Stack Oriented Arithmetic Processor

technical systems consultants, inc.

```
LOCN B1 B2 B3
                *
                *
                *
                *
                *    SL68-25
                *    STACK ORIENTED ARITHMETIC PROCESSOR
                *
                *    COPYRIGHT (C) 1977 BY
                *    TECHNICAL SYSTEMS CONSULTANTS
                *   BOX 2574  W. LAFAYETTE IN  47906
                *
                *
                * THE STACK ORIENTED ARITHMETIC PROCESSOR DERIVES
                * ITS NAME FROM THE FACT THAT THE OPERANDS ARE
                * PASSED TO THE PROCESSOR ON A STACK.  THE "STACK
                * POINTER" IS THE INDEX (X) REGISTER AND IT POINTS
                * TO THE OPERANDS AS FOLLOWS:
                *
                *       0000    LOW MEMORY
                *         .
                *         .
                *         .
                *   X-> LS  ARG2
                *       ARG2
                *       MS  ARG2
                *       EXP ARG2
                *       LS  ARG1
                *       ARG1
                *       MS  ARG1
                *       EXP ARG1
                *         .
                *         .
                *         .
                *       FFFF    HIGH MEMORY
                *
                *
                * THE X REGISTER IS RETURNED POINTING TO THE RESULT
                * (LS BYTE).  ARGUMENT 2 (IF ANY) IS REMOVED FROM
                * THE STACK AND ARGUMENT 1 IS OVERWRITTEN BY THE
                * RESULT.
                * THE BEST WAY TO TRANSFER OPERANDS TO THE STACK IS
                * SHOWN BELOW (ASSUMING X CONTAINS THE CURRENT OP-
                * OPERAND STACK TOP):
                *
                *        STS    SP          SAVE STACK POINTER
                *        TXS                TRANSFER CURRENT STACK
                *        LDX    #OPERAND    POINT TO EXP OF OPERAND
                *        LDA A  0,X
                *        PSH A
                *        LDA A  1,X
                *        PSH A
                *        LDA A  2,X
                *        PSH A
                *        LDA A  3,X
                *        PSH A              PUSH ONTO STACK
```

```
LOCN B1 B2 B3
                *         TSX                 RETURN NEW STACK POINTER
                *         LDS     SP          RESTORE ORIGNAL SP
                *
                *
                *  OPERAND FORMAT
                * AS IMPLIED ABOVE THE OPERANDS OCCUPY 4 BYTES.
                * FLOATING POINT VALUES CONSIST OF A 3 BYTE MAN-
                * TISSA (SIGN + MAGNITUDE) AND A ONE BYTE EXPONENT
                * (EXCESS 128 NOTATION).  NORMALIZED FORM ASSUMES
                * THE BINARY POINT TO THE RIGHT OF THE SIGN BIT
                * (MOST SIGNIFICANT BIT OF MOST SIGNIFICANT BYTE)
                * OF THE MANTISSA.  THIS RESTRICTS THE MAGNITUDE OF
                * THE MANTISSA TO  .5<= MANTISSA < 1.0.  THE EX-
                * PONENT IS THE POWER OF 2 BY WHICH THE MANTISSA
                * IS SCALED TO OBTAIN THE DESIRED VALUE.  EXCESS
                * 128 NOTATION MEANS REGULAR 2'S COMPLEMENT NOTATION
                * + 128 ($80).  THE EFFECTIVE RANGE OF FLOATING
                * POINT VALUES IS 10^+OR-38.
                * INTEGER VALUES CONSIST OF 3 BYTES (SIGN+MAGNITUDE)
                * WITH THE FOURTH BYTE (CORRESPONDING TO THE EXPONENT
                * IN FLOATING POINT VALUES) BEING ZERO.  THE RANGE OF
                * INTEGER VALUES IS 0<=INTEGER<=8388607.
                *
                * AVAILABLE OPERATIONS:
                *  FPADD  $0125  2 OPERANDS  FLOATING POINT SUM
                *  FPSUB  $011D  2 OPERANDS  FLOATING POINT DIFFERENCE
                *  FPMUL  $01A5  2 OPERANDS  FLOATING POINT PRODUCT
                *  FPDIV  $01C4  2 OPERANDS  FLOATING POINT QUOTIENT
                *  IADD   $0338  2 OPERANDS  INTEGER SUM
                *  ISUB   $0332  2 OPERANDS  INTEGER DIFFERENCE
                *  IMUL   $02EC  2 OPERANDS  INTEGER PRODUCT
                *  IDIV   $0313  2 OPERANDS  INTEGER QUOTIENT
                *  FIX    $03E5  1 OPERAND   CONVERT FP TO INTEGER
                *  FLOAT  $0414  1 OPERAND   CONVERT INTEGER TO FP
                *  SIGNUM $0379  1 OPERAND   COMPUTE SIGNUM FUNCTION
                *                            (INTEGER RESULT)
                *  ABSVAL $0372  1 OPERAND   ABSOLUTE VALUE
                *  RANDOM $0391  0 OPERANDS  RETURN A RANDOM FP NUMBER
                *                            BETWEEN 0 AND 1
                *  ITOA   $043B  1 OPERAND   CONVERT INTEGER TO ASCII
                *  FTOA   $04AA  1 OPERAND   CONVERT FP TO ASCII
                *  ATOI   $05D9  0 OPERANDS  CONVERT ASCII TO INTEGER
                *  ATOF   $0554  0 OPERANDS  CONVERT ASCII TO FP
                *  COPYUP $02D9  1 OPERAND   DUPLICATE THE TOP OPERAND
                *
                *   SEE DESCRIPTIONS OF THE INDIVIDUAL ROUTINES FOR
                * FURTHER DETAILS.
                *
                * EXAMPLES OF NORMALIZED FORM
                *     0         00 00 00 00
                *     .5        80 40 00 00
                *    -.5        80 C0 00 00
                *    12.0       84 60 00 00
                *
                * TIMING INFORMATION   (MACHINE CYCLES)
```

```
LOCN B1 B2 B3
                *        OPERATION    BEST CASE     WORST CASE
                *         FPADD         336           1450
                *         FPSUB         344           1458
                *         FPMUL        1702           3252
                *         FPDIV        3252           4602
                *         IADD          290            328
                *         ISUB          298            336
                *         IMUL         1702           3202
                *         IDIV         3302           4652
                *         FIX            46            664
                *         FLOAT          46           1275
                *         SIGNUM         51             77
                *         ABSVAL         18             18
                *         RANDOM                      2821
                *         FTOA                       67371
                *         ITOA                       47871
                *         ATOF                       29871
                *         ATOI                       15871
                *
                *  DESIGN PHILOSOPHY
                * THIS PACKAGE WAS DESIGNED FOR EASE OF USE AND
                * MAXIMUM OPERATING SPEED.  THE USER WILL NOTICE
                * THAT THE USUAL BYTE SAVING TRICKS ARE NOT USED
                * HERE.  BY THE SAME TOKEN, LOOPS WERE AVOIDED
                * EXCEPT WHERE ABSOLUTELY NECESSARY OR WHERE
                * SPEED WAS MODERATELY UNIMPORTANT (ASCII CON-
                * VERSION).
                *
                *  ILLEGAL OPERATIONS
                * THIS PACKAGE RETURNS FOUR DIFFERENT ERROR NUMBERS
                * TO INDICATE ILLEGAL OPERATIONS.  THE ERROR TYPES
                * ARE AS SHOWN BELOW.  IF THE BYTE "ERROR" IS ZERO
                * NO ERROR HAS OCCURRED AND RESULTS ARE VALID.  IF
                * ONE OF THE ERRORS BELOW IS RETURNED, RESULTS ARE
                * MEANINGLESS BUT THE STACK IS LEFT IN THE PROPER
                * CONDITION, THAT IS, AS IF NO ERROR HAD OCCURRED.
