## System calls

System calls to BASIC can be made through the System Call restart (RST 10H instruction).

The restart instruction is followed by inline function numbers (as with EXOS calls), but may consist of many bytes in a sequence which define the sequence of BASIC calls to be performed. The list of bytes is terminated with function number 0, which continues program execution with the following instruction.

Each byte in the sequence specified one routine, but some routines require another byte of data immediately following the function number.

Throughout this document the various system calls have been refered to mnemonically for convenience. These mnemonics are given below, together with the actual functions number and the Z80 registers used. Unless otherwise stated, the alternate register set (AF', BC', DE' and HL') are preserved. The term 'the stack' refers to BASICs system stack, not the Z80 stack.

END (0)

Terminates the sequence of system calls, to continue program execution.

Parameters:
Returns: as system call previously made
Preserves: all registers

LOADINT (2)

Puts the signed integer in HL onto BASICs stack.

Parameters: HL - number to put on stack.
Returns: IY - new value of STKPTR.
Preserves:

 

CFTOS (3)

Converts the top number on the stack into an ASCII string without removing the number from the stack. The ASCII representation of the number, with no leading or trailing spaces, but with a leading '-' if negative, is put into the buffer CHR, length byte first.

| | |
|---|---|
| Parameters: | On stack. |
| Returns: | IY - current value of STKPTR. |
| Preserves: | |

PSGN (5)

Tests the sign of the top number on the stack, without removing it from the stack.

| | |
|---|---|
| Parameters: | On stack. |
| Returns: | A - 80H if the number is -ve, else 0. |
| | Z - zero flag is set if the number is +ve. |
| | IY - current value of STKPTR. |
| Preserves: | A, BC, DE, HL |

FNEG (6)

Negates the top number on the stack.

| | |
|---|---|
| Parameters: | On stack. |
| Returns: | IY - current value of STKPTR. |
| Preserves: | BC |

?FNEG (7)

Negates the top number on the stack if it is negative, ie. takes its absolute value.

| | |
|---|---|
| Parameters: | On stack. |
| Returns: | IY - current value of STKPTR. |
| Preserves: | BC |

 

FCPL (8)

Ten's complements the top number on the stack.

| | |
|---|---|
| Parameters: | On stack. |
| Returns: | IY - current value of STKPTR. |
| Preserves: | BC |

?FCPL (9)

Tens's complements the top number on the stack if it is negative.

| | |
|---|---|
| Parameters: | On stack. |
| Returns: | IY - current value of STKPTR. |
| Preserves: | BC |

INT? (A hex., 10 decimal)

Tests the top number on the stack for being in the integer format, without removing it from the stack.

| | |
|---|---|
| Parameters: | On stack. |
| Returns: | A - exponent of top number.<br>Z - set if in the integer format.<br>IY - current value of STKPTR. |
| Preserves: | HL, DE, BC. |

FIX (B hex., 11 decimal)

Converts the top number on the stack into an integer in HL, and removes the number from the stack. An error is given if the number to too large to be converted.

| | |
|---|---|
| Parameters: | IY - points to the number on the stack. |
| Returns: | HL - the number converted. |
| Preserves: | BC, DE. |

 

FADD (C hex., 12 decimal)

Adds together the top two numbers on the stack, and replaces them with the result.

| | |
|---|---|
| Parameters: | Removed from stack. |
| Returns: | Result on stack. |
| Preserves: | AF. |

FSUB (D hex., 13 decimal)

Subtracts the top number on the stack from the second number on the stack, and replaces them with the result.

| | |
|---|---|
| Parameters: | Removed from stack. |
| Returns: | Result on stack. |
| Preserves: | AF. |

FMULT (E hex., 14 decimal)

Multiplies together the top two number on the stack, and replaces them with the result.

| | |
|---|---|
| Parameters: | Removed from stack. |
| Returns: | Result on stack. |
| Preserves: | AF. |

FDIV (F hex., 15 decimal)

Divides the second number on the stack by the top number, and replaces them by the result.

