```
                C16     :C64
ZØ1     =$Ø1    :MEMMAP 64
Z77     =$69    D - # char printed
Z78     =$6A    A - Pointer in Line
Z79     =$6B    A - Amode flag for relative
Z7A     =$6C    A - Stack pointer save
Z7B     =$6D    D
Z7C     =$6E    D   > opcode ROL's
Z7D     =$6F    Temp
Z99     =$99    :9A DFLTO
Z9A     =$9A    :9D MSGFLG
ZF3     =$26    Input Line - pointer
ZF4     =$27    Input Line - length
ZF5     =$28    Amode
ZF6     =$29    Length of instruction
ZAB     =$AB    :B7 FNLEN
ZAC     =$AC    :B8 LA    Log. File
ZAD     =$AD    :B9 SA    Sec. Adr.
ZAE     =$AE    :BA FA    Dev. No.
ZAF     =$AF    :BB FNADR/L
ZBØ     =$BØ    :BC FNADR/H
ZBB     =$2A    Temp
ZEF     =$EF    :C6 NDX   chrs in queue
Z9F     =$61    difference  A - Rel. temp.
ZAØ     =$62    "           A - Oped temp
ZF1     =$63    Input Hex
ZF2     =$64
ZA1     =$65    PC
ZA2     =$66
PTDNO   =$17    :$Ø4      Printer Dev. #
KEYBD   =$Ø527  :$Ø277    Keyboard Queue
SHIFT   =$Ø543  :$Ø28D    Shift flag
BASIC   =$8ØØ3  :$E37B    Warm start
        *=$1238 :$ØA38    (after Basic portion)
LØAFE   .B$EA             location choice poked
LØBØØ   .WLCF66,LFB38-1   renumber brackets
```

Install p.7

```
        LDX #$17
LØABC   LDA ENDMR,X
        STA SCNTP,X
        DEX
        BPL LØABC
        LDA LØAFE
        STA JMPEY+2
```

Renumber Monitor

```
        STA ZF2
        SEC
        SBC #>MNMSG
        STA ZF1
        LDX #3
LØADD   LDA LØBØØ,X
        STA ZF3,X
        DEX
        BPL LØADD
LØB31   LDY #Ø
        LDA (ZF3),Y
        LSR
        BCC LØB43
        LSR
        BCS LØB52
        CMP #$22
        BEQ LØB52
        AND #7
```

partial duplicate of LF7D4

```
        ORA #$8Ø
LØB43   LSR
        TAX
        LDA LF83D,X
        BCS LØB4E
        LSR
        LSR
        LSR
        LSR
LØB4E   AND #15
        BNE LØB54
LØB52   LDA #Ø
LØB54   TAX
        LDA LF881,X
        AND #3
        TAY
        CPY #2
        BNE LØB6D
        LDA (ZF3),Y
        CMP #>MNMSG
        BCC LØB6D
        CMP #>LØ558+$1ØØ
        BCS LØB6D
        ADC ZF1
        STA (ZF3),Y
LØB6D   TYA
        SEC
        ADC ZF3
        STA ZF3
        BCC LØB78
        INC ZF4
LØB78   CMP ZF5
        LDA ZF4
        SBC ZF6
        BCC LØB31
        LDY #Ø
Vector  LØB86 INY
Table
renumber
        CLC
        LDA LF58Ø,Y
        ADC ZF1
        STA LF58Ø,Y
        INY
        CPY #$1C
        BNE LØB86
Transfer
to
chosen
location
        LDX #8
        LDY #Ø
        STY ZF1
        STY ZF3
        LDA ZF2
        CMP #>MNMSG
        BCS LØBA4
        JMP L14ØØ  ;if transfer down
LØBA4   CLC        ;if transfer up
        ADC #7
        STA ZF2
        LDA #>LØ558
        STA ZF4
LØBAD   DEY
        LDA (ZF3),Y
        STA (ZF1),Y
```

```
            TYA
            BNE LØBAD
            DEC ZF4
            DEC ZF2
            DEX
            BNE LØBAD
Reset       LDA $2C
Basic       STA $2E
pointers    STA $3Ø
            STA $32
            LDA #3
            STA $2D
            STA $2F
            STA $31
            TYA
            STA ($2B),Y
            INY
            STA ($2B),Y
            JMP LØ7EE
MONITOR MNMSG .B$ØD;'MONITOR  (xxØØ)
        BKMSG .B$ØD;'BREAK
        RGMSG .B$ØD;'   PC  NVBDIZC
              .' AC XR YR SP';B$ØD
        RHMSG .';';B$AØ      :'/'
        ERMSG .B$ØD;'ERROR
Print   LCF66 LDA MNMSG,X
message       PHP
              AND #$7F
              JSR $FFD2
              INX
              PLP
              BPL LCF66
              RTS
for A   ALTIN LDA Z9F
              BCS ALRET
              LDA !ZAØ,Y
        ALRET RTS
Char.   LCF74 BIT $Ø7F8
fetch         BMI LCF9E
              LDA (ZA1),Y
              RTS
        LCF9E LDA #ZA1
              STA $Ø7DF
              JMP $Ø7D9
SYS or
BRK ->  ENTRY CLD
              LDX #Ø
              BCC LF464   SYS
BRK           LDX #3
        LF45B PLA
              STA LØ554,X
              DEX
              BPL LF45B
              PLA
              SBC #2
              STA LØ553
              PLA
              SBC #Ø
              STA LØ552
```

