLAST2  LI    R13,2                 BLAST TWO PIXELS
        JMP   STRA
BLAST3  LI    R13,3                 BLAST THREE PIXELS
STRA    MOV   R11,@RA+8
        MOVB  @XPOS,@SCRTCH+3         SAVE XPOS
CALLB   BL    @BLKBIT               BLANK THIS BIT
        INC   YXPOS                 NEXT XPOS
        DEC   R13
        JGT   CALLB
        MOVB  @SCRTCH+3,@XPOS         RESTORE XPOS
        MOV   @RA+8,R11
        B     *R11
        PAGE
*==============================================================
* TEST BIT                                                    *
*==============================================================
* FIND OUT WHETHER BIT (PIXEL) AT LOCATION YPOS,XPOS IS ON
* RETURN A ZERO IN FLAG IF NOT
TSTBIT  MOV   R11,@RA+10
        BL    @RDBYTE
        MOVB  @RBYTE,@FLAG
        MOVB  @BITMSK(XREM),@SCRTCH
        INV   @SCRTCH
        SZCB  @SCRTCH,@FLAG
        MOV   @RA+10,R11
        B     *R11
        PAGE
*==============================================================
* BLANK BIT                                                   *
*==============================================================
* GIVEN ANY YPOS,XPOS RESET THE BIT (PIXEL TO BACKGROUND)
* AT THAT SCREEN LOCATION.
* THIS ROUTINE IS USED TO EAT AWAY THE SHIELDS.
BLKBIT  MOV   R11,@RA+10
        BL    @RDBYTE
        MOVB  @RBYTE,@FLAG
        MOVB  @BITMSK(XREM),@SCRTCH
        SZCB  @SCRTCH,@FLAG
        MOVB  @FLAG,@RBYTE
```

```
        BL    @WRITR
        MOV   @RA+10,R11
        B     *R11
        PAGE
*===============================================================
* READ BYTE                                                    *
*===============================================================
* READ THE BYTE FOR PIXEL LOCATION YPOS,XPOS
RDBYTE  MOV   R11,R14
        BL    @FPTTRN             FIND PATTERN
        MOV   PTRNNO,VDPADD
        SLA   VDPADD,3            MULTIPLY BY 8
        AI    VDPADD,>800
        A     YREM,VDPADD         ADD TO GET EXACT BYTE
        BL    @READCL
        DATA  1,RBYTE
        B     *R14
        PAGE
*===============================================================
* FIND PATTERN                                                 *
*===============================================================
* GIVEN ANY YXPOS OF A SPRITE, RETURN THE PATTERN NUMBER
* UNDER THE SPRITE AND RETURN THE PIXEL WITHIN THE
* PATTERN (XREM,YREM)
FPTTRN  MOV   R11,R15
        MOV   YXPOS,Y       GET Y AND X FROM YXPOS
        SRL   Y,8
        MOV   YXPOS,X
        ANDI  X,>00FF
        MOV   Y,YREM
        MOV   X,XREM
        SRL   Y,3           DIVIDE BY 8
        SRL   X,3
        SLA   Y,3           MULTIPLY QUOTIENTS BY 8
        SLA   X,3
        S     Y,YREM        CALCULATE REMAINDER
        S     X,XREM
        SLA   Y,2           (Y*8)*4
        SRL   X,3           RESTORE X TO QUOTIENT
        A     X,Y
        CLR   PTRNNO
        BL    @READCL       Y IS THE VDP ADDRESS
        DATA  1,PTNOLB
        B     *R15
*===============================================================
* FIND YXPOS                                                   *
*===============================================================
* GIVEN A VDP ADDRESS, FIND THE YXPOS FOR
* A SPRITE (UPPER LEFT CORNER)
FYXPOS  MOV   VDPADD,X            FIND Y,X FOR VDPADD
        SRL   VDPADD,5
        SLA   VDPADD,5
        S     VDPADD,X
        SRL   VDPADD,2
        SLA   X,3
        MOV   X,YXPOS             XLB IN YXPOS LB
        MOVB  YLB,YXPOS           YLB IN YXPOS HB
        RT
        PAGE
*===============================================================
* CONVERT SCORE                                                *
