LA-UR-77-2383

MASTER

CONF-771023-5

TITLE: SCIENTIFIC CALCULATING SYSTEM
WITH A NUMBER-ORIENTED MICROPROCESSOR
SEQUENCED BY A SINGLE COMPONENT MICROCOMPUTER

AUTHOR(S): C. DWAYNE ETHRIDGE, GARY M. WORTH

SUBMITTED TO: W. C. Struven, Bin 50
Guest Editor 1977 NSS/NPSS
Stanford Linear Accelerator Center
P. O. Box 4349
Stanford, California 94305

1977 Nuclear Science Symposium.
The Institute of Electrical and
Electronic Engineers; Nuclear and
Plasma Science Society, San Francisco,
California, October 19-21, 1977.

By acceptance of this article for publication, the publisher recognizes the Government's (license) rights in any copyright and the Government and its authorized representatives have unrestricted right to reproduce in whole or in part said article under any copyright secured by the publisher.

The Los Alamos Scientific Laboratory requests that the publisher identify this article as work performed under the auspices of the USERDA.

los alamos
scientific laboratory
of the University of California
LOS ALAMOS, NEW MEXICO 87544

An Affirmative Action/Equal Opportunity Employer

NOTICE
This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

SCIENTIFIC CALCULATING SYSTEM WITH A NUMBER-ORIENTED MICROPROCESSOR
SEQUENCED BY A SINGLE COMPONENT MICROCOMPUTER*

C. Dwayne Ethridge

and

Gary M. Worth

University of California
Los Alamos Scientific Laboratory
Los Alamos, New Mexico 87545

Abstract

A scientific calculating system for large distributed-task processing systems and for small isolated intelligent data acquisition and instrumentation systems is established with a number-oriented microprocessor sequenced by a single component microcomputer. A MOS/LSI number-oriented microprocessor provides the scientific calculating capability with Reverse Polish Notation data entry. Input data sequencing, computation processing, intermediate result comparison, answer display and/or answer feedback to master processors is controlled by a single component microcomputer.

Introduction

A scientific calculating system for large distributed-task processing systems and for small isolated intelligent data acquisition and instrumentation systems is established with a number-oriented microprocessor sequenced by a single component microcomputer. A MOS/LSI number-oriented microprocessor provides the scientific calculating capability with Reverse Polish Notation data entry. Input data sequencing, computation processing, intermediate result comparison, answer display and/or answer feedback to master processors is controlled by a single component microcomputer.

This discussion involves a scientific calculation system for data and answers with a data format of floating point and/or scientific notation with an eight-digit mantissa and two-digit exponent. The display is a twelve-digit display which allows for the mantissa sign and exponent sign in addition to the mantissa and exponent. The 64 preprogrammed functions of the National Semiconductor MM 57109**MOS/LSI Number-Oriented Microprocessor^{1,2,3} include the following:

- . Arithmetic - plus, minus, multiply, divide, add to memory, subtract from memory, multiply memory, divide memory, reciprocal
- . Trigonometric - sin x, arc sin x, cos x, arc cos x, tan x, arc tan x, degrees to radians, radians to degrees
- . Logarithmic - log x, ln x, e^x , 10^x
- . Others - x^2 , y^x , π

*Work performed under the auspices of the USERDA.

**Reference to a company or product name here or elsewhere in this report does not imply approval or recommendation of the product by the University of California or the U. S. Energy Research and Development Administration to the exclusion of others that may be suitable.

The microprocessor contains 5 internal 12-digit registers X, Y, Z, T, and M, that are available to the user. Operational controls include number entry, decimal point, change sign, enter, halt, roll stack, pop stack, exchange X and Y, exchange X and M, store X to M, recall M to X, shift left or right mantissa digits, and master clear. All numbers and calculation instructions are entered into the microprocessor as a six-bit instruction code by the microcomputer.

The single component microcomputer sequencing the microprocessor is the Intel 8748 8-bit Single Component Microcomputer^{4,5,6}. The microcomputer is a totally self-sufficient 8-bit parallel computer fabricated on a single silicon chip using Intel's N-channel silicon gate MOS process. The microcomputer contains a 1K words by 8-bit byte EPROM program memory, a 64 words by 8-bit byte RAM data memory, 27 Input/Output lines, and an 8-bit timer/counter in addition to the on-board oscillator and clock circuitry. The 8,192-bit EPROM can be cleared by an ultraviolet light and then electrically programmed. The EPROM program memory storage provides the system with the capability of a fully programmable calculator with program retention when the system is turned off.

The microcomputer is designed to be an efficient controller as well as an arithmetic processor. The 96 instructions of the instruction set have extensive bit handling capability as well as facilities for both binary and BCD arithmetic. These instructions include decimal adjust, swap upper 4 bits with lower 4 bits of accumulator and exchange lower 4 bits of accumulator with those of a data memory register.

