A Dual-Parameter Multichannel Analyzer By a Dual-Computer System

TADASHI AKIMOTO, IKUO MURAI, SHINICHI KATO, SHIN CHAKI, ISAO SHOJI, AND YUICHI OGAWA

Abstract—**The design and feasibility of a simple and practical system for measuring two-parameter signals is reported. A dual-parameter multichannel pulse-height analyzer (MCA), which uses two central processing units (CPU's), was developed and designed. One computer (master computer) controls the other computer (slave computer), and also displays, records, and analyzes the data. The slave computer controls two analog-to-digital converters (ADC's), collects the dual-parameter data, and then transfers the collected data to the master computer. Three methods of data transfer between the two CPU's are examined and discussed. The count capacity at the data collection by the slave computer was 2^{16}-1 / ch (2-byte-long data), and this was expanded to 2^{32}-1 / ch (4-byte-long data) at the master computer. The data are analyzed and at any time stored in the storage apparatus by the dual CPU system. The system demonstrated stable operation over long measurement periods and the system was effectively utilized.**

I. Introduction

SPECTRA of leakage neutrons produced by pulsed-neutron sources from test bulk assemblies of reactor materials commonly have a time-dependent nature. Two parameters are needed to estimate the time-dependent behavior of such neutron spectra: the neutron energy and the time elapsed after injection of the pulsed neutrons [1], [2]. A dual-parameter multichannel pulse-height analyzer (MCA) is needed for measuring such two-parameter signals.

We have developed a simple and efficient computer-based dual-parameter MCA [3]. The computer in this system employed a 80286 central processing unit (CPU). The principal features of this previously developed MCA system are as follows.

- The system was relatively cheaply constructed from simple components without special hardware.
- Large data array handling was simplified by flexible application of the segment register of the 80286 CPU.
- The processing speed was improved by transferring data from analog-to-digital converter (ADC) to CPU in 16-bit words and by a simultaneous reading of the status flag and data.
- The graphic display processes were speeded up by directly writing the data corresponding to locations into the graphic video random access memory (VRAM). This made graphic display feasible without decreasing the efficiency of the measurements, and visual evaluation of data for long measurement times was thus facilitated.
- The maximum conversion gain was $X \times Y = 2^{18}$ch (for 16-bit data, X and Y: the memory size of the two parameters). The combination of X and Y can be determined arbitrarily and easily by suitable software.

Although the previous system has many outstanding features as mentioned above, further improvements were still possible. One was that, however short, the time required to renew the graphic data and to record the measured data onto floppy disks, etc., becomes dead time for measurements, because the computer was fully occupied when these operations are carried out. To alleviate this a dual-parameter MCA system using two CPU's was developed in the present study. With this new system only the time when data is transferred between the CPU's becomes dead time for data collection. This system has the good reliability of data preservation against some system failure because the data are at any time stored in a storage apparatus. In the following sections, an outline of the improved new system and its principal features are described and discussed.

II. Dual-parameter Multichannel Pulse-Height Analyzer Using Dual CPU's

A. Outline of the System

The system is composed of two ADC's (Toshiba NAIG E-551, 50-MHz Wilkinson-type), three parallel peripheral interface boards (Interface Co. Ltd. 98PPI(I/M)AZI-120 or ELM DATA Co. Ltd. EP-98DIOA), and two personal computers (NEC Corporation PC9801VX21 and PC9801UV2). The master computer (PC9801UV2, CPU: μPD70116-10, equivalent to an 8086) controls the slave computer (PC9801VX21, CPU:80286) and displays, records, and analyzes the data. The slave computer controls the two ADC's, collects the dual-parameter data, and transfers the data to the master computer under the control of the master computer. The slave computer is operated by the master computer except for booting of the operating system (OS) and loading of the program. A program written in Basic is mainly employed, but a program written in machine language and called by the Basic program was also used when high-speed processing such as collection of dual-parameter data, graphic display of the data, and data-transfer etc., are required.