C64

```
:STX ZBB
:LDA ZØ1
:TAX
:AND #$F8
:ORA LØ7FF
:SEI
:JSR LØ7F6
:CLI
:RTS
```

```
                LDX #<BKMSG
Both  LF464   JSR LCF66
              LDA #$CØ
              STA Z9A
              BCS LF45F
SYS           PHA       Adjust stack if SYS
              PHA
Both  LF45F   TSX
              STX LØ558
              CLI
              LDY #3
              BCC PRTSP
[R]   LF478   LDX #<RGMSG
      LF47A   JSR LCF66
              LDA LØ553
              LDX LØ552
              JSR LFAFF
              LDA LØ554
              LDY #8
Status ZRONE  ASL
Bits          PHA
              LDA #$18
              ROL
              CPY #6
              BEQ NUSED
              JSR $FFD2
      NUSED   PLA
              DEY
              BNE ZRONE
Both  PRTSP   JSR LFBØ8
Print LF485   LDA LØ555,Y
Regs          JSR LFBØ5
              INY
              CPY #4
              BCC LF485
              BCS LF495
[J]   LF5ØA   BCS DOBTS
              STA LØ553
Change        STX LØ552
Regs  DOBTS   LDY #8
      GTBIT   CPY #6
              BEQ SKIP1
              JSR NCHNS
              LSR
      SKIP1   ROL LØ554
              DEY
              BNE GTBIT
              JSR LFB3F
      LF518   JSR LFAAD
              BCS REPRT
              STA LØ555,Y
              INY
              CPY #4
              BCC LF518
      REPRT   JSR LFB35
              LDX #<RHMSG
              BNE LF47A
Bad→  LF492   JSR LFBØB
Next→ LF495   JSR LFB3A
              LDX #Ø
```

```
Input Line
        STX ZF3
LF49C   JSR $FFCF
        CPX #$59
        BCS LONGL
        STA $2ØØ,X
        STX ZF4
        INX
LONGL   CMP #13
        BNE LF49C
        JSR NCHNS
        BEQ LF495
        LDX #$13
matched letter
LF4B5   CMP LF57Ø,X
        BEQ LF4BF
        DEX
        BPL LF4B5
        BMI LF492
LF4BF   CPX #14
        BCS LF4D1  (LSVP$*)
        TXA
        ASL
        TAX
        LDA LF58Ø+1,X  Vectors
        PHA
        LDA LF58Ø,X
        PHA
        JMP LFAAD
LF4D1   STA ZF6
        JMP LF66E
[M]     BCS LF4E1
LF4D7   JSR F1NXT
        BCC LF4E7
LF4E1   LDA #$5F
        STA ZF1
        BNE LF4F5
LF4E7   JSR LFBAA
LF4F5   JSR $FFE1
        BEQ LF5Ø7
        JSR LFB3A
LF4FA   JSR LF59D
        BCS LF4F5
LF5Ø7   JMP LF495
[=>]    BCS POKE3
LF529   JSR LFB5B
        JSR GETNW
        DEY
        BMI POKE3
        STY ZF1
POKE5   LDA $2ØØ,Y
        STA (ZA1),Y
        DEY
        BPL POKE5
POKE3   JSR LFB35
        JMP LF4FA
[X]     LDX #>BASIC
LCF84   LDA #<BASIC
        BNE GODDD
[G]     BCS GOCCC
LF54B   STX LØ553
GODDD   STX LØ552
```

```
GOCCC LDX L0558
      INX
      BEQ GOAAA
      INX
      BNE GOBBB
GOAAA DEX
GOBBB TXS
      LDA #>L07EE-1
      PHA
      LDA #<L07EE-1
      PHA
      JSR LFB3A
      LDX #0
LF55D LDA L0552,X
      PHA
      INX
      CPX #6
      BNE LF55D
      PLA
      TAY
      PLA
      TAX
      PLA       :LDA #$37
      RTI       :JMP L07E8
      NOP       :
      NOP       :
```

