

fix non-ctrl

```

10 0000 ;*****
20 0000 ;
30 0000 ;   **** MARKSMAN DRIVER ****
40 0000 ;
50 0000 ;
60 0000 ; 10/18/79 L. SHUSTEK TEST DRIVER
70 0000 ; 12/ 2/79 L. SHUSTEK INTEGRATE INTO SYSTEM
80 0000 ; 1/13/80 L. SHUSTEK CHANGE I/O FOR 8255
90 0000 ;
100 0000 ;*****
110 0000 ;
120 0000 ;   HARD DISK CONFIGURATION
130 0000 ;
140 0000 HDSPT =76          SECTORS PER TRACK
150 0000 HDTPC =4          TRACKS PER CYLINDER
160 0000 HDSPC =HDSPT*HDTPC SECTORS PER CYLINDER
170 0000 ;
180 0000 ;   PIA PORTS
190 0000 ;
200 0000 B8255 =$BFF0      PET BASE ADDR (APPLE IS C0N0)
210 0000 PORTA =B8255+0    INPUT BITS
220 0000 PORTB =B8255+1    BIDIRECTIONAL DATA
230 0000 PORTC =B8255+2    HANDSHAKE AND OUTPUT BITS
240 0000 CNTRL =B8255+3    CONTROL REGISTER
250 0000 ;
260 0000 ;   PORT A: INPUT BITS
270 0000 ;
280 0000 HSRDY =$80        HANDSHAKE READY (WIRED TO PC0)
290 0000 READY =$40        DISK IS READY
300 0000 STATUS =$20       DISK IS PRESENTING STATUS
310 0000 DATA =$10        DISK IS PRESENTING DATA
320 0000 INPUT =$08        DISK IS IN INPUT MODE
330 0000 NACK =$04         NOT ACK FROM DISK
340 0000 NREQ =$02         NOT REQ TO DISK
350 0000 INTCH =$01        INT CHANGED SINCE OUR HS
360 0000 ;
370 0000 ;   PORT C: HANDSHAKE AND OUTPUT BITS
380 0000 ;
390 0000 ABORT =$80 (OUT)  HIGH CAUSES ABORT
400 0000 SEQNCE =$40 (OUT) - NOT IMPLEMENTED ON THIS DISK
410 0000 REQINV =$10 (OUT) HIGH INVERTS -OBF TO REQ
420 0000 NREQDS =$08 (OUT) LOW DISABLES REQ (FORCES 0)
430 0000 INTE =$04 (OUT)  INTERRUPT LINE (HSRDY) ENB
440 0000 REQ =$02 (HS)    REQ (OR NOT REQ) TO DISK
450 0000 INT =$01 (HS)    INTERRUPT LINE (PC0)
460 0000 ;
470 0000 ;   CONTROL WORDS
480 0000 CMDIN  =$97 W new - to be 96 A:MODE0 IN,C UPPER:OUT,B:MODE1 IN
490 0000 CMDOUT =$94      A:MODE0 IN,C UPPER:OUT,B:MODE1 OUT
500 0000 CMDDIS =$06      CLR PC3 (NREQDS)
510 0000 CMDENB =$07      SET PC3 (NREQDS)
520 0000 CMDNAB =$0E      CLR PC7 (ABORT)
530 0000 CMDINV =$09      SET PC4 (REQINV)
540 0000 CMDNIV =$08      CLR PC4 (REQINV)
550 0000 CMDINT =$05      SET PC2 (ENABLE INT =PC0=HSRDY)
560 0000 ;
570 0000 ;   MARKSMAN OPCODES
580 0000 ;
590 0000 STATOP =$00      GET STATUS
600 0000 SEEKOP =$48      SEEK CYL,HEAD