Although this scientific calculating system is a general purpose module satisfying a broad range of digital system needs, the design was accomplished specifically to provide the calculation functions of Special Nuclear Material Detection, Surveillance, Verification and Recovery projects of the Los Alamos Scientific Laboratory.

Programming the Calculation Sequence

The incorporation of the number-oriented microprocessor or calculator chip or number cruncher unit establishes a unique system design feature in the area of program sequence generation. Similarity of the microprocessor operation to hand-held calculators such as the Hewlett Packard HP-21 or the National Semiconductor Novus 4520, the Scientist, allows the experimenter, physicist or scientist, to develop his project modeling on readily available hand-held calculators. This feature eliminates the need of the experimenter to become familiar with the internal operation, the instruction sets, and/or the assembly languages of the microprocessor and microcomputer. Accuracy studies and parameter variation studies can therefore be accomplished before any firmware resources are allocated to a specific experiment⁷. Similarity

of the microprocessor's function execution times to the hand-held calculators allows the experimenter to easily relate to and generate system timing requirements.

After the experiment calculation sequence is derived, the calculation sequence instruction codes, Key-Level Language, of the microprocessor are integrated into the microcomputer main program which calls subroutines for firmware initialization, data input, microprocessor sequencing, answer read back, and display and/or answer feed back to master processors. One 256-word page of the program memory space is allocated to the Key-Level Language storage. This memory assignment prevents the main program software addresses from changing if the calculation sequence is changed or updated. The system operation flowchart of Figure 1 defines the steps of the main program.

FIGURE 1. SYSTEM OPERATION FLOWCHART

System Software

The microcomputer program memory software was written in Intel MCS-48 Assembly Language which is presently the highest level language available for the 8748 microcomputer. The software was developed on an Intel MDS-800 Intellect Microcomputer Development System equipped with the ISIS-II Diskette Operating System (software), the MDS-2DS Diskette Creating System (hardware), and a Texas Instruments 733 terminal. The program source file was developed and stored on a floppy disk. The source file was processed by an assembler, MDS-D48 8048 Software Package Version 1.2, to produce the object file. The software and hardware were debugged with the aid of the Intel PROMPT-48 MCS-48 Personal Programming Tool. The object file was down loaded from the MDS to the PROMPT-48 for EPROM programming.

The scientific calculating system software flow chart is defined in Figure 2. Initial application of power, activation of the local reset switch, or receipt of an external interrupt from the data source initializes the microcomputer and microprocessor with the "INIT" routine. Initial application of power or activation of the local reset switch vectors the microcomputer to address 0. The external interrupt vectors the microcomputer to address 3. The "Fake Return" sequence allows the microcomputer to continue into the "INIT" routine with the interrupt re-enabled and the stack pointer correspondingly decremented.

A programmed Input/Output loop waits for the data ready signal "RDY" to be applied to the microcomputer pin T0 by the data source. The "INSERT" routine selects and inputs data into the microcomputer data memory. The input data string is organized in bit-parallel digit-serial form. The microcomputer stores the data two binary-coded-digits per 8-bit byte in six bytes of RAM. In scientific notation, the data requires four bytes for the mantissa, one byte for the exponent, and one byte for the mantissa sign and exponent sign. This data structure is identical to the number-oriented microprocessor output string.

The "SYNC" routine provides the handshaking synchronization between the microcomputer "OLD" signal and the microprocessor "READY" signal. This initialization prepares the microprocessor for data/instruction entry.

The "KEYCMD ACTION" software utilizes a pointer to an address on the memory page containing the Key-Level Language to read the next key command action. A key command action can be one of four types as shown in Figure 2. The key command is examined and processed according to the flowchart until a Program End is detected. Then the "VIEWO" routine is entered which continually multiplexes the light-emitting diode display until an interrupt occurs.

If the key command is an instruction (multiply, for example), the six-bit code for multiply is entered into the microprocessor by the "KEY IN" subroutine. An error such as divide by zero results in an Error signal from the microprocessor which causes the microcomputer to display "ERROR" on the LED display.

If the key command is a variable, the "VARSEQ" variable sequence subroutine calls the "KEY IN" subroutine sequentially for each digit until the variable is entered. The "VARSEQ" subroutine processes a variable stored in RAM by requiring only the RAM starting address.

FIGURE 2. SOFTWARE FLOWCHART

If the key command is OUTPUT, the "RDSEQ" read sequence subroutine reads and stores the microprocessor output which is also in bit-parallel digit-serial form. The output is free running, and, therefore, the microcomputer must be fast enough to read and store each byte before the next output byte is presented.

