Source File C:\VB\User\PropLoad\PropTest.spin
==========

```
VAR

  byte b
  word w
  long l

  byte bOverlay[0]
  word wOverlay[0]
  long lOverlay[0]

  byte bOverlayEnd
  word wOverlayEnd
  long lOverlayEnd

DAT

  bData byte $12
  wData word $1234
  lData long $12345678

OBJ

  Obj1_A : "Obj1Test"
  Obj1_B : "Obj1Test"

  Obj2_A : "Obj2Test"
  Obj2_B : "Obj2Test"

PUB Main

  test_Arguments_1( 1 )
  test_Arguments_2( 1, 2 )
  test_BinaryOps
  test_BinaryOps_Assigned
  test_Case
  test_Constants
  test_Cog_Assembler
  test_Cog_Spin
  test_Cog_Spin_Args( 0, 1, 2 )
  test_If
  test_If_Else
  test_If_ElseIf
  test_IfNot_ElseIfNot
  b := test_Function_1
  b := test_Function_2
  b := test_Function_3
  if b := \test_function_Abort
    b := 0
  if b := \test_function_AbortValue
    b := 0
  b := test_Function_Nested( 1,2 )
  b := test_Function_Nested( test_Function_Nested(1,2) , test_Function_Nested(3,4) )
  b := test_Function_Nested( test_Function_Nested(1,2) , test_Function_Nested(3, test_Function_Nested(5,6) ) )
  test_Locks
  test_Lookup_And_LookDown
  test_Memory_Ops
  test_Objects
  test_One_Line_Commands
  test_ReBoot
  test_Registers
  test_RegisterBits
  test_Repeat
  test_Repeat_Count
  test_Repeat_FromTo
  test_Repeat_Until
  test_Repeat_While
  test_Repeat_With_Next
  test_Repeat_With_Quit
  test_Strings
  test_UnaryOps
  test_UnaryOps_Assigned
  test_Vars
  test_Vars_Casting
  test_Vars_Data
  test_Vars_Local
  test_Wait

PRI test_Arguments_1(a1)

  a1 := 1
  b  := a1

PRI test_Arguments_2(a1,a2)

  a1 := 1
  b  := a1
```

```
  a2 := a1

PRI test_BinaryOps

  b := b + b
  b := b - b
  b := b * b
  b := b / b
  b := b // b
  b := b ** b

  b := b & b
  b := b | b
  b := b ^ b

  b := b AND b
  b := b OR  b

  b := b #> b
  b := b <# b
  b := b << b
  b := b >> b
  b := b <- b
  b := b -> b
  b := b ~> b
  b := b >< b

  b := b == b
  b := b <> b
  b := b <  b
  b := b >  b
  b := b =< b
  b := b => b

PRI test_BinaryOps_Assigned

  b += b
  b -= b
  b *= b
  b /= b
  b //= b
  b **= b

  b &= b
  b |= b
  b ^= b

  b AND= b
  b OR=  b

  b #>= b
  b <#= b
  b <<= b
  b >>= b
  b <-= b
  b ->= b
  b ~>= b
  b ><= b

  b === b
  b <>= b
  b <=  b
  b >=  b
  b =<= b
  b =>= b

PRI test_Case

  Case b
    1 : b := 1
    2 : b := 2

  Case b
    1..2 : b := 1
    2..3 : b := 2

PRI test_Cog_Assembler

  CogInit(0, @Conds, 1)
  CogNew( @Conds, 2 )

  b := CogNew( @Conds, 5 )

  CogInit(0, @Opcodes, 1)
  CogNew( @Opcodes, 2 )

  b := CogNew( @Opcodes, 5 )

PRI test_Cog_Spin
```

```
  b := CogId

  CogInit(0, test_Cog_Spin, 1)
  CogNew( test_Cog_Spin, 2 )

  b := CogNew( test_Cog_Spin, 5 )

  CogInit(0, test_Cog_Spin_Args(0,1,2), 1)
  CogNew( test_Cog_Spin_Args(0,1,2), 2 )

  b := CogNew( test_Cog_Spin_Args(0,1,2), 5 )

  CogStop( 0 )
  CogStop( b )

PRI test_Cog_Spin_Args( arg1, arg2, arg3 )

  arg1 := arg2 + arg3

PRI test_Constants

  b := TRUE
  b := FALSE
  b := NEGX
  b := POSX

  b := $0
  b := $1
  b := $2

  b := $0E
  b := $0F
  b := $10
  b := $11
  b := $12

  b := $1E
  b := $1F
  b := $20
  b := $21
  b := $22

  b := $2E
  b := $2F
  b := $30
  b := $31
  b := $32

  b := $3E
  b := $3F
  b := $40
  b := $41
  b := $42

  b := $4E
  b := $4F
  b := $50
  b := $51
  b := $52

  b := $5E
  b := $5F
  b := $60
  b := $61
  b := $62

  b := $6E
  b := $6F
  b := $70
  b := $71
  b := $72

  b := $7E
  b := $7F
  b := $80
  b := $81
  b := $82

  b := $8E
  b := $8F
  b := $90
  b := $91
  b := $92

  b := $9E
  b := $9F
  b := $A0
  b := $A1
```

```
b := $A2

b := $AE
b := $AF
b := $B0
b := $B1
b := $B2

b := $BE
b := $BF
b := $C0
b := $C1
b := $C2

b := $CE
b := $CF
b := $D0
b := $D1
b := $D2

b := $DE
b := $DF
b := $E0
b := $E1
b := $E2

b := $EE
b := $EF
b := $F0
b := $F1
b := $F2

b := $0FE
b := $0FF
b := $100
b := $101
b := $102

b := $1FE
b := $1FF
b := $200
b := $201
b := $202

b := $2FE
b := $2FF
b := $300
b := $301
b := $302

b := $3FE
b := $3FF
b := $400
b := $401
b := $402

b := $4FE
b := $4FF
b := $500
b := $501
b := $502

b := $5FE
b := $5FF
b := $600
b := $601
b := $602

b := $6FE
b := $6FF
b := $700
b := $701
b := $702

b := $7FE
b := $7FF
b := $800
b := $801
b := $802

b := $8FE
b := $8FF
b := $900
b := $901
b := $902

b := $9FE
b := $9FF
b := $A00
```

```
b := $A01
b := $A02

b := $AFE
b := $AFF
b := $B00
b := $B01
b := $B02

b := $BFE
b := $BFF
b := $C00
b := $C01
b := $C02

b := $CFE
b := $CFF
b := $D00
b := $D01
b := $D02

b := $DFE
b := $DFF
b := $E00
b := $E01
b := $E02

b := $EFE
b := $EFF
b := $F00
b := $F01
b := $F02

b := $0FFE
b := $0FFF
b := $1000
b := $1001
b := $1002

b := $1FFE
b := $1FFF
b := $2000
b := $2001
b := $2002

b := $2FFE
b := $2FFF
b := $3000
b := $3001
b := $3002

b := $3FFE
b := $3FFF
b := $4000
b := $4001
b := $4002

b := $4FFE
b := $4FFF
b := $5000
b := $5001
b := $5002

b := $5FFE
b := $5FFF
b := $6000
b := $6001
b := $6002

b := $6FFE
b := $6FFF
b := $7000
b := $7001
b := $7002

b := $7FFE
b := $7FFF
b := $8000
b := $8001
b := $8002

b := $8FFE
b := $8FFF
b := $9000
b := $9001
b := $9002

b := $9FFE
b := $9FFF
```

```
b := $A000
b := $A001
b := $A002

b := $AFFE
b := $AFFF
b := $B000
b := $B001
b := $B002

b := $BFFE
b := $BFFF
b := $C000
b := $C001
b := $C002

b := $CFFE
b := $CFFF
b := $D000
b := $D001
b := $D002

b := $DFFE
b := $DFFF
b := $E000
b := $E001
b := $E002

b := $EFFE
b := $EFFF
b := $F000
b := $F001
b := $F002

b := $0FFFE
b := $0FFFF
b := $10000
b := $10001
b := $10002

b := $1FFFE
b := $1FFFF
b := $20000
b := $20001
b := $20002

b := $2FFFE
b := $2FFFF
b := $30000
b := $30001
b := $30002

b := $3FFFE
b := $3FFFF
b := $40000
b := $40001
b := $40002

b := $4FFFE
b := $4FFFF
b := $50000
b := $50001
b := $50002

b := $5FFFE
b := $5FFFF
b := $60000
b := $60001
b := $60002

b := $6FFFE
b := $6FFFF
b := $70000
b := $70001
b := $70002

b := $7FFFE
b := $7FFFF
b := $80000
b := $80001
b := $80002

b := $8FFFE
b := $8FFFF
b := $90000
b := $90001
b := $90002

b := $9FFFE
```

```
b := $9FFFF
b := $A0000
b := $A0001
b := $A0002

b := $AFFFE
b := $AFFFF
b := $B0000
b := $B0001
b := $B0002

b := $BFFFE
b := $BFFFF
b := $C0000
b := $C0001
b := $C0002

b := $CFFFE
b := $CFFFF
b := $D0000
b := $D0001
b := $D0002

b := $DFFFE
b := $DFFFF
b := $E0000
b := $E0001
b := $E0002

b := $EFFFE
b := $EFFFF
b := $F0000
b := $F0001
b := $F0002

b := $0FFFFE
b := $0FFFFF
b := $100000
b := $100001
b := $100002

b := $1FFFFE
b := $1FFFFF
b := $200000
b := $200001
b := $200002

b := $2FFFFE
b := $2FFFFF
b := $300000
b := $300001
b := $300002

b := $3FFFFE
b := $3FFFFF
b := $400000
b := $400001
b := $400002

b := $4FFFFE
b := $4FFFFF
b := $500000
b := $500001
b := $500002

b := $5FFFFE
b := $5FFFFF
b := $600000
b := $600001
b := $600002

b := $6FFFFE
b := $6FFFFF
b := $700000
b := $700001
b := $700002

b := $7FFFFE
b := $7FFFFF
b := $800000
b := $800001
b := $800002

b := $8FFFFE
b := $8FFFFF
b := $900000
b := $900001
b := $900002
```

```
b := $9FFFFE
b := $9FFFFF
b := $A00000
b := $A00001
b := $A00002

b := $AFFFFE
b := $AFFFFF
b := $B00000
b := $B00001
b := $B00002

b := $BFFFFE
b := $BFFFFF
b := $C00000
b := $C00001
b := $C00002

b := $CFFFFE
b := $CFFFFF
b := $D00000
b := $D00001
b := $D00002

b := $DFFFFE
b := $DFFFFF
b := $E00000
b := $E00001
b := $E00002

b := $EFFFFF
b := $EFFFFF
b := $F00000
b := $F00001
b := $F00002

b := $0FFFFFE
b := $0FFFFFF
b := $1000000
b := $1000001
b := $1000002

b := $1FFFFFE
b := $1FFFFFF
b := $2000000
b := $2000001
b := $2000002

b := $2FFFFFE
b := $2FFFFFF
b := $3000000
b := $3000001
b := $3000002

b := $3FFFFFE
b := $3FFFFFF
b := $4000000
b := $4000001
b := $4000002

b := $4FFFFFE
b := $4FFFFFF
b := $5000000
b := $5000001
b := $5000002

b := $5FFFFFE
b := $5FFFFFF
b := $6000000
b := $6000001
b := $6000002

b := $6FFFFFE
b := $6FFFFFF
b := $7000000
b := $7000001
b := $7000002

b := $7FFFFFE
b := $7FFFFFF
b := $8000000
b := $8000001
b := $8000002

b := $8FFFFFE
b := $8FFFFFF
b := $9000000
b := $9000001
b := $9000002
```

```
b := $9FFFFFE
b := $9FFFFFF
b := $A000000
b := $A000001
b := $A000002

b := $AFFFFFE
b := $AFFFFFF
b := $B000000
b := $B000001
b := $B000002

b := $BFFFFFE
b := $BFFFFFF
b := $C000000
b := $C000001
b := $C000002

b := $CFFFFFE
b := $CFFFFFF
b := $D000000
b := $D000001
b := $D000002

b := $DFFFFFE
b := $DFFFFFF
b := $E000000
b := $E000001
b := $E000002

b := $EFFFFFE
b := $EFFFFFF
b := $F000000
b := $F000001
b := $F000002

b := $0FFFFFFE
b := $0FFFFFFF
b := $10000000
b := $10000001
b := $10000002

b := $1FFFFFFE
b := $1FFFFFFF
b := $20000000
b := $20000001
b := $20000002

b := $2FFFFFFE
b := $2FFFFFFF
b := $30000000
b := $30000001
b := $30000002

b := $3FFFFFFE
b := $3FFFFFFF
b := $40000000
b := $40000001
b := $40000002

b := $4FFFFFFE
b := $4FFFFFFF
b := $50000000
b := $50000001
b := $50000002

b := $5FFFFFFE
b := $5FFFFFFF
b := $60000000
b := $60000001
b := $60000002

b := $6FFFFFFE
b := $6FFFFFFF
b := $70000000
b := $70000001
b := $70000002

b := $7FFFFFFE
b := $7FFFFFFF
b := $80000000
b := $80000001
b := $80000002

b := $8FFFFFFE
b := $8FFFFFFF
b := $90000000
b := $90000001
```

```
  b := $90000002

  b := $9FFFFFFE
  b := $9FFFFFFF
  b := $A0000000
  b := $A0000001
  b := $A0000002

  b := $AFFFFFFE
  b := $AFFFFFFF
  b := $B0000000
  b := $B0000001
  b := $B0000002

  b := $BFFFFFFE
  b := $BFFFFFFF
  b := $C0000000
  b := $C0000001
  b := $C0000002

  b := $CFFFFFFE
  b := $CFFFFFFF
  b := $D0000000
  b := $D0000001
  b := $D0000002

  b := $DFFFFFFE
  b := $DFFFFFFF
  b := $E0000000
  b := $E0000001
  b := $E0000002

  b := $EFFFFFFE
  b := $EFFFFFFF
  b := $F0000000
  b := $F0000001
  b := $F0000002

  b := $FFFFFFFE
  b := $FFFFFFFF

PRI test_Function_1

  Return b

PRI test_Function_2 : returnValue

  returnValue := b

PRI test_Function_3

  Result := b

PRI test_Function_Abort

  Abort

PRI test_Function_AbortValue

  Abort 0

PRI test_Function_Nested( arg1 ,  arg2 )

  RETURN arg1 + arg2

PRI test_If

  If b == 0
    b := 0

  If b <> 1
    b := 1

PRI test_If_Else

  If b == 0
    b := 0
  Else
    b := 1

PRI test_If_ElseIf

  If b == 0
    b := 0
  ElseIf b == 1
    b := 1
  Else
    b := 2
```

```
PRI test_IfNot_ElseIfNot

  IfNot b == 0
    b := 0
  ElseIfNot b == 1
    b := 1
  Else
    b := 2

PRI test_Locks

  LockClr(b)
  LockNew
  LockRet(b)
  LockSet(b)

  b := LockClr(b)
  b := LockNew
  b := LockSet(b)

PRI test_Lookup_And_LookDown

  b := LookUp( b : 1, 2, 3 )
  b := LookDown( b : 1, 2, 3 )

  b := LookUpZ( b : 1, 2, 3 )
  b := LookDownZ( b : 1, 2, 3 )

  b := LookUp( b : 1..2, 2..3, 3..4 )
  b := LookDown( b : 1..2, 2..3, 3..4 )

  b := LookUpZ( b : 1..2, 2..3, 3..4 )
  b := LookDownZ( b : 1..2, 2..3, 3..4 )

PRI test_Memory_Ops

  ByteFill( @b, 0, 1 )
  Wordfill( @b, 0, 2 )
  Longfill( @b, 0, 3 )

  ByteMove( @b, @b, 1 )
  WordMove( @b, @b, 2 )
  LongMove( @b, @b, 3 )

PRI test_Objects

  Obj1_A.Start
  Obj1_A.Stop

  Obj1_B.Start
  Obj1_B.Stop

  b := Obj1_A.Func
  b := Obj1_B.Func

  Obj2_A.Start
  Obj2_A.Stop

  Obj2_B.Start
  Obj2_B.Stop

  b := Obj2_A.Func
  b := Obj2_B.Func

PRI test_One_Line_Commands

  b := ChipVer
  b := ClkFreq
  b := ClkMode
  ClkSet( 0, 1 )
  b := Cnt

PRI test_ReBoot

  ReBoot

PRI test_Registers

  CTRA   := CTRB
  DIRA   := DIRB
  FRQA   := FRQB
  INA    := INA
  INB    := INB
  OUTA   := OUTA
  OUTB   := OUTB
  PAR    := PAR
  PHSA   := PHSB
  VCFG   := VCFG
  VSCL   := VSCL
```

```
  SPR[0] := SPR[1]
  SPR[2] ++

PRI test_RegisterBits

  OUTA[0] := 0
  OUTB[1] := 1

  OUTA[0..1] := 0
  OUTB[3..7] := 1

  OUTA[0]++
  OUTB[1]~~

  OUTA[0..1] := INB[3..7]

  OUTA[0..1] := INB[3..7]++

PRI test_Repeat

  Repeat
    b := 0
  Repeat
    b := 1
    Repeat
      b := 2

PRI test_Repeat_Count

  Repeat 10
    b++

PRI test_Repeat_FromTo

  Repeat b From 0 To 10
    b++

  Repeat b From 0 To 10 Step 1
    b++

  Repeat b From 10 To 0 Step -1
    b++

PRI test_Repeat_Until

  Repeat until b > 10
    b++

  Repeat
    b++
  Until b > 10

PRI test_Repeat_While

  Repeat while b > 10
    b++

  Repeat
    b++
  While b > 10

PRI test_Repeat_With_Next

  Repeat while b > 10
    b++
    Next
    b--

  Repeat
    b++
    Next
    b--
  While b > 10

PRI test_Repeat_With_Quit

  Repeat while b > 10
    b++
    Quit
    b--

  Repeat
    b++
    Quit
    b--
  While b > 10
```

```
PRI test_Strings

  b := StrComp( @b, @b )
  b := String( "Xyzzy" )
  b := StrSize( @b )

PRI test_UnaryOps

  -b
  ++b
  --b
  b++
  b--

  ! b
  NOT b

  ^^ b
  || b


  ~b
  ~~b
  b~
  b~~

  ?b
  b?

  |< b
  >| b

PRI test_UnaryOps_Assigned

  b := ( b := b )

  b := +b
  b := -b
  b := ++b
  b := --b
  b := b++
  b := b--

  b := ! b
  b := NOT b

  b := ^^ b
  b := || b

  b := ~b
  b := ~~b
  b := b~
  b := b~~

  b := ?b
  b := b?