| | |
|---|---|
| Parameters: | Removed from stack. |
| Returns: | Result on stack. |
| Preserves: | AF. |

 

FCOMP (10 hex., 16 decimal)

Compares the top two numbers on the stack, and removes them.

| | |
|---|---|
| Parameters: | Removed from stack. |
| Returns: | P, M, Z flags set as though the top number were subtracted from the second number.<br>H - contains a number consistent with the above flags. |
| Preserves: | |

FDIV10 (11 hex., 17 decimal)

Divides the mantissa part of the top number by 10 by shifting it right decimally. The number must not be in the integer form.

| | |
|---|---|
| Parameters: | A - the number of times to shift. |
| Returns: | IY - current value of STKPTR. |
| Preserves: | |

ZERO? (12 hex., 18 decimal)

Tests all the BCD digits of the top number on the stack, and converts the number to the proper zero representation (integer 0) if they are all zero.

| | |
|---|---|
| Parameters: | On stack. |
| Returns: | IY - current value of STKPTR.<br>Z flag set if number is zero. |
| Preserves: | HL. |

FLOAT (13 hex., 19 decimal)

Tests the top number on the stack, and if it is in the integer format converts it into a BCD number.

| | |
|---|---|
| Parameters: | On stack. |
| Returns: | IY - current value of STKPTR. |
| Preserves: | HL, BC. |

 

LOADNUM (14 hex., 20 decimal)

Copies the 6-byte floating number pointed to by HL to the stack, converting it to the 8-byte stack representation.

Parameters: HL - points to the 6-byte number.
Returns: HL - new value of STKPTR.
Preserves:

DUP (15 hex., 21 decimal)

Duplicates the top number or string on the stack.

Parameters: On stack.
Returns: DE - new value of STKPTR.
Preserves:

POP (16 hex., 22 decimal)

Removes the top number or string from the stack.

Parameters: On stack.
Returns: HL - new value of STKPTR.
Preserves:

SETZERO (17 hex., 23 decimal)

Sets the top number on the stack to zero, and ensures that the stack type byte is correct. Thus subtracting 8 from STKPTR and calling SETZERO has the effect of pushing a zero onto the stack.

Parameters: On stack.
Returns: IY - current STKPTR value.
Preserves: HL, DE, BC.

FSETZERO (18 hex., 24 decimal)

As SETZERO above, but sets the exponent byte to 3FH.

SETONE (19 hex., 25 decimal)

As SETZERO, but puts a one on the stack.

FSETONE (1A hex., 26 decimal)

As FSETZERO, but puts a one on the stack.

NUMASSIGN (1B hex., 27 decimal)

Copies the top number on the stack to the 6-byte variable pointed to by HL, converting from the 8-byte stack format to the 6-byte symbol table format. The number is not removed from the stack.

Parameters: On stack.
HL - points to the variable.
Returns:
Preserves:

STRASSIGN (1C hex., 28 decimal)

Copies the top string on the stack to the variable pointed to by HL, without removing it from the stack. An error is given if the string on the stack is longer than the maximum length of the variable.

Parameters: HL - points to the start of the variable. This must be the byte containing the maximum length of the string.
Returns:
Preserves:

 

PUTDIG (1D hex., 29 decimal)

Writes a value to a specified decimal digit within the top number on the stack. The most significant digit is numbered 4, and the least significant digit is numbered 13 (see section 2.6 (numbers) for details).

| | |
|---|---|
| Parameters: | B - the digit number. |
| | A - value to write (0 to 9) |
| Returns: | IY - current STKPTR value. |
| Preserves: | HL, DE, BC. |

GETDIG (1E hex., 30 decimal)

Reads a value from a specified decimal digit within the top number on the stack (see PUTDIG above).

| | |
|---|---|
| Parameters: | B - digit number to read. |
| Returns: | IY - current STKPTR value. |
| | A - digit value read (0 to 9). |
| | Z flag set according to A. |
| Preserves: | HL, DE, BC. |