```



```

010 0000 READOP = $A0 READ SECTOR
620 0000 WRITOP = $90 WRITE SECTOR WITHOUT VERIFY
630 0000 RBUFOP = $D0 READ BUFFER
640 0000 WBUFOP = $C0 WRITE BUFFER
650 0000 ;
660 0000 ;
670 9060 *=MLTPLY
680 9060 4C9DA8 JMP MULT
690 9063 *=DIVIDE
700 9063 4C4AA8 JMP DIV
710 905D *=HDSKOP
720 905D 4CC0A5 JMP START
730 A5C0 *=HDKRTN
740 A5C0 ;
750 A5C0 ;*****
760 A5C0 ;
770 A5C0 ; SETUP
780 A5C0 ;
790 A5C0 ; ENTRY/EXIT CONDITIONS ARE THE SAME AS FOR
800 A5C0 ; THE FLOPPY DISK LOW LEVEL ROUTINES,
810 A5C0 ; EXCEPT THAT Y CONTAINS THE OPERATION CODE
820 A5C0 ; (READ=0, WRITE=80, VERIFY=40), AND DRETRY=0
830 A5C0 ;
840 A5C0 ; SPECIAL RETURN CODES:
850 A5C0 ;
860 A5C0 ; FF DISK STATUS FAILURE
870 A5C0 ; FE ILLEGAL COMMAND TO DISK
880 A5C0 ; 03 VERIFY FAILED
890 A5C0 ;
900 A5C0 ;*****
910 A5C0 ;
920 A5C0 08 START PHP SAVE ENABLE STATUS
930 A5C1 EA NOP *** CHANGE TO SEI (78) TO DISABLE
940 A5C2 98 TYA SAVE OPERATION CODE
950 A5C3 48 PHA
960 A5C4 A53F LDA DDRV VALIDATE DRIVE #
970 A5C6 C903 CMP #3
980 A5C8 9004 BCC DRERR
990 A5CA C91C CMP #28
1000 A5CC 9005 BCC DROK LT
1010 A5CE A90B DRERR LDA #11 BAD DRIVE ERROR CODE
1020 A5D0 4C36A7 JMP RET
1030 A5D3 ;
1040 A5D3 20E7A7 DROK JSR PIAINI SETUP PIA
1050 A5D6 ADF0BF DRCHK LDA PORTA
1060 A5D9 2968 AND #READY+STATUS+INPUT POWERED ON?
1070 A5DB C968 CMP #READY+STATUS+INPUT
1080 A5DD F00B BEQ DRNR NO (ALL LINES FFLOATED)
1090 A5DF 2920 AND #STATUS
1100 A5E1 D014 BNE GETST TRYING TO GIVE US STATUS...
1110 A5E3 ADF0BF LDA PORTA
1120 A5E6 2940 AND #READY
1130 A5E8 D005 BNE DRRDY READY...
1140 A5EA A980 DRNR LDA #$80 "NOT READY" ERROR
1150 A5EC 4C36A7 JMP RET
1160 A5EF ADF0BF DRRDY LDA PORTA
1170 A5F2 2904 AND #NACK IS ACK ALREADY UP (NACK LOW)
1180 A5F4 D00A BNE RESTRT
1190 A5F6 00 BRK YES (ARRGH)
1200 A5F7 202FA8 GETST JSR RCV GET STATUS
1210 A5FA 202AF3 JSR STPTST
1220 A5FD D0D7 BNE DRCHK TRY AGAIN
1230 A5FF 00 BRK
1240 A600 ;
1250 A600 ; COPY INPUT PARMS
1260 A600 ;