*===============================================================
```

```
* GIVEN FOUR BYTES OF CHARACTER NUMBERS (R8,R9), CONVERT
* TO FORMAT NEEDED FOR SCORE ROUTINE
* EXAMPLE OF INPUT AND OUTPUT:
*    R8: >FF51    R9: >5255      RESULT: R10: >0125
CONVSC MOV  ORLINE,R8
       MOV  ORLINE+2,R9
       SLA  R8,12          NIBBLE 4 OF R8 TO NIBBLE 2
       SRL  R8,4
       ANDI R9,>0F0F       GET RID OF CHARACTER BIAS
       MOV  R9,R10
       ANDI R10,>000F      KEEP NIBBLE 4 OF R9 IN R10
       SRL  R9,4           NIBBLE 2 TO NIBBLE 3
       SOC  R9,R10         'OR' R9 AND R10
       MOVB R8,R10         GET HIGH BYTE
* CONVSC CONTINUES INTO SCORE ROUTINE
*==============================================================
* ADD TO SCORE                                                *
*==============================================================
* ADD R10 TO THE SCORE
* EXAMPLE OF FORMAT FOR THIS ROUTINE:
*   >0125 = DECIMAL 125
SCORE  MOV  R11,R14
       CLR  VDIGIT              CLEAR HIGH BYTE OF VDP DIGIT
       CLR  CARRY
       LI   VDPADD,12           RIGHT MOST SCORE DIGIT
RDIGIT BL   @READCL
       DATA 1,VDGTLB
       MOV  R10,SCRTCH          SAVE R10
       ANDI R10,>000F           GET RIGHT MOST DIGIT
       A    R10,VDIGIT          ADD DIGITS
       A    CARRY,VDIGIT
       CLR  CARRY
       CI   VDIGIT,>0059        DIGIT OVER NINE?
       JLE  WDIGIT
       LI   CARRY,1             SET CARRY
       AI   VDIGIT,-10
WDIGIT BL   @WRITR
       MOV  SCRTCH,R10          RESTORE R10
       SRL  R10,4               NEXT PAIR OF DIGITS
       DEC  VDPADD
       CI   VDPADD,8            PAST LEFT MOST SCORE DIGIT?
       JHE  RDIGIT
       BL   @RRSAB
       LI   VDPADD,8            CHECK SCORE FOR EXTRA GUN
       BL   @READCL
       DATA 2,REG8
       CB   @EGUN3+3,@CLEAR      SKIP IF ALREADY HAVE EXTRA GUN
       JNE  CLDIGT
       CI   R8,>5053            OVER 03000?
       JL   DSCORE
       MOVB @EGUNIP+3,@EGUN3+3  TURN GUN 3 COLOR ON
       LI   INDEX,4
SHFTGR CB   @EGUN1(INDEX)+1,@GUNIP+1  SHIFT OTHER GUNS RIGHT
       JH   NXTGN
       AB   @H18,@EGUN1(INDEX)+1
NXTGN  AI   INDEX,-4
       JEQ  SHFTGR
CLDIGT CB   R8,@LDIGIT           CHECK FOR NEW LEFT SCORE DIGIT
       JLE  WGUNS
       LI   INDEX,16            START WITH GUN 3
DECIND AI   INDEX,-4            FIND A GUN TO REPAIR
       JEQ  WGUNS
```

```
        CB    @EGUN1(INDEX)-2,@EGUNIP+2  IS THIS ONE DAMAGED?
        JEQ   DECIND
        SB    @H68,@EGUN1(INDEX)-3       MOVE GUN LEFT
        MOV   @EGUNIP+2,@EGUN1(INDEX)-2  SET DB PTR AND COLOR
        MOVB  R8,@LDIGIT
WGUNS   BL    @WESAB
DSCORE  B     *R14
        PAGE
*==============================================================
* AUTOMATIC MOVEMENT OF GUN                                   *
*==============================================================
AMGUN   MOV   R11,R14
CLRR8   CLR   R8
        CB    @REG10+1,@GUNPX      TEST FOR HORIZ OR VERTICL MOVE
        JEQ   ZTIME
        INC   R8
ZTIME   CLR   @TIMER
        CLR   @TCSCAN
        MOVB  @HO1,@GUNDIR         FIND DIRECTION
        CB    @GUNPY(R8),@REG10(R8)
        JL    MGUN
        MOVB  @HFF,@GUNDIR
        CLR   @KEYBRD
MGUN    CB    @TIMER,@TCSCAN       MOVE GUN AND LIFT VERTICALLY
        JL    MGUN
        AB    @HO1,@TCSCAN
        CB    @GUNPY(R8),@REG10(R8)
        JEQ   DONEM
        AB    @GUNDIR,@GUNPY(R8)
        CI    R8,1
        JEQ   BLW
        AB    @GUNDIR,@LIFTPY
BLW     BL    @WESAB
        CLR   @KEYBRD
        BL    @SCANI               ALLOW 'REDO' OR 'BACK'
        JMP   MGUN
DONEM   CB    @GUNPY,@REG10
