Cortland SANE Tool Set ERS

Apple Numerics

Version 0.02—12 December 1985

Modification History

0.00	6 Nov 85	JWThomas	Original.
0.01	11 Nov 85	JWThomas	Reflect change in tool set call mechanism. Specify initialization functions.
0.02	12 Dec 85	CRLewis	Parameter passing in SANE zero page, conformance with 11/21/85 Tool Locator ERS

Project Identification

Related Documents

Apple Numerics Manual
TheTool Locator ERS. Steve Glass

Product Abstract

This tool set will provide the Cortland assembly language programmer with the Standard Apple Numeric Environment (SANE). SANE is scrupulously-conforming extended-precision IEEE Standard (754) arithmetic, with elementary functions. SANE is designed for sufficient, convenient numeric support for most applications. It includes

- IEEE types single (32-bit), double (64-bit), and extended (80-bit)
- 64-bit type for exact fixed-point computations, as in accounting
- basic floating-point operations (+ - * / √ rem)
- comparisions
- binary-decimal and float-integer conversions
- scanning and formatting for ASCII numeric strings
- logs, trigs, and exponentials
- compound and annuity functions for financial computations
- random number generator
- functions for management of the floating-point environment
- other functions required or recommended by the IEEE Standard

The functionality of the SANE tool set matches that of the Macintosh SANE packages, as well as that of the 6502 Assembly Language SANE software from which it is derived.

Use of the SANE tool set follows the general scheme for accessing Cortland tools.

Use Environment

Hardware Environment

The SANE tool set is part of Cortland system software and requires only the basic Cortland hardware.

Software Environment

If any portions of the SANE tool set are not included in ROM then appropriate files for loading into RAM must be available to the Tool Set Manager.

Programming with the SANE tool set can be done using any assembler that generates code for the Cortland. A file of equs and macros facilitates use with the Cortland development system assembler.

Functional Specifications

General Overview

The SANE tool set is comprised of the usual tool set initialization functions, and functions -- FP816, Elems816, and DecStr816 -- that serve as entry points for the major pieces of SANE code. Each call to FP816, Elems816, or DecStr816 passes an opword parameter specifying the operation to be performed. For example, the opword 0206 (hex) passed to FP816 indicates *divide by a value of type single.*

FP816 contains --
- basic arithmetic operations
- comparisons
- conversions
- environment control
- IEEE auxiliary operations.

Elems816 contains --
- elementary functions
- financial functions
- random number generator.

DecStr816 contains --
- numeric scanners and formatter.

User Scenarios

An application that uses the SANE tool set must first reserve a 256-byte page aligned *SANE zero page*, and pass the address of the SANE zero page as an argument to the application initialization function for SANE.

The following code for a binary operation illustrates a typical invocation of a SANE tool set function:

```
PUSH4       <source operand address>
PUSH4       <destination operand address>
Fxxxx
```

PUSH4, and PUSH2 below, are appropriate macros and Fxxx is a macro for

```
PUSH        <OpWord (16-bit)>
LDX         #TSNum + FuncNum*256
JSL         <Dispatch>
```

Some SANE tool set operations require different numbers of arguments, some pass 16-bit integer arguments by value, and some return results in the X and Y registers and in the status bits.

Alternate function numbers permit passing source operands, the destination operand, and the SANE opword in the SANE zero page, instead of on the stack. Using these alternate function numbers will save the time required for the engine to copy from the stack to the SANE zero page and may permit reuse of one or more operand addresses. When passing parameters directly into SANE zero page, move the SANE opword and the addresses of dst, src, src2 as required for the call. Use the memory map:

```
^                    ^                    ^ upward in memory
|____________________|
|____src2 address____|
|____src address_____|
|____dst address_____|
|_______opword_______|
|_6 bytes reserved___|   _______          beginning of SANE zero page
```

When passed in this way, parameters in SANE zero page duplicate exactly the memory image of what would otherwise appear on the stack.

Performance Characteristics and Limitations

The SANE tool set reserves the high 16 bytes of the SANE zero page, for floating-point status—an application must preserve these sixteen bytes between calls to SANE. The remainder of the SANE zero page is scratch space used only during SANE execution—it is not preserved across calls to SANE and is available to the application.

The SANE tool set removes arguments from the stack and returns no results on the stack. Temporary stack growth during engine calls does not exceed 40 bytes.

The SANE tool set conforms to the general tool set rules for management of the CPU registers and modes.

If not in ROM, the parts of SANE will occupy RAM. Size estimates:

6K	-- FP816
4.5K	-- Elems816
1.5K	-- DecStr816

Typical timing estimates:

0.5 ms	-- add
3.0 ms	-- multiply
3.0 ms	-- divide
3.0 ms	-- scanner
3.0 ms	-- formatter
3.0 ms	-- decimal-to-extended
3.0 ms	-- extended-to-decimal
20.0 ms	-- sin
20.0 ms	-- exponential
20.0 ms	-- logarithm

Applicable Standards

The SANE tool set conforms to IEEE standard 754 for binary floating-point arithmetic and to the proposed IEEE standard 854, which is a radix- and word-length-independent standard for floating-point arithmetic. The SANE tool set fully supports the Standard Apple Numeric Environment.

Interface

Initialization Functions

The boot initialization function does nothing.

The application initialization function takes a single 2-byte input parameter, the address of the (page-aligned) application-allocated SANE zero page. The SANE tool set maintains its environment word and halt vector in the high sixteen bytes of the SANE zero page. The application initialization function clears the SANE environment word; this installs the default settings of round-to-nearest, round-to-extended-precision, all exceptions clear, and all halts disabled. The application initialization function also sets the halt vector to the address of the *system error routine.*

The application shutdown function does nothing. SANE does not call the memory manager to release the SANE zero page.

Version information is returned in the documented format.

SANE Functions

The (noninitialization) SANE functions, and their interfaces are documented in the *Apple Numerics Manual*, Part II, 6502 SANE Engine. The SANE tool set deviates from this documentation in these ways:

- Access is via the tool set dispatcher instead of by JSR. Thus, invocations end with

  ```
  LDX        #TSNum + FuncNum*256
  JSL        <Dispatch>
  ```

  instead of

  ```
  JSR        <xx6502>
  ```

- All address parameters are four bytes instead of two.

- The halt vector is four bytes instead of two.

- The halt vector is initialized to the address of the system error routine. An exception triggers a halt if the corresponding halt is enabled, and does not require the additional condition that the halt vector be nonzero.

- The halt mechanism is changed in that: when a halt occurs, the input parameters and SANE opcode are located in SANE zero page as shown in the diagram above, and the X–register contains a direct–register offset which points to a status information record. This record has the following fields:

halt vector-	2 words
environment	1 word
pending x–register	1 word

pending y–register	1 word
pending exceptions	1 word

<NOTE: the exact description of this record may change before final ROM code>