NORMALISE (1F hex., 31 decimal)

Shifts the top number on the stack left or right such that the most significant digit is in its normal position (ie. digit position 4). The number is shifted left if the overflow digits are non-zero (loosing those digits shifted out), or right if digit 4 is zero and the number is non-zero.

| | |
|---|---|
| Parameters: | On stack. |
| Returns: | IY - current STKPTR value. |
| Preserves: | |

 

GETITEM (20 hex., 32 decimal)

Reads the next item from the source code. The variables PTR and START define the current position within the source code, and GETITEM increments PTR to the next item, but does not read beyond the end of the line.

| | |
|---|---|
| Parameters: | Current item pointed to by PTR. |
| Returns: | Updated PTR.<br>A - new value of SIGNIS.<br>Z flag set if end of line. |
| Preserves: | HL, DE, BC, IY. |

The variables SYMBTYPE, SYMBLEN and SIGNIS are set according to what was read. The values put in these variables are related to the way the source code is tokenised (see section 2.2 (Program) for more details). For each type of item, the variables are set as follows:

| item | SIGNIS | SYMBLEN | SYMBTYPE |
|---|---|---|---|
| floating number | 127 | 6 | C0H |
| integer number | 127 | 2 | C0H |
| line number | 127 | 2 | A0H |
| string | 127 | length | 80H |
| numeric identifer | 127 | length | 20H |
| string identifier | 127 | length | 40H |
| keyword | 127 | keyword number | 60H |
| sign | token for sign 0 | | 0 |

In addition, the variable SYMB is set to point to the data after the item descriptor byte in the source code. This is the start of the identifier, number, string etc. The variable LN is set to 0 except when integer numbers and line numbers are read, in which case it is set to the value of the read item. If a number is read, then it is copied to the VALUE buffer.

GETSIGN (22 hex., 34 decimal)

If the sign token in a is not the same as that in SIGNIS, then an error is given. Otherwise, GETITEM is called.

| | |
|---|---|
| Parameters: | A - sign to compare. |
| Returns: | As GETITEM above. |
| Preserves: | HL, DE, BC, IY. |

 

EXPRESSION (23 hex., 35 decimal)

Evaluates a numeric expression, returning the result on the stack. Note that calling this routine may cause any other statements or functions to be called (via a user-defined function), including the calling routine if it as extension function or statement. Thus all extensions should be re-entrant.

Parameters:
Returns: Result on stack.
Preserves:

STREXPR (24 hex., 36 decimal)

As EXPRESSION above, but evaluates a string expression. Similar considerations about recursiveness apply.

Parameters:
Returns: Result string on stack.
Preserves:

GETCHAR (25 hex., 37 decimal)

Returns a character read from the default keyboard channel, or 0 if no character is ready.

Parameters:
Returns: A - character or 0.
flags set according to A.
Preserves: HL, IY.

CHKNUM (27 hex., 39 decimal)

Ensures that there is enough room on the stack to push a number on to it by moving the stack if necessary. If there is not enough room on the stack and no more memory is available, then an error is given.

Parameters:
Returns:
Preserves:

 

CHKSTR (28 hex., 40 decimal)

As CHKNUM above, but ensures that there is enough room for a string on the stack.

Parameters: DE - length of string to be put on stack.
Returns:
Preserves:

MINSTK (29 hex., 41 decimal)

Moves the stack as low in memory as possible, freeing up as much memory as possible to EXOS. Used before openning channels.

Parameters:
Returns:
Preserves:

DEBLANK (2A hex., 42 decimal)

Points the variable PTR passed tabs and spaces to point to the first non-blank character.

Parameters: PTR
Returns: Updated PTR.
Preserves: DE, BC, IY.

READNUMBER (2B hex., 43 decimal)

Reads in an ASCII representation of a floating point number from the text pointed to by the variable PTR and converts it into an 8-byte stack representation in the buffer VALUE, updating PTR. If the number was an integer in the range 1 to 9999, it is also put into the variable LN, which is otherwise zero.