Display one line

```
LF59D LDA #'>   :=
      JSR $FFD2
      JSR LFAFB
      LDY #0
LF5A7 JSR LCF74
      JSR LFB05
      INY
      CPY #8
      BCC LF5A7
      LDY #$FF
      LDA #':
      LDX Z99
      CPX #3
      BNE LF5C5
      JSR $FFD2
      LDA #$12
      BNE LF5C5
LF5B9 JSR LCF74
      PHA
      ASL
      CMP #$40
      PLA
      BCS LF5C5
      LDA #'.
LF5C5 JSR $FFD2
      INY
      CPY #8
      BCC LF5B9
      TYA
      JSR LFB96
      TYA
LFB74 SEC
LFB76 STA ZBB
      LDA ZF1
```

```
        SBC ZBB
        STA ZF1
        LDA ZF2
        SBC #Ø
        STA ZF2
        RTS
[C] LF5CE LDA #$8Ø
        .B$2C  :BIT
[T] LF5D1 LDA #$CØ
        STA ZF6
        JSR LFBAØ
        BCS LF6ØA
        JSR LFAAD
        BCS LF6ØA
        BIT ZF6
        BVC LF5E2
        LDA ZA1
        CMP ZF1
        LDA ZA2
        SBC ZF2
        ROR ZF6
        BMI LF5E2
        LDA Z9F
        ADC ZA1
        STA ZA1
        LDA ZAØ
        ADC ZA2
        STA ZA2
        CLC
        LDA Z9F
        ADC ZF1
        STA ZF1
        LDA ZAØ
        ADC ZF2
        STA ZF2
LF5E2   LDY #Ø
        STY ZF3
LF5E6   JSR LCF74
        BIT ZF6
        BVC LF5ED
        STA (ZF1),Y
        BVS BOBBY
LF5ED   CMP (ZF1),Y
        BEQ LF5F9
        JSR $FFE1
        BEQ LF6Ø7
        JSR LNBRK
BOBBY   BIT ZF6
        BMI LF5F9
        LDA #1
        JSR LFB74
        LDA ZA1
        BNE DECA1
        DEC ZA2
DECA1   DEC ZA1
        JMP LF5FF
LF5F9   JSR LFB94
        INC ZF1
        BNE LF5FF
        INC ZF2
```

ZF6 B7 B6
Tdn 1 0 C
TuporC 0 1 T

AAABB
LDX #2
CLC
LDA ZF1,X
ADC ZF1,X
STA ZF1,X
LDA ZA0
ADC ZF2,X
STA ZF2,X
DEX
DEX
BEQ AAABB

Δ6

```
     LF5FF JSR LFB86
           BCS LF5E6
     LF6Ø7 JMP LF495
     LF6ØA JMP LF492
```

Print address

```
     LNBRK LDA ZF3
           BNE DECF3
           JSR LFB3A
           LDA #8
           STA ZF3
     DECF3 DEC ZF3
           JMP LFAFB
```