```

CMP #B8255/253
   
 BEQ DRNR

```

1270 AC000 08 RESUME PLA GET ORIGINAL OPERATION
1280 A601 48 PHA
1290 A602 8D39AF NEXTOP STA CUROP SAVE CURRENT OPERATION
1300 A605 A544 LDA DSIZ
1310 A607 8D65AF STA HDSIZ #BYTES TO TRANSFER
1320 A60A A545 LDA DSIZ+1
1330 A60C 8D66AF STA HDSIZ+1
1340 A60F A546 LDA DADR PUT PTR ON PAGE ZERO
1350 A611 8549 STA PNTL
1360 A613 A547 LDA DADR+1
1370 A615 854A STA PNTH
1380 A617 ;
1390 A617 ; CONVERT DDRV/DBLK INTO CYL/HEAD/SECTOR
1400 A617 ;
1410 A617 A9A0 LDA #TBNUMD*256/256
1420 A619 8D69AF STA DIVDND+2 COMPUTE (DRV-3)*2464
1430 A61C A909 LDA #TBNUMD/256
1440 A61E 8D6AAF STA DIVDND+3
1450 A621 38 SEC
1460 A622 A53F LDA DDRV
1470 A624 E903 SBC #3
1480 A626 8D6BAF STA DIVISR
1490 A629 A900 LDA #0
1500 A62B 8D6CAF STA DIVISR+1
1510 A62E 209DA8 JSR MULT
1520 A631 18 CLC
1530 A632 AD67AF LDA DIVDND ADD DBLK
1540 A635 6542 ADC DBLK
1550 A637 8D67AF STA DIVDND
1560 A63A AD68AF LDA DIVDND+1
1570 A63D 6543 ADC DBLK+1
1580 A63F 8D68AF STA DIVDND+1
1590 A642 AD69AF LDA DIVDND+2
1600 A645 6900 ADC #0
1610 A647 8D69AF STA DIVDND+2
1620 A64A A930 LDA #HDSPC*256/256 DIVIDE BY 76*4
1630 A64C 8D6BAF STA DIVISR
1640 A64F A901 LDA #HDSPC/256
1650 A651 8D6CAF STA DIVISR+1
1660 A654 204AA8 JSR DIV
1670 A657 AD67AF LDA DIVDND
1680 A65A 8D62AF STA HDCYL QUOTIENT IS CYLINDER #
1690 A65D AD6AAF LDA DIVDND+3
1700 A660 8D68AF STA DIVDND+1 SHIFT REMAINDER TO DIVDND
1710 A663 AD69AF LDA DIVDND+2
1720 A666 8D67AF STA DIVDND+0
1730 A669 A900 LDA #0
1740 A66B 8D6AAF STA DIVDND+3
1750 A66E 8D69AF STA DIVDND+2
1760 A671 8D6CAF STA DIVISR+1
1770 A674 A94C LDA #HDSPT DIVIDE BY 76
1780 A676 8D6BAF STA DIVISR
1790 A679 204AA8 JSR DIV
1800 A67C AD67AF LDA DIVDND QUOTIENT IS HEAD #
1810 A67F 8D63AF STA HDHEAD
1820 A682 AD69AF LDA DIVDND+2 REMAINDER IS SECTOR #
1830 A685 8D64AF STA HDSECT
1840 A688 ;
1850 A688 A544 LDA DSIZ CHECK LENGTH
1860 A68A 0545 ORA DSIZ+1
1870 A68C D003 BNE HDSEEK
1880 A68E 4C2AA7 JMP GXIT NOTHING TO DO IF ZEROLENGTH
1890 A691 ;
1900 A691 ;
1910 A691 ; SEEK
1920 A691 ;