The transfer of data between the two CPU's is also important in this system. For data transfer, three methods were tentatively employed. These were 16-bit parallel data transfer

Manuscript received April 12, 1990; revised May 18, 1990.

The authors are with the Department of Nuclear Engineering, Faculty of Engineering, Hokkaido University, North 13, West 8, Kitaku, Sapporo, Japan 060.

IEEE Log Number 9038048.

0018-9499/90/1000-1594$01.00

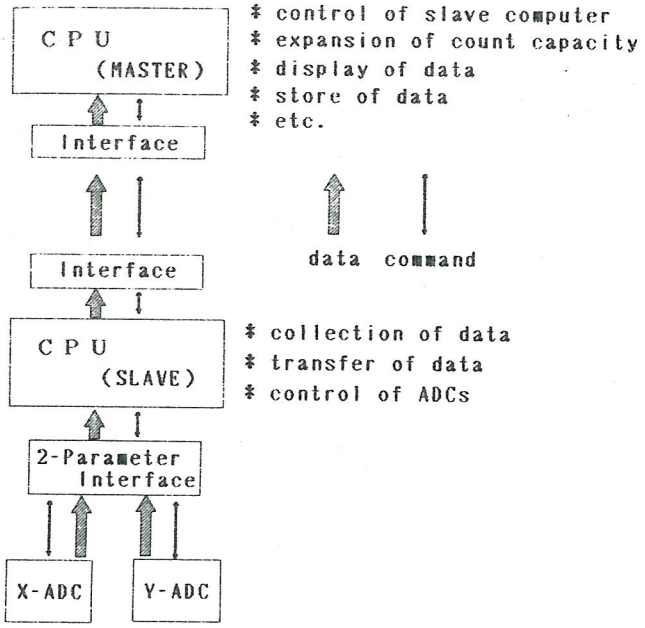

Fig. 1. Block diagram of system.

by software handshake or automatic handshake and direct memory access data transfer.

The use of a single board computer may appear better for the slave computer, but a personal computer is more cost-effective when the CPU characteristics such as memory capacity and program loading to ROM are considered. Therefore, a personal computer was employed as the slave computer in this system. A block diagram of the system is shown in Fig. 1.

B. Control of the Peripheral Instruments and Data Processing

1) *Data Reading and Processing*: In this system, the slave computer controls the two ADC's and collects the data through a parallel peripheral interface board. The speed of data processing is increased by transferring data from ADC to CPU in 16-bit words and by a simultaneous reading of the status flag (ADC-BUSY signal and DATA-READY signal) and data. This dual-parameter MCA system can handle large data arrays of 128 kbytes. Easy access in a large memory space becomes possible by allocating the data from X-ADC (row data) to the segment register and the data from Y-ADC (column data) to the offset register. The dual-parameter information is graphically displayed during measurement with a 16-color contour display or an isometric display at the master computer by transferring the data as necessary to the master computer. The process was speeded up by directly writing the bit data corresponding to the locations into the VRAM. Readers may refer to [3] for further details.

2) *Data-Transfer between two CPU's*: The process of data collection in the slave computer is interrupted when the data from the slave computer are transferred to the master computer. Therefore, the time required to transfer data strongly affects the performance of the system. The method of data transfer between the two CPU's was investigated with commercially available parallel peripheral interface boards. We have examined 1) 16-bit parallel data transfer by software handshake, or 2) automatic handshake, and 3) direct memory access data transfer. The characteristics of these methods are compared and discussed in the following sections.