                *
                *
                *
                *
                * ERROR TYPES
                *   1   ARITHMETIC OVERFLOW
                *   2   DIVIDE BY ZERO
                *   3   NUMBER TOO LARGE TO FIX
                *   4   ASCII CONVERSION ERROR
                *
                *
                *
                *
                *
                *
                *
                *
                *
                        ORG    $40
0040            SP      RMB    2          TEMP STACK PTR
```

```
LOCN B1 B2 B3
0042            RSIGN   RMB     1
0043            ACSIGN  RMB     1
0044            AXSIGN  RMB     1
0045            EXTEND  RMB     1
0046            ERROR   RMB     1
0047            TERROR  RMB     1
                *
0048            OBUFPT  RMB     2
004A            STACK   RMB     2
004C            CONPTR  RMB     2
004E            E10     RMB     1
004F            E1      RMB     1
0050            ESIGN   RMB     1
0051            RNDM    RMB     4
0055            OUTBUF  RMB     11
                *
                        ORG     $100
                *
                *
                *  CONSTANTS
                * ( FOR ASCII CONVERSION )
0100 A2 4A      TENTEN  FDB     $A24A,$817C
0102 81 7C
0104 84 50      TENONE  FDB     $8450,$0000
0106 00 00
0108 0F         CONST   FCB     $0F,$42,$40
0109 42
010A 40
010B 01                 FCB     $01,$86,$A0
010C 86
010D A0
010E 00                 FCB     $00,$27,$10
010F 27
0110 10
0111 00                 FCB     $00,$03,$E8
0112 03
0113 E8
0114 00                 FCB     $00,$00,$64
0115 00
0116 64
0117 00                 FCB     $00,$00,$0A
0118 00
0119 0A
011A 00                 FCB     $00,$00,$01
011B 00
011C 01
                *
                *
                ** FPSUB
                * FLOATING POINT SUBTRACT
                *   ARG1-ARG2
011D A6 02      FPSUB   LDA A   2,X         GET MS AX
011F 27 04              BEQ     FPADD       IF 0, OK
0121 88 80              EOR A   #$80        CHANGE SIGN
0123 A7 02              STA A   2,X         PUT BACK
```

```
LOCN B1 B2 B3
                    *
                    *
                    ** FPADD
                    * FLOATING POINT ADD
                    *   ARG1+ARG2
0125 BD 02 AF  FPADD   JSR     FIXUP     GO FIX STACK
0128 A6 03             LDA A   3,X       GET AC EXP
012A A0 0A     FPADD2  SUB A   10,X      GET DIFFERENCE
012C 27 2B             BEQ     FPADD7    IF SAME, GO ADD
012E 2A 14             BPL     FPADD4
0130 67 02     FPADD3  ASR     2,X
0132 66 01             ROR     1,X
0134 66 00             ROR     0,X       SHIFT IT
0136 4C                INC A             COUNT OFF
0137 27 20             BEQ     FPADD7    IF DIFFERENCE 0, DONE
0139 E6 00             LDA B   0,X
013B EA 01             ORA B   1,X
013D EA 02             ORA B   2,X       CHECK ZERO
013F 27 48             BEQ     FIXEND    IF SO, NO ADD
0141 7E 01 30          JMP     FPADD3
0144 E6 03     FPADD4  LDA B   3,X
0146 E7 0A             STA B   10,X      FIX NEW EXPONENT
0148 67 09     FPADD5  ASR     9,X
014A 66 08             ROR     8,X
014C 66 07             ROR     7,X       SHIFT IT
014E 4A                DEC A             COUNT OFF
014F 27 08             BEQ     FPADD7    IF DIFFERENCE 0, DONE
0151 E6 09             LDA B   9,X
0153 EA 08             ORA B   8,X
0155 EA 07             ORA B   7,X       SEE IF ZERO
0157 26 EF             BNE     FPADD5
0159 96 42     FPADD7  LDA A   RSIGN      GET INDICATOR
015B 2B 06             BMI     FPADD9    CHECK IF DIFFERENT SIGNS
015D BD 01 F8  FPAD7D  JSR     UADD      GO ADD
0160 7E 01 76          JMP     FPAD9C
0163 BD 01 E5  FPADD9  JSR     USUB      GO SUBTRACT
0166 96 43             LDA A   ACSIGN    GET PROPER SIGN
0168 24 0E             BCC     FPAD9B    IF NO CARRY, AX<AC
016A 4F                CLR A
016B 5F                CLR B             PREPARE FOR NEGATE
016C 60 07             NEG     7,X
016E A2 08             SBC A   8,X
0170 A7 08             STA A   8,X
0172 E2 09             SBC B   9,X
0174 E7 09             STA B   9,X       TWO'S COMPLEMENT
0176 96 44     FPAD9C  LDA A   AXSIGN    USE AX SIGN
0178 BD 02 6D  FPAD9B  JSR     NORMO     GO NORMALIZE
017B D6 45     FPAD9A  LDA B   EXTEND    GET EXTENSION
017D 26 12     FPAD10  BNE     OVER
017F 84 80     FPAD11  AND A   #$80       MASK BIT
0181 16                TAB               SAVE
0182 A6 09             LDA A   9,X       GET MS BYTE
0184 27 03             BEQ     FIXEND    IF ZERO, NO SIGN
0186 1B                ABA               TACK ON SIGN
0187 A7 09             STA A   9,X       PUT BACK
```

```
LOCN B1 B2 B3
                        *
0189 08                 FIXEND  INX
018A 08                         INX
018B 08                         INX
018C 08                         INX
018D 08                         INX
018E 08                         INX
018F 08                         INX
0190 39                         RTS
                        *
0191 2B 07              OVER    BMI     NOVER       IF UNDERFLOW SET 0
0193 86 01              OVER3   LDA A   #1
0195 97 46                      STA A   ERROR       SET ERROR NUMBER
0197 7E 01 89                   JMP     FIXEND      GO FINISH
019A 6F 09              NOVER   CLR     9,X
019C 6F 08                      CLR     8,X
019E 6F 07                      CLR     7,X
01A0 6F 0A                      CLR     10,X
01A2 7E 01 89                   JMP     FIXEND      GO FINISH
                        *
                        *
                        ** FPMUL
                        * FLOATING POINT MULTIPLY
                        *   ARG1*ARG2
01A5 BD 02 AF           FPMUL   JSR     FIXUP       GO FIX AREA
01A8 BD 02 0B                   JSR     UMUL        GO MULTIPLY
01AB BD 02 6F                   JSR     NORM        GO NORMALIZE
01AE D6 45                      LDA B   EXTEND      GET EXTENSION
01B0 A6 0A                      LDA A   10,X        GET EXPONENT
01B2 AB 03                      ADD A   3,X         ADD ON OTHER
01B4 C9 00                      ADC B   #0          PROPAGATE
01B6 80 80                      SUB A   #$80        TAKE OUT ONE BIAS
01B8 C2 00                      SBC B   #0          PROPAGATE
01BA D7 45                      STA B   EXTEND      SAVE EXTENSION
01BC A7 0A                      STA A   10,X        SAVE NEW EXPONENT
01BE 96 42                      LDA A   RSIGN       GET SIGN
01C0 5D                         TST B               SET FLAGS
01C1 7E 01 7D                   JMP     FPAD10
                        *
                        *
                        ** FPDIV
                        * FLOATING POINT DIVIDE
                        *   ARG1/ARG2
01C4 BD 02 AF           FPDIV   JSR     FIXUP       GO FIX AREA
01C7 6F 04                      CLR     4,X
01C9 6F 05                      CLR     5,X
01CB 6F 06                      CLR     6,X
