

```

1261                .PAGE
1262
1263                ; IRQ HANDLER
1264                ;
1265                ; JUMP      THRU IMMEDIATE IRQ VECTOR WHICH ORDINARILY POINTS TO
1266                ; SYSTEM IRQ:      DETERMINE & CLEAR CAUSE JUMP THRU SOFTWARE VECTOR.
1267
1268                *=INTORG
1269 E6D5 A9 40      IHINIT:    LDA    #$40          ;VBL ON BUF DLIST OFF***FOR NOW***
1270 E6D7 8D 0E D4      STA    NMIEB          ;ENABLE DISPLAY LIST. VERTICAL BLANK
1271 E6DA A9 38      LDA    #$38          ;LOOK AT DATA DIRECTION REGISTERS IN PIA
1272 E6DC 8D 02 D3      STA    PACTL
1273 E6DF SD 03 D3      STA    PBCTL
1274 E6E2 A9 00      LDA    #0           ;MAKE ALL INPUTS
1275 E6E4 8D 00 D3      STA    PORTA
1276 E6E7 8D 01 D3      STA    PORTB
1277 E6EA A9 3C      LDA    #$3C          ;BACK TO PORTS
1278 E6EC SD 02 D3      STA    PACTL
1279 E6EF 8D 03 D3      STA    PBCTL
1280 E6F2 60          RTS
1281 E6F3 6C 16 02      PIRA:    JMP    (VIMIRD)
1282 E6F6 80          CMPTAB:   .BYTE $80          ;BREAK KEY
1283 E6F7 40          .BYTE $40          ;KEY STROKE
1284 E6F8 04          .BYTE $04          ;TIMER 4
1285 E6F9 02          .BYTE $02          ;TIMER 2
1286 E6FA 01          .BYTE $01          ;TIMER 1
1287 E6FB 08          .BYTE $08          ;SERIAL OUT COMPLETE
1288 E6FC 10          .BYTE $10          ;SERIAL OUT READY
1289 E6FD 20          .BYTE $20          ;SERIAL IN READY
1290 E6FE
1291                ;
1292                ; THIS IS A TABLE OF OFFSETS INTO PAGE 2. THEY POINT TO
1292 E6FE 36      ADRTAB:   .BYTE BRKKEY-INTABS
1293 E6FF 08      .BYTE VKEYBD-INTABS
1294 E700 14      .BYTE VTIMR4-INTABS
1295 E701 12      .BYTE VTIMR2-INTABS
1296 E702 10      .BYTE VTIMR1-INTABS
1297 E703 0E      .BYTE VSEROC-INTABS
1298 E704 0C      .BYTE VSEROR-INTABS
1299 E705 0A      .BYTE VSERIN-INTABS