[H]

```
     LF6ØE JSR LFBAØ
           BCS LF6ØA
           JSR GETNW
           STA ZF3
           TYA
           BEQ LF6ØA
           STY ZF6
     LF649 LDY #Ø
     LF64D JSR LCF74
           CMP $2ØØ,Y
           BNE LF664
           INY
           CPY ZF6
           BNE LF64D
           JSR $FFE1
           BEQ LF6Ø7
           JSR LNBRK
     LF664 JSR LFB94
           JSR LFB86
           BCS LF649
           BCC LF6Ø7
```

Items for =+H

```
     GETNW LDY #Ø
           JSR NCHNS
           BEQ GNOK2
           CMP #''
           BNE ISHEX
     GNOK1 JSR LFB3F
           BEQ GNOK2
           STA $2ØØ,Y
           INY
           BPL GNOK1
     ISHEX DEC ZF3
     GNOK2 JSR LFAAD
           BCS GRET2
           STA $2ØØ,Y
           INY
           LDA ZF5
           BEQ GNOK2
           TXA
           STA $2ØØ,Y
           INY
     GNOK3 BPL GNOK2
     GRET2 RTS
```

[LSVP**]

```
     LF66E LDY #2
           LDX ZF3
           DEX
           LDA #Ø
           STA $9Ø
           JSR $FFBD
```

```
        DEY
        STY ZAE
        STY ZAD
        JSR NCHNS
        BEQ LF6E6
        CMP #'"
        BNE LF6A7
        LDX ZF3
        STX ZAF
LF692   INX
        CMP $1FF,X
        BEQ LF6AB
        INC ZAB
        CPX ZF4
        BCC LF692
LF6AB   STX ZF3
        JSR LFB3F
        JSR LFAAD
        BCS LF6E6
        STA ZAE
        JSR LFAAD
        BCS LF6E6
        DEC ZAD
        JSR F1NXT
        BCS LF6E6
        TAX
        LDY ZF2
        LDA ZF6
        CMP #'S
        BEQ NSAVE
LF6A7   JMP LF492
NSAVE   LDA #ZA1
        JSR $FFD8
        JMP NINTY
LF6E6   LDX ZF6
        TAY
        CPX #'P
        BEQ PRINT
        CPX #'L
        BEQ LF6F3
        CPX #'V
        BNE TSRST
        LDA #1
LF6F3   LDX ZA1
        LDY ZA2
        JSR $FFD5
NINTY   BCS TSTER
        LDA $9Ø
        AND #$BF
TSTER   CMP #1Ø
        BCC CLOSE
        LDX #<ERMSG
        JSR LCF66
CLOSE   JSR $FFE7
LF6E3   JMP LF495
LF7ØA   JSR LFBAØ
        BCS LF6A7
        JSR LFAAD
        BCS LF6A7
        LDY #Ø
```

[N]

[F]

moved to end of Asy

```
LF716 LDA ZF1
      STA (ZA1),Y
      JSR LFB94
      JSR LFB86
      BCS LF716
      BCC LF6E3
PRINT DEY
      LDX #PTDNO
      BNE DCMD1
TSRST CPX #'*
      BEQ DKCMD
      CPX #'$
      BNE LF6A7
      INC ZAB
      LDX #$FB
      BNE RCHAN
PRTCA JSR $FFD2
      BNE DIRLP
      LDY #$FD
ENTER JSR LFB3A
PAUSE LDA SHIFT
      LSR
      BCS PAUSE
DIRLP JSR $FFCF
      LDX $9Ø
      BNE ENDRD
BBBAA INY
      BMI DIRLP
      BNE PRTCA
      PHA
      JSR $FFCF
      TAX
      PLA
      JSR LFAFF
      BNE DIRLP
ENDRD JSR $FFCC
      LDA ZAC
      JSR $FFC3
      BCC LF6E3
DKCMD LDY #15
      LDX ZAB
      BEQ RCHAN
      STX ZF6
      LDX #8
DCMD1 STA ZAB
      JSR OPENP
      BCS ENDRD
      JSR $FFC9
      BCS ENDRD
      LDA Z99
      CMP #8
      BNE LF6E3
      LDY #Ø
SENDO LDA (ZAF),Y
      JSR $FFD2
      INY
      CPY ZF6
      BNE SENDO
      JSR LFB3A
      BNE ENDRD
```