Parameters: PTR.
Returns: PTR updated.
HL - the number if an integer in the range 0 to 9999.
Preserves:

 

RND (2D hex., 45 decimal)

Returns a 24-bit pseudo-random number.

Parameters:
Returns: AEL - pseudo-random number.
Preserves: DE, IY.

FINDLN (2E hex., 46 decimal)

Finds the specified line in the program.

Parameters: HL - line number to be found.
Returns: HL - points to the start of the line if found, or to the start of the next highest if not.
BC - offset from HL to start of next line.
Carry set if line not found.
Preserves: IY.

LISTLN (2F hex., 47 decimal)

Lists the program line pointed to by HL to the channel number in the variable CURCHAN. The lines are indented. If CURCHAN=255 then the line is listed to memory at OUCHPOS and OUCHPOS is incremented to the last character+1.

Parameters: HL - points to start of line to list.
Returns: HL - points to start of the next line.
Preserves:

DLTLN (30 hex., 48 decimal

Deletes the program line pointed to by HL and clears the stack and symbol table.

Should only be used for immediate mode commands.

Parameters: HL - points to start of line to delete.
Returns: HL - points to start of next line (ie. is uncorrupted).
Preserves:

 

CLRTAB (31 hex., 49 decimal)

Clears the symbol table and stack.

Should only be used for immediate mode commands.

Parameters:
Returns:
Preserves: HL, BC, AF.

LOADEFS (32 hex., 50 decimal)

Creates symbol table entries for all the DEF and HANDLER names in the program. This is used, for example, in the LIST <function> command - if <function> is not found, then LOADEFS is called. If it is still not found, then an error results.

Should only be used for immediate mode commands.

Parameters:
Returns:
Preserves:

STATLEV (33 hex., 51 decimal)

Searches the program from the line after the current line (pointed to by START) for the next line with the same value in the indentation byte. This is used by the block instructions (FOR, IF etc) to find and check for the end of the block. An error is given if the end of the program is reached.

Parameters:
Returns: BL - points to the start of the end of the block.
Preserves: IY

STOP (34 hex., 52 decimal)

Prints to the channel in CURCHAN 'STOP at line n' where n is the current line.

Parameters:
Returns:
Preserves: IY, DE, BC.

KEYMODE (35 hex., 53 decimal)

Returns the flags byte from the keyword table of the specified keyword.

| | |
|---|---|
| Parameters: | C - number of keyword. |
| Returns: | B - flags byte. |
| | HL - points to the flags byte. |
| Preserves: | IY. |

READLINE (36 hex., 54 decimal)

Reads a line of input (ending with an ASCII carriage return) into the buffer BUFF from the channel in CURCHAN. The number of characters read is in the first byte of the buffer.

| | |
|---|---|
| Parameters: | |
| Returns: | Carry set if end of file reached. |
| | HL - points to first character of buffer. |
| Preserves: | IY. |

OUTCHAR (38 hex., 56 decimal)

Outputs the character in A to the channel in CURCHAN, or if CURCHAN=255 then writes the character to the memory location pointed to by OUTCHPOS, and increments OUTCHPOS.

| | |
|---|---|
| Parameters: | A - character to output. |
| Returns: | |
| Preserves: | AF, BC, DE, HL, IY. |

PRTNO2 (39 hex., 57 decimal)

Outputs the integer in HL using OUTCHAR described above.

| | |
|---|---|
| Parameters: | HL - number to be printed (unsigned). |
| | C - ASCII character printed in the leading zeros positions. C=0 for no leading characters. |
| | Carry set if four digits are to be output, else five digits are output. |
| Returns: | |
| Preserves: | |

Takes a pointer to a variable in the symbol table, and if that variable is an array then adjusts the pointer to the array element specified by the index at the current position in the program.

Parameters: HL - points to the data area of the variable in the symbol table.
C - symbol table type byte of the variable.
B - bit 0 must be set for normal operation. If reset, then INDEX will just parse the following index expressions. See _EXPRESSION and _STREXPR.