```

```

1300 E706
1301 E706 48          SYIRQ:   PHA              ; SAVE ACCUMULATOR
1302 E707 AD 0E D2    LDA   IRQST          ; CHECK FOR SERIAL IN
1303 E70A 29 20      AND   #$20
1304 E70C D0 0D      BNE   SYIR02
1305 E70E A9 DF      LDA   #$DF          ; MASK ALL OTHERS
1306 E710 8D 0E D2    STA   IRGEN
1307 E713 A5 10      LDA   POKMSK
1308 E715 8D 0E D2    STA   IRQEN
1309 E718 6C 0A 02    JMP   (VSERIN)
1310 E71B 8A          SYIR02:  TXA              ; PUT X INTO ACC
1311 E71C 48          PHA              ; SAVE X ONTO STACK
1312 E71D A2 06      LDX   #$6          ; START WITH SIX OFFSET
1313 E71F BD F6 E6    LOOPM:   LDA   CMPTAB,X ; LOAD MASK
1314 E722 E0 05      CPX   #5          ; CHECK TO SEE IF COMPLETE IS SET
1315 E724 D0 04      BNE   LOOPM2
1316 E726 25 10      AND   POKMSK      ; IS THIS INTERRUPT ENABLED?
1317 E728 F0 05      BEQ   LL
1318 E72A 2C 0E D2    LOOPM2:  BIT   IRQST      ; IS IT THE INTERRUPT?
1319 E72D F0 06      BEQ   JMPP
1320 E72F CA          LL:       DEX              ; NO DEC X AND TRY NEXT MASK
1321 E730 10 ED      BPL   LOOPM      ; IF NOT NEG GOTO LOOPM
1322 E732 4C 62 E7    JMP   SYIRQ8     ; DONE BUT NO INTERRUPT
1323 E735 49 FF      JMPP:     EOR   #$FF      ; COMPLEMENT MASK
1324 E737 8D 0E D2    STA   IRQEN      ; ENABLE ALL OTHERS
1325 E73A A5 10      LDA   POKMSK      ; GET POKE MASK
1326 E73C 8D 0E D2    STA   IRQEN      ; ENABLE THOSE IN POKE MASK
1327 E73F BD FE E6    LDA   ADRTAB,X
1328 E742 AA          TAX
1329 E743 BD 00 02    LDA   INTABS,X    ; GET ADDRESS LOW PART
1330 E746 8D 8C 02    STA   JVECK      ; PUT IN VECTOR
1331 E749 BD 01 02    LDA   INTABS+I,X ; GET ADDRESS HIGH PART
1332 E74C 8D 8D 02    STA   JVECK+1    ; PUT IN VECTOR HIGH PART
1333 E74F 68          PLA              ; PULL X REGISTER FROM STACK
1334 E750 AA          TAX              ; PUT IT INTO X
1335 E751 6C 8C 02    JMP   (JVECK)    ; JUMP TO THE PROPER ROUTINE
1336 E754 A9 00      BRKKY2:  LDA   #0      ; BREAK KEY ROUTINE
1337 E756 85 11      STA   BRKKEY      ; SET BREAK KEY FLAG
1338 E758 8D FF 02    STA   SSFLAG      ; START/STOP FLAG

```

ERR LINE ADDR B1 B2 B3 B4

INTERRUPT HANDLER

Page 3

```

1339 E75B 8D F0 02          STA  CRSINH      ; CURSOR INHIBIT
1340 E75E 85 4D          STA  ATRACT      ; TURN OFF ATRACT MODE
1341 E760 68          PLA
1342 E761 40          RTI          ; EXIT FROM INT
1343 E762 68          SYIRQB: PLA
1344 E763 AA          TAX
1345 E764 2C 02 D3      BIT  PACTL      ; PROCEED ***I GUESS***
1346 E767 10 06      BPL  SYIRQ9
1347 E769 AD 00 D3      LDA  PORTA      ; CLEAR INT STATUS BIT
1348 E76C 6C 02 02      JMP  (VPRCED)
1349 E76F 2C 03 D3      SYIRQ9: BIT  PBCTL      ; INTERRUPT ***I GUESS***
1350 E772 10 06      BPL  SYIRQA
1351 E774 AD 01 D3      LDA  PORTS      ; CLEAR INT STATUS
1352 E777 6C 04 02      JMP  (VINTER)
1353 E77A 68          SYIRQA: PLA
1354 E77B 81 8C 02      STA  JVECK
1355 E77E 68          PLA
1356 E77F 48          PHA
1357 E780 29 10      AND  #$10      ; B BIT OF P REGISTER
1358 E782 F0 07      BEQ  SYRT12
1359 E784 AD 8C 02      LDA  JVECK
1360 E787 48          PHA
1361 E788 6C 06 02      JMP  (VBREAK)
1362 E78B AD 8C 02      SYRT12: LDA  JVECK
1363 E78E 48          PHA
1364 E78F 68          SYIRQB: PLA
1365 E790 40          SYRTI: RTI      ; UNIDENTIFIED INTERRUPT, JUST RETURN.
1366          .PAGE
1367          ;
1368          ; NMI HANDLER
1369          ;
1370          ; DETERMINE CAUSE AND JUMP      ; THRU VECTOR
1371
1372 E791 2C 0F D4      PNMI:  BIT  NMIST
1373 E794 10 03      BPL  PNMI1      ; SEE IF DISPLAY LIST
1374 E796 6C 00 02      JMP  (VDSLST)
1375 E799 48          PNMI1:  PHA
1376 E79A AD 0F D4      LDA  NMIST
1377 E79D 29 20      AND  #$20      ; SEE IF RESET