        JNE   CLRR8                REPEAT FOR VERT MOVE IF NEEDED
        B     *R14
        PAGE
*==============================================================
* CALL SOUND                                                  *
*==============================================================
SOUND   CB    @SHIPPY,@HC0    NO OTHER SOUNDS IF SAUCER OUT
        JNE   DSOUND
SOUND2  CB    @STOPSC,@HO2    NO OTHER SOUNDS IF GUN HIT
        JEQ   DSOUND
        CB    @STOPSC,@HO3
        JEQ   DSOUND
SOUND1  MOV   *R11,@SOUBLK  ENTRY POINT FOR HIGH PRIORITY SOUND
        SOCB  @HO1,@>83FD
        MOVB  @HO1,@SOUDC
DSOUND  INCT  R11
        RT
        PAGE
*==============================================================
* READ REMAINING SPRITE ATTRIBUTE BLOCK                       *
*==============================================================
RRSAB   MOV   R11,R15
        LI    VDPADD,>324
        BL    @READCL
        DATA  17,DROP+>24
```

```
        B     *R15
*==============================================================
* WRITE SPRITE ATTRIBUTE BLOCK                                 *
*==============================================================
WPSAB   LI    WCOUNT,36          WRITE PARTIAL SAB
        JMP   MOVR11
WESAB   LI    WCOUNT,53          WRITE ENTIRE SAB
MOVR11  MOV   R11,R15
        LI    VDPADD,>300
        BL    @WRITL
        DATA  DROP
        B     *R15
        PAGE
*==============================================================
* READ FROM VDP                                               *
*==============================================================
* THIS SUBROUTINE HAS THREE ENTRY POINTS
* ALL REQUIRE THAT VDPADD BE LOADED BEFORE THE BL
*
* 'READ' REQUIRES THAT RLOC,RCOUNT BE LOADED BEFORE THE BL
* 'READL' REQUIRES THAT RCOUNT BE LOADED BEFORE THE BL
*    RLOC IS FETCHED FROM A DATA STATEMENT FOLLOWING THE BL
* 'READCL' FETCHES RCOUNT AND RLOC FROM DATA STATEMENTS
*
RDVCL   MOV   *R11+,VDPADD
READCL  MOV   *R11+,RCOUNT
READL   MOV   *R11+,RLOC
READ    LIMI  0                  DISABLE INTERRUPTS
        MOVB  @VADDLB,@VDPADR      SET UP LOWER BYTE OF ADDRESS
        MOV   RCOUNT,COUNT
        MOVB  VDPADD,@VDPADR       SET UP UPPER BYTE OF ADDRESS
        MOV   RLOC,TEMP
R00010  MOVB  @VDPRD,*TEMP+        READ VDP BYTE INTO BUFFER
        DEC   COUNT               DECREMENT COUNTER
        JGT   R00010              AND LOOP IF NEEDED
        LIMI  2                   ENABLE INTERRUPTS
        RT
        PAGE
*==============================================================
* WRITE TO VDP                                                *
*==============================================================
* THIS SUBROUTINE HAS FOUR ENTRY POINTS
* ALL REQUIRE THAT VDPADD BE LOADED BEFORE THE BL
*
* 'WRITE' REQUIRES THAT WLOC,WCOUNT BE LOADED BEFORE THE BL
* 'WRITL' REQUIRES THAT WCOUNT BE LOADED BEFORE THE BL
*    WLOC IS FETCHED FROM A DATA STATEMENT FOLLOWING THE BL
* 'WRITCL' FETCHES WCOUNT AND WLOC FROM DATA STATEMENTS
* 'WRITR' REQUIRES THAT WCOUNT BE LOADED BEFORE THE BL
*    WLOC IS GIVEN THE VALUE OF THE LATEST RLOC
*
WRITR   MOV   RLOC,WLOC
        JMP   WRITE
WRTVCL  MOV   *R11+,VDPADD
WRITCL  MOV   *R11+,WCOUNT
WRITL   MOV   *R11+,WLOC
WRITE   LIMI  0                  DISABLE INTERRUPTS
        MOVB  @VADDLB,@VDPADR      SET UP LOWER BYTE OF ADDRESS
        MOV   WCOUNT,COUNT
        MOV   VDPADD,TEMP
        ORI   TEMP,>4000          SET VDP WRITE FLAG
        MOVB  TEMP,@VDPADR         SET UP UPPER BYTE OF ADDRESS
```

```
        MOV   WLOC,TEMP
W00010  MOVB  *TEMP+,@VDPWD       WRITE BYTE INTO VDP
        DEC   COUNT               DECREMENT COUNTER
        JGT   W00010