  b := |< b
  b := >| b

  b := @b

PRI test_Vars | bLocal, wLocal, lLocal

  b := 1
  w := 2
  l := 3

  bOverlay := 1
  wOverlay := 2
  lOverlay := 3

  bOverlayEnd := 1
  wOverlayEnd := 2
  lOverlayEnd := 3

  bLocal := 1
  wLocal := 2
  lLocal := 3

PRI test_Vars_Casting

  l.byte := l.byte
  l.word := l.word
  l.long := l.long

  l.byte[1] := l.byte[1]
  l.word[2] := l.word[2]
  l.long[3] := l.long[3]
```

```
PRI test_Vars_Data

  bData := bData[0]
  wData := wData[1]
  lData := lData[3]

  bData[0] := bData
  wData[1] := wData
  lData[3] := lData

PRI test_Vars_Local |t0, t1, t2, t3, t4, t5, t6, t7, t8

  t0 := t0
  t1 := t1
  t2 := t2
  t3 := t3
  t4 := t4
  t5 := t5
  t6 := t6
  t7 := t7
  t8 := t8

  t0++
  t1++
  t2++
  t3++
  t4++
  t5++
  t6++
  t7++
  t8++

PRI test_Wait

  WaitCnt( 0 )
  WaitPeq( 0, 1, 2)
  WaitPne( 0, 1, 2 )
  WaitVid( 0, 1 )

DAT

        ORG 0

Conds   IF_NEVER     ADD     dd,ss
        IF_NC_AND_NZ ADD     dd,ss
        IF_NC_AND_Z  ADD     dd,ss
        IF_NC        ADD     dd,ss
        IF_C_AND_NZ  ADD     dd,ss
        IF_NZ        ADD     dd,ss
        IF_C_NE_Z    ADD     dd,ss
        IF_NC_OR_NZ  ADD     dd,ss
        IF_C_AND_Z   ADD     dd,ss
        IF_C_EQ_Z    ADD     dd,ss
        IF_Z         ADD     dd,ss
        IF_NC_OR_Z   ADD     dd,ss
        IF_C         ADD     dd,ss
        IF_C_OR_NZ   ADD     dd,ss
        IF_C_OR_Z    ADD     dd,ss
        IF_ALWAYS    ADD     dd,ss

                     add     $40,$41  ' Outside the area here
                     add     $50,$52

CondsEnd             JMP     #CondsEnd

dd      long 0
ss      long 0

DAT

        ORG 0

Opcodes              ABS     d,s
                     ABSNEG  d,s
                     ADD     d,s
                     ADDABS  d,s
                     ADDS    d,s
                     ADDSX   d,s
                     ADDX    d,s
                     AND     d,s
                     ANDN    d,s
                     CLKSET  d
                     CMP     d,s
                     CMPS    d,s
                     CMPSUB  d,s
                     CMPSX   d,s
                     CMPX    d,s
                     COGID   d
```

```
COGINIT d
COGSTOP d
DJNZ    d,s
HUBOP   d,s
HUBOP   d,s
LOCKCLR d
LOCKNEW d
LOCKRET d
LOCKSET d
MAX     d,s
MAXS    d,s
MIN     d,s
MINS    d,s
MOV     d,s
MOVD    d,s
MOVI    d,s
MOVS    d,s
MUXC    d,s
MUXNC   d,s
MUXNZ   d,s
MUXZ    d,s
NEG     d,s
NEGC    d,s
NEGNC   d,s
NEGNZ   d,s
NEGZ    d,s
NOP
OR      d,s
RCL     d,s
RCR     d,s
RDBYTE  d,s
RDLONG  d,s
RDWORD  d,s
REV     d,s
ROL     d,s
ROR     d,s
SAR     d,s
SHL     d,s
SHR     d,s
SUB     d,s
SUBABS  d,s
SUBS    d,s
SUBSX   d,s
SUBX    d,s
SUMC    d,s
SUMNC   d,s
SUMNZ   d,s
SUMZ    d,s
TEST    d,s
TJNZ    d,s
TJZ     d,s
WAITCNT d,s
WAITPEQ d,s
WAITPNE d,s
WAITVID d,s
WRBYTE  d,s
WRLONG  d,s
WRWORD  d,s
XOR     d,s

ABS     d,#s
ABSNEG  d,#s
ADD     d,#s
ADDABS  d,#s
ADDS    d,#s
ADDSX   d,#s
ADDX    d,#s
AND     d,#s
ANDN    d,#s
CALL    #TestJmp
CALL    #TestJmpRet
CALL    #TestJmpReg
CLKSET  d
CMP     d,#s
CMPS    d,#s
CMPSUB  d,#s
CMPSX   d,#s
CMPX    d,#s
COGID   d
COGINIT d
COGSTOP d
DJNZ    d,#TestJmp
HUBOP   d,#s
HUBOP   d,#s
LOCKCLR d
LOCKNEW d
LOCKRET d
LOCKSET d
MAX     d,#s
```

```
                        MAXS    d,#s
                        MIN     d,#s
                        MINS    d,#s
                        MOV     d,#s
                        MOVD    d,#s
                        MOVI    d,#s
                        MOVS    d,#s
                        MUXC    d,#s
                        MUXNC   d,#s
                        MUXNZ   d,#s
                        MUXZ    d,#s
                        NEG     d,#s
                        NEGC    d,#s
                        NEGNC   d,#s
                        NEGNZ   d,#s
                        NEGZ    d,#s
                        NOP
                        OR      d,#s
                        RCL     d,#s
                        RCR     d,#s
                        RDBYTE  d,#s
                        RDLONG  d,#s
                        RDWORD  d,#s
                        REV     d,#s
                        ROL     d,#s
                        ROR     d,#s
                        SAR     d,#s
                        SHL     d,#s
                        SHR     d,#s
                        SUB     d,#s
                        SUBABS  d,#s
                        SUBS    d,#s
                        SUBSX   d,#s
                        SUBX    d,#s
                        SUMC    d,#s
                        SUMNC   d,#s
                        SUMNZ   d,#s
                        SUMZ    d,#s
                        TEST    d,#s
                        TJNZ    d,#TestJmp
                        TJZ     d,#TestJmp
                        WAITCNT d,#s
                        WAITPEQ d,#s
                        WAITPNE d,#s
                        WAITVID d,#s
                        WRBYTE  d,#s
                        WRLONG  d,#s
                        WRWORD  d,#s
                        XOR     d,#s

Opcodes_Ret             RET

TestJmp                 JMPRET  TestJmpRet_Ret,#TestJmpRet
                        JMP     #Opcodes
TestJmp_Ret             RET

TestJmpRet              JMPRET  Opcodes_ret,#TestJmp
TestJmpRet_Ret          RET

TestJmpReg              JMPRET  Opcodes_ret,s

                        add     $1D0,$1D1  ' Outside the area here
                        add     $1E0,$1E2

TestJmpReg_Ret          RET

d                       LONG    0
s                       LONG    0
```

```
Disassembly of C:\VB\User\PropLoad\PropTest.eeprom
=================================================

Unidentified '$3C' opcode
Incomplete 'Using' effects decoding

Addr   Org   Instruction     Label    Opcode   Operand
----   ---   -----------     -----    ------   -------

0000         00 1B B7 00             LONG     12000000       ; Clock Freq
0004         00                      BYTE     RCFAST         ; Clock Mode

0005         80                      BYTE     $80            ; Checksum

0006         10 00                   WORD     $0010          ; PBASE - Base of Program
0008         24 17                   WORD     $1724          ; VBASE - Base of Variables
000A         84 17                   WORD     $1784          ; SBASE - Base of Stack
000C         C0 03                   WORD     $03C0          ; PINIT - Initial Program Counter
000E         88 17                   WORD     $1788          ; SINIT - Initial Stack Pointer

0010                        PBASE:   ALIGN    OBJECT

0010         E8 15 2C 04    X1:      LINK     O134, 1068     ; +0 - First Object

0014         B0 03 00 00    X2:      LINK     S59            ; +1
0018         7B 04 00 00    X3:      LINK     S62            ; +2
001C         81 04 00 00    X4:      LINK     S63            ; +3
0020         89 04 00 00    X5:      LINK     S64            ; +4
0024         39 05 00 00    X6:      LINK     S65            ; +5
0028         B7 05 00 00    X7:      LINK     S66            ; +6
002C         E8 05 00 00    X8:      LINK     S73            ; +7
0030         15 06 00 00    X9:      LINK     S74            ; +8
0034         60 06 00 00    X10:     LINK     S75            ; +9
0038         65 06 00 00    X11:     LINK     S76            ; +10
003C         C8 11 00 00    X12:     LINK     F77            ; +11
0040         CC 11 00 00    X13:     LINK     F78            ; +12
0044         D0 11 00 00    X14:     LINK     F79            ; +13
0048         D4 11 00 00    X15:     LINK     S80            ; +14
004C         D6 11 00 00    X16:     LINK     S81            ; +15
0050         D9 11 00 00    X17:     LINK     F82            ; +16
0054         DE 11 00 00    X18:     LINK     S83            ; +17
0058         F1 11 00 00    X19:     LINK     S86            ; +18
005C         00 12 00 00    X20:     LINK     S89            ; +19
0060         1B 12 00 00    X21:     LINK     S93            ; +20
0064         36 12 00 00    X22:     LINK     S97            ; +21
0068         4E 12 00 00    X23:     LINK     S98            ; +22
006C         EF 12 00 00    X24:     LINK     S99            ; +23
0070         15 13 00 00    X25:     LINK     S100           ; +24
0074         4E 13 00 00    X26:     LINK     S101           ; +25
0078         63 13 00 00    X27:     LINK     S102           ; +26
007C         68 13 00 00    X28:     LINK     S103           ; +27
0080         9D 13 00 00    X29:     LINK     S104           ; +28
0084         CF 13 00 00    X30:     LINK     S105           ; +29
0088         E0 13 00 00    X31:     LINK     S108           ; +30
008C         EA 13 00 00    X32:     LINK     S111           ; +31
0090         15 14 00 00    X33:     LINK     S115           ; +32
0094         2C 14 00 00    X34:     LINK     S117           ; +33
0098         43 14 00 00    X35:     LINK     S119           ; +34
009C         64 14 00 00    X36:     LINK     S122           ; +35
00A0         85 14 00 00    X37:     LINK     S125           ; +36
00A4         9D 14 00 00    X38:     LINK     S127           ; +37
00A8         D1 14 00 00    X39:     LINK     S128           ; +38
00AC         36 15 0C 00    X40:     LINK     S129, 12       ; +39
00B0         5D 15 00 00    X41:     LINK     S130           ; +40
00B4         7E 15 00 00    X42:     LINK     S131           ; +41
00B8         AB 15 24 00    X43:     LINK     S132, 36       ; +42
00BC         D6 15 00 00    X44:     LINK     S133           ; +43
00C0         E8 15 10 00    X45:     LINK     O134, 16       ; +44
00C4         E8 15 20 00    X46:     LINK     O134, 32       ; +45
00C8         80 16 30 00    X47:     LINK     O152, 48       ; +46
00CC         80 16 44 00    X48:     LINK     O152, 68       ; +47

00D0         12             B49:     BYTE     18

00D1         00             ; ----   FRAME    CALL WITH RETURN VALUE

00D2         34 12          W50:     WORD     4660
00D4         78 56 34 12    L51:     LONG     305419896

                                     ALIGN          ORG      0

00D8   000   14 26 80 80    A52:     IF_NEVER       ADD      V1, V2
00DC   001   14 26 84 80             IF_NC_AND_NZ   ADD      V1, V2
00E0   002   14 26 88 80             IF_NC_AND_Z    ADD      V1, V2
00E4   003   14 26 8C 80             IF_NC          ADD      V1, V2
00E8   004   14 26 90 80             IF_C_AND_NZ    ADD      V1, V2
00EC   005   14 26 94 80             IF_NZ          ADD      V1, V2
00F0   006   14 26 98 80             IF_C_NE_Z      ADD      V1, V2
```

```
00F4   007   14 26 9C 80             IF_NC_OR_NZ     ADD     V1, V2
00F8   008   14 26 A0 80             IF_C_AND_Z      ADD     V1, V2
00FC   009   14 26 A4 80             IF_C_EQ_Z       ADD     V1, V2
0100   00A   14 26 A8 80             IF_Z            ADD     V1, V2
0104   00B   14 26 AC 80             IF_NC_OR_Z      ADD     V1, V2
0108   00C   14 26 B0 80             IF_C            ADD     V1, V2
010C   00D   14 26 B4 80             IF_C_OR_NZ      ADD     V1, V2
0110   00E   14 26 B8 80             IF_C_OR_Z       ADD     V1, V2
0114   00F   14 26 BC 80                             ADD     V1, V2
0118   010   41 80 BC 80                             ADD     V3, V4
011C   011   52 A0 BC 80                             ADD     V5, V6
0120   012   12 00 7C 5C     A53:                    JMP     #A53

0124   013   00 00 00 00     V1:                     LONG    0
0128   014   00 00 00 00     V2:                     LONG    0

                              ALIGN                 ORG     $40

       040   00 00 00 00     V3:                     LONG    0         ; $A446BD68 / 1757234852
       041   00 00 00 00     V4:                     LONG    0         ; $A446BD34 / 884819620

                              ALIGN                 ORG     $50

       050   00 00 00 00     V5:                     LONG    0         ; $A446BDCC / -860010844

                              ALIGN                 ORG     $52

       052   00 00 00 00     V6:                     LONG    0         ; $A446BD94 / -1799534940

                              ALIGN                 ORG     0

012C   000   A4 46 BD A8     A54:                    ABS     V7, V8
0130   001   A4 46 BD AC                             ABSNEG  V7, V8
0134   002   A4 46 BD 80                             ADD     V7, V8
0138   003   A4 46 BD 88                             ADDABS  V7, V8
013C   004   A4 46 BD D0                             ADDS    V7, V8
0140   005   A4 46 BD D8                             ADDSX   V7, V8
0144   006   A4 46 BD C8                             ADDX    V7, V8
0148   007   A4 46 BD 60                             AND     V7, V8
014C   008   A4 46 BD 64                             ANDN    V7, V8
0150   009   00 46 7D 0C                             CLKSET  V7
0154   00A   A4 46 3D 84                             CMP     V7, V8
0158   00B   A4 46 3D C0                             CMPS    V7, V8
015C   00C   A4 46 BD E0                             CMPSUB  V7, V8
0160   00D   A4 46 3D C4                             CMPSX   V7, V8
0164   00E   A4 46 3D CC                             CMPX    V7, V8
0168   00F   01 46 FD 0C                             COGID   V7
016C   010   02 46 7D 0C                             COGINIT V7
0170   011   03 46 7D 0C                             COGSTOP V7
0174   012   A4 46 BD E4                             DJNZ    V7, V8
0178   013   A4 46 3D 0C                             HUBOP   V7, V8
017C   014   A4 46 3D 0C                             HUBOP   V7, V8
0180   015   07 46 7D 0C                             LOCKCLR V7
0184   016   04 46 FD 0C                             LOCKNEW V7
0188   017   05 46 7D 0C                             LOCKRET V7
018C   018   06 46 7D 0C                             LOCKSET V7
0190   019   A4 46 BD 4C                             MAX     V7, V8
0194   01A   A4 46 BD 44                             MAXS    V7, V8
0198   01B   A4 46 BD 48                             MIN     V7, V8
019C   01C   A4 46 BD 40                             MINS    V7, V8
01A0   01D   A4 46 BD A0                             MOV     V7, V8
01A4   01E   A4 46 BD 54                             MOVD    V7, V8
01A8   01F   A4 46 BD 58                             MOVI    V7, V8
01AC   020   A4 46 BD 50                             MOVS    V7, V8
01B0   021   A4 46 BD 70                             MUXC    V7, V8
01B4   022   A4 46 BD 74                             MUXNC   V7, V8
01B8   023   A4 46 BD 7C                             MUXNZ   V7, V8
01BC   024   A4 46 BD 78                             MUXZ    V7, V8
01C0   025   A4 46 BD A4                             NEG     V7, V8
01C4   026   A4 46 BD B0                             NEGC    V7, V8
01C8   027   A4 46 BD B4                             NEGNC   V7, V8
01CC   028   A4 46 BD BC                             NEGNZ   V7, V8
01D0   029   A4 46 BD B8                             NEGZ    V7, V8
01D4   02A   00 00 00 00                             NOP
01D8   02B   A4 46 BD 68                             OR      V7, V8
01DC   02C   A4 46 BD 34                             RCL     V7, V8
01E0   02D   A4 46 BD 30                             RCR     V7, V8
01E4   02E   A4 46 BD 00                             RDBYTE  V7, V8
01E8   02F   A4 46 BD 08                             RDLONG  V7, V8
01EC   030   A4 46 BD 04                             RDWORD  V7, V8
01F0   031   A4 46 BD 3C                             REV     V7, V8
01F4   032   A4 46 BD 24                             ROL     V7, V8
01F8   033   A4 46 BD 20                             ROR     V7, V8
01FC   034   A4 46 BD 38                             SAR     V7, V8
0200   035   A4 46 BD 2C                             SHL     V7, V8
0204   036   A4 46 BD 28                             SHR     V7, V8
0208   037   A4 46 BD 84                             SUB     V7, V8
020C   038   A4 46 BD 8C                             SUBABS  V7, V8
0210   039   A4 46 BD D4                             SUBS    V7, V8
```

```
0214   03A   A4 46 BD DC                     SUBSX    V7, V8
0218   03B   A4 46 BD CC                     SUBX     V7, V8
021C   03C   A4 46 BD 90                     SUMC     V7, V8
0220   03D   A4 46 BD 94                     SUMNC    V7, V8
0224   03E   A4 46 BD 9C                     SUMNZ    V7, V8
0228   03F   A4 46 BD 98                     SUMZ     V7, V8
022C   040   A4 46 3D 60                     TEST     V7, V8
0230   041   A4 46 3D E8                     TJNZ     V7, V8
0234   042   A4 46 3D EC                     TJZ      V7, V8
0238   043   A4 46 BD F8                     WAITCNT  V7, V8
023C   044   A4 46 3D F0                     WAITPEQ  V7, V8
0240   045   A4 46 3D F4                     WAITPNE  V7, V8
0244   046   A4 46 3D FC                     WAITVID  V7, V8
0248   047   A4 46 3D 00                     WRBYTE   V7, V8
024C   048   A4 46 3D 08                     WRLONG   V7, V8
0250   049   A4 46 3D 04                     WRWORD   V7, V8
0254   04A   A4 46 BD 6C                     XOR      V7, V8
0258   04B   A4 46 FD A8                     ABS      V7, #$A4
025C   04C   A4 46 FD AC                     ABSNEG   V7, #$A4
0260   04D   A4 46 FD 80                     ADD      V7, #$A4
0264   04E   A4 46 FD 88                     ADDABS   V7, #$A4
0268   04F   A4 46 FD D0                     ADDS     V7, #$A4
026C   050   A4 46 FD D8                     ADDSX    V7, #$A4
0270   051   A4 46 FD C8                     ADDX     V7, #$A4
0274   052   A4 46 FD 60                     AND      V7, #$A4
0278   053   A4 46 FD 64                     ANDN     V7, #$A4
027C   054   9A 38 FD 5C                     CALL     #A56
0280   055   9D 3C FD 5C                     CALL     #A57
0284   056   9F 44 FD 5C                     CALL     #A58
0288   057   00 46 7D 0C                     CLKSET   V7
028C   058   A4 46 7D 84                     CMP      V7, #$A4
0290   059   A4 46 7D C0                     CMPS     V7, #$A4
0294   05A   A4 46 FD E0                     CMPSUB   V7, #$A4
0298   05B   A4 46 7D C4                     CMPSX    V7, #$A4
029C   05C   A4 46 7D CC                     CMPX     V7, #$A4
02A0   05D   01 46 FD 0C                     COGID    V7
02A4   05E   02 46 7D 0C                     COGINIT  V7
02A8   05F   03 46 7D 0C                     COGSTOP  V7
02AC   060   9A 46 FD E4                     DJNZ     V7, #A56
02B0   061   A4 46 7D 0C                     LOCKNEW  V7
02B4   062   A4 46 7D 0C                     LOCKNEW  V7
02B8   063   07 46 7D 0C                     LOCKCLR  V7
02BC   064   04 46 FD 0C                     LOCKNEW  V7
02C0   065   05 46 7D 0C                     LOCKRET  V7
02C4   066   06 46 7D 0C                     LOCKSET  V7
02C8   067   A4 46 FD 4C                     MAX      V7, #$A4
02CC   068   A4 46 FD 44                     MAXS     V7, #$A4
02D0   069   A4 46 FD 48                     MIN      V7, #$A4
02D4   06A   A4 46 FD 40                     MINS     V7, #$A4
02D8   06B   A4 46 FD A0                     MOV      V7, #$A4
02DC   06C   A4 46 FD 54                     MOVD     V7, #$A4
02E0   06D   A4 46 FD 58                     MOVI     V7, #$A4
02E4   06E   A4 46 FD 50                     MOVS     V7, #$A4
02E8   06F   A4 46 FD 70                     MUXC     V7, #$A4
02EC   070   A4 46 FD 74                     MUXNC    V7, #$A4
02F0   071   A4 46 FD 7C                     MUXNZ    V7, #$A4
02F4   072   A4 46 FD 78                     MUXZ     V7, #$A4
02F8   073   A4 46 FD A4                     NEG      V7, #$A4
02FC   074   A4 46 FD B0                     NEGC     V7, #$A4
0300   075   A4 46 FD B4                     NEGNC    V7, #$A4
0304   076   A4 46 FD BC                     NEGNZ    V7, #$A4
0308   077   A4 46 FD B8                     NEGZ     V7, #$A4
030C   078   00 00 00 00                     NOP
0310   079   A4 46 FD 68                     OR       V7, #$A4
0314   07A   A4 46 FD 34                     RCL      V7, #$A4
0318   07B   A4 46 FD 30                     RCR      V7, #$A4
031C   07C   A4 46 FD 00                     RDBYTE   V7, #X38.BYTE
0320   07D   A4 46 FD 08                     RDLONG   V7, #X38.LONG
0324   07E   A4 46 FD 04                     RDWORD   V7, #X38.WORD
0328   07F   A4 46 FD 3C                     REV      V7, #$A4
032C   080   A4 46 FD 24                     ROL      V7, #$A4
0330   081   A4 46 FD 20                     ROR      V7, #$A4
0334   082   A4 46 FD 38                     SAR      V7, #$A4
0338   083   A4 46 FD 2C                     SHL      V7, #$A4
033C   084   A4 46 FD 28                     SHR      V7, #$A4
0340   085   A4 46 FD 84                     SUB      V7, #$A4
0344   086   A4 46 FD 8C                     SUBABS   V7, #$A4
0348   087   A4 46 FD D4                     SUBS     V7, #$A4
034C   088   A4 46 FD DC                     SUBSX    V7, #$A4
0350   089   A4 46 FD CC                     SUBX     V7, #$A4
0354   08A   A4 46 FD 90                     SUMC     V7, #$A4
0358   08B   A4 46 FD 94                     SUMNC    V7, #$A4
035C   08C   A4 46 FD 9C                     SUMNZ    V7, #$A4
0360   08D   A4 46 FD 98                     SUMZ     V7, #$A4
0364   08E   A4 46 7D 60                     TEST     V7, #$A4
0368   08F   9A 46 7D E8                     TJNZ     V7, #A56
036C   090   9A 46 7D EC                     TJZ      V7, #A56
0370   091   A4 46 FD F8                     WAITCNT  V7, #$A4
0374   092   A4 46 7D F0                     WAITPEQ  V7, #$A4
```

```
0378   093   A4 46 7D F4                              WAITPNE  V7, #$A4
037C   094   A4 46 7D FC                              WAITVID  V7, #$A4
0380   095   A4 46 7D 00                              WRBYTE   V7, #X38.BYTE
0384   096   A4 46 7D 08                              WRLONG   V7, #X38.LONG
0388   097   A4 46 7D 04                              WRWORD   V7, #X38.WORD
038C   098   A4 46 FD 6C                              XOR      V7, #$A4
0390   099   00 00 7C 5C     R55_ret:                 RET

0394   09A   9D 3C FD 5C     A56:                     CALL     #A57
0398   09B   00 00 7C 5C                              JMP      #A54
039C   09C   00 00 7C 5C     A56_ret:                 RET

03A0   09D   9A 32 FD 5C     A57:                     JMPRET   V9, #A56
03A4   09E   00 00 7C 5C     A57_ret:                 RET

03A8   09F   A4 32 BD 5C     A58:                     JMPRET   V9, V8
03AC   0A0   D1 A1 BF 80                              ADD      V10, V11
03B0   0A1   E2 C1 BF 80                              ADD      V12, V13
03B4   0A2   00 00 7C 5C     A58_ret:                 RET

03B8   0A3   00 00 00 00     V7:                      LONG     0
03BC   0A4   00 00 00 00     V8:                      LONG     0

                                      ALIGN           ORG      $1D0

       1D0   00 00 00 00     V10:                     LONG     0         ; $890C3908 / 137956489
       1D1   00 00 00 00     V11:                     LONG     0         ; $01890C39 / 957122817

                                      ALIGN           ORG      $1E0

       1E0   00 00 00 00     V12:                     LONG     0         ; $FE890C39 / 957123070

                                      ALIGN           ORG      $1E2

       1E2   00 00 00 00     V13:                     LONG     0         ; $390B0089 / -1996485831

03C0                         PINIT:   ALIGN    SPIN

====                                  ; PUB Main
====                                  ;   test_Arguments_1( 1 )
====                                  ;   test_Arguments_2( 1, 2 )
====                                  ;   test_BinaryOps
====                                  ;   test_BinaryOps_Assigned
====                                  ;   test_Case
====                                  ;   test_Constants
====                                  ;   test_Cog_Assembler