01CD BD 02 37                   JSR     UDIV        GO DIVIDE
01D0 96 45                      LDA A   EXTEND      GET EXTENSION
01D2 E6 0A                      LDA B   10,X        GET EXPONENT
01D4 E0 03                      SUB B   3,X         SUBTRACT OTHER
01D6 82 00                      SBC A   #0          PROPAGATE
01D8 CB 80                      ADD B   #$80        ADD BACK IN BIAS
01DA 89 00                      ADC A   #0          FIX EXTENSION
01DC 97 45                      STA A   EXTEND      SAVE
```

```
LOCN B1 B2 B3
01DE E7 0A          STA B  10,X      STORE NEW EXPONENT
01E0 96 42          LDA A  RSIGN     GET RESULT SIGN
01E2 7E 01 78       JMP    FPAD9B    GO NORMALIZE
                    *
                    *
                    *
                    * UNSIGNED ARITHMETIC OPERATIONS
                    *
01E5 A6 07      USUB    LDA A  7,X
01E7 A0 00              SUB A  0,X
01E9 A7 07              STA A  7,X
01EB A6 08              LDA A  8,X
01ED A2 01              SBC A  1,X
01EF A7 08              STA A  8,X
01F1 A6 09              LDA A  9,X
01F3 A2 02              SBC A  2,X
01F5 A7 09              STA A  9,X      SUBTRACT EM
01F7 39                 RTS
                    *
01F8 A6 07      UADD    LDA A  7,X
01FA AB 00              ADD A  0,X
01FC A7 07              STA A  7,X
01FE A6 08              LDA A  8,X
0200 A9 01              ADC A  1,X
0202 A7 08              STA A  8,X
0204 A6 09              LDA A  9,X
0206 A9 02              ADC A  2,X
0208 A7 09              STA A  9,X      ADD EM UP
020A 39                 RTS
                    *
020B C6 17      UMUL    LDA B  #23      SET COUNTER
020D A6 09      UMUL0   LDA A  9,X
020F A7 06              STA A  6,X
0211 A6 08              LDA A  8,X
0213 A7 05              STA A  5,X
0215 A6 07              LDA A  7,X
0217 A7 04              STA A  4,X
0219 6F 09              CLR    9,X
021B 6F 08              CLR    8,X
021D 6F 07              CLR    7,X
021F A6 04      UMUL1   LDA A  4,X
0221 44                 LSR A           CHECK BIT
0222 24 03              BCC    UMUL2
0224 BD 01 F8           JSR    UADD     ADD IT IN
0227 64 09      UMUL2   LSR    9,X
0229 66 08              ROR    8,X
022B 66 07              ROR    7,X
022D 66 06              ROR    6,X
022F 66 05              ROR    5,X
0231 66 04              ROR    4,X      SHIFT EM
0233 5A                 DEC B           COUNT DOWM
0234 26 E9              BNE    UMUL1
0236 39                 RTS
                    *
0237 C6 18      UDIV    LDA B  #24      SET COUNTER
```

```
LOCN B1 B2 B3
0239 A6 00      UDIV0   LDA A   0,X
023B AA 01              ORA A   1,X
023D AA 02              ORA A   2,X         CHECK ZERO
023F 27 27              BEQ     OVER1
0241 BD 01 E5   UDIV1   JSR     USUB        GO SUBTRACT
0244 24 03              BCC     UDIV2       IF WENT , OK
0246 BD 01 F8           JSR     UADD        GO ADD BACK
0249 69 04      UDIV2   ROL     4,X
024B 69 05              ROL     5,X
024D 69 06              ROL     6,X
024F 69 07              ROL     7,X
0251 69 08              ROL     8,X
0253 69 09              ROL     9,X         SHIFT EM
0255 5A                 DEC B               KICK COUNTER
0256 26 E9              BNE     UDIV1       LOOP
0258 A6 04              LDA A   4,X
025A 43                 COM A
025B A7 07              STA A   7,X
025D A6 05              LDA A   5,X
025F 43                 COM A
0260 A7 08              STA A   8,X
0262 A6 06              LDA A   6,X
0264 43                 COM A
0265 A7 09              STA A   9,X         MOVE AND CORRECT
0267 39                 RTS
0268 86 02      OVER1   LDA A   #2
026A 97 46              STA A   ERROR       SET ERROR
026C 39                 RTS                 DONN
                *
                ** NORM
                * FLOATING POINT NORMALIZE
                *   GENERATES VALUE  .5<= N <1.0
026D 6F 06      NORM0   CLR     6,X
026F E6 09      NORM    LDA B   9,X         GET MS
0271 2B 1C              BMI     NORM3       SEE IF SHIFT RIGHT
0273 EA 08              ORA B   8,X
0275 EA 07              ORA B   7,X         SEE IF 0
0277 26 03              BNE     NORM2
0279 E7 0A              STA B   10,X        SET EXPONENT
027B 39         NORM1   RTS
027C E6 09      NORM2   LDA B   9,X
027E 58                 ASL B               SEE IF SHIFT LEFT
027F 2B FA              BMI     NORM1
0281 68 06              ASL     6,X
0283 69 07              ROL     7,X
0285 69 08              ROL     8,X
0287 69 09              ROL     9,X         TO THE LEFT
0289 BD 02 A2           JSR     DEXP        FIX EXPONENT
028C 7E 02 7C           JMP     NORM2       CHECK AGAIN
028F 64 09      NORM3   LSR     9,X
0291 66 08              ROR     8,X
0293 66 07              ROR     7,X         SHIFT IT
                *
                *
0295 C6 01      IEXP    LDA B   #1
```

```
LOCN B1 B2 B3
0297 EB 0A            ADD B  10,X      INCREMENT EXP
0299 E7 0A            STA B  10,X
029B C6 00            LDA B  #0
029D D9 45            ADC B  EXTEND    FIX EXTENSION
029F D7 45            STA B  EXTEND
02A1 39               RTS
                *
02A2 E6 0A      DEXP  LDA B  10,X
02A4 C0 01            SUB B  #1        DECREMENT
02A6 E7 0A            STA B  10,X
02A8 D6 45            LDA B  EXTEND    GET EXTENSION
02AA C2 00            SBC B  #0        PROPAGATE
02AC D7 45            STA B  EXTEND
02AE 39               RTS
                *
                *
                **FIXUP
                * VACATES A 3 BYTE WORKSPACE FOR ARITHMETIC
                * OPERATIONS.  RESULT SIGN (ARG1.XOR.ARG2)
                * IS COMPUTED AND SAVED.  SIGNS ARE RESET.
02AF 4F         FIXUP CLR A
02B0 97 46            STA A  ERROR     SET ERROR
02B2 97 45            STA A  EXTEND    CLEAR EXTENSION
02B4 9F 40            STS    SP        SAVE SP
02B6 35               TXS              GET PTR
02B7 A6 02            LDA A  2,X       GET MS AX
02B9 97 44            STA A  AXSIGN    SAVE
02BB 16               TAB              SAVE
02BC 84 7F            AND A  #$7F      RESET SIGN
02BE 36               PSH A            TRANSFER IT
02BF A6 01            LDA A  1,X
02C1 36               PSH A            TRANSFER
02C2 A6 00            LDA A  0,X
02C4 36               PSH A            TRANSFER
02C5 A6 03            LDA A  3,X       GET EXP
02C7 A7 00            STA A  0,X       TRANSFER
02C9 30               TSX              SET INDEX
02CA 9E 40            LDS    SP        RETRIEVE
02CC E8 09            EOR B  9,X       CALCULATE SIGN
02CE D7 42            STA B  RSIGN     SAVE
02D0 A6 09            LDA A  9,X       GET ACSIGN
02D2 97 43            STA A  ACSIGN    SAVE
02D4 84 7F            AND A  #$7F
02D6 A7 09            STA A  9,X       RESET SIGN
02D8 39               RTS
                *
                *
                ** COPYUP
                * COPY AN OPERAND UPWARD ON THE STACK LEAVING
                * TWO COPIES OF THE OPERAND ON THE STACK.