```

```

1930 AC91 AD03AF HDSSEK LDA HDHEAD
1940 A694 0948 ORA #SEEKOP
1950 A696 2041A8 JSR SEND "SEEK"
1960 A699 AD62AF LDA HDCYL
1970 A69C 2041A8 JSR SEND
1980 A69F 20ACA7 JSR CKSTAT
1990 A6A2 ;
2000 A6A2 ; SETUP BYTE COUNT IN X AND START
2010 A6A2 ;
2020 A6A2 AD66AF SETCNT LDA HDSIZ+1
2030 A6A5 F007 BEQ SETSML LT 256...
2040 A6A7 CE66AF DEC HDSIZ+1
2050 A6AA A200 LDX #0 DO 256
2060 A6AC F003 BEQ SETJMP (UNC)
2070 A6AE AE65AF SETSML LDX HDSIZ DO 1-255
2080 A6B1 AD39AF SETJMP LDA CUROP WHAT OPERATION?
2090 A6B4 1003 BPL RD
2100 A6B6 4C6AA7 JMP WR
2110 A6B9 ;
2120 A6B9 ; READ/VERIFY SECTORS
2130 A6B9 ;
2140 A6B9 A9A0 RD LDA #READOP
2150 A6BB 2041A8 JSR SEND "READ SECTOR"
2160 A6BE 2015A7 JSR SNDSFC SEND SECTOR
2170 A6C1 20ACA7 JSR CKSTAT
2180 A6C4 A9D0 ⊗ LDA #RBUFOP
2190 A6C6 2041A8 JSR SEND "READ BUFFER"
2200 A6C9 20D4A7 JSR BUSIN TURN BUS INWARD
2210 A6CC A000 LDY #0
2220 A6CE AD39AF LDA CUROP
2230 A6D1 D075 BNE VFLP VERIFY INNER LOOP...
2240 A6D3 ADF0BF RDLP LDA PORTA WAIT FOR INTR BIT
2250 A6D6 10FB BPL RDLP
2260 A6D8 ADF1BF LDA PORTB GET DATA; HANDSHAKE
2270 A6DB 9149 STA (PNTL),Y
2280 A6DD C8 INY
2290 A6DE CA DEX
2300 A6DF D0F2 BNE RDLP 23 USEC LOOP
2310 A6E1 98 RDEND TYA END OF SECTOR?
2320 A6E2 D03D BNE RDLST NO: FINISH LAST SECTOR
2330 A6E4 20ACA7 JSR CKSTAT YES: CHECK STATUS
2340 A6E7 ;
2350 A6E7 ; INCREMENT TO NEXT SECTOR
2360 A6E7 ;
2370 A6E7 AD65AF INCSEK LDA HDSIZ ←
2380 A6EA 0D66AF ORA HDSIZ+1
2390 A6ED F03B BEQ GXIT DONE WITH ALL
2400 A6EF E64A INC PNTH NEXT PAGE
2410 A6F1 EE64AF INC HDSECT NEXT SECTOR
2420 A6F4 AD64AF LDA HDSECT
2430 A6F7 C94C CMP #HDSPT
2440 A6F9 30A7 BMI SETCNT SAME TRK...
2450 A6FB ;
2460 A6FB ; INCREMENT HEAD/CYLINDER
2470 A6FB ;
2480 A6FB A900 LDA #0
2490 A6FD 8D64AF STA HDSECT
2500 A700 EE63AF INC HDHEAD NEXT HEAD
2510 A703 AD63AF LDA HDHEAD
2520 A706 C904 CMP #4
2530 A708 3087 BMI HDSEK
2540 A70A A900 LDA #00
2550 A70C 8D63AF STA HDHEAD
2560 A70F EE62AF INC HDCYL NEXT CYLINDER
2570 A712 4C91A6 JMP HDSEK
2580 A715 ;