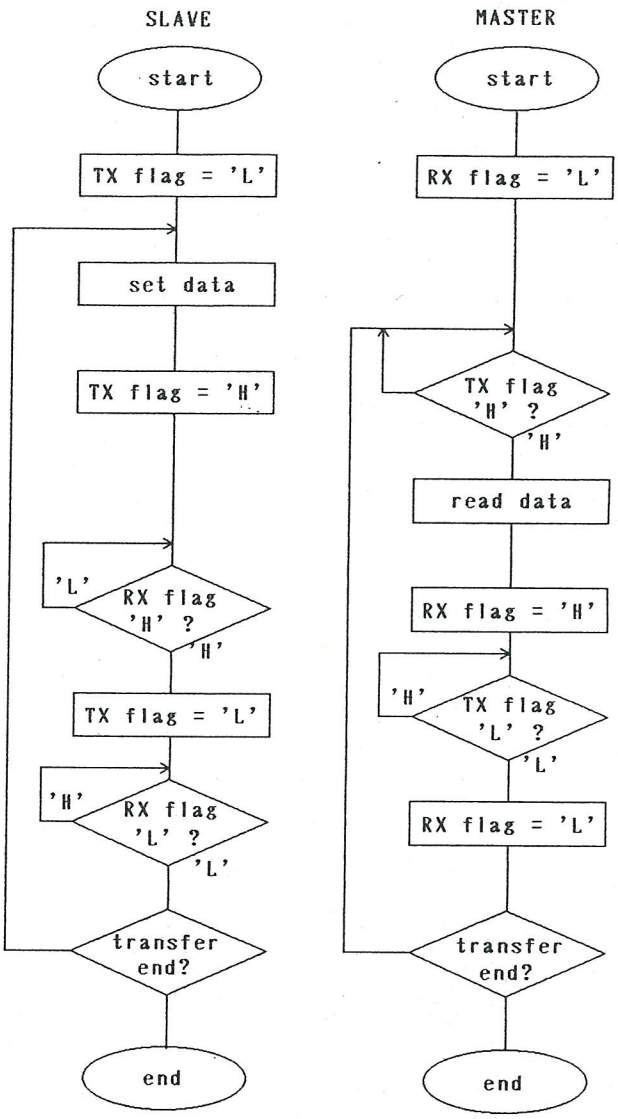

Fig. 2. Flowchart of data transfer routine by software handshaking.

a) *Software handshaking*: This method is highly flexible for data transfer because the logic of signal and timing of data transfer can be decided freely by the software. It is particularly effective when the processing speed of CPU and peripheral equipment are different. However, this method loses time when changing the state of flags by the software.

In the case of the method with software handshaking, commercially available parallel peripheral interface boards (Interface Co. Ltd. 98PPI(I/M)AZI-120) are employed as the interface between the master and slave computer. Mode 0 of the two programmable peripheral interfaces (PPI's) installed on the board are adopted. This PPI is a μPD71055 (equivalent to 8255) with three parallel 8-bit ports. Mode 0 is the basic operation mode that all ports can use as input or output lines. The flowchart and the timing diagram of the data transfer routine are shown in Fig. 2 and Fig. 3, respectively. Here the TX flag indicates transmission of data from the slave

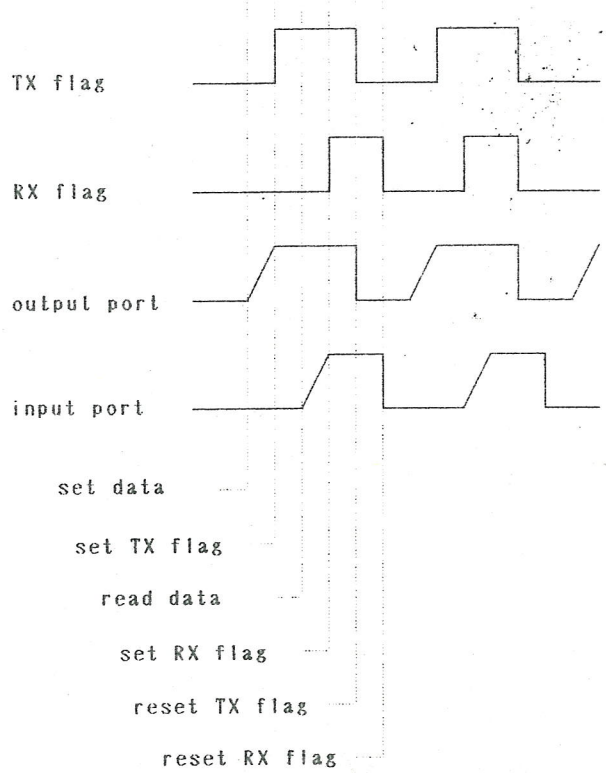

Fig. 3. Timing diagram of data transfer routine by software handshaking.