```

ERR LINE ADDR B1 B2 B3 B4

INTERRUPT HANDLER

Page 4

```

1378 E79F F0 03      BEQ    *+5
1379 E7A1 4C 74 E4    JMP    WARMSV      ; GO THRU WARM START JUMP
1380 E7A4 8A          TXA              ; SAVE REGISTERS
1381 E7A5 48          PHA
1382 E7A6 98          TYA
1383 E7A7 48          PHA
1384 E7A8 8D 0F D4    STA    NMIRES      ; RESET INTERRUPT STATUS
1385 E7AB 6C 22 02    JMP    (VVBLKI)    ; JUMP THRU VECTOR
1386                ;      .PAGE
1387                ;
1388                ; SYSTEM VBLANK ROUTINE
1389                ;
1390                ; INC FRAME COUNTER. PROCESS COUNTDOWN TIMERS. EXIT IF I WAS SET CLEAR
1391                ; SET DLISTL, DLISH. DMACTL FROM RAM CELLS. DO SOFTWARE REPEAT.
1392                ;
1393 E7AE E6 14    SYSVBL: INC    RTCLOK+2    ;INC FRAME COUNTER
1394 E7B0 D0 08    BNE    SYSVB1
1395 E7B2 E6 4D    INC    ATRACT      ;INCREMENT ATRACT (CAUSES ATRACT WHEN MINUS)
1396 E7B4 E6 13    INC    RTCLOK+1
1397 E7B6 D0 02    BNE    SYSVBI
1398 E7B8 E6 12    INC    RTCLOK
1399 E7BA A9 FE    SYSVB1: LDA    #$FE      ; {ATRACT} SET DARK MASK TO NORMAL
1400 E7BC A2 00    LDX    #0          ; SET COLRSH TO NORMAL
1401 E7BE A4 4D    LDY    ATRACT      ; TEST ATRACT FOR NEGATIVE
1402 E7C0 10 06    BPL    VBATRA      ; WHILE POSITIVE DONT GO INTO ATRACT
1403 E7C2 85 4D    STA    ATRACT      ; IN ATRACT, SO STAY BY STA $FE
1404 E7C4 A6 13    LDX    RTCLOK+1    ; COLOR SHIFT FOLLOWS RTCLOK+1
1405 E7C6 A9 F6    LDA    #$F6      ; SET DARK MASK TO DARK
1406 E7CB 85 4E    VBATRA: STA    DRKMSK
1407 E7CA 86 4F    STX    COLRSH
1408 E7CC A2 00    LDX    #0          ; POINT TO TIMER1
1409 E7CE 20 D0 E8    JSR    DCTIMR    ; GO DECREMENT TIMER1
1410 E7DI D0 03    BNE    SYSVB2      ; BRANCH IF STILL COUNTING
1411 E7D3 20 CA EB    JSR    JTIMRI    ; GO JUMP TO ROUTINE
1412 E7D6 A5 42    SYSVB2: LDA    CRITIC
1413 E7D8 D0 08    BNE    XXIT      ; GO IF CRITICAL SET
1414 E7DA BA        TSX              ; SEE IF I WAS SET
1415 E7DB BD 04 01    LDA    $104,X    ; GET STACKED P
1416 E7DE 29 04    AND    #$04      ; I BIT