An example of the Key Level Language for the following simple calculation is detailed in Table 1.

$$f = \sqrt{S^2 + \pi^2} = 3.1417827 \times 10^0$$

where $S = 3.456 \times 10^{-2}$

The overhead software to input the data, process the Key Level Language, and multiplex the display requires approximately 600 bytes of program memory. Therefore, roughly 400 bytes of the 1K program memory are uncommitted in this design.

Application of this calculating system to other data sources would require that the "INSERT" routine be modified to the specific input data requirements. The Key Level Language would also be a new sequence.

TABLE 1. SAMPLE INSTRUCTION SEQUENCE

HEXADECIMAL INSTRUCTION CODE	KEY-LEVEL LANGUAGE MNEMONIC	COMMENTS
2F	MCLR	Master clear
22	TOGM	Select scientific notation
03	3	Number 3 entry
0A	DP	Decimal Point*
04	4	Number 4 entry
05	5	Number 5 entry
06	6	Number 6 entry
08	EE	Enter Exponent
02	2	Number 2 entry
0C	CS	Change Sign
21	EN	$X = S = 3.456 \times 10^{-2}$
33	SQ	$X = S^2$
0D	PI	$X = \pi, Y = S^2$
3C	SQ	$X = \pi^2$
39	+	$X = \pi^2 + S^2$
34	SQRT	$X = \sqrt{S^2 + \pi^2}$
16	OUT	OUTPUT
7F	PEND	DISPLAY

*NOTE: Decimal point in scientific notation is assumed to the right of the MSD.

Hardware Description

The specific logic hardware devices of this design are shown in Figure 3. The microcomputer is supplied a 6 MHz external crystal to obtain instruction timer of 0.5 μ s (5.0 for 2 cycle instruction such as those involving moving immediate data.) Three input status signals are applied to the "INT, T0, and T1" inputs. Eight bits of input data (4 bits for BCD data and 4 bits, one each for digit identification) are applied to Port 1. A five bit input selection command is supplied to the input device by writing a Bus Port output into a 74LS174 hex latch. For answer display, the same latch supplies 5 bits of output digit select command and one bit of decimal point control.

The microprocessor data input and six-bit instruction codes are also written into a 74LS174 hex latch. Output data is read into the bus port through a 81LS95 octal tri-state buffer. This data consists of four bits for BCD data and three output status bits, Ready, Strobe, and Error.

Port 2 supplies the Hold signal to the microprocessor to synchronize the instruction and data input sequence. The remaining seven bits supply the seven segment data for the HP 4082-7414 LED displays.

The microprocessor requires +5 and -4 volt power supplies. All remaining devices require only +5 volts. The microprocessor also requires a 400 KHz oscillator to obtain typical computation execution times. Representative typical execution times for an add function and a sin function are 22 milliseconds and 562 milliseconds, respectively.

FIGURE 3. HARDWARE BLOCK DIAGRAM

Conclusion

A scientific calculation system for large distributed-task processing systems and for small isolated intelligent data acquisition and instrumentation systems is established with a number-oriented microprocessor sequenced by a single component microcomputer. A MOS/LSI number-oriented microprocessor provides the scientific calculating capability with Reverse Polish Notation data entry. Input data sequencing, computation processing, intermediate result comparison, answer display and/o. answer feedback to master processors is controlled by a single component microcomputer. The similarity of the system to hand-held calculators provides the system designer with an understanding of the system without requiring detailed system operation knowledge and/or assembly language programming of either the microprocessor or the microcomputer.

References

1. T. TOAL and A. J. WEISSBERGER, "Tough Mathematical Tasks Are Child's Play For Number Cruncher", *Electronics*, Vol. 50, No. 4, February 7, 1977, pp. 102-107.

2. MM57109 MOS/LSI Number-Oriented Microprocessor, IM-B50M37, National Semiconductor Corporation, March 1977.

3. S. N. KIM, "Number Cruncher (MM57109) Interface To Microprocessor", Application Note AN-186, IM-B2077, National Semiconductor Corporation, July 1977.

4. H. BLUME, D. BUDDE, H. RAPHAEL, and D. STAMM, "Single-Chip 8-Bit Microcomputer Fills Gap Between Calculator Types and Powerful Multichip Processors", *Electronics*, Vol. 49, No. 24, November 25, 1976, pp. 99-105.

5. MCS-48 Microcomputer User's Manual, 98-270A, Intel Corporation, November 1976.

6. HOO-MIN D. TOONG, "Microprocessors", *Scientific American*, Vol. 237, No. 3, September 1977, pp. 146-161.

7. R. ZUSSMAN, "Computer Simulation On A Pocket Calculator", *Computer Design*, Vol. 16, No. 5, May 1977, pp. 105-109.