computer and the RX flag indicates reception at the master computer.

First, both flags TX and RX are set to "L". The slave computer prepares the data and informs the master computer of the situation (TX flag = "H"). The master computer confirms the TX flag, reads the data, and informs the slave computer of data acceptance (RX flag = "H"). The slave computer confirms the data acceptance by the master computer and informs the master computer about the state of preparation for data setting (TX flag = "L"). The master computer confirms preparation of data transmission from the slave computer (RX flag = "L"). The slave computer sets the new data after confirming the state of the master computer. Reliable data transfer can be attained without duplication and loss of data by this handshaking.

b) *Automatic handshaking*: In this method, automatic handshaking is carried out by the hardware of the interface board. Data are transferred by automatic changes of flag states corresponding to the data input and output. The same board, 98PPI(I/M)AZI-120, as employed in software handshaking was also used as the interface board with mode 1. Mode 1 is the operation mode that each port can use as input or output lines and that has control and status lines corresponding to the input and output. In automatic handshaking, only a check for permission to input and output data and transfer of the data are performed by the software. Alterations to the hardware are limited because the handshake lines of PPI are already fixed. In this system the slave computer confirms that the WINT state (Write Interrupt, C3 port of the PPI on the board) becomes "H" and sets the 16-bit data to the A ports. The master computer confirms that RINT (Read Interrupt, C3 port of the PPI on the board) is "H" and reads the 16-bit data from the A ports. However, to ensure that WINT and RINT are usable, WIE (Write Interrupt Enable) and RIE (Read Interrupt Enable) must both be set to "H" with mode 1 at the interface board. The C6 port of the slave computer and C4 port of the master computer must be set to "H". The flow chart is shown in Fig. 4.

c) *DMA transfer*: In ordinary data transfer methods, data in the memory are put into the CPU register and are set on the external port by "OUT", etc. When data are accepted from external equipment, they are also treated through the register. In the DMA transfer method the data can be transferred directly between peripheral equipment and computer memory without passing through the register. The CPU is separated from the bus, and the DMA controller controls data during the transfer.

Control of data transfer must be performed by software in ordinary data transfer methods. Data transfer speed depends on the program language (Basic or assembler, etc.) and the program itself. In the DMA transfer method, the transfer speed depends on only the hardware, after setting the parameters on the DMA controller: the number of data to be transferred, the destination address, and other necessary specifications. Therefore, high-speed data transfer can be attained regardless of programming language.

In this system, channel 3 of DMA controller μPD8237 built in the master computer is used as the DMA controller. There are a few commercially available boards for DMA transfer. A DMA parallel I/O card EP-98DIOA was employed for economic reasons and because of its easy availability, although performance is not optimum. A DMA board is set in the master computer with a CPU processing speed that is lower than that of the slave computer. In the slave