02D9 9F 40      COPYUP STS    SP        SAVE SP
02DB 35               TXS              SET SP
02DC A6 03            LDA A  3,X
02DE 36               PSH A
02DF A6 02            LDA A  2,X
```

```
LOCN B1 B2 B3
02E1 36                PSH A
02E2 A6 01             LDA A  1,X
02E4 36                PSH A
02E5 A6 00             LDA A  0,X
02E7 36                PSH A
02E8 30                TSX                 RESET X
02E9 9E 40             LDS    SP           RESTORE SP
02EB 39                RTS
                  *
                  *
                  ** IMUL
                  * INTEGER MULTIPLY
                  * ARG1*ARG2
02EC BD 02 AF  IMUL    JSR    FIXUP        GO SET UP
02EF C6 18     IMUL0   LDA B  #24          SET COUNTER
02F1 BD 02 0D          JSR    UMUL0        GO MULTIPLY
02F4 E6 09             LDA B  9,X
02F6 EA 08             ORA B  8,X
02F8 EA 07             ORA B  7,X          CHECK ZERO
02FA A6 04             LDA A  4,X
02FC A7 07             STA A  7,X
02FE A6 05             LDA A  5,X
0300 A7 08             STA A  8,X
0302 A6 06             LDA A  6,X
0304 A7 09             STA A  9,X          MOVE PRODUCT
0306 2B 08             BMI    OVER2        IF NO 0, OVERFLOW
0308 5D                TST B               CHECK MS BYTE
0309 26 05             BNE    OVER2        IF BIT THERE, OVERFLOW
030B 96 42             LDA A  RSIGN
030D 7E 03 5D          JMP    IADD3        CLEAN UP
0310 7E 01 93  OVER2   JMP    OVER3        GO SET OVERFLOW
                  *
                  *
                  ** IDIV
                  * INTEGER DIVIDE
                  *   ARG1/ARG2
0313 BD 02 AF  IDIV    JSR    FIXUP        SET UP
0316 A6 09             LDA A  9,X
0318 A7 06             STA A  6,X
031A A6 08             LDA A  8,X
031C A7 05             STA A  5,X
031E A6 07             LDA A  7,X
0320 A7 04             STA A  4,X
0322 6F 09             CLR    9,X
0324 6F 08             CLR    8,X
0326 6F 07             CLR    7,X
0328 C6 19             LDA B  #25          SET COUNTER
032A BD 02 39          JSR    UDIV0        OPERATE
032D 96 42             LDA A  RSIGN        GET SIGN
032F 7E 03 5D          JMP    IADD3        GO FIX
                  *
                  *
                  ** ISUB
                  * INTEGER SUBTRACT
                  * ARG1-ARG2
```

```
LOCN B1 B2 B3
0332 A6 02      ISUB    LDA A  2,X
0334 88 80              EOR A  #$80       CHANGE SIGNS
0336 A7 02              STA A  2,X        PUT BACK
                *
                *
                ** IADD
                * INTEGER ADD
                *   ARG1+ARG2
0338 BD 02 AF   IADD    JSR    FIXUP      SET EM UP
033B 5D                 TST B             CHECK SIGNS
033C 2B 0A              BMI    IADD2      IF DIFFERENT, SKIP
033E BD 01 F8           JSR    UADD       GO ADD UP
0341 2B CD              BMI    OVER2      IF BIT SET, OVERFLOW
0343 96 43              LDA A  ACSIGN     GET SIGN
0345 7E 03 5D           JMP    IADD3
0348 BD 01 E5   IADD2   JSR    USUB       GO SUBTRACT
034B 96 43              LDA A  ACSIGN     GET SIGN
034D 24 0E              BCC    IADD3      IF AX>AC, SKIP
034F 4F                 CLR A
0350 5F                 CLR B
0351 60 07              NEG    7,X
0353 E2 08              SBC B  8,X
0355 E7 08              STA B  8,X
0357 A2 09              SBC A  9,X
0359 A7 09              STA A  9,X        COMPLEMENT
035B 96 44              LDA A  AXSIGN     GET OTHER SIGN
035D 84 80      IADD3   AND A  #$80       MASK DOWN
035F E6 09              LDA B  9,X        GET EXP
0361 1B                 ABA               TACK ON SIGN
0362 EA 08              ORA B  8,X
0364 EA 07              ORA B  7,X
0366 27 02              BEQ    IADD4      IF ZERO, NO SIGN
0368 A7 09              STA A  9,X
036A 08         IADD4   INX
036B 08                 INX
036C 08                 INX
036D 08                 INX
036E 08                 INX
036F 08                 INX
0370 08                 INX               DELETE ENTRIES
0371 39                 RTS               DONE
                *
                *
                ** ABSVAL
                * ABSOLUTE VALUE OF ARG1
0372 A6 02      ABSVAL  LDA A  2,X        GET SIGN
0374 84 7F              AND A  #$7F       SET PLUS
0376 A7 02              STA A  2,X        STORE
0378 39                 RTS               DONE
                *
                *
                ** SIGNUM
                * SIGNUM FUNCTION OF ARG1
                *   RETURNS THE FOLLOWING INTEGER VALUES
                *
```

```
LOCN B1 B2 B3
                    *       ARG1        VALUE
                    *     NEGATIVE       -1
                    *       ZERO          0
                    *     POSITIVE       +1
                    *
0379 A6 02          SIGNUM  LDA A  2,X
037B 16                     TAB                 SAVE
037C AA 01                  ORA A  1,X
037E AA 00                  ORA A  0,X          CHECK ZERO
0380 6F 03                  CLR    3,X
0382 6F 02          SIGNU2  CLR    2,X
0384 6F 01                  CLR    1,X
0386 6F 00                  CLR    0,X
0388 4D                     TST A               SEE IF ZERO
0389 27 05                  BEQ    SIGNU1
038B 6C 00                  INC    0,X          SET ONE
038D 58                     ASL B
038E 66 02                  ROR    2,X          FIX SIGN
0390 39             SIGNU1  RTS                 DONE
                    *
                    *
                    ** RANDOM
                    * RETURNS A RANDOM VALUE ON THE STACK
                    * RANGE OF VALUES  0.0<= N <=1.0
0391 C6 08          RANDOM  LDA B  #8           SET LOOP COUNTER
0393 96 51          RNDLP   LDA A  RNDM
0395 48                     ASL A
0396 48                     ASL A
0397 48                     ASL A
0398 98 51                  EOR A  RNDM         CALCULATE SHIFT
039A 48                     ASL A
039B 48                     ASL A               GET TO CARRY
039C 79 00 54               ROL    RNDM+3
039F 79 00 53               ROL    RNDM+2
03A2 79 00 52               ROL    RNDM+1
03A5 79 00 51               ROL    RNDM
03A8 5A                     DEC B               KICK COUNT
03A9 26 E8                  BNE    RNDLP        LOOP TILL DONE
03AB 09                     DEX
03AC 09                     DEX
03AD 09                     DEX
03AE 09                     DEX
03AF 09                     DEX
03B0 09                     DEX
03B1 09                     DEX
03B2 09                     DEX
03B3 09                     DEX
03B4 09                     DEX
03B5 09                     DEX
03B6 09                     DEX                 MAKE ROOM
03B7 4F                     CLR A