====                                  ;   test_Cog_Spin
====                                  ;   test_Cog_Spin_Args( 0, 1, 2 )
====                                  ;   test_If
====                                  ;   test_If_Else
====                                  ;   test_If_ElseIf
====                                  ;   test_IfNot_ElseIfNot
====                                  ;   b := test_Function_1
====                                  ;   b := test_Function_2
====                                  ;   b := test_Function_3
====                                  ;   if b := \test_function_Abort
====                                  ;     b := 0
====                                  ;   if b := \test_function_AbortValue
====                                  ;     b := 0
====                                  ;   b := test_Function_Nested( 1,2 )
====                                  ;   b := test_Function_Nested( test_Function_Nested(1,2) , test_Function_Nested(3,4) )
====                                  ;   b := test_Function_Nested( test_Function_Nested(1,2) , test_Function_Nested(3, tes
====                                  ;   test_Locks
====                                  ;   test_Lookup_And_LookDown
====                                  ;   test_Memory_Ops
====                                  ;   test_Objects
====                                  ;   test_One_Line_Commands
====                                  ;   test_ReBoot
====                                  ;   test_Registers
====                                  ;   test_RegisterBits
====                                  ;   test_Repeat
====                                  ;   test_Repeat_Count
====                                  ;   test_Repeat_FromTo
====                                  ;   test_Repeat_Until
====                                  ;   test_Repeat_While
====                                  ;   test_Repeat_With_Next
====                                  ;   test_Repeat_With_Quit
====                                  ;   test_Strings
====                                  ;   test_UnaryOps
====                                  ;   test_UnaryOps_Assigned
====                                  ;   test_Vars
====                                  ;   test_Vars_Casting
====                                  ;   test_Vars_Data
====                                  ;   test_Vars_Local
====                                  ;   test_Wait

03C0           01 36 05 02     S59:     CALLSUB  S62 ( 1 )
03C4           01 36 37 00 05          CALLSUB  S63 ( 1, 2 )
03C9            + 03
```

```
03CA        01 05 04                CALLSUB  S64
03CD        01 05 05                CALLSUB  S65
03D0        01 05 06                CALLSUB  S66
03D3        01 05 0A                CALLSUB  S76
03D6        01 05 07                CALLSUB  S73
03D9        01 05 08                CALLSUB  S74
03DC        01 35 36 37 00          CALLSUB  S75 ( 0, 1, 2 )
03E1         + 05 09
03E3        01 05 11                CALLSUB  S83
03E6        01 05 12                CALLSUB  S86
03E9        01 05 13                CALLSUB  S89
03EC        01 05 14                CALLSUB  S93
03EF        00 05 0B 89 0C          CALLFUN  F77, B174.BYTE
03F4        00 05 0C 89 0C          CALLFUN  F78, B174.BYTE
03F9        00 05 0D 89 0C          CALLFUN  F79, B174.BYTE
03FE        02 05 0E                CALLTRP  S80
0401        8A 0C 80                PUSH     B174.BYTE COPY
0404        0A 03                   JPF      N60
0406        35 89 0C                LET      B174.BYTE, 0
0409        02 05 0F        N60:    CALLTRP  S81
040C        8A 0C 80                PUSH     B174.BYTE COPY
040F        0A 03                   JPF      N61
0411        35 89 0C                LET      B174.BYTE, 0
0414        00 36 37 00 05  N61:    CALLFUN  F82 ( 1, 2 ), B174.BYTE
0419         + 10 89 0C
041C        00                      FRAME    CALL WITH RETURN VALUE
041D        00 36 37 00 05          CALLFUN  F82 ( 1, 2 )
0422         + 10
0423        00 37 21 37 01          CALLFUN  F82 ( 3, 4 )
0428         + 05 10
042A        05 10                   CALL     F82
042C        89 0C                   POP      B174.BYTE
042E        00                      FRAME    CALL WITH RETURN VALUE
042F        00 36 37 00 05          CALLFUN  F82 ( 1, 2 )
0434         + 10
0435        00                      FRAME    CALL WITH RETURN VALUE
0436        37 21                   PUSH     3
0438        00 38 05 38 06          CALLFUN  F82 ( 5, 6 )
043D         + 05 10
043F        05 10                   CALL     F82
0441        05 10                   CALL     F82
0443        89 0C                   POP      B174.BYTE
0445        01 05 15                CALLSUB  S97
0448        01 05 16                CALLSUB  S98
044B        01 05 17                CALLSUB  S99
044E        01 05 18                CALLSUB  S100
0451        01 05 19                CALLSUB  S101
0454        01 05 1A                CALLSUB  S102
0457        01 05 1B                CALLSUB  S103
045A        01 05 1C                CALLSUB  S104
045D        01 05 1D                CALLSUB  S105
0460        01 05 1E                CALLSUB  S108
0463        01 05 1F                CALLSUB  S111
0466        01 05 20                CALLSUB  S115
0469        01 05 21                CALLSUB  S117
046C        01 05 22                CALLSUB  S119
046F        01 05 23                CALLSUB  S122
0472        01 05 24                CALLSUB  S125
0475        01 05 25                CALLSUB  S127
0478        01 05 26                CALLSUB  S128
047B        01 05 27                CALLSUB  S129
047E        01 05 28                CALLSUB  S130
0481        01 05 29                CALLSUB  S131
0484        01 05 2A                CALLSUB  S132
0487        01 05 2B                CALLSUB  S133
048A        32                      RETURN

====                                ; PRI test_Arguments_1(a1)
====                                ;   a1 := 1
====                                ;   b  := a1

                                    ALIGN    STACK          ; For S62

     +0000                          LONG     0              ; Unused Result Variable
     +0004                  VL1:    LONG     0

                                    ALIGN    SPIN

048B        36 65           S62:    LET      VL1.LONG, 1
048D        64 89 0C                LET      B174.BYTE, VL1.LONG
0490        32                      RETURN

====                                ; PRI test_Arguments_2(a1,a2)
====                                ;   a1 := 1
====                                ;   b  := a1
====                                ;   a2 := a1

                                    ALIGN    STACK          ; For S63
```

```
     +0000                          LONG    0               ; Unused Result Variable
     +0004                VL2:      LONG    0
     +0008                VL3:      LONG    0

                                    ALIGN   SPIN

0491         36 65          S63:    LET     VL2.LONG, 1
0493         64 89 0C               LET     B174.BYTE, VL2.LONG
0496         64 69                  LET     VL3.LONG, VL2.LONG
0498         32                     RETURN

====                                ; PRI test_BinaryOps
====                                ;   b := b + b
====                                ;   b := b - b
====                                ;   b := b * b
====                                ;   b := b / b
====                                ;   b := b // b
====                                ;   b := b ** b
====                                ;   b := b & b
====                                ;   b := b | b
====                                ;   b := b ^ b
====                                ;   b := b AND b
====                                ;   b := b OR  b
====                                ;   b := b #> b
====                                ;   b := b <# b
====                                ;   b := b << b
====                                ;   b := b >> b
====                                ;   b := b <- b
====                                ;   b := b -> b
====                                ;   b := b ~> b
====                                ;   b := b >< b
====                                ;   b := b == b
====                                ;   b := b <> b
====                                ;   b := b <  b
====                                ;   b := b >  b
====                                ;   b := b =< b
====                                ;   b := b => b

0499         88 0C          S64:    PUSH    B174.BYTE
049B         88 0C                  PUSH    B174.BYTE
049D         EC                     ADD
049E         89 0C                  POP     B174.BYTE
04A0         88 0C                  PUSH    B174.BYTE
04A2         88 0C                  PUSH    B174.BYTE
04A4         ED                     SUB
04A5         89 0C                  POP     B174.BYTE
04A7         88 0C                  PUSH    B174.BYTE
04A9         88 0C                  PUSH    B174.BYTE
04AB         F4                     MPY
04AC         89 0C                  POP     B174.BYTE
04AE         88 0C                  PUSH    B174.BYTE
04B0         88 0C                  PUSH    B174.BYTE
04B2         F6                     DIV
04B3         89 0C                  POP     B174.BYTE
04B5         88 0C                  PUSH    B174.BYTE
04B7         88 0C                  PUSH    B174.BYTE
04B9         F7                     MOD
04BA         89 0C                  POP     B174.BYTE
04BC         88 0C                  PUSH    B174.BYTE
04BE         88 0C                  PUSH    B174.BYTE
04C0         F5                     MPY_MSW
04C1         89 0C                  POP     B174.BYTE
04C3         88 0C                  PUSH    B174.BYTE
04C5         88 0C                  PUSH    B174.BYTE
04C7         E8                     BIT_AND
04C8         89 0C                  POP     B174.BYTE
04CA         88 0C                  PUSH    B174.BYTE
04CC         88 0C                  PUSH    B174.BYTE
04CE         EA                     BIT_OR
04CF         89 0C                  POP     B174.BYTE
04D1         88 0C                  PUSH    B174.BYTE
04D3         88 0C                  PUSH    B174.BYTE
04D5         EB                     BIT_XOR
04D6         89 0C                  POP     B174.BYTE
04D8         88 0C                  PUSH    B174.BYTE
04DA         88 0C                  PUSH    B174.BYTE
04DC         F0                     LOG_AND
04DD         89 0C                  POP     B174.BYTE
04DF         88 0C                  PUSH    B174.BYTE
04E1         88 0C                  PUSH    B174.BYTE
04E3         F2                     LOG_OR
04E4         89 0C                  POP     B174.BYTE
04E6         88 0C                  PUSH    B174.BYTE
04E8         88 0C                  PUSH    B174.BYTE
04EA         E4                     MIN
04EB         89 0C                  POP     B174.BYTE
04ED         88 0C                  PUSH    B174.BYTE
04EF         88 0C                  PUSH    B174.BYTE
04F1         E5                     MAX
```

```
04F2        89 0C                   POP     B174.BYTE
04F4        88 0C                   PUSH    B174.BYTE
04F6        88 0C                   PUSH    B174.BYTE
04F8        E3                      SHL
04F9        89 0C                   POP     B174.BYTE
04FB        88 0C                   PUSH    B174.BYTE
04FD        88 0C                   PUSH    B174.BYTE
04FF        E2                      SHR
0500        89 0C                   POP     B174.BYTE
0502        88 0C                   PUSH    B174.BYTE
0504        88 0C                   PUSH    B174.BYTE
0506        E1                      ROL
0507        89 0C                   POP     B174.BYTE
0509        88 0C                   PUSH    B174.BYTE
050B        88 0C                   PUSH    B174.BYTE
050D        E0                      ROR
050E        89 0C                   POP     B174.BYTE
0510        88 0C                   PUSH    B174.BYTE
0512        88 0C                   PUSH    B174.BYTE
0514        EE                      SAR
0515        89 0C                   POP     B174.BYTE
0517        88 0C                   PUSH    B174.BYTE
0519        88 0C                   PUSH    B174.BYTE
051B        EF                      BIT_REV
051C        89 0C                   POP     B174.BYTE
051E        88 0C                   PUSH    B174.BYTE
0520        88 0C                   PUSH    B174.BYTE
0522        FC                      EQ
0523        89 0C                   POP     B174.BYTE
0525        88 0C                   PUSH    B174.BYTE
0527        88 0C                   PUSH    B174.BYTE
0529        FB                      NE
052A        89 0C                   POP     B174.BYTE
052C        88 0C                   PUSH    B174.BYTE
052E        88 0C                   PUSH    B174.BYTE
0530        F9                      LT
0531        89 0C                   POP     B174.BYTE
0533        88 0C                   PUSH    B174.BYTE
0535        88 0C                   PUSH    B174.BYTE
0537        FA                      GT
0538        89 0C                   POP     B174.BYTE
053A        88 0C                   PUSH    B174.BYTE
053C        88 0C                   PUSH    B174.BYTE
053E        FD                      LE
053F        89 0C                   POP     B174.BYTE
0541        88 0C                   PUSH    B174.BYTE
0543        88 0C                   PUSH    B174.BYTE
0545        FE                      GE
0546        89 0C                   POP     B174.BYTE
0548        32                      RETURN

====                                ; PRI test_BinaryOps_Assigned
====                                ;   b += b
====                                ;   b -= b
====                                ;   b *= b
====                                ;   b /= b
====                                ;   b //= b
====                                ;   b **= b
====                                ;   b &= b
====                                ;   b |= b
====                                ;   b ^= b
====                                ;   b AND= b
====                                ;   b OR=  b
====                                ;   b #>= b
====                                ;   b <#= b
====                                ;   b <<= b
====                                ;   b >>= b
====                                ;   b <-= b
====                                ;   b ->= b
====                                ;   b ~>= b
====                                ;   b ><= b
====                                ;   b === b
====                                ;   b <>= b
====                                ;   b <=  b
====                                ;   b >=  b
====                                ;   b =<= b
====                                ;   b =>= b

0549        88 0C           S65:    PUSH    B174.BYTE
054B        8A 0C 4C                ADD     B174.BYTE
054E        88 0C                   PUSH    B174.BYTE
0550        8A 0C 4D                SUB     B174.BYTE
0553        88 0C                   PUSH    B174.BYTE
0555        8A 0C 54                MPY     B174.BYTE
0558        88 0C                   PUSH    B174.BYTE
055A        8A 0C 56                DIV     B174.BYTE
055D        88 0C                   PUSH    B174.BYTE
055F        8A 0C 57                MOD     B174.BYTE
0562        88 0C                   PUSH    B174.BYTE
```

```
0564        8A 0C 55                MPY_BIG  B174.BYTE
0567        88 0C                   PUSH     B174.BYTE
0569        8A 0C 48                BIT_AND  B174.BYTE
056C        88 0C                   PUSH     B174.BYTE
056E        8A 0C 4A                BIT_OR   B174.BYTE
0571        88 0C                   PUSH     B174.BYTE
0573        8A 0C 4B                BIT_XOR  B174.BYTE
0576        88 0C                   PUSH     B174.BYTE
0578        8A 0C 50                LOG_AND  B174.BYTE
057B        88 0C                   PUSH     B174.BYTE
057D        8A 0C 52                LOG_OR   B174.BYTE
0580        88 0C                   PUSH     B174.BYTE
0582        8A 0C 44                MIN      B174.BYTE
0585        88 0C                   PUSH     B174.BYTE
0587        8A 0C 45                MAX      B174.BYTE
058A        88 0C                   PUSH     B174.BYTE
058C        8A 0C 43                SHL      B174.BYTE
058F        88 0C                   PUSH     B174.BYTE
0591        8A 0C 42                SHR      B174.BYTE
0594        88 0C                   PUSH     B174.BYTE
0596        8A 0C 41                ROL      B174.BYTE
0599        88 0C                   PUSH     B174.BYTE
059B        8A 0C 40                ROR      B174.BYTE
059E        88 0C                   PUSH     B174.BYTE
05A0        8A 0C 4E                SAR      B174.BYTE
05A3        88 0C                   PUSH     B174.BYTE
05A5        8A 0C 4F                BIT_REV  B174.BYTE
05A8        88 0C                   PUSH     B174.BYTE
05AA        8A 0C 5C                EQ       B174.BYTE
05AD        88 0C                   PUSH     B174.BYTE
05AF        8A 0C 5B                NE       B174.BYTE
05B2        88 0C                   PUSH     B174.BYTE
05B4        8A 0C 59                LT       B174.BYTE
05B7        88 0C                   PUSH     B174.BYTE
05B9        8A 0C 5A                GT       B174.BYTE
05BC        88 0C                   PUSH     B174.BYTE
05BE        8A 0C 5D                LE       B174.BYTE
05C1        88 0C                   PUSH     B174.BYTE
05C3        8A 0C 5E                GE       B174.BYTE
05C6        32                      RETURN

====                                ; PRI test_Case
====                                ;   Case b
====                                ;     1 : b := 1
====                                ;     2 : b := 2
====                                ;   Case b
====                                ;     1..2 : b := 1
====                                ;     2..3 : b := 2

05C7        39 05 CD        S66:    PUSH     1485
05CA        88 0C                   PUSH     B174.BYTE
05CC        36 0D 05                CASE     1, J67
05CF        37 00 0D 05             CASE     2, J68
05D3        0C                      GOTO[]
05D4        36 89 0C        J67:    LET      B174.BYTE, 1
05D7        0C                      GOTO[]
05D8        37 00 89 0C     J68:    LET      B174.BYTE, 2
05DC        0C                      GOTO[]
05DD        39 05 E7        J69:    PUSH     1511
05E0        88 0C                   PUSH     B174.BYTE
05E2        36 37 00 0E 07          CASE     1, 2, J70
05E7        37 00 37 21 0E          CASE     2, 3, J71
05EC         + 05
05ED        0C                      GOTO[]
05EE        36 89 0C        J70:    LET      B174.BYTE, 1
05F1        0C                      GOTO[]
05F2        37 00 89 0C     J71:    LET      B174.BYTE, 2
05F6        0C                      GOTO[]
05F7        32              J72:    RETURN

====                                ; PRI test_Cog_Assembler
====                                ;   CogInit(0, @Conds, 1)
====                                ;   CogNew( @Conds, 2 )
====                                ;   b := CogNew( @Conds, 5 )
====                                ;   CogInit(0, @Opcodes, 1)
====                                ;   CogNew( @Opcodes, 2 )
====                                ;   b := CogNew( @Opcodes, 5 )

05F8        35 C7 80 C8 36  S73:    COGISUB  0, #A52.LONG, 1
05FD         + 2C
05FE        34 C7 80 C8 37          COGISUB  -1, #A52.LONG, 2
0603         + 00 2C
0605        34 C7 80 C8 38          COGIFUN  ( -1, #A52.LONG, 5 ), B174.BYTE
060A         + 05 28 89 0C
060E        35 C7 81 1C 36          COGISUB  0, #A54.LONG, 1
0613         + 2C
0614        34 C7 81 1C 37          COGISUB  -1, #A54.LONG, 2
0619         + 00 2C
061B        34 C7 81 1C 38          COGIFUN  ( -1, #A54.LONG, 5 ), B174.BYTE
```

```
0620          + 05 28 89 0C
0624          32                        RETURN

====                                    ; PRI test_Cog_Spin
====                                    ;   b := CogId
====                                    ;   CogInit(0, test_Cog_Spin, 1)
====                                    ;   CogNew( test_Cog_Spin, 2 )
====                                    ;   b := CogNew( test_Cog_Spin, 5 )
====                                    ;   CogInit(0, test_Cog_Spin_Args(0,1,2), 1)
====                                    ;   CogNew( test_Cog_Spin_Args(0,1,2), 2 )
====                                    ;   b := CogNew( test_Cog_Spin_Args(0,1,2), 5 )
====                                    ;   CogStop( 0 )
====                                    ;   CogStop( b )

0625          3F 89 89 0C     S74:      LET       B174.BYTE, MEM+9.LONG
0629          37 02                     PUSH      8
062B          36                        PUSH      1
062C          15                        MARK
062D          35                        PUSH      0
062E          3F 8F                     PUSH      MEM+15.LONG
0630          37 61                     PUSH      $FFFFFFFC
0632          D1                        POP       MEM[][].LONG
0633          2C                        COGISUB
0634          37 02                     PUSH      8
0636          37 00                     PUSH      2
0638          15                        MARK
0639          2C                        COGISUB
063A          37 02                     PUSH      8
063C          38 05                     PUSH      5
063E          15                        MARK
063F          28 89 0C                  COGIFUN   (  ), B174.BYTE
0642          35                        PUSH      0
0643          36                        PUSH      1
0644          37 00                     PUSH      2
0646          39 03 09                  PUSH      777
0649          36                        PUSH      1
064A          15                        MARK
064B          35                        PUSH      0
064C          3F 8F                     PUSH      MEM+15.LONG
064E          37 61                     PUSH      $FFFFFFFC
0650          D1                        POP       MEM[][].LONG
0651          2C                        COGISUB
0652          35                        PUSH      0
0653          36                        PUSH      1
0654          37 00                     PUSH      2
0656          39 03 09                  PUSH      777
0659          37 00                     PUSH      2
065B          15                        MARK
065C          2C                        COGISUB
065D          35                        PUSH      0
065E          36                        PUSH      1
065F          37 00                     PUSH      2
0661          39 03 09                  PUSH      777
0664          38 05                     PUSH      5
0666          15                        MARK
0667          28 89 0C                  COGIFUN   (  ), B174.BYTE
066A          35                        PUSH      0
066B          21                        COGSTOP
066C          88 0C                     PUSH      B174.BYTE
066E          21                        COGSTOP
066F          32                        RETURN

====                                    ; PRI test_Cog_Spin_Args( arg1, arg2, arg3 )
====                                    ;   arg1 := arg2 + arg3

                                        ALIGN     STACK           ; For S75

     +0000                              LONG      0               ; Unused Result Variable
     +0004                VL4:          LONG      0
     +0008                VL5:          LONG      0
     +000C                VL6:          LONG      0

                                        ALIGN     SPIN

0670          68              S75:      PUSH      VL5.LONG