Returns: HL - points to the data of the variable ie. is unchanged if the variable is not an array, or points to the specified element if it is.

Preserves:

CHANNO (3C hex., 60 decimal)

Ensures that the current item in the program is a channel number (specified by '#'), giving an error if it is not, and then evaluates and returns the channel number.

Parameters:
Returns: A - channel number.
Preserves: BC, DE.

CPREST (3D hex., 61 decimal)

After the tokenisation routine of a keyword has been called, this call should be made to ensure that the remainder of the statement is tokenised correctly. If an extension statement does not require a tokenisation routine (which is the usual case) then this routine should be called instead.

Parameters:
Returns:
Preserves:

 

OUTSTR (3e hex., 62 decimal)

Outputs a string a character at a time using OUTCHAR described above. The number of characters in the string should be in the first byte.

Parameters: HL - points to the start of the string.
Returns:
Preserves: BC, DE, IY.

UPRINT (3F hex., 63 decimal)

Outputs in ASCII the top number on the stack using OUTCHAR described above, without removing it from the stack. The number is preceded by a leading space or '-' sign depending on its sign, and is followed by a trailing space.

Parameters: On stack.
Returns:
Preserves:

_STREXPR (40 hex., 64 decimal)

Parses a string expression, without returning a result and without calling any user-defined functions that may be referenced in the expression. Usually used in the keyword tokenising routines.

Parameters:
Returns:
Preserves:

_EXPRESSION (41 hex., 65 decimal)

Parses a string expression, without returning a result and without calling any user-defined functions that may be referenced in the expression. Usually used in the keyword tokenising routines.

Parameters:
Returns:
Preserves:

 

LINKSYM (80 hex., 128 decimal)

Links the symbol table pointed to by HL into the symbol table. This is used to link in extension functions, the entry (but not necessarily code) of which must be page 0. See section 4 (Extension Functions) for details of the format of the entries.

Parameters: HL - points to the entry in page 0 to be linked in.
Returns:
Preserves: IY.

ADDNAME (81 hex., 129 decimal)

Adds the flags byte and name of an identifier to the symbol table. The current program item (as got by GETITEM described above) must be an identifier.

Parameters: A - type byte of entry.
Returns:
Preserves: IY.

ADDNUM (82 hex., 130 decimal)

Adds to the symbol table a number in the symbol table format, and marks it as uninitialised. For example, if ADDNUM is called after ADDNAME described above, then a complete simple numeric variable will have been added to the symbol table. See section 2.3 (Symbol Table).

Parameters:
Returns:
Preserves: IY, HL.

ADDSTR (83 hex., 131 decimal)

Similar to ADDNUM described above, but adds a string data field to the symbol table, and marks it as uninitialised.

Parameters: C - maximum length of string.
Returns:
Preserves: HL, IY.

 

ADDENT (88 hex., 136 decimal)

Adds a complete entry for a simple numeric or string identifier, depending on the type of the identifier that is the current item as got by GETITEM (see description above). The maximum length of the string is the default (132 characters).

Parameters:
Returns: HL - points to start of new entry.
Preserves: IY.

ADDFN (89 hex,. 137 decimal)

Adds a user-defined function entry to the symbol table. A pointer to the current line (in START) is put after the numeric or string data. See section 2.3 (Symbol Table) for details of the format of the entry. The name of the function is the current program item as got by GETITEM.

Parameters:
Returns:
Preserves:

ADDREF (8A hex., 138 decimal)

Adds an entry to the symbol table which is a reference to another entry. Used in passing reference parameters to functions.

Parameters: HL - pointer to data part of entry being referenced.
A - type of entry being added.
Returns:
Preserves: IY.

FINDSYM (8B hex., 139 decimal)

Searches the symbol table for an identifier which matches the identifier that is the current item in the program source (as got by GETITEM). If not found, then a new entry is created but marked as uninitialised. Used in expression evaluation.