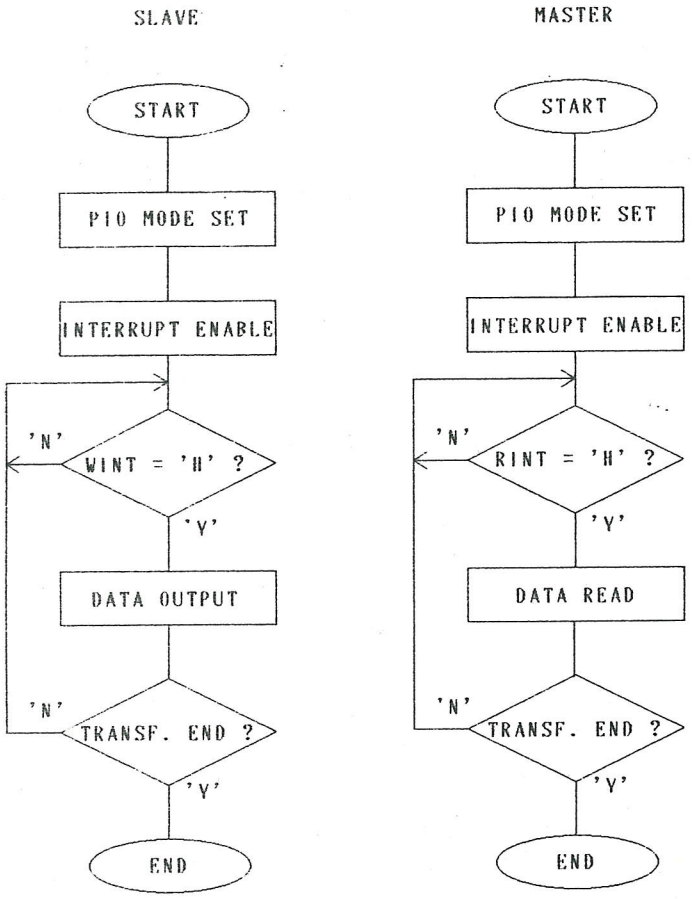

Fig. 4. Flowchart of data transfer routine by automatic handshaking.

TABLE I
COMPARISON OF DATA TRANSFER TIMES

Method of Transfer	Transfer Time per 128 kbytes (s)
software handshaking (16-bit parallel transfer)	1.70
DMA (single transfer mode)	0.98
Automatic handshaking (16-bit parallel transfer)	0.54

computer, the parallel interface board 98PPI(I/M)AZI-120 was used with mode 0 instead of a DMA board, because at present time only one DMA board was available in constructing the system. The data transfer is started by a DMA request signal and is stopped by evaluating the flag with the software. Since this DMA controller cannot handle memory addresses larger than 16 bits, a bank register (4 bits) must be added when 20-bit-size memory addresses are to be handled. Therefore, even when less than 64 kbytes of data are transferred, the controller may have to be set twice (when the upper 4 bit changes in the data transfer).

d) *Results*: A comparison of data transfer times with the three methods are shown in Table I. The software handshaking method was the most flexible, because it requires no special electric circuits. The logic of flags and signals can easily be changed by the software. In the automatic handshaking method, the handshake line is fixed because mode 1 of PPI is employed.

A superior data transfer time was not obtained with the DMA transfer method. The reason is considered to be that the single transfer mode in which only a one-byte-long datum is transferred at each DMA request is employed, and the end of the data transfer is determined by the software. High-speed data transfer may be realized by setting more sophisticated DMA boards, each which has its own DMA controller, on both the master and the slave computer.

The three data transfer methods needed similar amounts of programming works. The automatic handshake data transfer method was built by a commercially available cheap interface board and showed the best data transfer speed.

3) *Count Capacity*: Count capacity during collection of data by the slave computer was 2^{16}-1/ch (2-byte-long data) in the system, while it was expanded to 2^{32}-1/ch (4-byte-long data) at the master computer. High-speed data processing and high count capacity achieved with a two CPU system are attractive results that are attained without slowing down collection speed.