```

C A6E7  
 A940  
 LDA #540  
 STA curop  
 SD 39AF  
 JMP ⊕  
 4C C4A6



```

; SEND PERMITTED SECTOR # TO DISK
2600 A715 ;
2610 A715 AD64AF SNDSEC LDA HDSECT
2620 A718 18 CLC EVEN # 0-74 BECOME 0-37
2630 A719 6A ROR A ODD # 1-75 BECOME 38-75
2640 A71A 9002 BCC SNDSEZ
2650 A71C 6925 ADC #37 (ADDS 38)
2660 A71E 4C41A8 SNDSEZ JMP SEND SEND IT AND RETURN
2670 A721 ;
2680 A721 ; CLEANUP LAST SECTOR FOR READ
2690 A721 ;
2700 A721 RDLST
2710 A721 202FA8 RDNL JSR RCV SKIP TIL END OF SECTOR
2720 A724 C8 INY
2730 A725 D0FA BNE RDNL
2740 A727 20ACA7 JSR CKSTAT FINAL STATUS
2750 A72A ;
2760 A72A ; EXITS
2770 A72A ;
2780 A72A 2C39AF GXIT BIT CUROP WHAT OPERATION?
2790 A72D 7005 BVS GRET VERIFY: EXIT
2800 A72F ; CHANGE ABOVE TO BPL (10) TO VERIFY ONLY WRITES
2810 A72F ; CHANGE ABOVE TO BVS (70) TO VERIFY READS AND WRITES
2820 A72F A940 LDA #40 CHANGE TO VERIFY
2830 A731 4C02A6 JMP NEXTOP
2840 A734 ;
2850 A734 A900 GRET LDA #00 GOOD RETURN
2860 A736 ;
2870 A736 A8 RET TAY SAVE RETCODE
2880 A737 68 PLA PURGE OPERATION
2890 A738 28 PLP **** RESTORE ENB/DIS STATE
2900 A739 98 TYA RESTORE RETCODE, SET NE/EQ
2910 A73A 60 RTS RETURN TO CALLER
2920 A73B ;
2930 A73B EE38AF ERR INC DRETRY ERROR: RETCODE IS IN A
2940 A73E AE38AF LDX DRETRY
2950 A741 E004 CPX #4 *** NUMBER OF ATTEMPTS TO MAKE
2960 A743 F0F1 BEQ RET
2970 A745 4C00A6 JMP RESTRT RETRY FROM THE BEGINNING
2980 A748 ;
2990 A748 ; VERIFY INNER LOOP
3000 A748 ;
3010 A748 ADF0BF VFPL LDA PORTA WAIT FOR INTR BIT
3020 A74B 10FB BPL VFPL
3030 A74D ADF1BF LDA PORTB GET DATA; HANDSHAKE
3040 A750 D149 CMP (PNTL),Y
3050 A752 D006 BNE VFNG NG
3060 A754 C8 INY
3070 A755 CA DEX
3080 A756 D0F0 BNE VFPL
3090 A758 F087 BEQ RDEND CONTINUE AS IN READ
3100 A75A ;
3110 A75A C8 VFNG INY MORE IN THIS SECTOR?
3120 A75B F006 BEQ VFST NO
3130 A75D 202FA8 VFNGL JSR RCV YES: SKIP IT
3140 A760 C8 INY
3150 A761 D0FA BNE VFNGL
3160 A763 20ACA7 VFST JSR CKSTAT FINAL STATUS
3170 A766 A903 LDA #3 "VERIFY FAILED"
3180 A768 D0D1 BNE ERR (UNC)
3190 A76A ;
3200 A76A ; WRITE SECTORS
3210 A76A ;
3220 A76A A9C0 WR LDA #WBUFOP
3230 A76C 2041A8 JSR SEND "WRITE TO BUFFER"
3240 A76F A000 LDY #0