Parameters:
Returns: EL - points to data area of the entry.
C - type byte of the entry.
Preserves: IY.

 

DEFSYM (8C hex., 140 decimal)

As FINDSYM above, but if the entry found is a machine code function (ie. a built-in function or extension function), then a new entry is created. This is used in the LET and similar statements, in which built-in functions may be re-defined.

Parameters:
Returns: HL - points to data area of the entry.
C - type byte of the entry.
Preserves: IY.

SYM? (8E hex., 142 decimal)

As DEFSYM above, but does not create any new entries, and only searches the symbol table corresponding to the current user-defined function invocation ie. those symbols added by the current function call. This is the whole symbol table only if no user-defined functions are currently active.

Parameters:
Returns: Carry set if name not found.
Preserves: IY.

CLRSYM (8F hex., 143 decimal)

Clears all the non-machine code functions from the symbol table ie. all but the built-in and extension functions.

Parameters:
Returns:
Preserves: IY.

 

LOAD (91 hex., 145 decimal)

Puts a number in the tempory 7-byte format pointed to by HL onto the stack in the normal stack format for numbers.

| | |
|---|---|
| Parameters: | HL - points to the first byte of the no. |
| Results: | HL - new value of STKPTR. |
| Preserves: | IY. |

CLOAD (92 hex., 146 decimal)

Puts a 12-digit (7-byte as above) constant on the stack in the normal stack format for numbers.

| | |
|---|---|
| Parameters: | |
| Returns: | Constant on stack. |
| Preserves: | IY. |

 

The constant is specified by following the RST code for CLOAD (92H) by the constant number. The constants are stored in ROM, and the actual constants corresponding to the specified constant numbers are:

| name | constant number | actual value |
|---|---|---|
| HALF | 0 | 0.5 |
| ONE | 1 | 1 |
| INF | 2 | 0.999999999999E+63 |
| 32K | 3 | 32768 |
| PI | 4 | 3.14159265359 |
| 180/PI | 5 | 57.2957795131 |
| PI/180 | 6 | 0.01745329251994 |
| PI/2 | 7 | 1.57079632679 |
| PI/3 | 8 | 1.04719755120 |
| PI/6 | 9 | 0.523598775598 |
| 2PI | 10 | 6.28318530718 |
| SQRT(.05) approx. | 11 | 0.22367 |
| SQRT(0.8) approx. | 12 | 0.894427 |
| SQRT(0.1) | 13 | 0.316227766017 |
| -1/3! approx. | 14 | -0.166666666666 |
| 1/5! approx. | 15 | 8.33333072056E-3 |
| -1/7! approx. | 16 | -1.98408329231E-4 |
| 1/9! approx. | 17 | 2.75239710678E-6 |
| -1/11! approx. | 18 | -2.38683464060E-8 |
| | 19 | -1.44008344874 |
| | 20 | -0.720026848898 |
| | 21 | 4.32025038919 |
| | 22 | 4.75222584599 |
| | 23 | 0.267949192431 |
| SQRT(3) | 24 | 1.73205080757 |
| | 25 | 0.732050807569 |
| | 26 | -2.91568143790 |
| | 27 | 3.16303491557 |
| | 28 | -0.673581601478 |
| | 29 | -10.0704069542 |
| | 30 | 16.9669814021 |
| | 31 | -8.19080045467 |
| | 32 | 0.868588963807 |
| LN(10) | 33 | 2.30258509299 |
| 1/LOG(2) | 34 | 3.32192809489 |
| | 35 | -0.868588963807 |
| | 36 | 1.151 |
| | 37 | 0.292546497023E-3 |
| | 38 | 0.504464889506E+3 |
| | 39 | 0.140082997563E+2 |
| | 40 | 0.332873646516E-2 |
| | 41 | 0.100892977901E+4 |
| | 42 | 0.112094081097 |

NLOAD (93 hex., 147 decimal)

As LOAD described above, but the address of the number in page 0 follows the restart code for NLOAD (93H) in normal Z80 format.