0671          6C                        PUSH      VL6.LONG
0672          EC                        ADD
0673          65                        POP       VL4.LONG
0674          32                        RETURN

====                                    ; PRI test_Constants
====                                    ;   b := TRUE
====                                    ;   b := FALSE
====                                    ;   b := NEGX
====                                    ;   b := POSX
====                                    ;   b := $0
====                                    ;   b := $1
====                                    ;   b := $2
====                                    ;   b := $0E
```

```
====                                ;   b := $0F
====                                ;   b := $10
====                                ;   b := $11
====                                ;   b := $12
====                                ;   b := $1E
====                                ;   b := $1F
====                                ;   b := $20
====                                ;   b := $21
====                                ;   b := $22
====                                ;   b := $2E
====                                ;   b := $2F
====                                ;   b := $30
====                                ;   b := $31
====                                ;   b := $32
====                                ;   b := $3E
====                                ;   b := $3F
====                                ;   b := $40
====                                ;   b := $41
====                                ;   b := $42
====                                ;   b := $4E
====                                ;   b := $4F
====                                ;   b := $50
====                                ;   b := $51
====                                ;   b := $52
====                                ;   b := $5E
====                                ;   b := $5F
====                                ;   b := $60
====                                ;   b := $61
====                                ;   b := $62
====                                ;   b := $6E
====                                ;   b := $6F
====                                ;   b := $70
====                                ;   b := $71
====                                ;   b := $72
====                                ;   b := $7E
====                                ;   b := $7F
====                                ;   b := $80
====                                ;   b := $81
====                                ;   b := $82
====                                ;   b := $8E
====                                ;   b := $8F
====                                ;   b := $90
====                                ;   b := $91
====                                ;   b := $92
====                                ;   b := $9E
====                                ;   b := $9F
====                                ;   b := $A0
====                                ;   b := $A1
====                                ;   b := $A2
====                                ;   b := $AE
====                                ;   b := $AF
====                                ;   b := $B0
====                                ;   b := $B1
====                                ;   b := $B2
====                                ;   b := $BE
====                                ;   b := $BF
====                                ;   b := $C0
====                                ;   b := $C1
====                                ;   b := $C2
====                                ;   b := $CE
====                                ;   b := $CF
====                                ;   b := $D0
====                                ;   b := $D1
====                                ;   b := $D2
====                                ;   b := $DE
====                                ;   b := $DF
====                                ;   b := $E0
====                                ;   b := $E1
====                                ;   b := $E2
====                                ;   b := $EE
====                                ;   b := $EF
====                                ;   b := $F0
====                                ;   b := $F1
====                                ;   b := $F2
====                                ;   b := $0FE
====                                ;   b := $0FF
====                                ;   b := $100
====                                ;   b := $101
====                                ;   b := $102
====                                ;   b := $1FE
====                                ;   b := $1FF
====                                ;   b := $200
====                                ;   b := $201
====                                ;   b := $202
====                                ;   b := $2FE
====                                ;   b := $2FF
====                                ;   b := $300
====                                ;   b := $301
====                                ;   b := $302
```

```
====                                ;   b := $3FE
====                                ;   b := $3FF
====                                ;   b := $400
====                                ;   b := $401
====                                ;   b := $402
====                                ;   b := $4FE
====                                ;   b := $4FF
====                                ;   b := $500
====                                ;   b := $501
====                                ;   b := $502
====                                ;   b := $5FE
====                                ;   b := $5FF
====                                ;   b := $600
====                                ;   b := $601
====                                ;   b := $602
====                                ;   b := $6FE
====                                ;   b := $6FF
====                                ;   b := $700
====                                ;   b := $701
====                                ;   b := $702
====                                ;   b := $7FE
====                                ;   b := $7FF
====                                ;   b := $800
====                                ;   b := $801
====                                ;   b := $802
====                                ;   b := $8FE
====                                ;   b := $8FF
====                                ;   b := $900
====                                ;   b := $901
====                                ;   b := $902
====                                ;   b := $9FE
====                                ;   b := $9FF
====                                ;   b := $A00
====                                ;   b := $A01
====                                ;   b := $A02
====                                ;   b := $AFE
====                                ;   b := $AFF
====                                ;   b := $B00
====                                ;   b := $B01
====                                ;   b := $B02
====                                ;   b := $BFE
====                                ;   b := $BFF
====                                ;   b := $C00
====                                ;   b := $C01
====                                ;   b := $C02
====                                ;   b := $CFE
====                                ;   b := $CFF
====                                ;   b := $D00
====                                ;   b := $D01
====                                ;   b := $D02
====                                ;   b := $DFE
====                                ;   b := $DFF
====                                ;   b := $E00
====                                ;   b := $E01
====                                ;   b := $E02
====                                ;   b := $EFE
====                                ;   b := $EFF
====                                ;   b := $F00
====                                ;   b := $F01
====                                ;   b := $F02
====                                ;   b := $0FFE
====                                ;   b := $0FFF
====                                ;   b := $1000
====                                ;   b := $1001
====                                ;   b := $1002
====                                ;   b := $1FFE
====                                ;   b := $1FFF
====                                ;   b := $2000
====                                ;   b := $2001
====                                ;   b := $2002
====                                ;   b := $2FFE
====                                ;   b := $2FFF
====                                ;   b := $3000
====                                ;   b := $3001
====                                ;   b := $3002
====                                ;   b := $3FFE
====                                ;   b := $3FFF
====                                ;   b := $4000
====                                ;   b := $4001
====                                ;   b := $4002
====                                ;   b := $4FFE
====                                ;   b := $4FFF
====                                ;   b := $5000
====                                ;   b := $5001
====                                ;   b := $5002
====                                ;   b := $5FFE
====                                ;   b := $5FFF
====                                ;   b := $6000
====                                ;   b := $6001
```

```
====                                ;   b := $6002
====                                ;   b := $6FFE
====                                ;   b := $6FFF
====                                ;   b := $7000
====                                ;   b := $7001
====                                ;   b := $7002
====                                ;   b := $7FFE
====                                ;   b := $7FFF
====                                ;   b := $8000
====                                ;   b := $8001
====                                ;   b := $8002
====                                ;   b := $8FFE
====                                ;   b := $8FFF
====                                ;   b := $9000
====                                ;   b := $9001
====                                ;   b := $9002
====                                ;   b := $9FFE
====                                ;   b := $9FFF
====                                ;   b := $A000
====                                ;   b := $A001
====                                ;   b := $A002
====                                ;   b := $AFFE
====                                ;   b := $AFFF
====                                ;   b := $B000
====                                ;   b := $B001
====                                ;   b := $B002
====                                ;   b := $BFFE
====                                ;   b := $BFFF
====                                ;   b := $C000
====                                ;   b := $C001
====                                ;   b := $C002
====                                ;   b := $CFFE
====                                ;   b := $CFFF
====                                ;   b := $D000
====                                ;   b := $D001
====                                ;   b := $D002
====                                ;   b := $DFFE
====                                ;   b := $DFFF
====                                ;   b := $E000
====                                ;   b := $E001
====                                ;   b := $E002
====                                ;   b := $EFFE
====                                ;   b := $EFFF
====                                ;   b := $F000
====                                ;   b := $F001
====                                ;   b := $F002
====                                ;   b := $0FFFE
====                                ;   b := $0FFFF
====                                ;   b := $10000
====                                ;   b := $10001
====                                ;   b := $10002
====                                ;   b := $1FFFE
====                                ;   b := $1FFFF
====                                ;   b := $20000
====                                ;   b := $20001
====                                ;   b := $20002
====                                ;   b := $2FFFE
====                                ;   b := $2FFFF
====                                ;   b := $30000
====                                ;   b := $30001
====                                ;   b := $30002
====                                ;   b := $3FFFE
====                                ;   b := $3FFFF
====                                ;   b := $40000
====                                ;   b := $40001
====                                ;   b := $40002
====                                ;   b := $4FFFE
====                                ;   b := $4FFFF
====                                ;   b := $50000
====                                ;   b := $50001
====                                ;   b := $50002
====                                ;   b := $5FFFE
====                                ;   b := $5FFFF
====                                ;   b := $60000
====                                ;   b := $60001
====                                ;   b := $60002
====                                ;   b := $6FFFE
====                                ;   b := $6FFFF
====                                ;   b := $70000
====                                ;   b := $70001
====                                ;   b := $70002
====                                ;   b := $7FFFE
====                                ;   b := $7FFFF
====                                ;   b := $80000
====                                ;   b := $80001
====                                ;   b := $80002
====                                ;   b := $8FFFE
====                                ;   b := $8FFFF
====                                ;   b := $90000
```

```
====                                ;   b := $90001
====                                ;   b := $90002
====                                ;   b := $9FFFE
====                                ;   b := $9FFFF
====                                ;   b := $A0000
====                                ;   b := $A0001
====                                ;   b := $A0002
====                                ;   b := $AFFFE
====                                ;   b := $AFFFF
====                                ;   b := $B0000
====                                ;   b := $B0001
====                                ;   b := $B0002
====                                ;   b := $BFFFE
====                                ;   b := $BFFFF
====                                ;   b := $C0000
====                                ;   b := $C0001
====                                ;   b := $C0002
====                                ;   b := $CFFFE
====                                ;   b := $CFFFF
====                                ;   b := $D0000
====                                ;   b := $D0001
====                                ;   b := $D0002
====                                ;   b := $DFFFE
====                                ;   b := $DFFFF
====                                ;   b := $E0000
====                                ;   b := $E0001
====                                ;   b := $E0002
====                                ;   b := $EFFFE
====                                ;   b := $EFFFF
====                                ;   b := $F0000
====                                ;   b := $F0001
====                                ;   b := $F0002
====                                ;   b := $0FFFFE
====                                ;   b := $0FFFFF
====                                ;   b := $100000
====                                ;   b := $100001
====                                ;   b := $100002
====                                ;   b := $1FFFFE
====                                ;   b := $1FFFFF
====                                ;   b := $200000
====                                ;   b := $200001
====                                ;   b := $200002
====                                ;   b := $2FFFFE
====                                ;   b := $2FFFFF
====                                ;   b := $300000
====                                ;   b := $300001
====                                ;   b := $300002
====                                ;   b := $3FFFFE
====                                ;   b := $3FFFFF
====                                ;   b := $400000
====                                ;   b := $400001
====                                ;   b := $400002
====                                ;   b := $4FFFFE
====                                ;   b := $4FFFFF
====                                ;   b := $500000
====                                ;   b := $500001
====                                ;   b := $500002
====                                ;   b := $5FFFFE
====                                ;   b := $5FFFFF
====                                ;   b := $600000
====                                ;   b := $600001
====                                ;   b := $600002
====                                ;   b := $6FFFFE
====                                ;   b := $6FFFFF
====                                ;   b := $700000
====                                ;   b := $700001
====                                ;   b := $700002
====                                ;   b := $7FFFFE
====                                ;   b := $7FFFFF
====                                ;   b := $800000
====                                ;   b := $800001
====                                ;   b := $800002
====                                ;   b := $8FFFFE
====                                ;   b := $8FFFFF
====                                ;   b := $900000
====                                ;   b := $900001
====                                ;   b := $900002
====                                ;   b := $9FFFFE
====                                ;   b := $9FFFFF
====                                ;   b := $A00000
====                                ;   b := $A00001
====                                ;   b := $A00002
====                                ;   b := $AFFFFE
====                                ;   b := $AFFFFF
====                                ;   b := $B00000
====                                ;   b := $B00001
====                                ;   b := $B00002
====                                ;   b := $BFFFFE
====                                ;   b := $BFFFFF
```

```
====                                ;   b := $C00000
====                                ;   b := $C00001
====                                ;   b := $C00002
====                                ;   b := $CFFFFE
====                                ;   b := $CFFFFF
====                                ;   b := $D00000
====                                ;   b := $D00001
====                                ;   b := $D00002
====                                ;   b := $DFFFFE
====                                ;   b := $DFFFFF
====                                ;   b := $E00000
====                                ;   b := $E00001
====                                ;   b := $E00002
====                                ;   b := $EFFFFF
====                                ;   b := $EFFFFF
====                                ;   b := $F00000
====                                ;   b := $F00001
====                                ;   b := $F00002
====                                ;   b := $0FFFFFE
====                                ;   b := $0FFFFFF
====                                ;   b := $1000000
====                                ;   b := $1000001
====                                ;   b := $1000002
====                                ;   b := $1FFFFFE
====                                ;   b := $1FFFFFF
====                                ;   b := $2000000
====                                ;   b := $2000001
====                                ;   b := $2000002
====                                ;   b := $2FFFFFE
====                                ;   b := $2FFFFFF
====                                ;   b := $3000000
====                                ;   b := $3000001
====                                ;   b := $3000002
====                                ;   b := $3FFFFFE
====                                ;   b := $3FFFFFF
====                                ;   b := $4000000
====                                ;   b := $4000001
====                                ;   b := $4000002
====                                ;   b := $4FFFFFE
====                                ;   b := $4FFFFFF
====                                ;   b := $5000000
====                                ;   b := $5000001
====                                ;   b := $5000002
====                                ;   b := $5FFFFFE
====                                ;   b := $5FFFFFF
====                                ;   b := $6000000
====                                ;   b := $6000001
====                                ;   b := $6000002
====                                ;   b := $6FFFFFE
====                                ;   b := $6FFFFFF
====                                ;   b := $7000000
====                                ;   b := $7000001
====                                ;   b := $7000002
====                                ;   b := $7FFFFFE
====                                ;   b := $7FFFFFF
====                                ;   b := $8000000
====                                ;   b := $8000001
====                                ;   b := $8000002
====                                ;   b := $8FFFFFE
====                                ;   b := $8FFFFFF
====                                ;   b := $9000000
====                                ;   b := $9000001
====                                ;   b := $9000002
====                                ;   b := $9FFFFFE
====                                ;   b := $9FFFFFF
====                                ;   b := $A000000
====                                ;   b := $A000001
====                                ;   b := $A000002
====                                ;   b := $AFFFFFE
====                                ;   b := $AFFFFFF
====                                ;   b := $B000000
====                                ;   b := $B000001
====                                ;   b := $B000002
====                                ;   b := $BFFFFFE
====                                ;   b := $BFFFFFF
====                                ;   b := $C000000
====                                ;   b := $C000001
====                                ;   b := $C000002
====                                ;   b := $CFFFFFE
====                                ;   b := $CFFFFFF
====                                ;   b := $D000000
====                                ;   b := $D000001
====                                ;   b := $D000002
====                                ;   b := $DFFFFFE
====                                ;   b := $DFFFFFF
====                                ;   b := $E000000
====                                ;   b := $E000001
====                                ;   b := $E000002
====                                ;   b := $EFFFFFE
```

```
====                                ;   b := $EFFFFFF
====                                ;   b := $F000000
====                                ;   b := $F000001
====                                ;   b := $F000002
====                                ;   b := $0FFFFFFE