03B8 A7 00                  STA A  0,X
03BA A7 01                  STA A  1,X
03BC A7 08                  STA A  8,X
03BE A7 09                  STA A  9,X          SET ZERO
```

```
LOCN B1 B2 B3
03C0 86 80          LDA A  #$80
03C2 A7 03          STA A  3,X
03C4 A7 07          STA A  7,X       SET EXPONENTS
03C6 86 82          LDA A  #$82
03C8 A7 0B          STA A  11,X      SET SCALE EXP
03CA 86 40          LDA A  #$40
03CC A7 02          STA A  2,X
03CE A7 0A          STA A  10,X      SET MS MANTISSA
03D0 D6 51          LDA B  RNDM
03D2 C4 3F          AND B  #$3F      MASK DOWN
03D4 1B             ABA              NORMALIZE
03D5 A7 06          STA A  6,X
03D7 96 52          LDA A  RNDM+1
03D9 A7 05          STA A  5,X
03DB 96 53          LDA A  RNDM+2
03DD A7 04          STA A  4,X
03DF BD 01 1D       JSR    FPSUB     SHIFT 0-1/2
03E2 7E 01 A5       JMP    FPMUL     SCALE 0-1
                    *
                    *
                    ** FIX
                    * CONVERT AG1 TO INTEGER
                    * ERROR 3 IF ARG1>8388607
03E5 4F             FIX    CLR A
03E6 97 46          STA A  ERROR     RESET ERROR
03E8 E6 03          LDA B  3,X       GET EXPONENT
03EA A7 03          STA A  3,X       SET ZERO
03EC C0 81          SUB B  #$81      REMOVE BIAS
03EE 25 92          BCS    SIGNU2    IF <0 SET 0
03F0 50             NEG B
03F1 CB 16          ADD B  #22       FUDGE
03F3 27 19          BEQ    FIX2      IF, NO SHIFTS, DONE
03F5 2B 18          BMI    FIX3      IF EXP >2^23, OVERFL
03F7 A6 02          LDA A  2,X       GET SIGN
03F9 36             PSH A            SAVE
03FA 84 7F          AND A  #$7F      MASK
03FC A7 02          STA A  2,X       STOR
03FE 64 02          FIX1   LSR  2,X
0400 66 01          ROR    1,X
0402 66 00          ROR    0,X       MOVE IT
0404 5A             DEC B            COUNT OFF
0405 26 F7          BNE    FIX1      LOOP TILL DONE
0407 32             PUL A            GET SIGN
0408 84 80          AND A  #$80      MASK
040A AA 02          ORA A  2,X       TACK ON
040C A7 02          STA A  2,X       STORE
040E 39             FIX2   RTS       DONE
040F 86 03          FIX3   LDA A  #3
0411 97 46          STA A  ERROR     SET ERROR
0413 39             RTS
                    *
                    *
                    ** FLOAT
                    * CONVERT ARG1 TO FLOATING POINT
0414 7F 00 46       FLOAT  CLR    ERROR     RESET ERROR
```

```
LOCN B1 B2 B3
0417 E6 02          LDA B  2,X        GET SIGN
0419 86 97          LDA A  #$97
041B A7 03          STA A  3,X        SET EXPONENT
041D A6 02     FLOAT1 LDA A  2,X       ET MS BYTE
041F 48             ASL A
0420 2B 11          BMI    FLOAT2     IF BIT, DONE
0422 AA 01          ORA A  1,X
0424 AA 00          ORA A  0,X
0426 27 10          BEQ    FLOAT3
0428 68 00          ASL    0,X
042A 69 01          ROL    1,X
042C 69 02          ROL    2,X
042E 6A 03          DEC    3,X
0430 7E 04 1D       JMP    FLOAT1
0433 58        FLOAT2 ASL B
0434 46             ROR A              SET SIGN
0435 A7 02          STA A  2,X         STORE
0437 39             RTS                DONE
0438 A7 03     FLOAT3 STA A  3,X        SET EXPONENT
043A 39             RTS                DONE
                    *
                    *
                    ** ITOA
                    * CONVERTS ARG1 TO ASCII IN OUTBUF
                    * AND REMOVES FROM STACK
                    * OUTBUF FORMAT IS SIGN FOLLOWED BY 7
                    * DECIMAL ASCII DIGITS
043B DF 4A     ITOA   STX    STACK      SAVE PTR
043D A6 02            LDA A  2,X        GET SIGN
043F 16               TAB
0440 C4 7F            AND B  #$7F       RESET
0442 E7 02            STA B  2,X        FIX
0444 CE 01 08         LDX    #CONST
0447 DF 4C            STX    CONPTR     SET POINTER
0449 CE 00 55         LDX    #OUTBUF    POINT TO BUFFER
044C C6 2B     ITOA0  LDA B  #'+        GET PLUS
044E E7 00            STA B  0,X
0450 4D               TST A              CHECK SIGN
0451 2A 04            BPL    ITOA1
0453 C6 2D            LDA B  #'-
0455 E7 00            STA B  0,X
0457 C6 07     ITOA1  LDA B  #7          SET COUNT
0459 08        ITOA2  INX                BUMP POINTER
045A DF 48            STX    OBUFPT     SAVE IT
045C DE 4A            LDX    STACK      GET OTHER PTR
045E 9F 40            STS    SP         SAVE SP
0460 35               TXS               TRANSFER
0461 A6 03            LDA A  3,X
0463 36               PSH A
0464 A6 02            LDA A  2,X
0466 36               PSH A
0467 A6 01            LDA A  1,X
0469 36               PSH A
046A A6 00            LDA A  0,X
046C 36               PSH A              COPY UP
```

```
LOCN B1 B2 B3
046D DE 4C              LDX     CONPTR    GET CONSTANT PTR
046F 4F                 CLR A
0470 36                 PSH A
0471 A6 00              LDA A   0,X
0473 36                 PSH A
0474 A6 01              LDA A   1,X
0476 36                 PSH A
0477 A6 02              LDA A   2,X
0479 36                 PSH A
047A 08                 INX
047B 08                 INX
047C 08                 INX               ADVANCE PTR
047D DF 4C              STX     CONPTR
047F 30                 TSX               RESET
0480 9E 40              LDS     SP        RESTORE SP
0482 37                 PSH B             SAVE COUNT
0483 BD 03 13           JSR     IDIV      GO DIVIDE
0486 A6 00              LDA A   0,X       GET QUOTIENT
0488 8B 30              ADD A   #$30      ADD ASCII
048A 36                 PSH A             SAVE
048B 09                 DEX
048C 09                 DEX
048D 09                 DEX
048E 09                 DEX
048F 09                 DEX
0490 09                 DEX
0491 09                 DEX               ADJUST FOR MUL
0492 BD 02 EF           JSR     IMUL0     GO MULTIPLY
0495 BD 03 32           JSR     ISUB      GO SUBTRACT
0498 32                 PUL A
0499 33                 PUL B
049A DF 4A              STX     STACK     SAVE PTR
049C DE 48              LDX     OBUFPT    GET OUTPUT PTR
049E A7 00              STA A   0,X       SAVE
04A0 5A                 DEC B             KICK COUNTER
04A1 26 B6              BNE     ITOA2
04A3 DE 4A              LDX     STACK     GET PTR
04A5 08                 INX
04A6 08                 INX
04A7 08                 INX
04A8 08                 INX               DELETE ENTRY
04A9 39                 RTS
                *
                *
                ** FTOA
                * CONVERTS ARG1 TO ASCII IN OUTBUF
                * AND REMOVES FROM STACK.  OUTBUF FORMAT
                * SAME AS FOR ITOA FOLLOWED BY EXPONENT
                * SIGN AND 2 DIGIT POWER OF TEN IN THAT
                * ORDER