Moved to End of Msg

601 OK

:DELETE/64

BIT Z7A
BMI BBBAA
CMP #$ØD
BEQ ENDRD

```
RCHAN  STX Z7A
       JSR OPENF
       BCS ENDRD
       JSR $FFC6
       BCS ENDRD
       LDY Z7A
       ~~BNE~~ ENTER   BCC
OPENF  LDX #8
OPENP  TXA
       JSR $FFBA
       JSR $FFCØ
       LDX ZAC
       RTS
```

Disassemble + Print one line

```
LF752  LDA #'.
LF754  JSR PTADR      prints "." , space, address, space
       LDY #Ø
       JSR LCF74
```

Get Objek.

```
       JSR LF7D4
       PHA
```

Print byte values

```
       LDX ZF6
LF76B  BPL LF779
       LDX #2
GOSPC  JSR LFBØ8      space
       DEX
       BPL GOSPC
       BMI LF77F
LF779  JSR LCF74
       JSR LFBØ5      byte + space
LF77F  INY
       CPY #3
       DEX
       BCC LF76B
       PLA
       JSR INSTR
       LDA ZF6
       JMP LFB98      add to A1 #A2
```

Print Instruction

```
INSTR  TSX
       STX Z7A
       LDX #3
       TAY
       LDA LF89B,Y
       STA Z7B
       LDA LF8DB,Y
       STA Z7C
LF826  LDA #Ø
       LDY #5
LF82A  ASL Z7C
       ROL Z7B
       ROL
       DEY
       BNE LF82A
       ADC #$3F
       JSR NOUTX
       DEX
       BNE LF826
       JSR LFBØ8
       LDX #6
       STX Z77
LF78D  CPX #3
       BNE LF7A5
```

```
        LDY ZF6
        BEQ LF7A5
LF795   LDA ZF5
        CMP #$E8
ALTGT   JSR LCF74       "JSR ALTIN" during Assembly
        BCS LF7BC
        JSR LFB1Ø
        DEY
        BNE LF795
LF7A5   ASL ZF5
        BCC LF7B8
        LDA LF88F-1,X
        JSR NOUTX
        LDA LF895-1,X
        BEQ LF7B8
        JSR NOUTX
LF7B8   DEX
        BNE LF78D
FILSP   LDX Z77
        BMI UPDA1
FILLS   JSR LFBØ8
        DEX
        BPL FILLS
UPDA1   SEC
        RTS
if Relative LF7BC LDX ZA2
        TAY
        BPL LF7CE
        DEX
LF7CE   ADC ZA1
        BCC ADONE
        INX
        CLC
ADONE   ADC #1
        BNE LF7C5
        INX
LF7C5   JSR LFAFF
        JMP FILSP
```

```
Get OP  LF7D4 TAY          produces: F5 = Amode
        LSR                          F6 = Length
        BCC LF7E3                    A = Mnemonic index
        LSR
        BCS LF7F2
        CMP #$22
        BEQ LF7F2
        AND #7
        ORA #$8Ø
LF7E3   LSR
        TAX
        LDA LF83D,X
        BCS LF7EE
        LSR
        LSR
        LSR
        LSR
LF7EE   AND #15
        BNE LF7F6
LF7F2   LDY #$8Ø
        LDA #Ø
LF7F6   TAX
```

```
        LDA LF881,X
        STA ZF5        Amode
        AND #3
        STA ZF6        Length
        TYA
        AND #$8F
        TAX
        TYA
        LDY #3
        CPX #$8A
        BEQ LF817
LF8ØC   LSR
        BCC LF817
        LSR
LF81Ø   LSR
        ORA #$2Ø
        DEY
        BNE LF81Ø
        INY
LF817   DEY
        BNE LF8ØC
        STY Z78        (=Ø)
        RTS
```