====                                ;   b := $0FFFFFFF
====                                ;   b := $10000000
====                                ;   b := $10000001
====                                ;   b := $10000002
====                                ;   b := $1FFFFFFE
====                                ;   b := $1FFFFFFF
====                                ;   b := $20000000
====                                ;   b := $20000001
====                                ;   b := $20000002
====                                ;   b := $2FFFFFFE
====                                ;   b := $2FFFFFFF
====                                ;   b := $30000000
====                                ;   b := $30000001
====                                ;   b := $30000002
====                                ;   b := $3FFFFFFE
====                                ;   b := $3FFFFFFF
====                                ;   b := $40000000
====                                ;   b := $40000001
====                                ;   b := $40000002
====                                ;   b := $4FFFFFFE
====                                ;   b := $4FFFFFFF
====                                ;   b := $50000000
====                                ;   b := $50000001
====                                ;   b := $50000002
====                                ;   b := $5FFFFFFE
====                                ;   b := $5FFFFFFF
====                                ;   b := $60000000
====                                ;   b := $60000001
====                                ;   b := $60000002
====                                ;   b := $6FFFFFFE
====                                ;   b := $6FFFFFFF
====                                ;   b := $70000000
====                                ;   b := $70000001
====                                ;   b := $70000002
====                                ;   b := $7FFFFFFE
====                                ;   b := $7FFFFFFF
====                                ;   b := $80000000
====                                ;   b := $80000001
====                                ;   b := $80000002
====                                ;   b := $8FFFFFFE
====                                ;   b := $8FFFFFFF
====                                ;   b := $90000000
====                                ;   b := $90000001
====                                ;   b := $90000002
====                                ;   b := $9FFFFFFE
====                                ;   b := $9FFFFFFF
====                                ;   b := $A0000000
====                                ;   b := $A0000001
====                                ;   b := $A0000002
====                                ;   b := $AFFFFFFE
====                                ;   b := $AFFFFFFF
====                                ;   b := $B0000000
====                                ;   b := $B0000001
====                                ;   b := $B0000002
====                                ;   b := $BFFFFFFE
====                                ;   b := $BFFFFFFF
====                                ;   b := $C0000000
====                                ;   b := $C0000001
====                                ;   b := $C0000002
====                                ;   b := $CFFFFFFE
====                                ;   b := $CFFFFFFF
====                                ;   b := $D0000000
====                                ;   b := $D0000001
====                                ;   b := $D0000002
====                                ;   b := $DFFFFFFE
====                                ;   b := $DFFFFFFF
====                                ;   b := $E0000000
====                                ;   b := $E0000001
====                                ;   b := $E0000002
====                                ;   b := $EFFFFFFE
====                                ;   b := $EFFFFFFF
====                                ;   b := $F0000000
====                                ;   b := $F0000001
====                                ;   b := $F0000002
====                                ;   b := $FFFFFFFE
====                                ;   b := $FFFFFFFF

0675        34 89 0C        S76:    LET     B174.BYTE, -1
0678        35 89 0C                LET     B174.BYTE, 0
067B        37 1E 89 0C             LET     B174.BYTE, $80000000
067F        37 3E 89 0C             LET     B174.BYTE, $7FFFFFFF
0683        35 89 0C                LET     B174.BYTE, 0
0686        36 89 0C                LET     B174.BYTE, 1
0689        37 00 89 0C             LET     B174.BYTE, 2
```

```
068D    38 0E 89 0C             LET     B174.BYTE, 14
0691    37 23 89 0C             LET     B174.BYTE, 15
0695    37 03 89 0C             LET     B174.BYTE, $10
0699    38 11 89 0C             LET     B174.BYTE, 17
069D    38 12 89 0C             LET     B174.BYTE, 18
06A1    38 1E 89 0C             LET     B174.BYTE, $1E
06A5    37 24 89 0C             LET     B174.BYTE, $1F
06A9    37 04 89 0C             LET     B174.BYTE, $20
06AD    38 21 89 0C             LET     B174.BYTE, 33
06B1    38 22 89 0C             LET     B174.BYTE, 34
06B5    38 2E 89 0C             LET     B174.BYTE, 46
06B9    38 2F 89 0C             LET     B174.BYTE, 47
06BD    38 30 89 0C             LET     B174.BYTE, $30
06C1    38 31 89 0C             LET     B174.BYTE, 49
06C5    38 32 89 0C             LET     B174.BYTE, 50
06C9    38 3E 89 0C             LET     B174.BYTE, $3E
06CD    37 25 89 0C             LET     B174.BYTE, $3F
06D1    37 05 89 0C             LET     B174.BYTE, $40
06D5    38 41 89 0C             LET     B174.BYTE, 65
06D9    38 42 89 0C             LET     B174.BYTE, 66
06DD    38 4E 89 0C             LET     B174.BYTE, 78
06E1    38 4F 89 0C             LET     B174.BYTE, 79
06E5    38 50 89 0C             LET     B174.BYTE, 80
06E9    38 51 89 0C             LET     B174.BYTE, 81
06ED    38 52 89 0C             LET     B174.BYTE, 82
06F1    38 5E 89 0C             LET     B174.BYTE, 94
06F5    38 5F 89 0C             LET     B174.BYTE, 95
06F9    38 60 89 0C             LET     B174.BYTE, $60
06FD    38 61 89 0C             LET     B174.BYTE, 97
0701    38 62 89 0C             LET     B174.BYTE, 98
0705    38 6E 89 0C             LET     B174.BYTE, 110
0709    38 6F 89 0C             LET     B174.BYTE, 111
070D    38 70 89 0C             LET     B174.BYTE, $70
0711    38 71 89 0C             LET     B174.BYTE, 113
0715    38 72 89 0C             LET     B174.BYTE, 114
0719    38 7E 89 0C             LET     B174.BYTE, $7E
071D    37 26 89 0C             LET     B174.BYTE, $7F
0721    37 06 89 0C             LET     B174.BYTE, $80
0725    38 81 89 0C             LET     B174.BYTE, 129
0729    38 82 89 0C             LET     B174.BYTE, 130
072D    38 8E 89 0C             LET     B174.BYTE, 142
0731    38 8F 89 0C             LET     B174.BYTE, 143
0735    38 90 89 0C             LET     B174.BYTE, 144
0739    38 91 89 0C             LET     B174.BYTE, 145
073D    38 92 89 0C             LET     B174.BYTE, 146
0741    38 9E 89 0C             LET     B174.BYTE, 158
0745    38 9F 89 0C             LET     B174.BYTE, 159
0749    38 A0 89 0C             LET     B174.BYTE, 160
074D    38 A1 89 0C             LET     B174.BYTE, 161
0751    38 A2 89 0C             LET     B174.BYTE, 162
0755    38 AE 89 0C             LET     B174.BYTE, 174
0759    38 AF 89 0C             LET     B174.BYTE, 175
075D    38 B0 89 0C             LET     B174.BYTE, 176
0761    38 B1 89 0C             LET     B174.BYTE, 177
0765    38 B2 89 0C             LET     B174.BYTE, 178
0769    38 BE 89 0C             LET     B174.BYTE, 190
076D    38 BF 89 0C             LET     B174.BYTE, 191
0771    38 C0 89 0C             LET     B174.BYTE, $C0
0775    38 C1 89 0C             LET     B174.BYTE, 193
0779    38 C2 89 0C             LET     B174.BYTE, 194
077D    38 CE 89 0C             LET     B174.BYTE, 206
0781    38 CF 89 0C             LET     B174.BYTE, 207
0785    38 D0 89 0C             LET     B174.BYTE, 208
0789    38 D1 89 0C             LET     B174.BYTE, 209
078D    38 D2 89 0C             LET     B174.BYTE, 210
0791    38 DE 89 0C             LET     B174.BYTE, 222
0795    38 DF 89 0C             LET     B174.BYTE, 223
0799    38 E0 89 0C             LET     B174.BYTE, $E0
079D    38 E1 89 0C             LET     B174.BYTE, 225
07A1    38 E2 89 0C             LET     B174.BYTE, 226
07A5    38 EE 89 0C             LET     B174.BYTE, 238
07A9    38 EF 89 0C             LET     B174.BYTE, 239
07AD    38 F0 89 0C             LET     B174.BYTE, $F0
07B1    38 F1 89 0C             LET     B174.BYTE, 241
07B5    38 F2 89 0C             LET     B174.BYTE, 242
07B9    38 FE 89 0C             LET     B174.BYTE, $FE
07BD    37 27 89 0C             LET     B174.BYTE, $FF
07C1    37 07 89 0C             LET     B174.BYTE, $0100
07C5    39 01 01 89 0C          LET     B174.BYTE, 257
07CA    39 01 02 89 0C          LET     B174.BYTE, 258
07CF    39 01 FE 89 0C          LET     B174.BYTE, $01FE
07D4    37 28 89 0C             LET     B174.BYTE, $01FF
07D8    37 08 89 0C             LET     B174.BYTE, $0200
07DC    39 02 01 89 0C          LET     B174.BYTE, 513
07E1    39 02 02 89 0C          LET     B174.BYTE, 514
07E6    39 02 FE 89 0C          LET     B174.BYTE, 766
07EB    39 02 FF 89 0C          LET     B174.BYTE, 767
07F0    39 03 00 89 0C          LET     B174.BYTE, $0300
07F5    39 03 01 89 0C          LET     B174.BYTE, 769
```

```
07FA        39 03 02 89 0C          LET     B174.BYTE, 770
07FF        39 03 FE 89 0C          LET     B174.BYTE, $03FE
0804        37 29 89 0C             LET     B174.BYTE, $03FF
0808        37 09 89 0C             LET     B174.BYTE, $0400
080C        39 04 01 89 0C          LET     B174.BYTE, 1025
0811        39 04 02 89 0C          LET     B174.BYTE, 1026
0816        39 04 FE 89 0C          LET     B174.BYTE, 1278
081B        39 04 FF 89 0C          LET     B174.BYTE, 1279
0820        39 05 00 89 0C          LET     B174.BYTE, 1280
0825        39 05 01 89 0C          LET     B174.BYTE, 1281
082A        39 05 02 89 0C          LET     B174.BYTE, 1282
082F        39 05 FE 89 0C          LET     B174.BYTE, 1534
0834        39 05 FF 89 0C          LET     B174.BYTE, 1535
0839        39 06 00 89 0C          LET     B174.BYTE, $0600
083E        39 06 01 89 0C          LET     B174.BYTE, 1537
0843        39 06 02 89 0C          LET     B174.BYTE, 1538
0848        39 06 FE 89 0C          LET     B174.BYTE, 1790
084D        39 06 FF 89 0C          LET     B174.BYTE, 1791
0852        39 07 00 89 0C          LET     B174.BYTE, $0700
0857        39 07 01 89 0C          LET     B174.BYTE, 1793
085C        39 07 02 89 0C          LET     B174.BYTE, 1794
0861        39 07 FE 89 0C          LET     B174.BYTE, $07FE
0866        37 2A 89 0C             LET     B174.BYTE, $07FF
086A        37 0A 89 0C             LET     B174.BYTE, $0800
086E        39 08 01 89 0C          LET     B174.BYTE, 2049
0873        39 08 02 89 0C          LET     B174.BYTE, 2050
0878        39 08 FE 89 0C          LET     B174.BYTE, 2302
087D        39 08 FF 89 0C          LET     B174.BYTE, 2303
0882        39 09 00 89 0C          LET     B174.BYTE, 2304
0887        39 09 01 89 0C          LET     B174.BYTE, 2305
088C        39 09 02 89 0C          LET     B174.BYTE, 2306
0891        39 09 FE 89 0C          LET     B174.BYTE, 2558
0896        39 09 FF 89 0C          LET     B174.BYTE, 2559
089B        39 0A 00 89 0C          LET     B174.BYTE, 2560
08A0        39 0A 01 89 0C          LET     B174.BYTE, 2561
08A5        39 0A 02 89 0C          LET     B174.BYTE, 2562
08AA        39 0A FE 89 0C          LET     B174.BYTE, 2814
08AF        39 0A FF 89 0C          LET     B174.BYTE, 2815
08B4        39 0B 00 89 0C          LET     B174.BYTE, 2816
08B9        39 0B 01 89 0C          LET     B174.BYTE, 2817
08BE        39 0B 02 89 0C          LET     B174.BYTE, 2818
08C3        39 0B FE 89 0C          LET     B174.BYTE, 3070
08C8        39 0B FF 89 0C          LET     B174.BYTE, 3071
08CD        39 0C 00 89 0C          LET     B174.BYTE, $0C00
08D2        39 0C 01 89 0C          LET     B174.BYTE, 3073
08D7        39 0C 02 89 0C          LET     B174.BYTE, 3074
08DC        39 0C FE 89 0C          LET     B174.BYTE, 3326
08E1        39 0C FF 89 0C          LET     B174.BYTE, 3327
08E6        39 0D 00 89 0C          LET     B174.BYTE, 3328
08EB        39 0D 01 89 0C          LET     B174.BYTE, 3329
08F0        39 0D 02 89 0C          LET     B174.BYTE, 3330
08F5        39 0D FE 89 0C          LET     B174.BYTE, 3582
08FA        39 0D FF 89 0C          LET     B174.BYTE, 3583
08FF        39 0E 00 89 0C          LET     B174.BYTE, $0E00
0904        39 0E 01 89 0C          LET     B174.BYTE, 3585
0909        39 0E 02 89 0C          LET     B174.BYTE, 3586
090E        39 0E FE 89 0C          LET     B174.BYTE, 3838
0913        39 0E FF 89 0C          LET     B174.BYTE, 3839
0918        39 0F 00 89 0C          LET     B174.BYTE, $0F00
091D        39 0F 01 89 0C          LET     B174.BYTE, 3841
0922        39 0F 02 89 0C          LET     B174.BYTE, 3842
0927        39 0F FE 89 0C          LET     B174.BYTE, $0FFE
092C        37 2B 89 0C             LET     B174.BYTE, $0FFF
0930        37 0B 89 0C             LET     B174.BYTE, $1000
0934        39 10 01 89 0C          LET     B174.BYTE, 4097
0939        39 10 02 89 0C          LET     B174.BYTE, 4098
093E        39 1F FE 89 0C          LET     B174.BYTE, $1FFE
0943        37 2C 89 0C             LET     B174.BYTE, $1FFF
0947        37 0C 89 0C             LET     B174.BYTE, $2000
094B        39 20 01 89 0C          LET     B174.BYTE, 8193
0950        39 20 02 89 0C          LET     B174.BYTE, 8194
0955        39 2F FE 89 0C          LET     B174.BYTE, 12286
095A        39 2F FF 89 0C          LET     B174.BYTE, 12287
095F        39 30 00 89 0C          LET     B174.BYTE, $3000
0964        39 30 01 89 0C          LET     B174.BYTE, 12289
0969        39 30 02 89 0C          LET     B174.BYTE, 12290
096E        39 3F FE 89 0C          LET     B174.BYTE, $3FFE
0973        37 2D 89 0C             LET     B174.BYTE, $3FFF
0977        37 0D 89 0C             LET     B174.BYTE, $4000
097B        39 40 01 89 0C          LET     B174.BYTE, 16385
0980        39 40 02 89 0C          LET     B174.BYTE, 16386
0985        39 4F FE 89 0C          LET     B174.BYTE, 20478
098A        39 4F FF 89 0C          LET     B174.BYTE, 20479
098F        39 50 00 89 0C          LET     B174.BYTE, 20480
0994        39 50 01 89 0C          LET     B174.BYTE, 20481
0999        39 50 02 89 0C          LET     B174.BYTE, 20482
099E        39 5F FE 89 0C          LET     B174.BYTE, 24574
09A3        39 5F FF 89 0C          LET     B174.BYTE, 24575
09A8        39 60 00 89 0C          LET     B174.BYTE, $6000
```

```
09AD        39 60 01 89 0C          LET     B174.BYTE, 24577
09B2        39 60 02 89 0C          LET     B174.BYTE, 24578
09B7        39 6F FE 89 0C          LET     B174.BYTE, 28670
09BC        39 6F FF 89 0C          LET     B174.BYTE, 28671
09C1        39 70 00 89 0C          LET     B174.BYTE, $7000
09C6        39 70 01 89 0C          LET     B174.BYTE, 28673
09CB        39 70 02 89 0C          LET     B174.BYTE, 28674
09D0        39 7F FE 89 0C          LET     B174.BYTE, $7FFE
09D5        37 2E 89 0C             LET     B174.BYTE, $7FFF
09D9        37 0E 89 0C             LET     B174.BYTE, $8000
09DD        39 80 01 89 0C          LET     B174.BYTE, 32769
09E2        39 80 02 89 0C          LET     B174.BYTE, 32770
09E7        39 8F FE 89 0C          LET     B174.BYTE, 36862
09EC        39 8F FF 89 0C          LET     B174.BYTE, 36863
09F1        39 90 00 89 0C          LET     B174.BYTE, 36864
09F6        39 90 01 89 0C          LET     B174.BYTE, 36865
09FB        39 90 02 89 0C          LET     B174.BYTE, 36866
0A00        39 9F FE 89 0C          LET     B174.BYTE, 40958
0A05        39 9F FF 89 0C          LET     B174.BYTE, 40959
0A0A        39 A0 00 89 0C          LET     B174.BYTE, 40960
0A0F        39 A0 01 89 0C          LET     B174.BYTE, 40961
0A14        39 A0 02 89 0C          LET     B174.BYTE, 40962
0A19        39 AF FE 89 0C          LET     B174.BYTE, 45054
0A1E        39 AF FF 89 0C          LET     B174.BYTE, 45055
0A23        39 B0 00 89 0C          LET     B174.BYTE, 45056
0A28        39 B0 01 89 0C          LET     B174.BYTE, 45057
0A2D        39 B0 02 89 0C          LET     B174.BYTE, 45058
0A32        39 BF FE 89 0C          LET     B174.BYTE, 49150
0A37        39 BF FF 89 0C          LET     B174.BYTE, 49151
0A3C        39 C0 00 89 0C          LET     B174.BYTE, $C000
0A41        39 C0 01 89 0C          LET     B174.BYTE, 49153
0A46        39 C0 02 89 0C          LET     B174.BYTE, 49154
0A4B        39 CF FE 89 0C          LET     B174.BYTE, 53246
0A50        39 CF FF 89 0C          LET     B174.BYTE, 53247
0A55        39 D0 00 89 0C          LET     B174.BYTE, 53248
0A5A        39 D0 01 89 0C          LET     B174.BYTE, 53249
0A5F        39 D0 02 89 0C          LET     B174.BYTE, 53250
0A64        39 DF FE 89 0C          LET     B174.BYTE, 57342
0A69        39 DF FF 89 0C          LET     B174.BYTE, 57343
0A6E        39 E0 00 89 0C          LET     B174.BYTE, $E000
0A73        39 E0 01 89 0C          LET     B174.BYTE, 57345
0A78        39 E0 02 89 0C          LET     B174.BYTE, 57346
0A7D        39 EF FE 89 0C          LET     B174.BYTE, 61438
0A82        39 EF FF 89 0C          LET     B174.BYTE, 61439
0A87        39 F0 00 89 0C          LET     B174.BYTE, $F000
0A8C        39 F0 01 89 0C          LET     B174.BYTE, 61441
0A91        39 F0 02 89 0C          LET     B174.BYTE, 61442
0A96        39 FF FE 89 0C          LET     B174.BYTE, $FFFE
0A9B        37 2F 89 0C             LET     B174.BYTE, $FFFF
0A9F        37 0F 89 0C             LET     B174.BYTE, $010000
0AA3        3A 01 00 01 89          LET     B174.BYTE, 65537
0AA8         + 0C
0AA9        3A 01 00 02 89          LET     B174.BYTE, 65538
0AAE         + 0C
0AAF        3A 01 FF FE 89          LET     B174.BYTE, $01FFFE
0AB4         + 0C
0AB5        37 30 89 0C             LET     B174.BYTE, $01FFFF
0AB9        37 10 89 0C             LET     B174.BYTE, $020000
0ABD        3A 02 00 01 89          LET     B174.BYTE, 131073
0AC2         + 0C
0AC3        3A 02 00 02 89          LET     B174.BYTE, 131074
0AC8         + 0C
0AC9        3A 02 FF FE 89          LET     B174.BYTE, 196606
0ACE         + 0C
0ACF        3A 02 FF FF 89          LET     B174.BYTE, 196607
0AD4         + 0C
0AD5        3A 03 00 00 89          LET     B174.BYTE, $030000
0ADA         + 0C
0ADB        3A 03 00 01 89          LET     B174.BYTE, 196609
0AE0         + 0C
0AE1        3A 03 00 02 89          LET     B174.BYTE, 196610
0AE6         + 0C
0AE7        3A 03 FF FE 89          LET     B174.BYTE, $03FFFE
0AEC         + 0C
0AED        37 31 89 0C             LET     B174.BYTE, $03FFFF
0AF1        37 11 89 0C             LET     B174.BYTE, $040000
0AF5        3A 04 00 01 89          LET     B174.BYTE, 262145
0AFA         + 0C
0AFB        3A 04 00 02 89          LET     B174.BYTE, 262146
0B00         + 0C
0B01        3A 04 FF FE 89          LET     B174.BYTE, 327678
0B06         + 0C
0B07        3A 04 FF FF 89          LET     B174.BYTE, 327679
0B0C         + 0C
0B0D        3A 05 00 00 89          LET     B174.BYTE, 327680
0B12         + 0C
0B13        3A 05 00 01 89          LET     B174.BYTE, 327681
0B18         + 0C
0B19        3A 05 00 02 89          LET     B174.BYTE, 327682
```

```
0B1E          + 0C
0B1F          3A 05 FF FE 89          LET     B174.BYTE, 393214
0B24          + 0C
0B25          3A 05 FF FF 89          LET     B174.BYTE, 393215
0B2A          + 0C
0B2B          3A 06 00 00 89          LET     B174.BYTE, $060000
0B30          + 0C
0B31          3A 06 00 01 89          LET     B174.BYTE, 393217
0B36          + 0C
0B37          3A 06 00 02 89          LET     B174.BYTE, 393218
0B3C          + 0C
0B3D          3A 06 FF FE 89          LET     B174.BYTE, 458750
0B42          + 0C
0B43          3A 06 FF FF 89          LET     B174.BYTE, 458751
0B48          + 0C
0B49          3A 07 00 00 89          LET     B174.BYTE, $070000
0B4E          + 0C
0B4F          3A 07 00 01 89          LET     B174.BYTE, 458753
0B54          + 0C
0B55          3A 07 00 02 89          LET     B174.BYTE, 458754
0B5A          + 0C
0B5B          3A 07 FF FE 89          LET     B174.BYTE, $07FFFE
0B60          + 0C
0B61          37 32 89 0C             LET     B174.BYTE, $07FFFF
0B65          37 12 89 0C             LET     B174.BYTE, $080000
0B69          3A 08 00 01 89          LET     B174.BYTE, 524289
0B6E          + 0C
0B6F          3A 08 00 02 89          LET     B174.BYTE, 524290
0B74          + 0C
0B75          3A 08 FF FE 89          LET     B174.BYTE, 589822
0B7A          + 0C
0B7B          3A 08 FF FF 89          LET     B174.BYTE, 589823
0B80          + 0C
0B81          3A 09 00 00 89          LET     B174.BYTE, 589824
0B86          + 0C
0B87          3A 09 00 01 89          LET     B174.BYTE, 589825
0B8C          + 0C
0B8D          3A 09 00 02 89          LET     B174.BYTE, 589826
0B92          + 0C
0B93          3A 09 FF FE 89          LET     B174.BYTE, 655358
0B98          + 0C
0B99          3A 09 FF FF 89          LET     B174.BYTE, 655359
0B9E          + 0C
0B9F          3A 0A 00 00 89          LET     B174.BYTE, 655360
0BA4          + 0C
0BA5          3A 0A 00 01 89          LET     B174.BYTE, 655361
0BAA          + 0C
0BAB          3A 0A 00 02 89          LET     B174.BYTE, 655362
0BB0          + 0C
0BB1          3A 0A FF FE 89          LET     B174.BYTE, 720894
0BB6          + 0C
0BB7          3A 0A FF FF 89          LET     B174.BYTE, 720895
0BBC          + 0C
0BBD          3A 0B 00 00 89          LET     B174.BYTE, 720896
0BC2          + 0C
0BC3          3A 0B 00 01 89          LET     B174.BYTE, 720897
0BC8          + 0C
0BC9          3A 0B 00 02 89          LET     B174.BYTE, 720898
0BCE          + 0C
0BCF          3A 0B FF FE 89          LET     B174.BYTE, 786430
0BD4          + 0C
0BD5          3A 0B FF FF 89          LET     B174.BYTE, 786431
0BDA          + 0C
0BDB          3A 0C 00 00 89          LET     B174.BYTE, $0C0000
0BE0          + 0C
0BE1          3A 0C 00 01 89          LET     B174.BYTE, 786433
0BE6          + 0C
0BE7          3A 0C 00 02 89          LET     B174.BYTE, 786434
0BEC          + 0C
0BED          3A 0C FF FE 89          LET     B174.BYTE, 851966
0BF2          + 0C
0BF3          3A 0C FF FF 89          LET     B174.BYTE, 851967
0BF8          + 0C
0BF9          3A 0D 00 00 89          LET     B174.BYTE, 851968
0BFE          + 0C
0BFF          3A 0D 00 01 89          LET     B174.BYTE, 851969
0C04          + 0C
0C05          3A 0D 00 02 89          LET     B174.BYTE, 851970
0C0A          + 0C
0C0B          3A 0D FF FE 89          LET     B174.BYTE, 917502
0C10          + 0C
0C11          3A 0D FF FF 89          LET     B174.BYTE, 917503
0C16          + 0C
0C17          3A 0E 00 00 89          LET     B174.BYTE, $0E0000
0C1C          + 0C
0C1D          3A 0E 00 01 89          LET     B174.BYTE, 917505
0C22          + 0C
0C23          3A 0E 00 02 89          LET     B174.BYTE, 917506
0C28          + 0C
```

```
0C29        3A 0E FF FE 89          LET     B174.BYTE, 983038
0C2E         + 0C
0C2F        3A 0E FF FF 89          LET     B174.BYTE, 983039
0C34         + 0C
0C35        3A 0F 00 00 89          LET     B174.BYTE, $0F0000
0C3A         + 0C
0C3B        3A 0F 00 01 89          LET     B174.BYTE, 983041
0C40         + 0C
0C41        3A 0F 00 02 89          LET     B174.BYTE, 983042
0C46         + 0C
0C47        3A 0F FF FE 89          LET     B174.BYTE, $0FFFFE
0C4C         + 0C
0C4D        37 33 89 0C             LET     B174.BYTE, $0FFFFF
0C51        37 13 89 0C             LET     B174.BYTE, $100000