A 4-segment memory (256 kbytes) is required at a conversion gain of 256 × 256 ch to expand 2-byte-long data to 4 bytes. Moreover, a 2-segment memory (128 kbytes) is required to store data from the slave computer temporarily. In this system, 128 kbytes of data from the slave computer are transferred temporarily to the VRAM and are accumulated in the memory as the 4-byte-long data at the corresponding address. Two VRAM's were employed with page switching technology, since 128-kbyte data cannot be stored in one

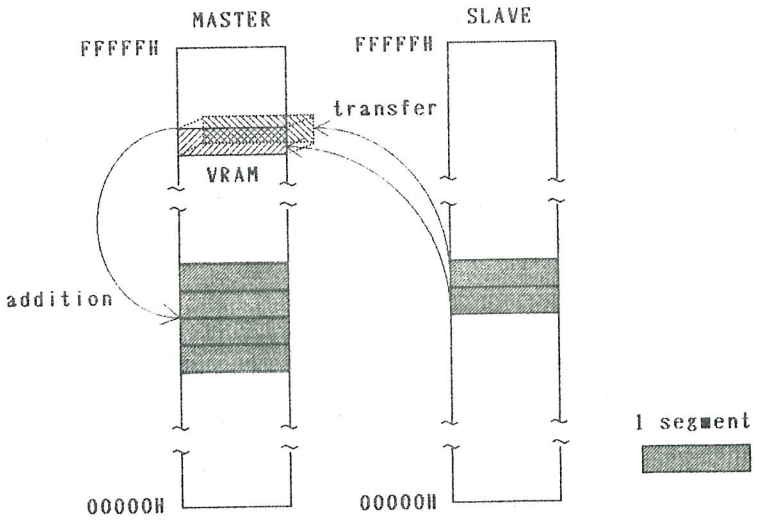

Fig. 5. Expansion of count capacity (from 2^{16}-/ch to 2^{32-1}/ch).

VRAM. The memory on the slave computer is reset after every data transfer. The outline is shown in Fig. 5.

III. Conclusion

The performance of this system is as follows:

maximum conversion gain	$X \times Y = 2^{18}$ channels (for example, 512 × 512 ch); combinations of X and Y can be determined arbitrarily
count capacity	2^{32}-1/ch
minimum data processing time	13.6 ± 1 μs (at 256 × 256 ch)
time to renew graphic data	3 ± 0.5 s/2^{16} dots (contour display); about 0.10 s/(2^{16}/16) dots (isometric display)
data transfer time	0.54 s/128 kbytes.

Data transfer time was 0.54 s with 128 kbytes of data, depending on the amount of data and performance of the computer. The progress of data collection is interrupted during data transfer, but the dead time of the new system is only one seventh of the previous "one CPU dual-parameter MCA" system (time to renew graphic data was estimated to be about 3.5 s). These results show that the new system is much superior to the old one.

The master computer had the expanded count capacity of 2^{32}-1/ch (4-byte-long data) at data collection, while the slave computer had the capacity of 2^{16}-1/ch. High-speed data processing and high count capacity was possible without loss of collection speed by this expansion.

The system shows good stability during long measurements and is safe against accidents such as electric power failures, because the data are stored in a storage apparatus such as floppy disks at any time. The information gained after the data analysis is effectively fed back into the setting of the next measuring condition. The further reduction in transfer time might be attained by using a bank selection method or dual-port memory.

Acknowledgment

The authors would like to thank Dr. T. Mizuno, Mr. T. Christensen, and Professor Y. Abe for their helpful suggestions.

References

[1] N. Pieroni, D. Rusch, and E. Wattecamps, "Measurement of time-dependent fast neutron spectra with an NE213 scintillator," *Nucl. Instrum. Methods*, vol. 115, pp. 317–323, 1974.

[2] T. Akimoto, O. Togawa, and Y. Ogawa, "Theoretical analysis of time-dependent neutron spectra in bulk assemblies" (in Japanese), in *Bulletin of Faculty of Engineering*, Hokkido University, no. 139, pp. 29–37, 1988.

[3] T. Akimoto, I. Murai, S. Chaki, Y. Ogawa, and I. Shoji, "A dual-parameter multichannel analyzer using a personal computer," *IEEE Trans. Nucl. Sci.*, vol. 36, no. 3, June 1989.