Parameters:
Returns: Number on stack.
Preserves: IY.

XLOAD (94 hex., 148 decimal)

Puts the number in the floating point register X onto the stack.

Parameters:
Returns: X on stack.
Preserves: IY.

YLOAD (95 hex., 149 decimal)

Similar to XLOAD above, but puts the number in the Y register on the stack.

Parameters:
Returns: Y on stack.
Preserves: IY.

NASSIGN (96 hex., 150 decimal)

Copies the top number on the stack into the page 0 buffer pointed to by the address following the RST code for NASSIGN (96H). The number is converted from the stack format into the tempory 7-byte format described above, and the number is removed from the stack.

Parameters: Removed from stack.
Returns:
Preserves: IY.

XASSIGN (97 hex., 151 decimal)

Copies and removes from the stack the top number on the stack into the X floating point register.

Parameters: Removed from stack.
Returns:
Preserves: IY.

P_NUM (99 hex., 153 decimal)

Parses the syntax ' ( <expression> ) '. The result of the expression is left on the stack.

Parameters:
Returns: Result of <expression> on the stack.
Preserves:

P1_NUM (9A hex., 154 decimal)

Parses the syntax ' ( <expression> '. The result of the expression is left on the stack.

Parameters:
Returns: Result of <expression> on the stack.
Preserves:

P2_NUM (9B hex., 155 decimal)

Parses the syntax ' ,<expression> '. The result of the expression is left on the stack.

Parameters:
Returns: Result of <expression> on the stack.
Preserves:

P_STR (9C hex., 156 decimal)

Parses the syntax ' ( <string-expression> ) '. The result of the string-expression is left on the stack.

Parameters:
Returns: Result of <string-expression> on stack.
Preserves:

P1_STR (9D hex., 157 decimal)

Parses the syntax ' ( <string-expression> '. The result of the <string-expression> is left on the stack.

Parameters:
Returns: Result of <string-expression> on stack.
Preserves:

P2_STR (9E hex., 158 decimal)

Parses the syntax ' , <string-expression> '. The result of the <string-expression> is left on the stack.

Parameters:
Returns: Result of <string-expression> on stack.
Preserves:

P_ARRAY (9F hex., 159 decimal)

Parses the syntax ' ( <array-identifier> '.

Parameters:
Returns: HL - points to data area of the entry.
C - type byte of the entry.
Preserves:

P_END (A0 hex., 160 decimal)

Parses the syntax ' ) ' ie. checks for a closing ).

Parameters:
Returns:
Preserves:

FRE (A1 hex., 161 decimal)

Obtains the number of bytes available for the current program.

Parameters:
Returns: The number of bytes on the stack.
Preserves:

ATN (A2 hex., 162 decimal)

Calculates the arctangent of the top number on the stack in radians. The X and Y floating point registers are corrupted.

Parameters: Removed from stack.
Returns: Result on the stack.
Preserves:

ASIN (A3 hex., 163 decimal)

Calculates the arcsine of the top number on the stack in radians. The X and Y floating-point registers are corrupted.

Parameters: Removed from stack.
Returns: Result on stack.
Preserves:

SIN (A4 hex., 164 decimal)

Calculates the sine of the top number on the stack in radians. The X floating-point register is corrupted.

Parameters: Removed from stack.
Returns: Result on stack.
Preserves:

COS (A5 hex., 165 decimal)

Calculates the cosine of the top number on the stack in radians. The X floating-point register is corrupted.

Parameters: Removed from stack.
Returns: Result on stack.
Preserves:

TAN (A6 hex., 166 decimal)

Calculates the tangent of the top number on the stack in radians. The X and Y floating-point registers are corrupted.

Parameters: Removed from stack.
Returns: Result on stack.
Preserves:

 

EXP (A7 hex., 167 decimal)

Calculates e raised to the power of the top number on the stack. The X and Y floating-point registers are corrupted.