04AA 86 30      FTOA    LDA A   #$30      SET ASCII
04AC 97 4E              STA A   E10
04AE 97 4F              STA A   E1        SET UP EXPONENT DIGITS
04B0 86 2B              LDA A   #'+
04B2 97 50              STA A   ESIGN     SET SIGN
```

```
LOCN B1 B2 B3
04B4 A6 02          LDA A  2,X
04B6 27 39          BEQ    FTOA3      IF 0, DONE
04B8 A6 03          LDA A  3,X        GET SIGN
04BA 81 80          CMP A  #$80
04BC 22 4C          BHI    FTOA5
04BE 86 2D          LDA A  #'-
04C0 97 50          STA A  ESIGN      SET ASCII
04C2 A6 03    FTOA1 LDA A  3,X        GET EXPONENT
04C4 81 76          CMP A  #$76
04C6 22 0B          BHI    FTOA2
04C8 BD 05 39       JSR    PSHTEN     SET STACK
04CB BD 01 A5       JSR    FPMUL      GO SCALE
04CE 7C 00 4E       INC    E10        ADD EXPONENT
04D1 20 EF          BRA    FTOA1
04D3 A6 03    FTOA2 LDA A  3,X        GET EXPONENT
04D5 81 93          CMP A  #$93
04D7 22 18          BHI    FTOA3
04D9 BD 05 4C       JSR    PSHONE     SET STACK
04DC BD 01 A5       JSR    FPMUL      SCALE
04DF 96 4F          LDA A  E1         GET EXPONENT
04E1 4C             INC A             ADD ONE
04E2 97 4F          STA A  E1
04E4 81 3A          CMP A  #$3A       CHECK OVERFLOW
04E6 26 EB          BNE    FTOA2
04E8 86 30          LDA A  #$30
04EA 97 4F          STA A  E1         RESET TO 0
04EC 7C 00 4E       INC    E10        INCREMENT HIGH
04EF 20 E2          BRA    FTOA2
04F1 BD 03 E5 FTOA3 JSR    FIX        GO FIX NUMBER
04F4 BD 04 3B FTOA3C JSR   ITOA       CONVERT
04F7 DF 4A          STX    STACK
04F9 DE 48          LDX    OBUFPT     GET PTR
04FB 96 50          LDA A  ESIGN
04FD A7 01          STA A  1,X        STORE SIGN
04FF 96 4E          LDA A  E10        GET EXP (10)
0501 A7 02          STA A  2,X
0503 96 4F          LDA A  E1
0505 A7 03          STA A  3,X        SET EXP (1)
0507 DE 4A          LDX    STACK
0509 39             RTS               DONE
050A A6 03    FTOA5 LDA A  3,X        GET
050C 81 B5          CMP A  #$B5       CHECK
050E 25 0B          BCS    FTOA6
0510 BD 05 39       JSR    PSHTEN
0513 BD 01 C4       JSR    FPDIV      REMOVE FACTORS OF TEN
0516 7C 00 4E       INC    E10
0519 20 EF          BRA    FTOA5
051B A6 03    FTOA6 LDA A  3,X        GET EXP
051D 81 97          CMP A  #$80+23
051F 23 D0          BLS    FTOA3
0521 BD 05 4C       JSR    PSHONE     SET STACK
0524 BD 01 C4       JSR    FPDIV      GO REMOVE FACTORS
0527 96 4F          LDA A  E1
0529 4C             INC A
052A 97 4F          STA A  E1
```

```
LOCN B1 B2 B3
052C 81 3A            CMP A  #$3A
052E 26 EB            BNE    FTOA6
0530 7C 00 4E         INC    E10
0533 86 30            LDA A  #$30
0535 97 4F            STA A  E1
0537 20 E2            BRA    FTOA6
                 *
                 **
0539 9F 40       PSHTEN STS    SP
053B 35                 TXS                 SET POINTERS
053C CE 01 00           LDX    #TENTEN      POINT CONSTANT
053F C6 04       PSH1   LDA B  #4           SET COUNTER
0541 A6 00       PSH2   LDA A  0,X          GET DATA
0543 36                 PSH A               STORE
0544 08                 INX
0545 5A                 DEC B               COUNT DOWN
0546 26 F9              BNE    PSH2         LOOP TILL DONE
0548 30                 TSX                 RESET PTR
0549 9E 40              LDS    SP           RESET SP
054B 39                 RTS
                 *
                 **
054C 9F 40       PSHONE STS    SP           SAVE
054E 35                 TXS                 SET PTR
054F CE 01 04           LDX    #TENONE      POINT TO CONSTANT
0552 20 EB              BRA    PSH1         GO MOVE
                 *
                 **
                 *
                 *
                 ** ATOF
                 * CONVERTS ASCII IN OUTBUF TO FLOATING
                 * POINT ON STACK.  ERROR 4 RETURNED
                 * IF NUMBER TOO LARGE TO REPRESENT.
                 * FORMAT EXPECTED SAME AS PRODUCED BY
                 * FTOA
0554 BD 05 D9    ATOF   JSR    ATOI
0557 BD 04 14           JSR    FLOAT        FLOAT IT
055A DF 4A       ATOF4  STX    STACK
055C DE 48              LDX    OBUFPT
055E 5F                 CLR B
055F A6 00              LDA A  0,X
0561 81 2B              CMP A  #'+          CHECK FOR +
0563 27 01              BEQ    ATOF5
0565 53                 COM B
0566 A6 02       ATOF5  LDA A  2,X
0568 84 0F              AND A  #$0F          REMOVE BIAS
056A 97 4F              STA A  E1
056C A6 01              LDA A  1,X
056E DE 4A              LDX    STACK
0570 84 0F              AND A  #$0F
0572 97 4E              STA A  E10
0574 37                 PSH B
0575 27 2A              BEQ    ATOF9A
0577 9F 40       ATOF6  STS    SP
```

```
LOCN B1 B2 B3
0579 35          TXS
057A C6 04       LDA B  #4
057C CE 01 00    LDX    #TENTEN
057F A6 00  ATOF7  LDA A  0,X
0581 36          PSH A
0582 08          INX
0583 5A          DEC B
0584 26 F9       BNE    ATOF7
0586 30          TSX
0587 9E 40       LDS    SP
0589 33          PUL B
058A 37          PSH B
058B 5D          TST B
058C 2B 05       BMI    ATOF8
058E BD 01 A5    JSR    FPMUL
0591 20 03       BRA    ATOF9
0593 BD 01 C4  ATOF8  JSR    FPDIV
0596 96 47     ATOF9  LDA A  TERROR
0598 9A 46            ORA A  ERROR      MERGE ERRORS
059A 97 47            STA A  TERROR
059C 7A 00 4E         DEC    E10
059F 26 D6            BNE    ATOF6
05A1 96 4F     ATOF9A LDA A  E1
05A3 27 2A            BEQ    ATOF15
05A5 9F 40     ATOF10 STS    SP
05A7 35               TXS
05A8 C6 04            LDA B  #4
05AA CE 01 04         LDX    #TENONE
05AD A6 00     ATOF11 LDA A  0,X
05AF 36               PSH A
05B0 08               INX
05B1 5A               DEC B
05B2 26 F9            BNE    ATOF11
05B4 30        ATOF14 TSX
05B5 9E 40            LDS    SP
05B7 33               PUL B
05B8 37               PSH B
05B9 5D               TST B
05BA 2B 05            BMI    ATOF12
05BC BD 01 A5         JSR    FPMUL
05BF 20 03            BRA    ATOF13
05C1 BD 01 C4  ATOF12 JSR    FPDIV
05C4 96 47     ATOF13 LDA A  TERROR
05C6 9A 46            ORA A  ERROR
05C8 97 47            STA A  TERROR
05CA 7A 00 4F         DEC    E1
05CD 26 D6            BNE    ATOF10
05CF 33        ATOF15 PUL B
05D0 96 47            LDA A  TERROR
05D2 27 04            BEQ    ATOF16
05D4 86 04            LDA A  #4
05D6 97 46            STA A  ERROR
05D8 39        ATOF16 RTS               DONE
               *
               *
```

```
LOCN B1 B2 B3
                    *
                    ** ATOI
                    * CONVERTS ASCII IN OUTBUF TO INTEGER
                    * VALUE ON STACK.  ERROR 4 RETURNED IF
                    * NUMBER TOO LARGE TO REPRESENT.  FORMAT
                    * EXPECTED SAME AS PRODUCED BY ITOA.