0C55        3A 10 00 01 89          LET     B174.BYTE, 1048577
0C5A         + 0C
0C5B        3A 10 00 02 89          LET     B174.BYTE, 1048578
0C60         + 0C
0C61        3A 1F FF FE 89          LET     B174.BYTE, $1FFFFE
0C66         + 0C
0C67        37 34 89 0C             LET     B174.BYTE, $1FFFFF
0C6B        37 14 89 0C             LET     B174.BYTE, $200000
0C6F        3A 20 00 01 89          LET     B174.BYTE, 2097153
0C74         + 0C
0C75        3A 20 00 02 89          LET     B174.BYTE, 2097154
0C7A         + 0C
0C7B        3A 2F FF FE 89          LET     B174.BYTE, 3145726
0C80         + 0C
0C81        3A 2F FF FF 89          LET     B174.BYTE, 3145727
0C86         + 0C
0C87        3A 30 00 00 89          LET     B174.BYTE, $300000
0C8C         + 0C
0C8D        3A 30 00 01 89          LET     B174.BYTE, 3145729
0C92         + 0C
0C93        3A 30 00 02 89          LET     B174.BYTE, 3145730
0C98         + 0C
0C99        3A 3F FF FE 89          LET     B174.BYTE, $3FFFFE
0C9E         + 0C
0C9F        37 35 89 0C             LET     B174.BYTE, $3FFFFF
0CA3        37 15 89 0C             LET     B174.BYTE, $400000
0CA7        3A 40 00 01 89          LET     B174.BYTE, 4194305
0CAC         + 0C
0CAD        3A 40 00 02 89          LET     B174.BYTE, 4194306
0CB2         + 0C
0CB3        3A 4F FF FE 89          LET     B174.BYTE, 5242878
0CB8         + 0C
0CB9        3A 4F FF FF 89          LET     B174.BYTE, 5242879
0CBE         + 0C
0CBF        3A 50 00 00 89          LET     B174.BYTE, 5242880
0CC4         + 0C
0CC5        3A 50 00 01 89          LET     B174.BYTE, 5242881
0CCA         + 0C
0CCB        3A 50 00 02 89          LET     B174.BYTE, 5242882
0CD0         + 0C
0CD1        3A 5F FF FE 89          LET     B174.BYTE, 6291454
0CD6         + 0C
0CD7        3A 5F FF FF 89          LET     B174.BYTE, 6291455
0CDC         + 0C
0CDD        3A 60 00 00 89          LET     B174.BYTE, $600000
0CE2         + 0C
0CE3        3A 60 00 01 89          LET     B174.BYTE, 6291457
0CE8         + 0C
0CE9        3A 60 00 02 89          LET     B174.BYTE, 6291458
0CEE         + 0C
0CEF        3A 6F FF FE 89          LET     B174.BYTE, 7340030
0CF4         + 0C
0CF5        3A 6F FF FF 89          LET     B174.BYTE, 7340031
0CFA         + 0C
0CFB        3A 70 00 00 89          LET     B174.BYTE, $700000
0D00         + 0C
0D01        3A 70 00 01 89          LET     B174.BYTE, 7340033
0D06         + 0C
0D07        3A 70 00 02 89          LET     B174.BYTE, 7340034
0D0C         + 0C
0D0D        3A 7F FF FE 89          LET     B174.BYTE, $7FFFFE
0D12         + 0C
0D13        37 36 89 0C             LET     B174.BYTE, $7FFFFF
0D17        37 16 89 0C             LET     B174.BYTE, $800000
0D1B        3A 80 00 01 89          LET     B174.BYTE, 8388609
0D20         + 0C
0D21        3A 80 00 02 89          LET     B174.BYTE, 8388610
0D26         + 0C
0D27        3A 8F FF FE 89          LET     B174.BYTE, 9437182
0D2C         + 0C
0D2D        3A 8F FF FF 89          LET     B174.BYTE, 9437183
0D32         + 0C
0D33        3A 90 00 00 89          LET     B174.BYTE, 9437184
0D38         + 0C
0D39        3A 90 00 01 89          LET     B174.BYTE, 9437185
```

```
0D3E          + 0C
0D3F          3A 90 00 02 89          LET     B174.BYTE, 9437186
0D44          + 0C
0D45          3A 9F FF FE 89          LET     B174.BYTE, 10485758
0D4A          + 0C
0D4B          3A 9F FF FF 89          LET     B174.BYTE, 10485759
0D50          + 0C
0D51          3A A0 00 00 89          LET     B174.BYTE, 10485760
0D56          + 0C
0D57          3A A0 00 01 89          LET     B174.BYTE, 10485761
0D5C          + 0C
0D5D          3A A0 00 02 89          LET     B174.BYTE, 10485762
0D62          + 0C
0D63          3A AF FF FE 89          LET     B174.BYTE, 11534334
0D68          + 0C
0D69          3A AF FF FF 89          LET     B174.BYTE, 11534335
0D6E          + 0C
0D6F          3A B0 00 00 89          LET     B174.BYTE, 11534336
0D74          + 0C
0D75          3A B0 00 01 89          LET     B174.BYTE, 11534337
0D7A          + 0C
0D7B          3A B0 00 02 89          LET     B174.BYTE, 11534338
0D80          + 0C
0D81          3A BF FF FE 89          LET     B174.BYTE, 12582910
0D86          + 0C
0D87          3A BF FF FF 89          LET     B174.BYTE, 12582911
0D8C          + 0C
0D8D          3A C0 00 00 89          LET     B174.BYTE, $C00000
0D92          + 0C
0D93          3A C0 00 01 89          LET     B174.BYTE, 12582913
0D98          + 0C
0D99          3A C0 00 02 89          LET     B174.BYTE, 12582914
0D9E          + 0C
0D9F          3A CF FF FE 89          LET     B174.BYTE, 13631486
0DA4          + 0C
0DA5          3A CF FF FF 89          LET     B174.BYTE, 13631487
0DAA          + 0C
0DAB          3A D0 00 00 89          LET     B174.BYTE, 13631488
0DB0          + 0C
0DB1          3A D0 00 01 89          LET     B174.BYTE, 13631489
0DB6          + 0C
0DB7          3A D0 00 02 89          LET     B174.BYTE, 13631490
0DBC          + 0C
0DBD          3A DF FF FE 89          LET     B174.BYTE, 14680062
0DC2          + 0C
0DC3          3A DF FF FF 89          LET     B174.BYTE, 14680063
0DC8          + 0C
0DC9          3A E0 00 00 89          LET     B174.BYTE, $E00000
0DCE          + 0C
0DCF          3A E0 00 01 89          LET     B174.BYTE, 14680065
0DD4          + 0C
0DD5          3A E0 00 02 89          LET     B174.BYTE, 14680066
0DDA          + 0C
0DDB          3A EF FF FF 89          LET     B174.BYTE, 15728639
0DE0          + 0C
0DE1          3A EF FF FF 89          LET     B174.BYTE, 15728639
0DE6          + 0C
0DE7          3A F0 00 00 89          LET     B174.BYTE, $F00000
0DEC          + 0C
0DED          3A F0 00 01 89          LET     B174.BYTE, 15728641
0DF2          + 0C
0DF3          3A F0 00 02 89          LET     B174.BYTE, 15728642
0DF8          + 0C
0DF9          3A FF FF FE 89          LET     B174.BYTE, $FFFFFE
0DFE          + 0C
0DFF          37 37 89 0C             LET     B174.BYTE, $FFFFFF
0E03          37 17 89 0C             LET     B174.BYTE, $01000000
0E07          3B 01 00 00 01          LET     B174.BYTE, 16777217
0E0C          + 89 0C
0E0E          3B 01 00 00 02          LET     B174.BYTE, 16777218
0E13          + 89 0C
0E15          3B 01 FF FF FE          LET     B174.BYTE, $01FFFFFE
0E1A          + 89 0C
0E1C          37 38 89 0C             LET     B174.BYTE, $01FFFFFF
0E20          37 18 89 0C             LET     B174.BYTE, $02000000
0E24          3B 02 00 00 01          LET     B174.BYTE, 33554433
0E29          + 89 0C
0E2B          3B 02 00 00 02          LET     B174.BYTE, 33554434
0E30          + 89 0C
0E32          3B 02 FF FF FE          LET     B174.BYTE, 50331646
0E37          + 89 0C
0E39          3B 02 FF FF FF          LET     B174.BYTE, 50331647
0E3E          + 89 0C
0E40          3B 03 00 00 00          LET     B174.BYTE, $03000000
0E45          + 89 0C
0E47          3B 03 00 00 01          LET     B174.BYTE, 50331649
0E4C          + 89 0C
0E4E          3B 03 00 00 02          LET     B174.BYTE, 50331650
0E53          + 89 0C
```

```
0E55        3B 03 FF FF FE          LET     B174.BYTE, $03FFFFFE
0E5A         + 89 0C
0E5C        37 39 89 0C             LET     B174.BYTE, $03FFFFFF
0E60        37 19 89 0C             LET     B174.BYTE, $04000000
0E64        3B 04 00 00 01          LET     B174.BYTE, 67108865
0E69         + 89 0C
0E6B        3B 04 00 00 02          LET     B174.BYTE, 67108866
0E70         + 89 0C
0E72        3B 04 FF FF FE          LET     B174.BYTE, 83886078
0E77         + 89 0C
0E79        3B 04 FF FF FF          LET     B174.BYTE, 83886079
0E7E         + 89 0C
0E80        3B 05 00 00 00          LET     B174.BYTE, 83886080
0E85         + 89 0C
0E87        3B 05 00 00 01          LET     B174.BYTE, 83886081
0E8C         + 89 0C
0E8E        3B 05 00 00 02          LET     B174.BYTE, 83886082
0E93         + 89 0C
0E95        3B 05 FF FF FE          LET     B174.BYTE, 100663294
0E9A         + 89 0C
0E9C        3B 05 FF FF FF          LET     B174.BYTE, 100663295
0EA1         + 89 0C
0EA3        3B 06 00 00 00          LET     B174.BYTE, $06000000
0EA8         + 89 0C
0EAA        3B 06 00 00 01          LET     B174.BYTE, 100663297
0EAF         + 89 0C
0EB1        3B 06 00 00 02          LET     B174.BYTE, 100663298
0EB6         + 89 0C
0EB8        3B 06 FF FF FE          LET     B174.BYTE, 117440510
0EBD         + 89 0C
0EBF        3B 06 FF FF FF          LET     B174.BYTE, 117440511
0EC4         + 89 0C
0EC6        3B 07 00 00 00          LET     B174.BYTE, $07000000
0ECB         + 89 0C
0ECD        3B 07 00 00 01          LET     B174.BYTE, 117440513
0ED2         + 89 0C
0ED4        3B 07 00 00 02          LET     B174.BYTE, 117440514
0ED9         + 89 0C
0EDB        3B 07 FF FF FE          LET     B174.BYTE, $07FFFFFE
0EE0         + 89 0C
0EE2        37 3A 89 0C             LET     B174.BYTE, $07FFFFFF
0EE6        37 1A 89 0C             LET     B174.BYTE, $08000000
0EEA        3B 08 00 00 01          LET     B174.BYTE, 134217729
0EEF         + 89 0C
0EF1        3B 08 00 00 02          LET     B174.BYTE, 134217730
0EF6         + 89 0C
0EF8        3B 08 FF FF FE          LET     B174.BYTE, 150994942
0EFD         + 89 0C
0EFF        3B 08 FF FF FF          LET     B174.BYTE, 150994943
0F04         + 89 0C
0F06        3B 09 00 00 00          LET     B174.BYTE, 150994944
0F0B         + 89 0C
0F0D        3B 09 00 00 01          LET     B174.BYTE, 150994945
0F12         + 89 0C
0F14        3B 09 00 00 02          LET     B174.BYTE, 150994946
0F19         + 89 0C
0F1B        3B 09 FF FF FE          LET     B174.BYTE, 167772158
0F20         + 89 0C
0F22        3B 09 FF FF FF          LET     B174.BYTE, 167772159
0F27         + 89 0C
0F29        3B 0A 00 00 00          LET     B174.BYTE, 167772160
0F2E         + 89 0C
0F30        3B 0A 00 00 01          LET     B174.BYTE, 167772161
0F35         + 89 0C
0F37        3B 0A 00 00 02          LET     B174.BYTE, 167772162
0F3C         + 89 0C
0F3E        3B 0A FF FF FE          LET     B174.BYTE, 184549374
0F43         + 89 0C
0F45        3B 0A FF FF FF          LET     B174.BYTE, 184549375
0F4A         + 89 0C
0F4C        3B 0B 00 00 00          LET     B174.BYTE, 184549376
0F51         + 89 0C
0F53        3B 0B 00 00 01          LET     B174.BYTE, 184549377
0F58         + 89 0C
0F5A        3B 0B 00 00 02          LET     B174.BYTE, 184549378
0F5F         + 89 0C
0F61        3B 0B FF FF FE          LET     B174.BYTE, 201326590
0F66         + 89 0C
0F68        3B 0B FF FF FF          LET     B174.BYTE, 201326591
0F6D         + 89 0C
0F6F        3B 0C 00 00 00          LET     B174.BYTE, $0C000000
0F74         + 89 0C
0F76        3B 0C 00 00 01          LET     B174.BYTE, 201326593
0F7B         + 89 0C
0F7D        3B 0C 00 00 02          LET     B174.BYTE, 201326594
0F82         + 89 0C
0F84        3B 0C FF FF FE          LET     B174.BYTE, 218103806
0F89         + 89 0C
0F8B        3B 0C FF FF FF          LET     B174.BYTE, 218103807
```

```
0F90          + 89 0C
0F92          3B 0D 00 00 00          LET     B174.BYTE, 218103808
0F97          + 89 0C
0F99          3B 0D 00 00 01          LET     B174.BYTE, 218103809
0F9E          + 89 0C
0FA0          3B 0D 00 00 02          LET     B174.BYTE, 218103810
0FA5          + 89 0C
0FA7          3B 0D FF FF FE          LET     B174.BYTE, 234881022
0FAC          + 89 0C
0FAE          3B 0D FF FF FF          LET     B174.BYTE, 234881023
0FB3          + 89 0C
0FB5          3B 0E 00 00 00          LET     B174.BYTE, $0E000000
0FBA          + 89 0C
0FBC          3B 0E 00 00 01          LET     B174.BYTE, 234881025
0FC1          + 89 0C
0FC3          3B 0E 00 00 02          LET     B174.BYTE, 234881026
0FC8          + 89 0C
0FCA          3B 0E FF FF FE          LET     B174.BYTE, 251658238
0FCF          + 89 0C
0FD1          3B 0E FF FF FF          LET     B174.BYTE, 251658239
0FD6          + 89 0C
0FD8          3B 0F 00 00 00          LET     B174.BYTE, $0F000000
0FDD          + 89 0C
0FDF          3B 0F 00 00 01          LET     B174.BYTE, 251658241
0FE4          + 89 0C
0FE6          3B 0F 00 00 02          LET     B174.BYTE, 251658242
0FEB          + 89 0C
0FED          3B 0F FF FF FE          LET     B174.BYTE, $0FFFFFFE
0FF2          + 89 0C
0FF4          37 3B 89 0C             LET     B174.BYTE, $0FFFFFFF
0FF8          37 1B 89 0C             LET     B174.BYTE, $10000000
0FFC          3B 10 00 00 01          LET     B174.BYTE, 268435457
1001          + 89 0C
1003          3B 10 00 00 02          LET     B174.BYTE, 268435458
1008          + 89 0C
100A          3B 1F FF FF FE          LET     B174.BYTE, $1FFFFFFE
100F          + 89 0C
1011          37 3C 89 0C             LET     B174.BYTE, $1FFFFFFF
1015          37 1C 89 0C             LET     B174.BYTE, $20000000
1019          3B 20 00 00 01          LET     B174.BYTE, 536870913
101E          + 89 0C
1020          3B 20 00 00 02          LET     B174.BYTE, 536870914
1025          + 89 0C
1027          3B 2F FF FF FE          LET     B174.BYTE, 805306366
102C          + 89 0C
102E          3B 2F FF FF FF          LET     B174.BYTE, 805306367
1033          + 89 0C
1035          3B 30 00 00 00          LET     B174.BYTE, $30000000
103A          + 89 0C
103C          3B 30 00 00 01          LET     B174.BYTE, 805306369
1041          + 89 0C
1043          3B 30 00 00 02          LET     B174.BYTE, 805306370
1048          + 89 0C
104A          3B 3F FF FF FE          LET     B174.BYTE, $3FFFFFFE
104F          + 89 0C
1051          37 3D 89 0C             LET     B174.BYTE, $3FFFFFFF
1055          37 1D 89 0C             LET     B174.BYTE, $40000000
1059          3B 40 00 00 01          LET     B174.BYTE, 1073741825
105E          + 89 0C
1060          3B 40 00 00 02          LET     B174.BYTE, 1073741826
1065          + 89 0C
1067          3B 4F FF FF FE          LET     B174.BYTE, 1342177278
106C          + 89 0C
106E          3B 4F FF FF FF          LET     B174.BYTE, 1342177279
1073          + 89 0C
1075          3B 50 00 00 00          LET     B174.BYTE, 1342177280
107A          + 89 0C
107C          3B 50 00 00 01          LET     B174.BYTE, 1342177281
1081          + 89 0C
1083          3B 50 00 00 02          LET     B174.BYTE, 1342177282
1088          + 89 0C
108A          3B 5F FF FF FE          LET     B174.BYTE, 1610612734
108F          + 89 0C
1091          3B 5F FF FF FF          LET     B174.BYTE, 1610612735
1096          + 89 0C
1098          3B 60 00 00 00          LET     B174.BYTE, $60000000
109D          + 89 0C
109F          3B 60 00 00 01          LET     B174.BYTE, 1610612737
10A4          + 89 0C
10A6          3B 60 00 00 02          LET     B174.BYTE, 1610612738
10AB          + 89 0C
10AD          3B 6F FF FF FE          LET     B174.BYTE, 1879048190
10B2          + 89 0C
10B4          3B 6F FF FF FF          LET     B174.BYTE, 1879048191
10B9          + 89 0C
10BB          3B 70 00 00 00          LET     B174.BYTE, $70000000
10C0          + 89 0C
10C2          3B 70 00 00 01          LET     B174.BYTE, 1879048193
10C7          + 89 0C
```

```
10C9        3B 70 00 00 02          LET     B174.BYTE, 1879048194
10CE         + 89 0C
10D0        3B 7F FF FF FE          LET     B174.BYTE, $7FFFFFFE
10D5         + 89 0C
10D7        37 3E 89 0C             LET     B174.BYTE, $7FFFFFFF
10DB        37 1E 89 0C             LET     B174.BYTE, $80000000
10DF        3B 80 00 00 01          LET     B174.BYTE, -2147483647
10E4         + 89 0C
10E6        3B 80 00 00 02          LET     B174.BYTE, -2147483646
10EB         + 89 0C
10ED        3B 8F FF FF FE          LET     B174.BYTE, -1879048194
10F2         + 89 0C
10F4        3B 8F FF FF FF          LET     B174.BYTE, -1879048193
10F9         + 89 0C
10FB        3B 90 00 00 00          LET     B174.BYTE, -1879048192
1100         + 89 0C
1102        3B 90 00 00 01          LET     B174.BYTE, -1879048191
1107         + 89 0C
1109        3B 90 00 00 02          LET     B174.BYTE, -1879048190
110E         + 89 0C
1110        3B 9F FF FF FE          LET     B174.BYTE, -1610612738
1115         + 89 0C
1117        3B 9F FF FF FF          LET     B174.BYTE, -1610612737
111C         + 89 0C
111E        3B A0 00 00 00          LET     B174.BYTE, -1610612736
1123         + 89 0C
1125        3B A0 00 00 01          LET     B174.BYTE, -1610612735
112A         + 89 0C
112C        3B A0 00 00 02          LET     B174.BYTE, -1610612734
1131         + 89 0C
1133        3B AF FF FF FE          LET     B174.BYTE, -1342177282
1138         + 89 0C
113A        3B AF FF FF FF          LET     B174.BYTE, -1342177281
113F         + 89 0C
1141        3B B0 00 00 00          LET     B174.BYTE, -1342177280
1146         + 89 0C
1148        3B B0 00 00 01          LET     B174.BYTE, -1342177279
114D         + 89 0C
114F        3B B0 00 00 02          LET     B174.BYTE, -1342177278
1154         + 89 0C
1156        3B BF FF FF FE          LET     B174.BYTE, -1073741826
115B         + 89 0C
115D        37 5D 89 0C             LET     B174.BYTE, -1073741825
1161        37 7D 89 0C             LET     B174.BYTE, $C0000000
1165        3B C0 00 00 01          LET     B174.BYTE, -1073741823
116A         + 89 0C
116C        3B C0 00 00 02          LET     B174.BYTE, -1073741822
1171         + 89 0C
1173        3B CF FF FF FE          LET     B174.BYTE, -805306370
1178         + 89 0C
117A        3B CF FF FF FF          LET     B174.BYTE, -805306369
117F         + 89 0C
1181        3B D0 00 00 00          LET     B174.BYTE, -805306368
1186         + 89 0C
1188        3B D0 00 00 01          LET     B174.BYTE, -805306367
118D         + 89 0C
118F        3B D0 00 00 02          LET     B174.BYTE, -805306366
1194         + 89 0C
1196        3B DF FF FF FE          LET     B174.BYTE, -536870914
119B         + 89 0C
119D        37 5C 89 0C             LET     B174.BYTE, -536870913
11A1        37 7C 89 0C             LET     B174.BYTE, $E0000000
11A5        3B E0 00 00 01          LET     B174.BYTE, -536870911
11AA         + 89 0C
11AC        3B E0 00 00 02          LET     B174.BYTE, -536870910
11B1         + 89 0C
11B3        3B EF FF FF FE          LET     B174.BYTE, -268435458
11B8         + 89 0C
11BA        37 5B 89 0C             LET     B174.BYTE, -268435457
11BE        37 7B 89 0C             LET     B174.BYTE, $F0000000
11C2        3B F0 00 00 01          LET     B174.BYTE, -268435455
11C7         + 89 0C
11C9        3B F0 00 00 02          LET     B174.BYTE, -268435454
11CE         + 89 0C
11D0        37 60 89 0C             LET     B174.BYTE, $FFFFFFFE
11D4        34 89 0C                LET     B174.BYTE, -1
11D7        32                      RETURN

====                                ; PRI test_Function_1
====                                ;   Return b

11D8        88 0C 33        F77:    RETURN  B174.BYTE

11DB        32              ; ----  RETURN

====                                ; PRI test_Function_2 : returnValue
====                                ;   returnValue := b

                                    ALIGN   STACK           ; For F78
```

```
      +0000                 VL7:     LONG     0              ; Result Variable

                                     ALIGN    SPIN

11DC          88 0C 61        F78:     LET      VL7.LONG, B174.BYTE
11DF          32                       RETURN

====                                 ; PRI test_Function_3
====                                 ;   Result := b

                                     ALIGN    STACK          ; For F79

      +0000                 VL8:     LONG     0              ; Result Variable

                                     ALIGN    SPIN

11E0          88 0C 61        F79:     LET      VL8.LONG, B174.BYTE
11E3          32                       RETURN

====                                 ; PRI test_Function_Abort
====                                 ;   Abort

11E4          30              S80:     ABORT

11E5          32              ; ----   RETURN

====                                 ; PRI test_Function_AbortValue
====                                 ;   Abort 0

11E6          35 31           S81:     ABORT    0

11E8          32              ; ----   RETURN

====                                 ; PRI test_Function_Nested( arg1 ,  arg2 )
====                                 ;   RETURN arg1 + arg2

                                     ALIGN    STACK          ; For F82

      +0000                 VL9:     LONG     0              ; Result Variable
      +0004                 VL10:    LONG     0
      +0008                 VL11:    LONG     0

                                     ALIGN    SPIN

11E9          64              F82:     PUSH     VL10.LONG
11EA          68                       PUSH     VL11.LONG
11EB          EC                       ADD
11EC          33                       RETVAL

11ED          32              ; ----   RETURN

====                                 ; PRI test_If
====                                 ;   If b == 0
====                                 ;     b := 0
====                                 ;   If b <> 1
====                                 ;     b := 1

11EE          88 0C           S83:     PUSH     B174.BYTE
11F0          35                       PUSH     0
11F1          FC                       EQ
11F2          0A 03                    JPF      N84
11F4          35 89 0C                 LET      B174.BYTE, 0
11F7          88 0C           N84:     PUSH     B174.BYTE
11F9          36                       PUSH     1
11FA          FB                       NE
11FB          0A 03                    JPF      N85
11FD          36 89 0C                 LET      B174.BYTE, 1
1200          32              N85:     RETURN

====                                 ; PRI test_If_Else
====                                 ;   If b == 0
====                                 ;     b := 0
====                                 ;   Else
====                                 ;     b := 1

1201          88 0C           S86:     PUSH     B174.BYTE
1203          35                       PUSH     0
1204          FC                       EQ
1205          0A 05                    JPF      N87
1207          35 89 0C                 LET      B174.BYTE, 0
120A          04 03                    GOTO     J88
120C          36 89 0C        N87:     LET      B174.BYTE, 1
120F          32              J88:     RETURN

====                                 ; PRI test_If_ElseIf
====                                 ;   If b == 0
====                                 ;     b := 0
====                                 ;   ElseIf b == 1
```

```
====                                ;     b := 1
====                                ;   Else
====                                ;     b := 2

1210        88 0C           S89:    PUSH    B174.BYTE
1212        35                      PUSH    0
1213        FC                      EQ
1214        0A 05                   JPF     N90
1216        35 89 0C                LET     B174.BYTE, 0
1219        04 0F                   GOTO    J92
121B        88 0C           N90:    PUSH    B174.BYTE
121D        36                      PUSH    1
121E        FC                      EQ
121F        0A 05                   JPF     N91
1221        36 89 0C                LET     B174.BYTE, 1
1224        04 04                   GOTO    J92
1226        37 00 89 0C     N91:    LET     B174.BYTE, 2
122A        32              J92:    RETURN

====                                ; PRI test_IfNot_ElseIfNot
====                                ;   IfNot b == 0
====                                ;     b := 0
====                                ;   ElseIfNot b == 1
====                                ;     b := 1
====                                ;   Else
====                                ;     b := 2

122B        88 0C           S93:    PUSH    B174.BYTE