```

ERR LINE ADDR B1 B2 B3 B4

INTERRUPT HANDLER

Page 5

```

1417 E7E0 F0 03          BEQ  SYSVB3      ; CBRANCH IF OK
1418 E7E2 4C 05 E9      XXIT:  JMP  XITVBL      ; I WAS SET EXIT
1419 E7E5 AD 0D D4      SYSVB3: LDA  PENV
1420 E7EB 8D 35 02          STA  LPENV
1421 E7EB AD 0C D4          LDA  PENH
1422 E7EE 8D 34 02          STA  LPENH
1423 E7F1 AD 31 02          LDA  SDLSTH
1424 E7F4 8D 03 D4          STA  DLISTH
1425 E7F7 AD 30 02          LDA  SDLSTL
1426 E7FA 8D 02 D4          STA  DLISTL
1427 E7FD AD 2F 02          LDA  SDMCTL
1428 EB00 BD 00 D4          STA  DMACTL
1429 E803 AD 6F 02          LDA  GPRIOR      ; GLOBAL PRIOR
1430 E806 8D 1B D0          STA  PRIOR
1431 E809 A2 08          LDX  #$08      ; TURN OFF KEYBOARD SPEAKER
1432 EB0B 8E 1F D0          STX  CONSOL
1433 EB0E 58          SCOLLP: CLI      ; DISABLE INTERRUPTS
1434 EB0F BD C0 02          LDA  PCOLR0,X    ; LOAD COLOR REGISTERS FROM RAM
1435 E812 45 4F          EOR  COLRSH      ; DO COLOR SHIFT .
1436 E814 25 4E          AND  DRKMSK      ; AND DARK ATTRACT
1437 E816 9D 12 D0          STA  COLPMO,X
1438 E819 CA          DEX
1439 E81A 10 F2          BPL  SCOLLP
1440 E81C AD F4 02          LDA  CHBAS
1441 E81F 8D 09 D4          STA  CHBASE
1442 E822 AD F3 02          LDA  CHACT
1443 E825 8D 01 D4          STA  CHACTL
1444 E828 A2 02          LDX  #2      ; POINT TO TIMER 2
1445 E82A 20 D0 ES          JSR  DCTIMR
1446 E82D D0 03          BNE  SYSVB4      ; IF DIDNT GO ZERO
1447 E82F 20 CD ES          JSR  JTIMR2      ; GO JUMP TO TIMER2 ROUTINE
1448 E832 A2 02          SYSVB4: LDX  #2      ; RESTORE X
1449 E834 E8          SYSVB: INX
1450 E835 E8          INX
1451 E836 BD 18 02          LDA  CDTMVI,X
1452 E839 ID 19 02          ORA  CDTMVI+I,X
1453 E83C F0 06          BEQ  SYSVBA
1454 E83E 20 D0 E8          JSR  DCTIMR      ; DECREMENT AND SET FLAG IF NONZERO
1455 E841 9D 26 02          STA  CDTMF3-4,X