```

```

3260 A774 10FB      BPL WRLP
3270 A776 B149      LDA (PNTL),Y
3280 A778 8DF1BF    STA PORTB      SEND; HANDSHAKE
3290 A77B C8        INY
3300 A77C CA        DEX
3310 A77D D0F2      BNE WRLP      23 USEC LOOP
3320 A77F 98        TYA          END OF SECTOR?
3330 A780 D011      BNE WRLST    NO: FINISH LAST SECTOR
3340 A782 20ACA7    JSR CKSTAT
3350 A785 A990      LDA #WRITOP
3360 A787 2041A8    JSR SEND     "WRITE SECTOR"
3370 A78A 2015A7    JSR SNDSEC  SEND SECTOR
3380 A78D 20ACA7    JSR CKSTAT
3390 A790 4CE7A6    JMP INCSEC
3400 A793          ;
3410 A793          ;      CLEANUP WRITE OF LAST SECTOR
3420 A793          ;
3430 A793 A900      WRLST  LDA #0      FILL WITH ZZEROS
3440 A795 2041A8    WRNL   JSR SEND
3450 A798 C8        INY
3460 A799 D0FA      BNE WRNL
3470 A79B 20ACA7    JSR CKSTAT
3480 A79E A990      LDA #WRITOP
3490 A7A0 2041A8    JSR SEND     "WRITE SECTOR"
3500 A7A3 2015A7    JSR SNDSEC  SEND SECTOR
3510 A7A6 20ACA7    JSR CKSTAT
3520 A7A9 4C2AA7    JMP GXIT    GOOD EXIT
3530 A7AC          ;*****
3540 A7AC          ;
3550 A7AC          ; EVERYTHING ABOVE HERE IS OPEN CODE.
3560 A7AC          ; THE STACK CONTAINS THE RETURN ADDRESS, SAVED STATUS
3570 A7AC          ; REGISTER, AND THE ORIGINAL OPERATION CODE.
3580 A7AC          ;
3590 A7AC          ; ROUTINES BELOW HERE ARE CLOSED SUBROUTINES.
3600 A7AC          ;
3610 A7AC          ;*****
3620 A7AC          ;
3630 A7AC          ;      CHECK STATUS
3640 A7AC          ;
3650 A7AC          ;      READS STATUS FROM PREVIOUS OPERATION.
3660 A7AC          ;      IF BAD, PURGE RETURN ADDR AND GO TO ERR
3670 A7AC          ;      DESTROYS ONLY A
3680 A7AC          ;
3690 A7AC 202FA8    CKSTAT JSR RCV      GET STATUS 0
3700 A7AF 2988      AND #588     ILLEGAL CMD + ERROR
3710 A7B1 F020      BEQ CKSTZ    NEITHER
3720 A7B3 3005      BMI CKSTS1  ERROR...
3730 A7B5 A9FE      LDA #5FE     ILLEGAL CMD
3740 A7B7 4CC8A7    JMP CKSTER
3750 A7BA A901      CKSTS1 LDA #STATOP+1
3760 A7BC 2041A8    JSR SEND     "GET STATUS 1"
3770 A7BF 202FA8    JSR RCV
3780 A7C2 C900      CMP #0
3790 A7C4 D002      BNE CKSTER
3800 A7C6 A9FF      LDA #5FF     ??? NO ERROR STATUS
3810 A7C8 8D34AF    CKSTER STA TEMP
3820 A7CB 68        PLA          PURGE RETURN ADDR
3830 A7CC 68        PLA
3840 A7CD AD34AF    LDA TEMP
3850 A7D0 4C3BA7    JMP ERR      ERROR EXIT
3860 A7D3 60        CKSTZ  RTS
3870 A7D4          ;
3880 A7D4          ; BUSIN  INITIALIZE PIA FOR INPUT
3890 A7D4          ;
3900 A7D4 ADF2BF    BUSIN  LDA PORTC