05D9 9F 40          ATOI    STS     SP          SAVE SP
05DB 35                     TXS                 SET PTR
05DC CE 00 55               LDX     #OUTBUF     POINT TO BUFFER
05DF 4F                     CLR A
05E0 36                     PSH A
05E1 36                     PSH A
05E2 36                     PSH A
05E3 36                     PSH A
05E4 97 46                  STA A   ERROR
05E6 97 47                  STA A   TERROR
05E8 C6 80                  LDA B   #$80        SET MINUS
05EA A6 00                  LDA A   0,X         GET CHR
05EC 08                     INX
05ED DF 48                  STX     OBUFPT      SAVE PTR
05EF 30                     TSX
05F0 9E 40                  LDS     SP
05F2 81 2D                  CMP A   #'-         CHECK FOR MINUS
05F4 27 01                  BEQ     ATOI1
05F6 5F                     CLR B
05F7 D7 4E          ATOI1   STA B   E10         SAVE SIGN
05F9 C6 07                  LDA B   #7          SET COUNT
05FB 37             ATOI2   PSH B               SAVE COUNT
05FC 9F 40                  STS     SP
05FE 35                     TXS
05FF 5F                     CLR B
0600 37                     PSH B
0601 37                     PSH B
0602 37                     PSH B
0603 C6 0A                  LDA B   #10
0605 37                     PSH B               SET STACK
0606 30                     TSX
0607 9E 40                  LDS     SP
0609 BD 02 EC               JSR     IMUL        GO MULTIPLY
060C 96 46                  LDA A   ERROR
060E 9A 47                  ORA A   TERROR
0610 97 47                  STA A   TERROR
0612 9F 40                  STS     SP
0614 35                     TXS
0615 4F                     CLR A
0616 36                     PSH A
0617 36                     PSH A
0618 36                     PSH A
0619 DE 48                  LDX     OBUFPT      GET POINTER
061B A6 00                  LDA A   0,X         GET DIGIT
061D 80 30                  SUB A   #$30        REMOVE BIAS
061F 36                     PSH A               PUT IN STACK
0620 08                     INX
0621 DF 48                  STX     OBUFPT
0623 30                     TSX
```

```
LOCN B1 B2 B3
0624 9E 40          LDS     SP
0626 BD 03 38       JSR     IADD
0629 96 47          LDA A   TERROR
062B 9A 46          ORA A   ERROR
062D 97 47          STA A   TERROR
062F 33             PUL B
0630 5A             DEC B               COUNT OFF
0631 26 C8          BNE     ATOI2
0633 96 4E          LDA A   E10
0635 AA 02          ORA A   2,X
0637 A7 02          STA A   2,X
0639 A6 02    ATOI3 LDA A   2,X
063B 48             ASL A
063C AA 01          ORA A   1,X
063E AA 00          ORA A   0,X
0640 26 02          BNE     ATOI5
0642 A7 02          STA A   2,X
0644 96 47    ATOI5 LDA A   TERROR
0646 27 04          BEQ     ATOI4
0648 86 04          LDA A   #4
064A 97 46          STA A   ERROR
064C 39       ATOI4 RTS
                    END
```

SYMBOL TABLE:

```
ABSVAL 0372   ACSIGN 0043   ATOF   0554   ATOF10 05A5   ATOF11 05AD
ATOF12 05C1   ATOF13 05C4   ATOF14 05B4   ATOF15 05CF   ATOF16 05D8
ATOF4  055A   ATOF5  0566   ATOF6  0577   ATOF7  057F   ATOF8  0593
ATOF9  0596   ATOF9A 05A1   ATOI   05D9   ATOI1  05F7   ATOI2  05FB
ATOI3  0639   ATOI4  064C   ATOI5  0644   AXSIGN 0044   CONPTR 004C
CONST  0108   COPYUP 02D9   DEXP   02A2   ERROR  0046   ESIGN  0050
EXTEND 0045   E1     004F   E10    004E   FIX    03E5   FIXEND 0189
FIXUP  02AF   FIX1   03FE   FIX2   040E   FIX3   040F   FLOAT  0414
FLOAT1 041D   FLOAT2 0433   FLOAT3 0438   FPADD  0125   FPADD2 012A
FPADD3 0130   FPADD4 0144   FPADD5 0148   FPADD7 0159   FPADD9 0163
FPAD10 017D   FPAD11 017F   FPAD7D 015D   FPAD9A 017B   FPAD9B 0178
FPAD9C 0176   FPDIV  01C4   FPMUL  01A5   FPSUB  011D   FTOA   04AA
FTOA1  04C2   FTOA2  04D3   FTOA3  04F1   FTOA3C 04F4   FTOA5  050A
FTOA6  051B   IADD   0338   IADD2  0348   IADD3  035D   IADD4  036A
IDIV   0313   IEXP   0295   IMUL   02EC   IMUL0  02EF   ISUB   0332
ITOA   043B   ITOA0  044C   ITOA1  0457   ITOA2  0459   NORM   026F
NORM0  026D   NORM1  027B   NORM2  027C   NORM3  028F   NOVER  019A
OBUFPT 0048   OUTBUF 0055   OVER   0191   OVER1  0268   OVER2  0310
OVER3  0193   PSHONE 054C   PSHTEN 0539   PSH1   053F   PSH2   0541
RANDOM 0391   RNDLP  0393   RNDM   0051   RSIGN  0042   SIGNUM 0379
SIGNU1 0390   SIGNU2 0382   SP     0040   STACK  004A   TENONE 0104
TENTEN 0100   TERROR 0047   UADD   01F8   UDIV   0237   UDIV0  0239
UDIV1  0241   UDIV2  0249   UMUL   020B   UMUL0  020D   UMUL1  021F
UMUL2  0227   USUB   01E5
```

OBJECT CODE:

```
S1 13 0100 A2 4A 81 7C 84 50 00 00 0F 42 40 01 86 A0 00 27 4F
S1 13 0110 10 00 03 E8 00 00 64 00 00 0A 00 00 01 A6 02 27 A2
S1 13 0120 04 88 80 A7 02 BD 02 AF A6 03 A0 0A 27 2B 2A 14 C5
S1 13 0130 67 02 66 01 66 00 4C 27 20 E6 00 EA 01 EA 02 27 0E
S1 13 0140 48 7E 01 30 E6 03 E7 0A 67 09 66 08 66 07 4A 27 1E
S1 13 0150 08 E6 09 EA 08 EA 07 26 EF 96 42 2B 06 BD 01 F8 ED