[D]

```
LF724   BCS LF72E
        JSR F1NXT
        BCC LF734
LF72E   LDA #$14
        STA ZF1
        BNE LF737
LF734   JSR LFBAA
LF737   JSR LFB3A
        JSR $FFE1
        BEQ ENDOK
        JSR LF752
        LDA ZF6
        CLC
        JSR LFB76
        BCS LF737
        BCC ENDOK
```

[A]

```
LF91F   BCC LF924
LF921   JMP LF492
LF924   JSR LFB5B
        JSR LFBAA
LF927   LDX #Ø
LF92B   JSR LFB3F
        BNE LF937
        TXA
        BNE LF921
ENDOK   JMP LF495
LF937   CMP #$2Ø
        BEQ LF927
        CMP #'?
        BEQ LF921
        STA $2ØØ,X
        INX
        CPX #3
        BNE LF92B
        LDY ZF3
NTEND   JSR GVAL1
        BNE NTEND
```

Try Opcodes one at a time (0-FF), See if Disassembly matches input.

Initialize Z9F, ZA9, ZF1, ZF2 = 9

"

Look for 3 chars in row ≠ space

Get Hex value, if any, in input instruction → ZF1, ZF2

```
        STY ZF3          Transfer instruction as typed to $200 →
        LDX #3
LF958   LDA #$2Ø
LF95A   STA $2ØØ,X
        INX
        JSR NCHNS
        BNE LF95A
        CPX #12          tack spaces on and
        BCC LF958
        LDA #$2C         change input and output routines
        LDX #<ALTIN
        JSR REST1
NXTOP   LDA ZAØ          try opcode
        JSR LF7D4        get Amode, Length, mnem.
        LDX ZF5
        CPX #$9D
        ROR Z79          Z79 negative if relative
        JSR INSTR        "Print" instruction
        BCS OKFND        CS = all characters matched
        BIT Z79          If mismatch, check for where
        BPL NTREL        "
        CPY #4
        BCC NTREL        But if relative:
        INC Z9F            If opcode matched, but rest
        BNE NXTOP            did not, go to next rel.temp
NTREL   INC ZAØ          Next opcode
        BNE NXTOP
        JSR RESTR        tried them all, restore in/out,
        BNE LF921          error
OKFND   LDY ZF6          If match, store values
        LDA Z9F  -rel. temp
        BIT Z79
        BMI STBY2
STBYT   LDA !ZAØ,Y       Aø, F1, F2 are contiguous
STBY2   STA (ZA1),Y
        DEY
        BPL STBYT
        JSR RESTR
        JSR LFB35        return, cursor up
        LDA #'A          Disassemble what was just
        JSR LF754          assembled + the PC
        LDA #$2C         change output to KB buffer
        STA NOUTX
        LDY #Ø           put next address in buffer
        LDA #'A
        JSR PTADR
        LDA #$4C
        STA NOUTX
        JMP LF495  GOTOK

Subs in/out
RESTR   LDX #<LCF74
        LDA #$4C
REST1   STA NOUTA
        STX ALTGT+1
        RTS
PTADR   JSR NOUTX        Print Acc contents
        JSR LFBØ8          space
        JMP LFAFB          PC

F1NXT   JSR LFB5B

Convert Hex digits to bytes and on ',' or space.
LFAAD   JSR GETVL
        CMP #',
```

[F] LF7ØA →

```
                BEQ LFAF1
                AND #$DF
                BEQ LFAF1
                PLA
                PLA
        LFA72   JMP LF492
        LFAF1   CLC
                DEC ZF5
                BPL LFAF5
                SEC
        LFAF5   LDX ZF2
                LDA ZF1
                RTS
```

Convert hex digits to bytes. # bytes in F5 (Ø-2)

```
        GETVL   LDX #Ø
                STX ZF1
                STX ZF2
        GVAL1   LDX #1
                STX ZF5
                JSR NCHNS
        LFABF   CMP #'Ø
                BCC LFAF6
                CMP #'G
                BCS LFAF6
                CMP #':
                BCC LFAD9
                CMP #'A
                BCC LFAF6
                SBC #8
        LFAD9   SBC #$2F
                ASL
                ASL
                ASL
                ASL
                LDX #4
        LFAE1   ASL
                ROL ZF1
                ROL ZF2
                DEX
                BNE LFAE1
                INC ZF5
                JSR LFB3F
                BNE LFABF
        LFAF6   PHA
                LDA ZF5
                LSR
                CMP #3
                BCC TWOOK
                LDA #2
        TWOOK   STA ZF5
                PLA
                RTS
```