```

```

3910 A7D7 2910      AND #REQINV  ALREADY IN?
3920 A7D9 F025      BEQ BUSINZ   YES: EXIT
3930 A7DB 202AF3  BUSINL JSR STPTST *TEMP
3940 A7DE F021      BEQ INSTP   *TEMP
3950 A7E0 ADF0BF    LDA PORTA   WAIT FOR PREV XMIT
3960 A7E3 2901      AND #INTCH  BY SEEING IF INT HAD A TRANSITION
3970 A7E5 F0F4      BEQ BUSINL
3980 A7E7          ; (YOU CAN'T JUST LOOK AT INT, SINCE THE INCOMING
3990 A7E7          ; HANDSHAKE FROM THE DISK MAY ALREADY HAVE SET IT TO
4000 A7E7          ; INDICATE THAT THE NEW DATA HAS ARRIVED.)
4010 A7E7 A906      PIAINI LDA #CMDDIS DISABLE REQ LINE
4020 A7E9 8DF3BF    STA CNTRL
4030 A7EC A997      LDA #CMDIN  PROGRAM INPUT MODE
4040 A7EE 8DF3BF    STA CNTRL  (ALSO CLRS PORT C)
4050 A7F1 A908      LDA #CMDNIV DON'T INVERT PC1 (IBF)
4060 A7F3 8DF3BF    STA CNTRL
4070 A7F6 A905      LDA #CMDINT ENABLE INTR BIT
4080 A7F8 8DF3BF    STA CNTRL
4090 A7FB A907      LDA #CMDENB ENABLE REQ
4100 A7FD 8DF3BF    STA CNTRL
4110 A800 60        BUSINZ RTS
4120 A801 00        INSTP BRK          *TEMP
4130 A802          ;
4140 A802          ; BUSOUT  INITIALIZE PIA FOR OUTPUT
4150 A802          ;          DESTROYS ONLY A
4160 A802          ;
4170 A802 ADF2BF  BUSOUT LDA PORTC
4180 A805 2910      AND #REQINV ALREADY OUT?
4190 A807 D01A      BNE BUSWAI  YES: WAIT FOR PREV XMIT
4200 A809 A906      LDA #CMDDIS DISABLE REQ LINE
4210 A80B 8DF3BF    STA CNTRL
4220 A80E A994      LDA #CMDOUT PROGRAM OUTPUT MODE
4230 A810 8DF3BF    STA CNTRL  (ALSO CLRS PORT C)
4240 A813 A909      LDA #CMDINV INVERT PC1 (-OBF)
4250 A815 8DF3BF    STA CNTRL
4260 A818 A905      LDA #CMDINT ENABLE INTR BIT
4270 A81A 8DF3BF    STA CNTRL
4280 A81D A907      LDA #CMDENB ENABLE REQ
4290 A81F 8DF3BF    STA CNTRL
4300 A822 60        BUSOTZ RTS
4310 A823 ADF0BF  BUSWAI LDA PORTA   WAIT FOR PREV XMIT
4320 A826 30FA      BMI BUSOTZ  TO FINISH
4330 A828 202AF3    JSR STPTST *TEMP
4340 A82B D0F6      BNE BUSWAI
4350 A82D 00        BRK
4360 A82E 00        OUTSTP BRK
4370 A82F          ;
4380 A82F          ; RECEIVE BYTE IN A.  DESTROYS ONLY A.
4390 A82F          ;
4400 A82F 20D4A7  RCV    JSR BUSIN
4410 A832          RCVQ
4420 A832 202AF3    JSR STPTST *TEMP
4430 A835 F009      BEQ RCVSTP *TEMP
4440 A837 ADF0BF    LDA PORTA   WAIT FOR INTR. LINE
4450 A83A 10F6      BPL RCVQ
4460 A83C ADF1BF    LDA PORTB   GET DATA; DO HANDSHAKE
4470 A83F 60        RTS
4480 A840 00        RCVSTP BRK
4490 A841          ;
4500 A841          ; SEND BYTE IN A.  DESTROYS ONLY A.
4510 A841          ;
4520 A841 48        SEND  PHA          SAVE BYTE
4530 A842 2002A8    JSR BUSOUT  TURN BUS OUT
4540 A845 68        PLA
4550 A846 8DF1BF    STA PORTB   SEND DATA; DO HANDSHAKE
4560 A849 60        RTS

```

```

4580 A84A ;
4590 A84A ; DIVIDE DIVDND (32 BITS) BY DIVISR (16 BITS)
4600 A84A ; REMNDR (16 BITS) AND QUOTNT (16 BITS)
4610 A84A ; REPLACE THE DIVDND.
4620 A84A ;
4630 A84A ; ALL QUANTITIES ARE STORED LSB FIRST AND ARE POSITIVE.
4640 A84A ; NO DIVIDE-CHECK TEST IS DONE. A AND X ARE DESTROYED.
4650 A84A ; THIS TAKES ABOUT 1.1 MSEC.
4660 A84A ;
4670 A84A ;