```

```

1456 E844 E0 08      SYSVBA:    CPX    #8          ; SEE IF DONE ALL 3
1457 E846 D0 EC      BNE    SYSVBB      ; LOOP
1458                ; CHECK    DEBOUNCE COUNTER
1459 E848 AD 0F D2    LDA    SKSTAT
1460 E84B 29 04      AND    #$04          ; KEY DOWN BIT
1461 E84D F0 08      BEQ    SYVB6A      ; IF KEY DOWN
1462                ; KEY UP SO COUNT IT
1463 E84F AD F1 02    LDA    KEYDEL      ; KEY DELAY COUNTER
1464 E852 F0 03      BEQ    SYVB6A      ; IF COUNTED DOWN ALREADY
1465 E854 CE F1 02    DEC    KEYDEL      ; COUNT IT
1466                ; CHECK    SOFTWARE REPEAT TIMER
1467 E857 AD 2B 02    SYVB6A:    LDA    SRTIMR
1468 E85A F0 17      BEQ    SYSVB7      ; DOESN'T COUNT
1469 E85C AD 0F D2    LDA    SKSTAT
1470 E85F 29 04      AND    #$04          ; CHECK KEY DOWN BIT
1471 E861 D0 60      BNE    SYSVB6      ; BRANCH IF NO LONGER DOWN
1472 E863 CE 2B 02    DEC    SRTIMR      ; COUNT FRAME OF KEY DOWN
1473 E866 D0 0B      BNE    SYSVB7      ; BRANCH IF NOT RUN OUT
1474                ; TIMER RAN OUT - RESET AND SIMULATE KEYBOARD IRQ
1475 E868 A9 06      LDA    #SRTIM2     ; TIMER VALUE
1476 E86A 8D 2B 02    STA    SRTIMR      ; SET TIMER
1477 E86D AD 09 D2    LDA    KBCODE     ; GET THE KEY
1478 E870 8D FC 02    STA    CH          ; PUT INTO CH
1479                ; READ GAME CONTROLLERS
1480 E873 A0 01      SYSVB7:    LDY    #1
1481 E875 A2 03      LDX    #3
1482 E877 B9 00 D3    STLOOP:    LDA    PORTA,Y
1483 E87A 4A          LSR    A
1484 E87B 4A          LSR    A
1485 E87C 4A          LSR    A
1486 E87D 4A          LSR    A
1487 E87E 9D 78 02    STA    STICK0,X    ; STORE JOYSTICK
1488 E881 CA          DEX
1489 E882 B9 00 D3    LDA    PORTA.Y
1490 E885 29 0F      AND    #$F
1491 E887 9D 78 02    STA    STICK0,X    ; STORE JOYSTICK
1492 E88A CA          DEX
1493 E88B 88          DEY
1494 EBBC 10 E9      BPL    STLOOP

```

```

1495
1496 E88E A2 03
1497 E890 BD 10 D0
1498 E893 9D 84 02
1499 E896 BD 00 D2
1500 E899 9D 70 02
1501 E89C BD 04 D2
1502 E89F 9D 74 02
1503 EBA2 CA
1504 EBA3 10 EB
1505 EBA5 8D 0B D2
1506
1507 EBAB A2 06
1508 E8AA A0 03
1509 E8AC 89 78 02
1510 EBAF 4A
1511 EBB0 4A
1512 EBB1 4A
1513 EBB2 9D 7D 02
1514 E8B5 A9 00
1515 ESB7 2A
1516 EBB5 9D 7C 02
1517 EBBB CA
1518 EBBC CA
1519 EBBD 88
1520 EBBE 10 EC
1521
1522 EBC0 6C 24 02
1523 00E8
1524 0073
1525 EBC3 A9 00
1526 EBC5 8D 2B 02
1527 ESCS F0 A9
1528 EBCA 6C 26 02
1529 EBCD 6C 28 02
1530
1531
1532
1533

;
LDX #3
STRL: LDA TRIGO,X ; MOVE JOYSTICK TRIGGERS
STA STRIG0,X
LDA POT0,X ; MOVE POT VALUES
STA PADDL0,X
LDA POT4,X
STA PADDL4,X
DEX
BPL STRL
STA POTGO ; START POTS FOR NEXT TIME
;
LDX #6
LDY #3
PTRLP: LDA STICKO,Y ; TRANSFER BITS FROM JOYSTICKS
LSR A ; TO PADDLE TRIGGERS
LSR A
LSR A
STA PTRIG1,X
LDA #0
ROL A
STA PTRIG0,X
DEX
DEX
DEY
BPL PTRLP
;
JMP (VVBLKD) ; GO TO DEFERRED VBLANK ROUTINE
SV7H = SYSVB7/256
SV7L = (-256)*SV7H+SYSVB7
SYSVB6: LDA #0
STA SRTIMR ; ZERO TIMER
BEQ SYSVB7 ; UNCOND
JTIMR1: JMP (CDTMA1)
JTIMR2: JMP (CDTMA2)
;
; SUBROUTINE TO DECREMENT A COUNTDOWN TIMER
: ENTRY X=OFFSET FROM TIMER 1
: EXIT A,P=ZERO IF WENT ZERO, FF OTHERWISE