122D        35                      PUSH    0
122E        FC                      EQ
122F        0B 05                   JPT     T94
1231        35 89 0C                LET     B174.BYTE, 0
1234        04 0F                   GOTO    J96
1236        88 0C           T94:    PUSH    B174.BYTE
1238        36                      PUSH    1
1239        FC                      EQ
123A        0B 05                   JPT     T95
123C        36 89 0C                LET     B174.BYTE, 1
123F        04 04                   GOTO    J96
1241        37 00 89 0C     T95:    LET     B174.BYTE, 2
1245        32              J96:    RETURN

====                                ; PRI test_Locks
====                                ;   LockClr(b)
====                                ;   LockNew
====                                ;   LockRet(b)
====                                ;   LockSet(b)
====                                ;   b := LockClr(b)
====                                ;   b := LockNew
====                                ;   b := LockSet(b)

1246        88 0C 2F        S97:    LCLRSUB  B174.BYTE
1249        2D                      LNEWSUB
124A        88 0C 22                LRETSUB  B174.BYTE
124D        88 0C 2E                LSETSUB  B174.BYTE
1250        88 0C 2B 89 0C          LCLRFUN  ( B174.BYTE ), B174.BYTE
1255        29 89 0C                LNEWFUN  B174.BYTE
1258        88 0C 2A 89 0C          LSETFUN  ( B174.BYTE ), B174.BYTE
125D        32                      RETURN

====                                ; PRI test_Lookup_And_LookDown
====                                ;   b := LookUp( b : 1, 2, 3 )
====                                ;   b := LookDown( b : 1, 2, 3 )
====                                ;   b := LookUpZ( b : 1, 2, 3 )
====                                ;   b := LookDownZ( b : 1, 2, 3 )
====                                ;   b := LookUp( b : 1..2, 2..3, 3..4 )
====                                ;   b := LookDown( b : 1..2, 2..3, 3..4 )
====                                ;   b := LookUpZ( b : 1..2, 2..3, 3..4 )
====                                ;   b := LookDownZ( b : 1..2, 2..3, 3..4 )

125E        36              S98:    PUSH    1
125F        39 12 5D                PUSH    4701
1262        88 0C                   PUSH    B174.BYTE
1264        36 10                   LOOKUP  1
1266        37 00 10                LOOKUP  2
1269        37 21 10                LOOKUP  3
126C        0F                      LOOKEND
126D        89 0C                   POP     B174.BYTE
126F        36                      PUSH    1
1270        39 12 6E                PUSH    4718
1273        88 0C                   PUSH    B174.BYTE
1275        36 11                   LOOKDN  1
1277        37 00 11                LOOKDN  2
127A        37 21 11                LOOKDN  3
127D        0F                      LOOKEND
127E        89 0C                   POP     B174.BYTE
1280        35                      PUSH    0
1281        39 12 7F                PUSH    4735
1284        88 0C                   PUSH    B174.BYTE
```

```
1286          36 10                     LOOKUP   1
1288          37 00 10                  LOOKUP   2
128B          37 21 10                  LOOKUP   3
128E          0F                        LOOKEND
128F          89 0C                     POP      B174.BYTE
1291          35                        PUSH     0
1292          39 12 90                  PUSH     4752
1295          88 0C                     PUSH     B174.BYTE
1297          36 11                     LOOKDN   1
1299          37 00 11                  LOOKDN   2
129C          37 21 11                  LOOKDN   3
129F          0F                        LOOKEND
12A0          89 0C                     POP      B174.BYTE
12A2          36                        PUSH     1
12A3          39 12 A7                  PUSH     4775
12A6          88 0C                     PUSH     B174.BYTE
12A8          36 37 00 12               LOOKUP   1, 2
12AC          37 00 37 21 12            LOOKUP   2, 3
12B1          37 21 37 01 12            LOOKUP   3, 4
12B6          0F                        LOOKEND
12B7          89 0C                     POP      B174.BYTE
12B9          36                        PUSH     1
12BA          39 12 BE                  PUSH     4798
12BD          88 0C                     PUSH     B174.BYTE
12BF          36 37 00 13               LOOKDN   1, 2
12C3          37 00 37 21 13            LOOKDN   2, 3
12C8          37 21 37 01 13            LOOKDN   3, 4
12CD          0F                        LOOKEND
12CE          89 0C                     POP      B174.BYTE
12D0          35                        PUSH     0
12D1          39 12 D5                  PUSH     4821
12D4          88 0C                     PUSH     B174.BYTE
12D6          36 37 00 12               LOOKUP   1, 2
12DA          37 00 37 21 12            LOOKUP   2, 3
12DF          37 21 37 01 12            LOOKUP   3, 4
12E4          0F                        LOOKEND
12E5          89 0C                     POP      B174.BYTE
12E7          35                        PUSH     0
12E8          39 12 EC                  PUSH     4844
12EB          88 0C                     PUSH     B174.BYTE
12ED          36 37 00 13               LOOKDN   1, 2
12F1          37 00 37 21 13            LOOKDN   2, 3
12F6          37 21 37 01 13            LOOKDN   3, 4
12FB          0F                        LOOKEND
12FC          89 0C                     POP      B174.BYTE
12FE          32                        RETURN

====                                    ; PRI test_Memory_Ops
====                                    ;   ByteFill( @b, 0, 1 )
====                                    ;   Wordfill( @b, 0, 2 )
====                                    ;   Longfill( @b, 0, 3 )
====                                    ;   ByteMove( @b, @b, 1 )
====                                    ;   WordMove( @b, @b, 2 )
====                                    ;   LongMove( @b, @b, 3 )

12FF          8B 0C 35 36 18  S99:      BYTEFIL  #B174.BYTE, 0, 1
1304          8B 0C 35 37 00            WORDFIL  #B174.BYTE, 0, 2
1309           + 19
130A          8B 0C 35 37 21            LONGFIL  #B174.BYTE, 0, 3
130F           + 1A
1310          8B 0C 8B 0C 36            BYTEMOV  #B174.BYTE, #B174.BYTE, 1
1315           + 1C
1316          8B 0C 8B 0C 37            WORDMOV  #B174.BYTE, #B174.BYTE, 2
131B           + 00 1D
131D          8B 0C 8B 0C 37            LONGMOV  #B174.BYTE, #B174.BYTE, 3
1322           + 21 1E
1324          32                        RETURN

====                                    ; PRI test_Objects
====                                    ;   Obj1_A.Start
====                                    ;   Obj1_A.Stop
====                                    ;   Obj1_B.Start
====                                    ;   Obj1_B.Stop
====                                    ;   b := Obj1_A.Func
====                                    ;   b := Obj1_B.Func
====                                    ;   Obj2_A.Start
====                                    ;   Obj2_A.Stop
====                                    ;   Obj2_B.Start
====                                    ;   Obj2_B.Stop
====                                    ;   b := Obj2_A.Func
====                                    ;   b := Obj2_B.Func

1325          01 06 2C 02     S100:     OBJSUB   O134, +2
1329          01 06 2C 03               OBJSUB   O134, +3
132D          01 06 2D 02               OBJSUB   O134, +2
1331          01 06 2D 03               OBJSUB   O134, +3
1335          00 06 2C 04 89            OBJFUN   O134, +4, B174.BYTE
133A           + 0C
133B          00 06 2D 04 89            OBJFUN   O134, +4, B174.BYTE
```

```
1340          + 0C
1341         01 06 2E 02              OBJSUB   O152, +2
1345         01 06 2E 03              OBJSUB   O152, +3
1349         01 06 2F 02              OBJSUB   O152, +2
134D         01 06 2F 03              OBJSUB   O152, +3
1351         00 06 2E 04 89           OBJFUN   O152, +4, B174.BYTE
1356          + 0C
1357         00 06 2F 04 89           OBJFUN   O152, +4, B174.BYTE
135C          + 0C
135D         32                       RETURN

====                                  ; PRI test_One_Line_Commands
====                                  ;   b := ChipVer
====                                  ;   b := ClkFreq
====                                  ;   b := ClkMode
====                                  ;   ClkSet( 0, 1 )
====                                  ;   b := Cnt

135E         34              S101:    PUSH     -1
135F         80                       PUSH     MEM[].BYTE
1360         89 0C                    POP      B174.BYTE
1362         35                       PUSH     0
1363         C0                       PUSH     MEM[].LONG
1364         89 0C                    POP      B174.BYTE
1366         38 04                    PUSH     4
1368         80                       PUSH     MEM[].BYTE
1369         89 0C                    POP      B174.BYTE
136B         35                       PUSH     0
136C         36                       PUSH     1
136D         20                       CLKSET
136E         3F 91 89 0C              LET      B174.BYTE, CNT
1372         32                       RETURN

====                                  ; PRI test_ReBoot
====                                  ;   ReBoot

1373         37 06           S102:    PUSH     $80
1375         35                       PUSH     0
1376         20                       CLKSET
1377         32                       RETURN

====                                  ; PRI test_Registers
====                                  ;   CTRA   := CTRB
====                                  ;   DIRA   := DIRB
====                                  ;   FRQA   := FRQB
====                                  ;   INA    := INA
====                                  ;   INB    := INB
====                                  ;   OUTA   := OUTA
====                                  ;   OUTB   := OUTB
====                                  ;   PAR    := PAR
====                                  ;   PHSA   := PHSB
====                                  ;   VCFG   := VCFG
====                                  ;   VSCL   := VSCL
====                                  ;   SPR[0] := SPR[1]
====                                  ;   SPR[2] ++

1378         3F 99 3F B8     S103:    LET      CTRA, CTRB
137C         3F 97 3F B6              LET      DIRA, DIRB
1380         3F 9B 3F BA              LET      FRQA, FRQB
1384         3F 92 3F B2              LET      INA, INA
1388         3F 93 3F B3              LET      INB, INB
138C         3F 94 3F B4              LET      OUTA, OUTA
1390         3F 95 3F B5              LET      OUTB, OUTB
1394         3F 90 3F B0              LET      PAR, PAR
1398         3F 9D 3F BC              LET      PHSA, PHSB
139C         3F 9E 3F BE              LET      VCFG, VCFG
13A0         3F 9F 3F BF              LET      VSCL, VSCL
13A4         36 24 35 25              LET      PAR, PAR
13A8         37 00 26 2E              INC      PAR
13AC         32                       RETURN

====                                  ; PRI test_RegisterBits
====                                  ;   OUTA[0] := 0
====                                  ;   OUTB[1] := 1
====                                  ;   OUTA[0..1] := 0
====                                  ;   OUTB[3..7] := 1
====                                  ;   OUTA[0]++
====                                  ;   OUTB[1]~~
====                                  ;   OUTA[0..1] := INB[3..7]
====                                  ;   OUTA[0..1] := INB[3..7]++

13AD         35              S104:    PUSH     0
13AE         35                       PUSH     0
13AF         3D B4                    POP      OUTA[]
13B1         36                       PUSH     1
13B2         36                       PUSH     1
13B3         3D B5                    POP      OUTB[]
13B5         35                       PUSH     0
13B6         35                       PUSH     0
```

```
13B7        36                      PUSH     1
13B8        3E B4                   POP      OUTA[..]
13BA        36                      PUSH     1
13BB        37 21                   PUSH     3
13BD        37 22                   PUSH     7
13BF        3E B5                   POP      OUTB[..]
13C1        35                      PUSH     0
13C2        3D D4 28                INC      OUTA[]
13C5        36                      PUSH     1
13C6        3D D5 1C                POSTSET  OUTB[]
13C9        37 21                   PUSH     3
13CB        37 22                   PUSH     7
13CD        3E 93                   PUSH     INB[..]
13CF        35                      PUSH     0
13D0        36                      PUSH     1
13D1        3E B4                   POP      OUTA[..]
13D3        37 21                   PUSH     3
13D5        37 22                   PUSH     7
13D7        3E D3 A8                PUSH     INB[..] POSTINC
13DA        35                      PUSH     0
13DB        36                      PUSH     1
13DC        3E B4                   POP      OUTA[..]
13DE        32                      RETURN

====                                ; PRI test_Repeat
====                                ;   Repeat
====                                ;     b := 0
====                                ;   Repeat
====                                ;     b := 1
====                                ;     Repeat
====                                ;       b := 2

13DF        35 89 0C        S105:   LET      B174.BYTE, 0
13E2        04 7B                   GOTO     S105
13E4        36 89 0C        J106:   LET      B174.BYTE, 1
13E7        37 00 89 0C     J107:   LET      B174.BYTE, 2
13EB        04 7A                   GOTO     J107
13ED        04 75                   GOTO     J106
13EF        32                      RETURN

====                                ; PRI test_Repeat_Count
====                                ;   Repeat 10
====                                ;     b++

13F0        38 0A           S108:   PUSH     10
13F2        08 05                   LOOPJPF  N110
13F4        8A 0C 2A        J109:     INC      B174.BYTE
13F7        09 7B                   LOOPRPT  J109
13F9        32              N110:   RETURN

====                                ; PRI test_Repeat_FromTo
====                                ;   Repeat b From 0 To 10
====                                ;     b++
====                                ;   Repeat b From 0 To 10 Step 1
====                                ;     b++
====                                ;   Repeat b From 10 To 0 Step -1
====                                ;     b++

13FA        35 89 0C        S111:   LET      B174.BYTE, 0
13FD        8A 0C 2A        J112:   INC      B174.BYTE
1400        35                      PUSH     0
1401        38 0A                   PUSH     10
1403        8A 0C 02 76             RPTINCJ  B174.BYTE, J112
1407        35 89 0C                LET      B174.BYTE, 0
140A        8A 0C 2A        J113:   INC      B174.BYTE
140D        36                      PUSH     1
140E        35                      PUSH     0
140F        38 0A                   PUSH     10
1411        8A 0C 06 75             RPTADDJ  B174.BYTE, J113
1415        38 0A 89 0C             LET      B174.BYTE, 10
1419        8A 0C 2A        J114:   INC      B174.BYTE
141C        34                      PUSH     -1
141D        38 0A                   PUSH     10
141F        35                      PUSH     0
1420        8A 0C 06 75             RPTADDJ  B174.BYTE, J114
1424        32                      RETURN

====                                ; PRI test_Repeat_Until
====                                ;   Repeat until b > 10
====                                ;     b++
====                                ;   Repeat
====                                ;     b++
====                                ;   Until b > 10

1425        88 0C           S115:   PUSH     B174.BYTE
1427        38 0A                   PUSH     10
1429        FA                      GT
142A        0B 05                   JPT      J116
142C        8A 0C 2A                INC      B174.BYTE
```

```
142F        04 74                   GOTO     S115
1431        8A 0C 2A        J116:   INC      B174.BYTE
1434        88 0C                   PUSH     B174.BYTE
1436        38 0A                   PUSH     10
1438        FA                      GT
1439        0A 76                   JPF      J116
143B        32                      RETURN

====                                ; PRI test_Repeat_While
====                                ;   Repeat while b > 10
====                                ;     b++
====                                ;   Repeat
====                                ;     b++
====                                ;   While b > 10

143C        88 0C           S117:   PUSH     B174.BYTE
143E        38 0A                   PUSH     10
1440        FA                      GT
1441        0A 05                   JPF      J118
1443        8A 0C 2A                INC      B174.BYTE
1446        04 74                   GOTO     S117
1448        8A 0C 2A        J118:   INC      B174.BYTE
144B        88 0C                   PUSH     B174.BYTE
144D        38 0A                   PUSH     10
144F        FA                      GT
1450        0B 76                   JPT      J118
1452        32                      RETURN

====                                ; PRI test_Repeat_With_Next
====                                ;   Repeat while b > 10
====                                ;     b++
====                                ;     Next
====                                ;     b--
====                                ;   Repeat
====                                ;     b++
====                                ;     Next
====                                ;     b--
====                                ;   While b > 10

1453        88 0C           S119:   PUSH     B174.BYTE
1455        38 0A                   PUSH     10
1457        FA                      GT
1458        0A 0A                   JPF      J120
145A        8A 0C 2A                INC      B174.BYTE
145D        04 74                   GOTO     S119
145F        8A 0C 3A                DEC      B174.BYTE
1462        04 6F                   GOTO     S119
1464        8A 0C 2A        J120:   INC      B174.BYTE
1467        04 03                   GOTO     J121
1469        8A 0C 3A                DEC      B174.BYTE
146C        88 0C           J121:   PUSH     B174.BYTE
146E        38 0A                   PUSH     10
1470        FA                      GT
1471        0B 71                   JPT      J120
1473        32                      RETURN

====                                ; PRI test_Repeat_With_Quit
====                                ;   Repeat while b > 10
====                                ;     b++
====                                ;     Quit
====                                ;     b--
====                                ;   Repeat
====                                ;     b++
====                                ;     Quit
====                                ;     b--
====                                ;   While b > 10

1474        88 0C           S122:   PUSH     B174.BYTE
1476        38 0A                   PUSH     10
1478        FA                      GT
1479        0A 0A                   JPF      J123
147B        8A 0C 2A                INC      B174.BYTE
147E        04 05                   GOTO     J123
1480        8A 0C 3A                DEC      B174.BYTE
1483        04 6F                   GOTO     S122
1485        8A 0C 2A        J123:   INC      B174.BYTE
1488        04 0A                   GOTO     J124
148A        8A 0C 3A                DEC      B174.BYTE
148D        88 0C                   PUSH     B174.BYTE
148F        38 0A                   PUSH     10
1491        FA                      GT
1492        0B 71                   JPT      J123
1494        32              J124:   RETURN

====                                ; PRI test_Strings
====                                ;   b := StrComp( @b, @b )
====                                ;   b := String( "Xyzzy" )
====                                ;   b := StrSize( @b )
```

```
1495        8B 0C 8B 0C 17  S125:    STRCOMP  #B174.BYTE, #B174.BYTE
149A        89 0C                     POP      B174.BYTE
149C        87 94 97 89 0C            LET      B174.BYTE, #B126.BYTE
14A1        8B 0C                     PUSH     #B174.BYTE
14A3        16                        STRSIZE
14A4        89 0C                     POP      B174.BYTE
14A6        32                        RETURN

14A7        58              B126:     BYTE     "X"            ; 88
14A8        79                        BYTE     "y"            ; 121
14A9        7A                        BYTE     "z"            ; 122
14AA        7A                        BYTE     "z"            ; 122
14AB        79                        BYTE     "y"            ; 121
14AC        00                        BYTE     0

====                                  ; PRI test_UnaryOps
====                                  ;   -b
====                                  ;   ++b
====                                  ;   --b
====                                  ;   b++
====                                  ;   b--
====                                  ;   ! b
====                                  ;   NOT b
====                                  ;   ^^ b
====                                  ;   || b
====                                  ;   ~b
====                                  ;   ~~b
====                                  ;   b~
====                                  ;   b~~
====                                  ;   ?b
====                                  ;   b?
====                                  ;   |< b
====                                  ;   >| b

14AD        8A 0C 46        S127:     NEG      B174.BYTE
14B0        8A 0C 22                  INC      B174.BYTE
14B3        8A 0C 32                  DEC      B174.BYTE
14B6        8A 0C 2A                  INC      B174.BYTE
14B9        8A 0C 3A                  DEC      B174.BYTE
14BC        8A 0C 47                  BIT_NOT  B174.BYTE
14BF        8A 0C 5F                  LOG_NOT  B174.BYTE
14C2        8A 0C 58                  SQR      B174.BYTE
14C5        8A 0C 49                  ABS      B174.BYTE
14C8        8A 0C 10                  SEXBYTE  B174.BYTE
14CB        8A 0C 14                  SEXWORD  B174.BYTE
14CE        8A 0C 18                  POSTCLR  B174.BYTE
14D1        8A 0C 1C                  POSTSET  B174.BYTE
14D4        8A 0C 08                  FWDRAND  B174.BYTE
14D7        8A 0C 0C                  REVRAND  B174.BYTE
14DA        8A 0C 53                  DECODE   B174.BYTE
14DD        8A 0C 51                  ENCODE   B174.BYTE
14E0        32                        RETURN

====                                  ; PRI test_UnaryOps_Assigned
====                                  ;   b := ( b := b )
====                                  ;   b := +b
====                                  ;   b := -b
====                                  ;   b := ++b
====                                  ;   b := --b
====                                  ;   b := b++
====                                  ;   b := b--
====                                  ;   b := ! b
====                                  ;   b := NOT b
====                                  ;   b := ^^ b
====                                  ;   b := || b
====                                  ;   b := ~b
====                                  ;   b := ~~b
====                                  ;   b := b~
====                                  ;   b := b~~
====                                  ;   b := ?b
====                                  ;   b := b?