```

```

-----
4680 A84A A211 DIV LDX #17 LOOP COUNT
4690 A84C 38 DIVSBT SEC SUBTRACT CYCLE
4700 A84D AD69AF LDA DIVDND+2
4710 A850 ED6BAF SBC DIVISR+0
4720 A853 8D69AF STA DIVDND+2
4730 A856 AD6AAF LDA DIVDND+3
4740 A859 ED6CAF SBC DIVISR+1
4750 A85C 8D6AAF STA DIVDND+3
4760 A85F 08 DIVROT PHP SAVE CARRY
4770 A860 2E67AF ROL DIVDND+0 ROTATE CARRY INTO QUOTIENT
4780 A863 2E68AF ROL DIVDND+1
4790 A866 CA DEX
4800 A867 F01E BEQ DIVFXP ALL CYCLES DONE: DO FIXUP
4810 A869 2E69AF ROL DIVDND+2 OTHERWISE
4820 A86C 2E6AAF ROL DIVDND+3 SHIFT DIVDND LEFT
4830 A86F 28 PLP RETRIEVE CARRY
4840 A870 B0DA BCS DIVSBT MORE: DO NEXT CYCLE (SUBT)
4850 A872 AD69AF DIVADD LDA DIVDND+2 MORE: DO NEXT CYCLE (ADD)
4860 A875 6D6BAF ADC DIVISR+0
4870 A878 8D69AF STA DIVDND+2
4880 A87B AD6AAF LDA DIVDND+3
4890 A87E 6D6CAF ADC DIVISR+1
4900 A881 8D6AAF STA DIVDND+3
4910 A884 4C5FA8 JMP DIVROT
4920 A887 ;
4930 A887 28 DIVFXP PLP RETRIEVE CARRY
4940 A888 B012 BCS DIVZ NO FIXUP NECESSARY...
4950 A88A AD69AF LDA DIVDND+2 FIXUP REMAINDER
4960 A88D 6D6BAF ADC DIVISR+0
4970 A890 8D69AF STA DIVDND+2
4980 A893 AD6AAF LDA DIVDND+3
4990 A896 6D6CAF ADC DIVISR+1
5000 A899 8D6AAF STA DIVDND+3
5010 A89C 60 DIVZ RTS
5020 A89D ;
5030 A89D ;

```

```

-----
5040 A89D ; MULTIPLY THE HIGH-ORDER 16 BITS OF DIVDND (REMNDR)
5050 A89D ; BY DIVISR (16 BITS).
5060 A89D ; THE RESULT REPLACES DIVDND (32 BITS).
5070 A89D ; ALL QUANTITIES ARE STORED LSB FIRST.
5080 A89D ;
5090 A89D ;

```

```

-----
5100 A89D A900 MULT LDA #0
5110 A89F 8D68AF STA DIVDND+1 A HAS LOW-ORDER DIVDND
5120 A8A2 A010 LDY #16
5130 A8A4 0A MPYSHF ASL A SHIFT SUM AND MLTPLICAND
5140 A8A5 2E68AF ROL DIVDND+1
5150 A8A8 2E69AF ROL DIVDND+2
5160 A8AB 2E6AAF ROL DIVDND+3
5170 A8AE 900F BCC MPYNXT NO SUM...
5180 A8B0 18 CLC
5190 A8B1 6D6BAF ADC DIVISR ADD MULTIPLIER
5200 A8B4 AA TAX
5210 A8B5 AD68AF LDA DIVDND+1
5220 A8B8 6D6CAF ADC DIVISR+1

```

```
5230 A8BB 8D08AF STA DIVDND+1
5240 A8BE 8A TXA
5250 A8BF 88 MPYNXT DEY
5260 A8C0 D0E2 BNE MPYSHF
5270 A8C2 8D67AF STA DIVDND STORE LOW-ORDER SUM
5280 A8C5 60 RTS
5290 A8C6 ;
```