        LIMI  2                   ENABLE INTERRUPTS
        RT
*==============================================================
* WRITE TEXT TO VDP                                           *
*==============================================================
*
WTEXT   MOV   *R11+,VDPADD
        MOV   *R11+,WLOC
        MOV   *R11+,BIAS
        SLA   BIAS,8
        LIMI  0                   DISABLE INTERRUPTS
        MOVB  @VADDLB,@VDPADR       SET UP LOWER BYTE OF ADDRESS
        MOV   WCOUNT,COUNT
        MOV   VDPADD,TEMP
        ORI   TEMP,>4000          SET VDP WRITE FLAG
        MOVB  TEMP,@VDPADR         SET UP UPPER BYTE OF ADDRESS
        MOV   WLOC,TEMP
CKFRF   CB    *TEMP,@HFD           CHECK FOR TERMINATION CHARACTER
        JEQ   DWTEXT
        CB    *TEMP,@HFE           CHECK FOR REPEAT FLAG
        JNE   W00030
        INC   TEMP
        MOVB  *TEMP+,R13          GET BYTE TO REPEAT
        MOVB  *TEMP+,R14          GET REPEAT COUNT
        SRL   R14,8
REPEAT  MOVB  R13,@VDPWD           WRITE BYTE
        DEC   R14                 DECREMENT REPEAT COUNT
        JGT   REPEAT
        JMP   CKFRF
W00030  MOVB  *TEMP+,R13
        CB    R13,@H20
        JNE   ABIAS
        LI    R13,>FF00           MAKE BLANKS >FF
        JMP   WSB
ABIAS   AB    BIAS,R13            ADD BIAS
WSB     MOVB  R13,@VDPWD           WRITE SINGLE BYTE
        JMP   CKFRF
DWTEXT  LIMI  2                   ENABLE INTERRUPTS
        RT
*==============================================================
* CHECK TIME TO MOVE INVADER - USED AT END OF GAME
*==============================================================
DELAY   MOV   R11,@RA+6
        CLR   @TIMER
        CLR   @KEYBRD
BSCANI  BL    @SCANI              ALLOW 'REDO' AND 'BACK'
        CB    @TIMER,@H01           CHECK TIMER
        JL    BSCANI
        MOV   @RA+6,R11
        B     *R11
*==============================================================
* MOVE ONE INVADER DOWN
*==============================================================
MOVE1D  MOV   R11,@RA+8
        BL    @DELAY
        AI    R10,>0808               TOP HALF
        BL    @WRITCL
        DATA  2,REG10
```

```
        AI    VDPADD,32
        AI    R10,>0202             BOTTOM HALF
        BL    @WRITE
        BL    @DELAY
        AI    VDPADD,-32            ERASE TOP
        BL    @WRITL
        DATA  HFFFF
        AI    VDPADD,32             BACK TO PATTERN 0
        AI    R10,>F5F6
        BL    @WRITL
        DATA  REG10
        MOV   @RA+8,R11
        B     *R11
*==============================================================
* CHECK FOR NEW HIGH SCORE
*==============================================================
***
***  CHANGED HISC TO BSS 5 HSCORE TO HOLD HI-SCORE
***
HISCOR MOV   R11,R15
        BL    @RDVCL                          READ PLAYER'S SCORE
        DATA  >08,5,PSCORE
        BL    @RDVCL
        DATA  >19,5,HSCORE
        CLR   INDEX
CDIGTS CB    @PSCORE(INDEX),@HSCORE(INDEX)   COMPARE DIGITS
        JH    NEWHI
        JL    NONEW
        INC   INDEX                           CHECK NEXT DIGIT
        CI    INDEX,5
        JL    CDIGTS
        JMP   NONEW
NEWHI   CLR   INDEX                           SET NEW HIGH SCORE
MOVEHI MOVB  @PSCORE(INDEX),@HSCORE(INDEX)
        INC   INDEX
        CI    INDEX,5
        JL    MOVEHI
***
***  MOVE PLAYER'S SCORE TO HIGH SCORE AND THEN SAVE AT VPD >1008
***
        LI    VDPADD,8
        LI    WCOUNT,5
        LI    RLOC,PSCORE
        LI    WLOC,PSCORE
        BL    @READ
        LI    VDPADD,25
        BL    @WRITE
        LI    VDPADD,>1008
        BL    @WRITE
NONEW   B     *R15
SLAST   END
```