Parameters: Removed from stack.
Returns: Result on stack.
Preserves:

RAD_DEG (A8 hex., 168 decimal)

Convert the top number on the stack from radians to degrees if OPTION ANGLE DEGREES is in effect. Generally used at the end of the calculation of a trigonometric function to ensure that the answer is in the correct mode.

Parameters: Removed from stack.
Returns: Result on stack.
Preserves:

DEG_RAD (A9 hex., 169 decimal)

Convert the top number on the stack from degrees to radians if OPTION ANGLE DEGREES is in effect. Generally used before calculation of a trigonometric function to ensure that the argument is in radians before using it for calculation.

Parameters: Removed from stack.
Returns: Result on stack.
Preserves:

LOG10 (AA hex., 170 decimal)

Calculates the logarithm to the base 10 of the top number on the stack. The X floating-point register is corrupted.

Parameters: Removed from stack.
Returns: Result on stack.
Preserves:

MOD (AB hex., 171 decimal)

Calculates a MOD b where b is the top and a is the second number on the stack. The X and Y floating-point registers are not affected.

Parameters: Removed from stack.
Returns: Result on stack.
Preserves:

SQR (AC hex., 172 decimal)

Calculates the square root of the top number on the stack. The X and Y floating-point registers are corrupted.

Parameters: Removed from stack.
Returns: Result on stack.
Preserves:

RECIP (AD hex., 173 decimal)

Calculates the reciprocal of the top number on the stack. The Y floating-point register is corrupted.

Parameters: Removed from stack.
Returns: Result on stack.
Preserves:

INT (AE hex., 174 decimal)

Calculates the integer part of the top number on the stack. This is the floor of the number. The X and Y floating-point registers are not affected.

Parameters: Removed from stack.
Returns: Result on stack.
Preserves:

ROUND (AF hex., 175 decimal)

Rounds the number pointed to by IY to the specified number of decimal places.

Parameters: IY - points to number (normally the current value of STKPTR).
A is the number of places to round to.
Returns: IY - points to result (ie. uncorrupted).
Preserves: IY, AF.

INVOL (B0 hex., 176 decimal)

Raises a to the power b, the b is the top and a is the second number on the stack. The X and Y floating-point registers are corrupted.

Parameters: Removed from stack.
Returns: Result on stack.
Preserves:

IP (B1 hex., 177 decimal)

Calculates the integer part of the top number on the stack. Note that this is not the same as INT for negative numbers. The X floating-point register is corrupted.

Parameters: Removed from stack.
Result: Result on stack.
Preserves:

SGN (B2 hex., 178 decimal)

Returns -1, 0 or 1 depending on whether the top number on the stack is negative, zero or positive respectively. The X and Y floating-point registers are not affected.

Parameters: Removed from stack.
Results: Result on stack.
Preserves:

DELIST (BA hex., 186 decimal)

Deletes or lists a range of line numbers. If deleting, then the symbol table and stack are cleared.

Parameters: BL - first line number to delete.
DE - last line number to delete.
Alternate carry (in AF') set for list.
Returns:
Preserves:

ERRM (BF hex., 191 decimal)

Gets the error message corresponding to the error number in HL into the MSGBUFF buffer. The first byte of the buffer is the length byte of the message. If no specific message is available, then BASIC returns a more general message describibg the type. Error numbers in the range 9000 to 9255 are given to EXOS to explain.

Parameters: HL - error number.
Returns: Carry set if a specific message is available.
Preserves: IY.

SL (C1 hex., 193 decimal)

Scans the whole program, ensuring that the indentation byte of each line has the correct. Used for example before LIST to ensure that the program lists with correctly indented lines.

Parameters:
Returns:
Preserves: IY.

CASE (C4 hex., 196 decimal)

Convers the top string on the stack to upper or lower case.

Parameters: B - 60H to uppercase, 40H to lowercase.
Returns: Result on stack.
Preserves: IY.

FKEY (C5 hex., 197 decimal)

Re-programs all the function keys on the keyboard to contain BASICs default settings.

Parameters:
Returns:
Preserves: IY.

 