```

```

1534      ;
1535 ESD0  BC 18 02      DCTIMR:  LDY  CDTMVI,X      ; LO BYTE
1536 EBD3  D0 08      BNE  DCTIM1      ; NONZERO, GO DEC IT
1537 ESD5  BC 19 02      LDY  CDTMV1+1,X      ; SEE IF BOTH ZERO
1538 ESD8  F0 10      BEQ  DCTXF      ; YES, EXIT NONZERO
1539 EBDA  DE 19 02      DEC  CDTMVI+1,X      ; DEC HI BYTE
1540 EBDD  DE 18 02      DCTIM1:  DEC  CDTMVI,X      ; DEC LO BYTE
1541 EBE0  D0 08      BNE  DCTXF
1542 EBE2  BC 19 02      LDY  CDTMV1+1,X
1543 EBE5  D0 03      BNE  DCTXF
1544 EBE7  A9 00      LDA  #0      ; WENT ZERO, RETURN ZERO
1545 EBE9  60      RTS
1546 E8EA  A9 FF      DCTXF:  LDA  #$FF      ; RETURN NONZERO
1547 EBEC  60      RTS
1548      ;      .PAGE
1549      ;
1550      ; SUBROUTINE TO SET VERTIC      AL BLANK VECTORS AND TIMERS
1551      ; ENTRY X=HI, Y=LO BYTE TOSET
1552      ;      A=1-5 TIMERS 1-5
1553      ;      6 IMM VBLANK
1554      ;      7 DEF VBLANK
1555      ;
1556 EKED  0A      SETVBL:  ASL  A      ; MUL BY 2
1557 EBEE  8D 2D 02      STA  INTEMP
1558 ESF1  8A      TXA
1559 E8F2  A2 05      LDX  #5
1560 ESF4  8D 0A D4      STA  WSYNC      ; WASTE 20 CPU CYCLES
1561 EBF7  CA      SETLOP:  DEX      ; TO ALOWD VBLANK TO HAPPEN
1562 EBF8  D0 FD      BNE  SETLOP      ; IF THIS IS LINE "7C"
1563 EBFA  AE 2D 02      LDX  INTEMP
1564 EBFD  9D 17 02      STA  CDTMV1-1,X
1565 E900  98      TYA
1566 E901  9D 16 02      STA  CDTMV1-2,X
1567 E904  60      RTS
1568      ;
1569      ; EXIT FROM VERTICAL BLANK
1570      ;
1571 E905  68      XITVBL:  PLA      ; UNSTACK Y
1572 E906  A8      TAY

```



ERR LINE ADDR B1 B2 B3 B4

INTERRUPT HANDLER

Page 9

```

1573 E907 68          PLA          ; UNSTACK X
1574 E908 AA          TAX
1575 E909 68          PLA          ; UNSTACK A
1576 E90A 40          RTI          ; AND GO BACK FROM WHENCE.
1577 00E6             PIRQH        =      PIRG/256
1578 00F3             PIRQL        =      (-256)*PIRQH+PIRQ
1579 00E7             PNMIH        =      PNMI/256
1580 0091             PNMIL        =      (-256)*PNMIH+PNMI
1581             ; SPARE          BYTE OR MODULE TOO      LONG FLAG
1582 E90B             CRNTP2        =*
1583             *=$14
1584 0014 39          INTSPR:      .BYTE SIOORG-CRNTP2 ; ^GINTHV IS T00 LONG
1585             ;

```