====                                  ;   b := |< b
====                                  ;   b := >| b
====                                  ;   b := @b

14E1        88 0C           S128:     PUSH     B174.BYTE
14E3        8A 0C 80 89 0C            LET      B174.BYTE, B174.BYTE COPY
14E8        88 0C 89 0C               LET      B174.BYTE, B174.BYTE
14EC        88 0C                     PUSH     B174.BYTE
14EE        E6                        NEG
14EF        89 0C                     POP      B174.BYTE
14F1        8A 0C A2 89 0C            LET      B174.BYTE, B174.BYTE PREINC
14F6        8A 0C B2 89 0C            LET      B174.BYTE, B174.BYTE PREDEC
14FB        8A 0C AA 89 0C            LET      B174.BYTE, B174.BYTE POSTINC
1500        8A 0C BA 89 0C            LET      B174.BYTE, B174.BYTE POSTDEC
1505        88 0C                     PUSH     B174.BYTE
1507        E7                        BIT_NOT
1508        89 0C                     POP      B174.BYTE
150A        88 0C                     PUSH     B174.BYTE
```

```
150C        FF                      LOG_NOT
150D        89 0C                   POP     B174.BYTE
150F        88 0C                   PUSH    B174.BYTE
1511        F8                      SQRT
1512        89 0C                   POP     B174.BYTE
1514        88 0C                   PUSH    B174.BYTE
1516        E9                      ABS
1517        89 0C                   POP     B174.BYTE
1519        8A 0C 90 89 0C          LET     B174.BYTE, B174.BYTE SEXBYTE
151E        8A 0C 94 89 0C          LET     B174.BYTE, B174.BYTE SEXWORD
1523        8A 0C 98 89 0C          LET     B174.BYTE, B174.BYTE POSTCLR
1528        8A 0C 9C 89 0C          LET     B174.BYTE, B174.BYTE POSTSET
152D        8A 0C 88 89 0C          LET     B174.BYTE, B174.BYTE FWDRAND
1532        8A 0C 8C 89 0C          LET     B174.BYTE, B174.BYTE REVRAND
1537        88 0C                   PUSH    B174.BYTE
1539        F3                      DECODE
153A        89 0C                   POP     B174.BYTE
153C        88 0C                   PUSH    B174.BYTE
153E        F1                      ENCODE
153F        89 0C                   POP     B174.BYTE
1541        8B 0C 89 0C             LET     B174.BYTE, #B174.BYTE
1545        32                      RETURN

====                                ; PRI test_Vars | bLocal, wLocal, lLocal
====                                ;   b := 1
====                                ;   w := 2
====                                ;   l := 3
====                                ;   bOverlay := 1
====                                ;   wOverlay := 2
====                                ;   lOverlay := 3
====                                ;   bOverlayEnd := 1
====                                ;   wOverlayEnd := 2
====                                ;   lOverlayEnd := 3
====                                ;   bLocal := 1
====                                ;   wLocal := 2
====                                ;   lLocal := 3

                                    ALIGN   STACK           ; For S129

     +0000                          LONG    0               ; Unused Result Variable
     +0004              VL12:       LONG    0
     +0008              VL13:       LONG    0
     +000C              VL14:       LONG    0

                                    ALIGN   SPIN

1546        36 89 0C        S129:   LET     B174.BYTE, 1
1549        37 00 A9 08             LET     W172.WORD, 2
154D        37 21 41                LET     L170.LONG, 3
1550        36 89 0D                LET     B175.BYTE, 1
1553        37 00 A9 0A             LET     W173.WORD, 2
1557        37 21 45                LET     L171.LONG, 3
155A        36 89 0D                LET     B175.BYTE, 1
155D        37 00 A9 0A             LET     W173.WORD, 2
1561        37 21 45                LET     L171.LONG, 3
1564        36 65                   LET     VL12.LONG, 1
1566        37 00 69                LET     VL13.LONG, 2
1569        37 21 6D                LET     VL14.LONG, 3
156C        32                      RETURN

====                                ; PRI test_Vars_Casting
====                                ;   l.byte := l.byte
====                                ;   l.word := l.word
====                                ;   l.long := l.long
====                                ;   l.byte[1] := l.byte[1]
====                                ;   l.word[2] := l.word[2]
====                                ;   l.long[3] := l.long[3]

156D        88 00 89 00     S130:   LET     L170.BYTE, L170.BYTE
1571        A8 00 A9 00             LET     L170.WORD, L170.WORD
1575        40 41                   LET     L170.LONG, L170.LONG
1577        36                      PUSH    1
1578        98 00                   PUSH    L170[].BYTE
157A        36                      PUSH    1
157B        99 00                   POP     L170[].BYTE
157D        37 00                   PUSH    2
157F        B8 00                   PUSH    L170[].WORD
1581        37 00                   PUSH    2
1583        B9 00                   POP     L170[].WORD
1585        37 21                   PUSH    3
1587        D8 00                   PUSH    L170[].LONG
1589        37 21                   PUSH    3
158B        D9 00                   POP     L170[].LONG
158D        32                      RETURN

====                                ; PRI test_Vars_Data
====                                ;   bData := bData[0]
====                                ;   wData := wData[1]
====                                ;   lData := lData[3]
```

```
====                                ;   bData[0] := bData
====                                ;   wData[1] := wData
====                                ;   lData[3] := lData

158E        35              S131:   PUSH    0
158F        94 80 C0                PUSH    B49[].BYTE
1592        85 80 C0                POP     B49.BYTE
1595        36                      PUSH    1
1596        B4 80 C2                PUSH    W50[].WORD
1599        A5 80 C2                POP     W50.WORD
159C        37 21                   PUSH    3
159E        D4 80 C4                PUSH    L51[].LONG
15A1        C5 80 C4                POP     L51.LONG
15A4        84 80 C0                PUSH    B49.BYTE
15A7        35                      PUSH    0
15A8        95 80 C0                POP     B49[].BYTE
15AB        A4 80 C2                PUSH    W50.WORD
15AE        36                      PUSH    1
15AF        B5 80 C2                POP     W50[].WORD
15B2        C4 80 C4                PUSH    L51.LONG
15B5        37 21                   PUSH    3
15B7        D5 80 C4                POP     L51[].LONG
15BA        32                      RETURN

====                                ; PRI test_Vars_Local |t0, t1, t2, t3, t4, t5, t6, t7, t8
====                                ;   t0 := t0
====                                ;   t1 := t1
====                                ;   t2 := t2
====                                ;   t3 := t3
====                                ;   t4 := t4
====                                ;   t5 := t5
====                                ;   t6 := t6
====                                ;   t7 := t7
====                                ;   t8 := t8
====                                ;   t0++
====                                ;   t1++
====                                ;   t2++
====                                ;   t3++
====                                ;   t4++
====                                ;   t5++
====                                ;   t6++
====                                ;   t7++
====                                ;   t8++

                                    ALIGN   STACK           ; For S132

     +0000                          LONG    0               ; Unused Result Variable
     +0004              VL15:       LONG    0
     +0008              VL16:       LONG    0
     +000C              VL17:       LONG    0
     +0010              VL18:       LONG    0
     +0014              VL19:       LONG    0
     +0018              VL20:       LONG    0
     +001C              VL21:       LONG    0
     +0020              VL22:       LONG    0
     +0024              VL23:       LONG    0

                                    ALIGN   SPIN

15BB        64 65           S132:   LET     VL15.LONG, VL15.LONG
15BD        68 69                   LET     VL16.LONG, VL16.LONG
15BF        6C 6D                   LET     VL17.LONG, VL17.LONG
15C1        70 71                   LET     VL18.LONG, VL18.LONG
15C3        74 75                   LET     VL19.LONG, VL19.LONG
15C5        78 79                   LET     VL20.LONG, VL20.LONG
15C7        7C 7D                   LET     VL21.LONG, VL21.LONG
15C9        CC 20 CD 20             LET     VL22.LONG, VL22.LONG
15CD        CC 24 CD 24             LET     VL23.LONG, VL23.LONG
15D1        66 2E                   INC     VL15.LONG
15D3        6A 2E                   INC     VL16.LONG
15D5        6E 2E                   INC     VL17.LONG
15D7        72 2E                   INC     VL18.LONG
15D9        76 2E                   INC     VL19.LONG
15DB        7A 2E                   INC     VL20.LONG
15DD        7E 2E                   INC     VL21.LONG
15DF        CE 20 2E                INC     VL22.LONG
15E2        CE 24 2E                INC     VL23.LONG
15E5        32                      RETURN

====                                ; PRI test_Wait
====                                ;   WaitCnt( 0 )
====                                ;   WaitPeq( 0, 1, 2)
====                                ;   WaitPne( 0, 1, 2 )
====                                ;   WaitVid( 0, 1 )

15E6        35              S133:   PUSH    0
15E7        23                      WAITCNT
15E8        35                      PUSH    0
15E9        36                      PUSH    1
```

```
15EA         37 00                  PUSH     2
15EC         1B                     WAITPEQ
15ED         35                     PUSH     0
15EE         36                     PUSH     1
15EF         37 00                  PUSH     2
15F1         1F                     WAITPNE
15F2         35                     PUSH     0
15F3         36                     PUSH     1
15F4         27                     WAITVID
15F5         32                     RETURN

15F6         00 00                  WORD     0              ; Alignment Padding

                                    ALIGN    OBJECT

15F8         98 00 09 00    O134:   LINK     O152, 9        ; +0 = Next Object

15FC         38 00 00 00    X135:   LINK     F144           ; +1
1600         57 00 00 00    X136:   LINK     S145           ; +2
1604         5B 00 00 00    X137:   LINK     S146           ; +3
1608         5F 00 00 00    X138:   LINK     F147           ; +4
160C         64 00 00 00    X139:   LINK     S148           ; +5
1610         68 00 00 00    X140:   LINK     S149           ; +6
1614         6C 00 00 00    X141:   LINK     F150           ; +7
1618         70 00 0C 00    X142:   LINK     S151, 12       ; +8

                                    ALIGN          ORG      0

161C  000    04 06 BC 80    A143:                ADD      V14, V15
1620  001    21 40 BC 80                         ADD      V16, V17
1624  002    02 00 3C 5C                         JMP      V18

1628  003    00 00 00 00    V14:                 LONG     0
162C  004    00 00 00 00    V15:                 LONG     0

                                    ALIGN          ORG      $20

      020    00 00 00 00    V16:                 LONG     0         ; $59000000 / 89
      021    00 00 00 00    V17:                 LONG     0         ; $5D000000 / 93

1630         01 05 02       F144:   CALLSUB  S145
1633         01 05 03               CALLSUB  S146
1636         00 05 04 A9 08         CALLFUN  F147, W172.WORD
163B         01 05 05               CALLSUB  S148
163E         01 05 06               CALLSUB  S149
1641         00 05 07 A9 08         CALLFUN  F150, W172.WORD
1646         01 05 08               CALLSUB  S151
1649         34 C7 24 35 2C         COGISUB  -1, #A143.LONG, 0
164E         32                     RETURN

164F         01 05 05       S145:   CALLSUB  S148
1652         32                     RETURN

1653         01 05 06       S146:   CALLSUB  S149
1656         32                     RETURN

1657         00 05 07       F147:   CALLFUN  F150
165A         33                     RETVAL

165B         32             ; ----  RETURN

165C         35 A9 08       S148:   LET      W172.WORD, 0
165F         32                     RETURN

1660         36 A9 08       S149:   LET      W172.WORD, 1
1663         32                     RETURN

                                    ALIGN    STACK          ; For F150

     +0000                  VL24:   LONG     0              ; Result Variable

                                    ALIGN    SPIN

1664         A8 08 61       F150:   LET      VL24.LONG, W172.WORD
1667         32                     RETURN

                                    ALIGN    STACK          ; For S151

     +0000                          LONG     0              ; Unused Result Variable
     +0004                  VL25:   LONG     0
     +0008                  VL26:   LONG     0
     +000C                  VL27:   LONG     0

                                    ALIGN    SPIN

1668         36 89 0E       S151:   LET      VAR+14.BYTE, 1
166B         37 00 A9 0A            LET      W173.WORD, 2
166F         37 21 41               LET      L170.LONG, 3
1672         36 89 0F               LET      VAR+15.BYTE, 1
```

```
1675         37 00 A9 0C              LET      B174.WORD, 2
1679         37 21 45                 LET      L171.LONG, 3
167C         36 89 0F                 LET      VAR+15.BYTE, 1
167F         37 00 A9 0C              LET      B174.WORD, 2
1683         37 21 45                 LET      L171.LONG, 3
1686         36 65                    LET      VL25.LONG, 1
1688         37 00 69                 LET      VL26.LONG, 2
168B         37 21 6D                 LET      VL27.LONG, 3
168E         32                       RETURN

168F         00                       BYTE     0              ; Alignment Padding

                                      ALIGN    OBJECT

1690         94 00 09 00     O152:    LINK     VBASE, 9       ; +0 = Next Object

1694         38 00 00 00     X153:    LINK     F162           ; +1
1698         55 00 00 00     X154:    LINK     S163           ; +2
169C         59 00 00 00     X155:    LINK     S164           ; +3
16A0         5D 00 00 00     X156:    LINK     F165           ; +4
16A4         62 00 00 00     X157:    LINK     S166           ; +5
16A8         65 00 00 00     X158:    LINK     S167           ; +6
16AC         68 00 00 00     X159:    LINK     F168           ; +7
16B0         6B 00 0C 00     X160:    LINK     S169, 12       ; +8

                                      ALIGN           ORG      0

16B4   000   04 06 BC 80     A161:                    ADD      V19, V20
16B8   001   21 40 BC 80                              ADD      V21, V22
16BC   002   02 00 3C 5C                              JMP      V23

16C0   003   00 00 00 00     V19:                     LONG     0
16C4   004   00 00 00 00     V20:                     LONG     0

                                      ALIGN           ORG      $20

       020   00 00 00 00     V21:                     LONG     0
       021   00 00 00 00     V22:                     LONG     0

16C8         01 05 02        F162:    CALLSUB  S163
16CB         01 05 03                 CALLSUB  S164
16CE         00 05 04 41              CALLFUN  F165, L170.LONG
16D2         01 05 05                 CALLSUB  S166
16D5         01 05 06                 CALLSUB  S167
16D8         00 05 07 41              CALLFUN  F168, L170.LONG
16DC         01 05 08                 CALLSUB  S169
16DF         34 C7 24 35 2C           COGISUB  -1, #A161.LONG, 0
16E4         32                       RETURN

16E5         01 05 05        S163:    CALLSUB  S166
16E8         32                       RETURN

16E9         01 05 06        S164:    CALLSUB  S167
16EC         32                       RETURN

16ED         00 05 07        F165:    CALLFUN  F168
16F0         33                       RETVAL

16F1         32              ; ----   RETURN

16F2         35 41           S166:    LET      L170.LONG, 0
16F4         32                       RETURN

16F5         36 41           S167:    LET      L170.LONG, 1
16F7         32                       RETURN

                                      ALIGN    STACK          ; For F168

     +0000                   VL28:    LONG     0              ; Result Variable

                                      ALIGN    SPIN

16F8         40 61           F168:    LET      VL28.LONG, L170.LONG
16FA         32                       RETURN

                                      ALIGN    STACK          ; For S169

     +0000                            LONG     0              ; Unused Result Variable
     +0004                   VL29:    LONG     0
     +0008                   VL30:    LONG     0
     +000C                   VL31:    LONG     0

                                      ALIGN    SPIN

16FB         36 89 10        S169:    LET      L176.BYTE, 1
16FE         37 00 A9 0C              LET      B174.WORD, 2
1702         37 21 45                 LET      L171.LONG, 3
1705         36 89 11                 LET      VAR+17.BYTE, 1
1708         37 00 A9 0E              LET      VAR+14.WORD, 2
```

```
170C         37 21 49                LET      W172.LONG, 3
170F         36 89 11                LET      VAR+17.BYTE, 1
1712         37 00 A9 0E             LET      VAR+14.WORD, 2
1716         37 21 49                LET      W172.LONG, 3
1719         36 65                   LET      VL29.LONG, 1
171B         37 00 69                LET      VL30.LONG, 2
171E         37 21 6D                LET      VL31.LONG, 3
1721         32                      RETURN

1722         00 00                   WORD     0              ; Alignment Padding

1724                        VBASE:   ALIGN    LONG

1724         00 00 00 00    L170:    LONG     0

1724                        W170_H   EQU      L170+0

1724                        B170_0   EQU      L170+0

1728         00 00 00 00    L171:    LONG     0
172C         00 00          W172:    WORD     0
172E         00 00          W173:    WORD     0
1730         00             B174:    BYTE     0
1731         00             B175:    BYTE     0
1732         00 00                   WORD     0
1734         00 00 00 00    L176:    LONG     0

1738   021   00 00 00 00    L177:                    LONG     0

173C         00 00          W178:    WORD     0
173E         00 00          W179:    WORD     0
1740         00 00          W180:    WORD     0
1742         00             B181:    BYTE     0
1743         00             B182:    BYTE     0
1744         00 00 00 00             LONG     0
1748         00 00 00 00             LONG     0
174C         00 00 00 00             LONG     0
1750         00 00 00 00             LONG     0
1754         00 00 00 00    L183:    LONG     0
1758         00 00 00 00    L184:    LONG     0
175C         00 00 00 00    L185:    LONG     0
1760         00 00          W186:    WORD     0
1762         00 00          W187:    WORD     0
1764         00             B188:    BYTE     0
1765         00             B189:    BYTE     0
1766         00 00                   WORD     0
1768         00 00 00 00             LONG     0
176C         00 00 00 00             LONG     0
1770         00 00 00 00             LONG     0
1774         00 00 00 00             LONG     0
1778         00 00 00 00             LONG     0

177C                        VENDS:   ALIGN    LONG

177C         FF FF F9 FF             LONG     -393217
1780         FF FF F9 FF             LONG     -393217

1784                        SBASE:

1784         00 00 00 00             LONG     0

1788                        SINIT:
```

Symbol Table
==========

197 symbols used

A = Assembler Jump Destination
B = Byte Variable
F = Function Entry Point
J = Jump Destination
L = Long Variable
N = Jump if False (Not True) Destination
O = Object
R = Assembler RET Location
S = Subroutine Entry Point
T = Jump if True Destination
V = Assembler Variable
W = Word Variable
X = Link Vector

```
0010   PBASE    Base of Program
1724   VBASE    Base of Variables
1784   SBASE    Base of Stack
03C0   PINIT    Initial Program Counter
1788   SINIT    Initial Stack Pointer

0010   X1       Link   O134
```

```
0014   X2       Link   S59
0018   X3       Link   S62
001C   X4       Link   S63
0020   X5       Link   S64
0024   X6       Link   S65
0028   X7       Link   S66
002C   X8       Link   S73
0030   X9       Link   S74
0034   X10      Link   S75
0038   X11      Link   S76
003C   X12      Link   F77
0040   X13      Link   F78
0044   X14      Link   F79
0048   X15      Link   S80
004C   X16      Link   S81
0050   X17      Link   F82
0054   X18      Link   S83
0058   X19      Link   S86
005C   X20      Link   S89
0060   X21      Link   S93
0064   X22      Link   S97
0068   X23      Link   S98
006C   X24      Link   S99
0070   X25      Link   S100
0074   X26      Link   S101
0078   X27      Link   S102
007C   X28      Link   S103
0080   X29      Link   S104
0084   X30      Link   S105
0088   X31      Link   S108
008C   X32      Link   S111
0090   X33      Link   S115
0094   X34      Link   S117
0098   X35      Link   S119
009C   X36      Link   S122
00A0   X37      Link   S125
00A4   X38      Link   S127  Long Word Byte
00A8   X39      Link   S128
00AC   X40      Link   S129
00B0   X41      Link   S130
00B4   X42      Link   S131
00B8   X43      Link   S132
00BC   X44      Link   S133
00C0   X45      Link   O134
00C4   X46      Link   O134
00C8   X47      Link   O152
00CC   X48      Link   O152
00D0   B49      Byte
00D2   W50      Word
00D4   L51      Long
00D8   A52      Asmb   Org0  Call  Long
0120   A53      Asmb   Goto
012C   A54      Asmb   Org0  Call  Long
0390   R55_ret  Asmb   Goto
0394   A56      Asmb   Call  JmpT         Called from 027C  Paired with 039C = A56_ret
039C   A56_ret  Asmb   Goto               Jumped from 027C  Paired with 0394 = A56
03A0   A57      Asmb   Call               Called from 0394  Paired with 03A4 = A57_ret
03A4   A57_ret  Asmb   Goto               Jumped from 0394  Paired with 03A0 = A57
03A8   A58      Asmb   Call               Called from 0284  Paired with 03B4 = A58_ret
03B4   A58_ret  Asmb   Goto               Jumped from 0284  Paired with 03A8 = A58
03C0   S59      Spin   Call
0409   N60      Spin   JmpF               Jumped from 0404
0414   N61      Spin   JmpF               Jumped from 040F
048B   S62      Spin   Call               Called from 03C2
0491   S63      Spin   Call               Called from 03C8
0499   S64      Spin   Call               Called from 03CB
0549   S65      Spin   Call               Called from 03CE
05C7   S66      Spin   Call               Called from 03D1
05D4   J67      Spin   Goto
05D8   J68      Spin   Goto
05DD   J69      Spin   Goto
05EE   J70      Spin   Goto
05F2   J71      Spin   Goto
05F7   J72      Spin   Goto
05F8   S73      Spin   Call               Called from 03D7
0625   S74      Spin   Call               Called from 03DA
0670   S75      Spin   Call               Called from 03E1
0675   S76      Spin   Call               Called from 03D4
11D8   F77      Spin   Call               Called from 03F0
11DC   F78      Spin   Call               Called from 03F5
11E0   F79      Spin   Call               Called from 03FA
11E4   S80      Spin   Call               Called from 03FF
11E6   S81      Spin   Call               Called from 040A
11E9   F82      Spin   Call               Called from 0441
11EE   S83      Spin   Call               Called from 03E4
11F7   N84      Spin   JmpF               Jumped from 11F2
1200   N85      Spin   JmpF               Jumped from 11FB
1201   S86      Spin   Call               Called from 03E7
120C   N87      Spin   JmpF               Jumped from 1205
```

```
120F   J88      Spin   Goto               Jumped from 120A
1210   S89      Spin   Call               Called from 03EA
121B   N90      Spin   JmpF               Jumped from 1214
1226   N91      Spin   JmpF               Jumped from 121F
122A   J92      Spin   Goto               Jumped from 1219
122B   S93      Spin   Call               Called from 03ED
1236   T94      Spin   JmpT               Jumped from 122F
1241   T95      Spin   JmpT               Jumped from 123A
1245   J96      Spin   Goto               Jumped from 1234
1246   S97      Spin   Call               Called from 0446
125E   S98      Spin   Call               Called from 0449
12FF   S99      Spin   Call               Called from 044C
1325   S100     Spin   Call               Called from 044F
135E   S101     Spin   Call               Called from 0452
1373   S102     Spin   Call               Called from 0455
1378   S103     Spin   Call               Called from 0458
13AD   S104     Spin   Call               Called from 045B
13DF   S105     Spin   Call  Goto         Called from 13E2
13E4   J106     Spin   Goto               Jumped from 13ED
13E7   J107     Spin   Goto               Jumped from 13EB
13F0   S108     Spin   Call               Called from 0461
13F4   J109     Spin   Goto               Jumped from 13F7
13F9   N110     Spin   JmpF               Jumped from 13F2
13FA   S111     Spin   Call               Called from 0464
13FD   J112     Spin   Goto
140A   J113     Spin   Goto
1419   J114     Spin   Goto
1425   S115     Spin   Call  Goto         Called from 142F
1431   J116     Spin   Goto               Jumped from 1439
143C   S117     Spin   Call  Goto         Called from 1446
1448   J118     Spin   Goto               Jumped from 1450
1453   S119     Spin   Call  Goto         Called from 1462
1464   J120     Spin   Goto               Jumped from 1471
146C   J121     Spin   Goto               Jumped from 1467
1474   S122     Spin   Call  Goto         Called from 1483
1485   J123     Spin   Goto               Jumped from 147E
1494   J124     Spin   Goto               Jumped from 1488
1495   S125     Spin   Call               Called from 0473
14A7   B126     Byte
14AD   S127     Spin   Call               Called from 0476
14E1   S128     Spin   Call               Called from 0479
1546   S129     Spin   Call               Called from 047C
156D   S130     Spin   Call               Called from 047F
158E   S131     Spin   Call               Called from 0482
15BB   S132     Spin   Call               Called from 0485
15E6   S133     Spin   Call               Called from 0488
15F8   O134     Link   O152
15FC   X135     Link   F144
1600   X136     Link   S145
1604   X137     Link   S146
1608   X138     Link   F147
160C   X139     Link   S148
1610   X140     Link   S149
1614   X141     Link   F150
1618   X142     Link   S151
161C   A143     Asmb   Org0  Call  Long
1630   F144     Spin   Call
164F   S145     Spin   Call               Called from 1631
1653   S146     Spin   Call               Called from 1634
1657   F147     Spin   Call               Called from 1637
165C   S148     Spin   Call               Called from 163C
1660   S149     Spin   Call               Called from 163F
1664   F150     Spin   Call               Called from 1642
1668   S151     Spin   Call               Called from 1647
1690   O152     Link   L170
1694   X153     Link   F162
1698   X154     Link   S163
169C   X155     Link   S164
16A0   X156     Link   F165 Asmb
16A4   X157     Link   S166
16A8   X158     Link   S167
16AC   X159     Link   F168
16B0   X160     Link   S169
16B4   A161     Asmb   Org0  Call  Long
16C8   F162     Spin   Call
16E5   S163     Spin   Call               Called from 16C9
16E9   S164     Spin   Call               Called from 16CC
16ED   F165     Spin   Call               Called from 16CF
16F2   S166     Spin   Call               Called from 16D3
16F5   S167     Spin   Call               Called from 16D6
16F8   F168     Spin   Call               Called from 16D9
16FB   S169     Spin   Call               Called from 16DD
1724   L170     Long Word Byte
1728   L171     Long
172C   W172     Word
172E   W173     Word
1730   B174     Byte
1731   B175     Byte
1734   L176     Long
```

```
1738   L177     Asmb   Long
173C   W178     Word
173E   W179     Word
1740   W180     Word
1742   B181     Byte
1743   B182     Byte
1754   L183     Long
1758   L184     Long
175C   L185     Long
1760   W186     Word
1762   W187     Word
1764   B188     Byte
1765   B189     Byte

Object List
===========

2 objects used

0010   PBASE    15F8   O134
0014   +1       03C0   S59
0018   +2       048B   S62
001C   +3       0491   S63
0020   +4       0499   S64
0024   +5       0549   S65
0028   +6       05C7   S66
002C   +7       05F8   S73
0030   +8       0625   S74
0034   +9       0670   S75
0038   +10      0675   S76
003C   +11      11D8   F77
0040   +12      11DC   F78
0044   +13      11E0   F79
0048   +14      11E4   S80
004C   +15      11E6   S81
0050   +16      11E9   F82
0054   +17      11EE   S83
0058   +18      1201   S86
005C   +19      1210   S89
0060   +20      122B   S93
0064   +21      1246   S97
0068   +22      125E   S98
006C   +23      12FF   S99
0070   +24      1325   S100
0074   +25      135E   S101
0078   +26      1373   S102
007C   +27      1378   S103
0080   +28      13AD   S104
0084   +29      13DF   S105
0088   +30      13F0   S108
008C   +31      13FA   S111
0090   +32      1425   S115
0094   +33      143C   S117
0098   +34      1453   S119
009C   +35      1474   S122
00A0   +36      1495   S125
00A4   +37      14AD   S127
00A8   +38      14E1   S128
00AC   +39      1546   S129
00B0   +40      156D   S130
00B4   +41      158E   S131
00B8   +42      15BB   S132
00BC   +43      15E6   S133
00C0   +44      15F8   O134
00C4   +45      15F8   O134
00C8   +46      1690   O152
00CC   +47      1690   O152

15F8   O134     1690   O152
15FC   +1       1630   F144
1600   +2       164F   S145
1604   +3       1653   S146
1608   +4       1657   F147
160C   +5       165C   S148
1610   +6       1660   S149
1614   +7       1664   F150
1618   +8       1668   S151

1690   O152     1724   VBASE
1694   +1       16C8   F162
1698   +2       16E5   S163
169C   +3       16E9   S164
16A0   +4       16ED   F165
16A4   +5       16F2   S166
16A8   +6       16F5   S167
16AC   +7       16F8   F168
16B0   +8       16FB   S169
```