S1 13 0160 7E 01 76 BD 01 E5 96 43 24 0E 4F 5F 60 07 A2 08 29
S1 13 0170 A7 08 E2 09 E7 09 96 44 BD 02 6D D6 45 26 12 84 14
S1 13 0180 80 16 A6 09 27 03 1B A7 09 08 08 08 08 08 08 08 F9
S1 13 0190 39 2B 07 86 01 97 46 7E 01 89 6F 09 6F 08 6F 07 1F
S1 13 01A0 6F 0A 7E 01 89 BD 02 AF BD 02 0B BD 02 6F D6 45 49
S1 13 01B0 A6 0A AB 03 C9 00 80 80 C2 00 D7 45 A7 0A 96 42 AD
S1 13 01C0 5D 7E 01 7D BD 02 AF 6F 04 6F 05 6F 06 BD 02 37 12
S1 13 01D0 96 45 E6 0A E0 03 82 00 CB 80 89 00 97 45 E7 0A 4A
S1 13 01E0 96 42 7E 01 78 A6 07 A0 00 A7 07 A6 08 A2 01 A7 49
S1 13 01F0 08 A6 09 A2 02 A7 09 39 A6 07 AB 00 A7 07 A6 08 03
S1 13 0200 A9 01 A7 08 A6 09 A9 02 A7 09 39 C6 17 A6 09 A7 1B
S1 13 0210 06 A6 08 A7 05 A6 07 A7 04 6F 09 6F 08 6F 07 A6 17
S1 13 0220 04 44 24 03 BD 01 F8 64 09 66 08 66 07 66 06 66 8B
S1 13 0230 05 66 04 5A 26 E9 39 C6 18 A6 00 AA 01 AA 02 27 A7
S1 13 0240 27 BD 01 E5 24 03 BD 01 F8 69 04 69 05 69 06 69 50
S1 13 0250 07 69 08 69 09 5A 26 E9 A6 04 43 A7 07 A6 05 43 BE
S1 13 0260 A7 08 A6 06 43 A7 09 39 86 02 97 46 39 6F 06 E6 0A
S1 13 0270 09 2B 1C EA 08 EA 07 26 03 E7 0A 39 E6 09 58 2B 82
S1 13 0280 FA 68 06 69 07 69 08 69 09 BD 02 A2 7E 02 7C 64 EE
S1 13 0290 09 66 08 66 07 C6 01 EB 0A E7 0A C6 00 D9 45 D7 0E
S1 13 02A0 45 39 E6 0A C0 01 E7 0A D6 45 C2 00 D7 45 39 4F A9
S1 13 02B0 97 46 97 45 9F 40 35 A6 02 97 44 16 84 7F 36 A6 F5
S1 13 02C0 01 36 A6 00 36 A6 03 A7 00 30 9E 40 E8 09 D7 42 AF
S1 13 02D0 A6 09 97 43 84 7F A7 09 39 9F 40 35 A6 03 36 A6 0C
S1 13 02E0 02 36 A6 01 36 A6 00 36 30 9E 40 39 BD 02 AF C6 9E
S1 13 02F0 18 BD 02 0D E6 09 EA 08 EA 07 A6 04 A7 07 A6 05 41
S1 13 0300 A7 08 A6 06 A7 09 2B 08 5D 26 05 96 42 7E 03 5D 6D
S1 13 0310 7E 01 93 BD 02 AF A6 09 A7 06 A6 08 A7 05 A6 07 F6
S1 13 0320 A7 04 6F 09 6F 08 6F 07 C6 19 BD 02 39 96 42 7E 8C
S1 13 0330 03 5D A6 02 88 80 A7 02 BD 02 AF 5D 2B 0A BD 01 42
S1 13 0340 F8 2B CD 96 43 7E 03 5D BD 01 E5 96 43 24 0E 4F 05
S1 13 0350 5F 60 07 E2 08 E7 08 A2 09 A7 09 96 44 84 80 E6 DB
S1 13 0360 09 1B EA 08 EA 07 27 02 A7 09 08 08 08 08 08 08 79
S1 13 0370 08 39 A6 02 84 7F A7 02 39 A6 02 16 AA 01 AA 00 98
S1 13 0380 6F 03 6F 02 6F 01 6F 00 4D 27 05 6C 00 58 66 02 02
S1 13 0390 39 C6 08 96 51 48 48 48 98 51 48 48 79 00 54 79 D4
S1 13 03A0 00 53 79 00 52 79 00 51 5A 26 E8 09 09 09 09 09 CC
S1 13 03B0 09 09 09 09 09 09 09 4F A7 00 A7 01 A7 08 A7 09 FD
S1 13 03C0 86 80 A7 03 A7 07 86 82 A7 0B 86 40 A7 02 A7 0A F1
S1 13 03D0 D6 51 C4 3F 1B A7 06 96 52 A7 05 96 53 A7 04 BD 42
S1 13 03E0 01 1D 7E 01 A5 4F 97 46 E6 03 A7 03 C0 81 25 92 10
S1 13 03F0 50 CB 16 27 19 2B 18 A6 02 36 84 7F A7 02 64 02 55
S1 13 0400 66 01 66 00 5A 26 F7 32 84 80 AA 02 A7 02 39 86 5A
S1 13 0410 03 97 46 39 7F 00 46 E6 02 86 97 A7 03 A6 02 48 5B
S1 13 0420 2B 11 AA 01 AA 00 27 10 68 00 69 01 69 02 6A 03 56
S1 13 0430 7E 04 1D 58 46 A7 02 39 A7 03 39 DF 4A A6 02 16 CF
S1 13 0440 C4 7F E7 02 CE 01 08 DF 4C CE 00 55 C6 2B E7 00 7F
S1 13 0450 4D 2A 04 C6 2D E7 00 C6 07 08 DF 48 DE 4A 9F 40 40
S1 13 0460 35 A6 03 36 A6 02 36 A6 01 36 A6 00 36 DE 4C 4F 64
S1 13 0470 36 A6 00 36 A6 01 36 A6 02 36 08 08 08 DF 4C 30 3B
S1 13 0480 9E 40 37 BD 03 13 A6 00 8B 30 36 09 09 09 09 09 BC
```

```
S1 13 0490 09 09 BD 02 EF BD 03 32 32 33 DF 4A DE 48 A7 00 4B
S1 13 04A0 5A 26 B6 DE 4A 08 08 08 08 39 B6 30 97 4E 97 4F 10
S1 13 04B0 86 2B 97 50 A6 02 27 39 A6 03 81 80 22 4C 86 2D CD
S1 13 04C0 97 50 A6 03 81 76 22 0B BD 05 39 BD 01 A5 7C 00 9A
S1 13 04D0 4E 20 EF A6 03 81 93 22 18 BD 05 4C BD 01 A5 96 BD
S1 13 04E0 4F 4C 97 4F 81 3A 26 EB B6 30 97 4F 7C 00 4E 20 35
S1 13 04F0 E2 BD 03 E5 BD 04 3B DF 4A DE 48 96 50 A7 01 96 02
S1 13 0500 4E A7 02 96 4F A7 03 DE 4A 39 A6 03 81 B5 25 0B F1
S1 13 0510 BD 05 39 BD 01 C4 7C 00 4E 20 EF A6 03 81 97 23 9D
S1 13 0520 D0 BD 05 4C BD 01 C4 96 4F 4C 97 4F 81 3A 26 EB 84
S1 13 0530 7C 00 4E B6 30 97 4F 20 E2 9F 40 35 CE 01 00 C6 A6
S1 13 0540 04 A6 00 36 08 5A 26 F9 30 9E 40 39 9F 40 35 CE 1D
S1 13 0550 01 04 20 EB BD 05 D9 BD 04 14 DF 4A DE 48 5F A6 C3
S1 13 0560 00 81 2B 27 01 53 A6 02 84 0F 97 4F A6 01 DE 4A 70
S1 13 0570 84 0F 97 4E 37 27 2A 9F 40 35 C6 04 CE 01 00 A6 24
S1 13 0580 00 36 08 5A 26 F9 30 9E 40 33 37 5D 2B 05 BD 01 ED
S1 13 0590 A5 20 03 BD 01 C4 96 47 9A 46 97 47 7A 00 4E 26 84
S1 13 05A0 D6 96 4F 27 2A 9F 40 35 C6 04 CE 01 04 A6 00 36 AE
S1 13 05B0 08 5A 26 F9 30 9E 40 33 37 5D 2B 05 BD 01 A5 20 2E
S1 13 05C0 03 BD 01 C4 96 47 9A 46 97 47 7A 00 4F 26 D6 33 0F
S1 13 05D0 96 47 27 04 86 04 97 46 39 9F 40 35 CE 00 55 4F E9
S1 13 05E0 36 36 36 36 97 46 97 47 C6 80 A6 00 08 DF 48 30 29
S1 13 05F0 9E 40 81 2D 27 01 5F D7 4E C6 07 37 9F 40 35 5F 48
S1 13 0600 37 37 37 C6 0A 37 30 9E 40 BD 02 EC 96 46 9A 47 C4
S1 13 0610 97 47 9F 40 35 4F 36 36 36 DE 48 A6 00 80 30 36 41
S1 13 0620 08 DF 48 30 9E 40 BD 03 38 96 47 9A 46 97 47 33 C3
S1 13 0630 5A 26 C8 96 4E AA 02 A7 02 A6 02 48 AA 01 AA 00 F0
S1 10 0640 26 02 A7 02 96 47 27 04 86 04 97 46 39 30
S9
```