(P) PC + space

```
        LFAFB   LDA ZA1
                LDX ZA2
```

(P) Word + space

```
        LFAFF   PHA
                TXA
                JSR LFB1Ø
                PLA
```

(P) Byte + space

(P) space

```
        LFBØ5   JSR LFB1Ø
        LFBØ8   LDA #$2Ø
                .B$2C   :BIT
```

convert Hex bytes to ASCII digits

```
LFBØB LDA #'?          "BIT NOUTB" if to Keyboard Buf.
NOUTX JMP NOUTB
      STA KEYBD,Y
      INY
      STY ZEF
      RTS
NOUTB DEC Z77
NOUTA JMP $FFD2        "BIT $FFD2" if match test
      STY Z7D
      LDY Z78          does "D" output match typed
      INC Z78            instruction.
      CMP $2ØØ,Y
      BEQ MATCH
      CLC              on 1st mismatch, abort "D", go back
      LDX Z7A            to Assembly with CC,
      TXS                Y reg has pointer value
      RTS
MATCH LDY Z7D
      RTS
```

```
LFB1Ø STX ZBB
      JSR LFB2Ø
      JSR NOUTX
      TXA
      LDX ZBB
      JMP NOUTX
LFB2Ø PHA
      JSR LFB2A
      TAX
      PLA
      LSR
      LSR
      LSR
      LSR
LFB2A AND #15
      CMP #1Ø
      BCC LFB32
      ADC #6
LFB32 ADC #$3Ø
      RTS
```

```
NCHNS JSR LFB3F
      BEQ RETRN
      CMP #$2Ø
      BEQ NCHNS
RETRN RTS
```

```
LFB3F STX ZBB
      LDX ZF3
      LDA $2ØØ,X
      CPX ZF4
      LDX ZBB
      BCS LFB51
      CMP #':
      BNE LFB57
LFB51 LDA #Ø
      RTS
LFB57 INC ZF3
      RTS
```

```
LFB5B LDA ZF1
      STA ZA1
      LDA ZF2
      STA ZA2
```

```
                RTS
Dec. 9F,A0  LFB86 SEC
                  LDA Z9F
                  SBC #1
                  STA Z9F
                  LDA ZA0
                  SBC #0
                  STA ZA0
                  RTS
Inc PC      LFB94 LDA #1
Add A to PC LFB96 CLC
            LFB98 ADC ZA1        ;if CS, Add A+1 to PC
                  STA ZA1
                  BCC LFB9F
                  INC ZA2
            LFB9F RTS
Move F1,2 to LFBA0 BCS LFBB6
A1,2                JSR F1NXT
Get next addr,      BCS LFBB6
Diff to F1,2 LFBAA SEC
and 9F,A0           LDA ZF1
                    SBC ZA1
                    STA ZF1
                    STA Z9F
                    LDA ZF2
                    SBC ZA2
                    STA ZF2
                    STA ZA0
                    CLC
            LFBB6 RTS
(P) Ret,↑   LFB35 JSR LFB3A
(P) ↑       LFB38 LDA #$91
                  .B$2C  :BIT
(P) Ret     LFB3A LDA #13
                  JMP $FFD2
[Commands]  LF57Ø .'XMRGTCDA.HF>;NLSVP$*' [:=]/NLS..
[Vectors]   LF58Ø .WLCF84-1,LF4D7-1,LF478-1,LF54B-1
                  .WLF5D1-1,LF5CE-1,LF724-1,LF91F-1
                  .WLF91F-1,LF6ØE-1,LF7ØA-1,LF529-1
                  .WLF5ØA-1,CLOSE-1
Opcode indexes LF83D .B$4Ø,$Ø2,$45,$Ø3,$DØ,$Ø8,$4Ø,$Ø9
for D                .B$3Ø,$22,$45,$33,$DØ,$Ø8,$4Ø,$Ø9
(computed-nibbles    .B$4Ø,$Ø2,$45,$33,$DØ,$Ø8,$4Ø,$Ø9
are index for        .B$4Ø,$Ø2,$45,$B3,$DØ,$Ø8,$4Ø,$Ø9
LF881)               .B$ØØ,$22,$44,$33,$DØ,$8C,$44,$ØØ
                     .B$11,$22,$44,$33,$DØ,$8C,$44,$9A
                     .B$1Ø,$22,$44,$33,$DØ,$Ø8,$4Ø,$Ø9
                ???  .B$1Ø,$22,$44,$33,$DØ,$Ø8,$4Ø,$Ø9
                     .B$62,$13,$78,$A9   Imp  Acc
Addr. Modes LF881    .B$ØØ,$21,$81,$82,$ØØ,$ØØ,$59,$4D
                     .B$91,$92,$86,$4A,$85,$9D
Characters  LF88F .B$2C,$29,$2C,$23,$28,$24   , ) , # ( $
            LF895 .B$59,$ØØ,$58,$24,$24,$ØØ   X _ Y # $ —
Opcode      LF89B .B$1C,$8A,$1C,$23,$5D,$8B,$1B,$A1
mnemonic             .B$9D,$8A,$1D,$23,$9D,$8B,$1D,$A1
(computed - 5 bits each)  .B$ØØ,$29,$19,$AE,$69,$A8,$19,$23
(one from LF89B and       .B$24,$53,$1B,$23,$24,$53,$19,$A1
one from LF8DB =          .B$ØØ,$1A,$5B,$5B,$A5,$69,$24,$24
3 ASCII chars.)           .B$AE,$AE,$A8,$AD,$29,$ØØ,$7C,$ØØ
                          .B$15,$9C,$6D,$9C,$A5,$69,$29,$53
```

```
                .B$84,$13,$34,$11,$A5,$69,$23,$AØ
          LF8DB .B$D8,$62,$5A,$48,$26,$62,$94,$88
                .B$54,$44,$C8,$54,$68,$44,$E8,$94
                .B$ØØ,$B4,$Ø8,$84,$74,$B4,$28,$6E
                .B$74,$F4,$CC,$4A,$72,$F2,$A4,$8A
                .B$ØØ,$AA,$A2,$A2,$74,$74,$74,$72
                .B$44,$68,$B2,$32,$B2,$ØØ,$22,$ØØ
                .B$1A,$1A,$26,$26,$72,$72,$88,$C8
Phantoms!       .B$C4,$CA,$26,$48,$44,$44,$A2,$C8
PCH       LØ552 .BØ
PCL       LØ553 .BØ
status    LØ554 .BØ
A,X,Y     LØ555 .BØ,Ø,Ø
SP        LØ558 .BØ                    END of Relocated MONITOR
Transfer down L14ØØ LDA #>MNMSG
                STA ZF4
          L14Ø2 LDA (ZF3),Y
                STA (ZF1),Y
                INY
                BNE L14Ø2
                INC ZF4
                INC ZF2
                DEX
                BNE L14Ø2
                TYA
                STA (ZF1),Y
                STA ($2B),Y
                INY
                STA ($2B),Y
                LDA $2C
                STA $2E
                STA $3Ø
                STA $32
                LDA #3
                STA $2D
                STA $2F
                STA $31
                JMP LØ7EE
Low mem   ENDMR *=$ØBE8          :$Ø7E8
routines  SCNTP =*
                                 :LØ7E8 STA ZØ1
                                 :PLA
                                 :RTI
BRk →     LO7EC SEC
                .B$24 :BIT
SYS →     LØ7EE CLC
                                 :LDA #$36
                                 :STA ZØ1
          JMPEY JMP ENTRY
                .'**********'    :LØ7F6 STA ZØ1
                .'**'            :LDA (ZA1),Y
                .'**'            :STX ZØ1
                .'**'            :LDX ZBB
                .'*'             :RTS
                .'*'             :LØ7FF .B7
```

to