LA-UR -81-1568

CONF-810523--4

TITLE: BULK-MEMORY PROCESSOR FOR DATA ACQUISITION

MASTER

AUTHOR(S): R. O. Nelson, D. E. McMillan, J. W. Sunier, M. Meier, and R. V. Poore

SUBMITTED TO: Topical Conference on Computerized Data Acquisition in Particle and Nuclear Physics, Oak Ridge, TN, May 28-30, 1981

DISCLAIMER

By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes.

The Los Alamos Scientific Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy

University of California

LOS ALAMOS SCIENTIFIC LABORATORY
Post Office Box 1663 Los Alamos, New Mexico 87545
An Affirmative Action/Equal Opportunity Employer

Form No. 836 R3
St. No. 2629

UNITED STATES
DEPARTMENT OF ENERGY

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

BULK-MEMORY PROCESSOR FOR DATA ACQUISITION

R. O. Nelson, D. E. McMillan, J. W. Sunier, M. Meier and R. V. Poore

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

ABSTRACT

To meet the diverse needs and data rate requirements at the Van de Graaff and Weapons Neutron Research (WNR) facilities, a bulk memory system has been implemented which includes a fast and flexible processor. This bulk memory processor (BMP) utilizes bit slice and microcode techniques and features a 24 bit wide internal architecture allowing direct addressing of up to 16 megawords of memory and histogramming up to 16 million counts per channel without overflow. The BMP is interfaced to the MOSTEK MK 8000 bulk memory system and to the standard MODCOMP computer I/O bus. Coding for the BMP both at the microcode level and with macro instructions is supported. The generalized data acquisition system[1] has been extended to support the BMP in a manner transparent to the user.

I. INTRODUCTION

ear physics studies with "white" neutron sources often met with requirements for large data age. This primarily results from the use of time-light techniques for neutron energy measurements h produce many bits of information characterizing vent. Over the past decade this information was cally handled by forming histograms stored on fast s since large random access memories were prohibi-ly expensive. Naturally, the event data rates in systems were low, typically less than 10K events second.

nt advances in neutron detector technology (no-y two dimensional area detectors), more complex riments, and the influx of new experiments from condensed matter physics community have increased pressure for greater data acquisition rates and er histogramming space. Fortunately, advances in iconductor technology have provided the random ss memories capable of solving both problems at an rdable price. The adaptation of these products to acquisition is now of wide interest.

II. IMPLEMENTATION OF HARDWARE

implementing a memory system for use in experi-al physics, the first design question should ress the possible need for intelligence in the lk memory" system. In a simple histogramming ironment no such requirement exists. Also, if t rates exceed 1M events per second, little essing is be possible. But for the broad range of lications often found in the physics laboratory, a processing capability will find use.

second design question should address the sort of essor which will manage the bulk memory and form the preprocessing. An obvious option is to orporate a microprocessor on a chip. The micro-essor approach offers primarily a reduction in the ign effort and some flexibility in the choice of essor, but suffers in processor speed. An rnative approach using bipolar microprocessor see yields greater speed and all the flexibility in hitecture possible at the expense of greater design ort. As our group supports diverse experimental grams at both a tandem Van de Graaff accelerator a spallation neutron source at the Weapons Neutron search facility (WNR), the bit slice design is ored to insure both speed and flexibility.

Since the bit slice processor is to be used in the CAMAC environment of our laboratories , it is natural to select a matching word width of 24 bits. For histogramming this offers a sufficient width so that overflows can largely be ignored. Furthermore, the 24 bit width allows single word addressing of 16 Mega-words of memory.

The AMD 2903, selected as the arithmetic/logic unit (ALU) for the bulk memory processor (BMP), offers standard arithmetic and logic functions plus extended functions to facilitate multiplication, division, sign extension, etc. Furthermore, the architecture of the 2903 allows expansion of the general purpose register set beyond the internal 16 registers so that an exper-iment of modest complexity may retain conditional sorting or windowing limits for multiparameter data acquisition entirely in the register set. Therefore, the design includes an additional 80 general registers beyond those internal to the 2903. Thirty-two con-stants registers are included primarily for facili-tating bit operations bringing the total register count to 128.

For main memory the MOSTEK MK 8000 system is utilized. It offers 128K words of 24 bits on each card with card cages of 1 or 2 Megaword capacity. The card is based on the MK 4116 dynamic RAM chip and has on board refresh logic. The memory access times are 250ns and 450ns for read and full access, respectively. To increase memory speeds for sequential access, odd and even addresses are interleaved resulting in an effective cycle time of 330ns including all processor request and control overhead. Three access modes are supported: read (330ns), write (330ns) and read/modi-fy/write (variable).

To interface the BMP to the experiment, the processor contains a 32-bit wide I/O data register fully inter-faced to the MODCOMP bus which is in turn interfaced to the experiment via CAMAC. The control logic of the BMP interface supports direct memory access as well as register driven I/O. Use of the "double word" re-gister simplifies matching the 16-bit wide mini-com-puter to the 24-bit wide processor. The MODCOMP computer concentrates and derandomises the data from several CAMAC interfaced experiments and at the option of the programmer, may manipulate the data before passing it along.

The flow of data from the interface through the BMP to memory and eventually from memory back to the

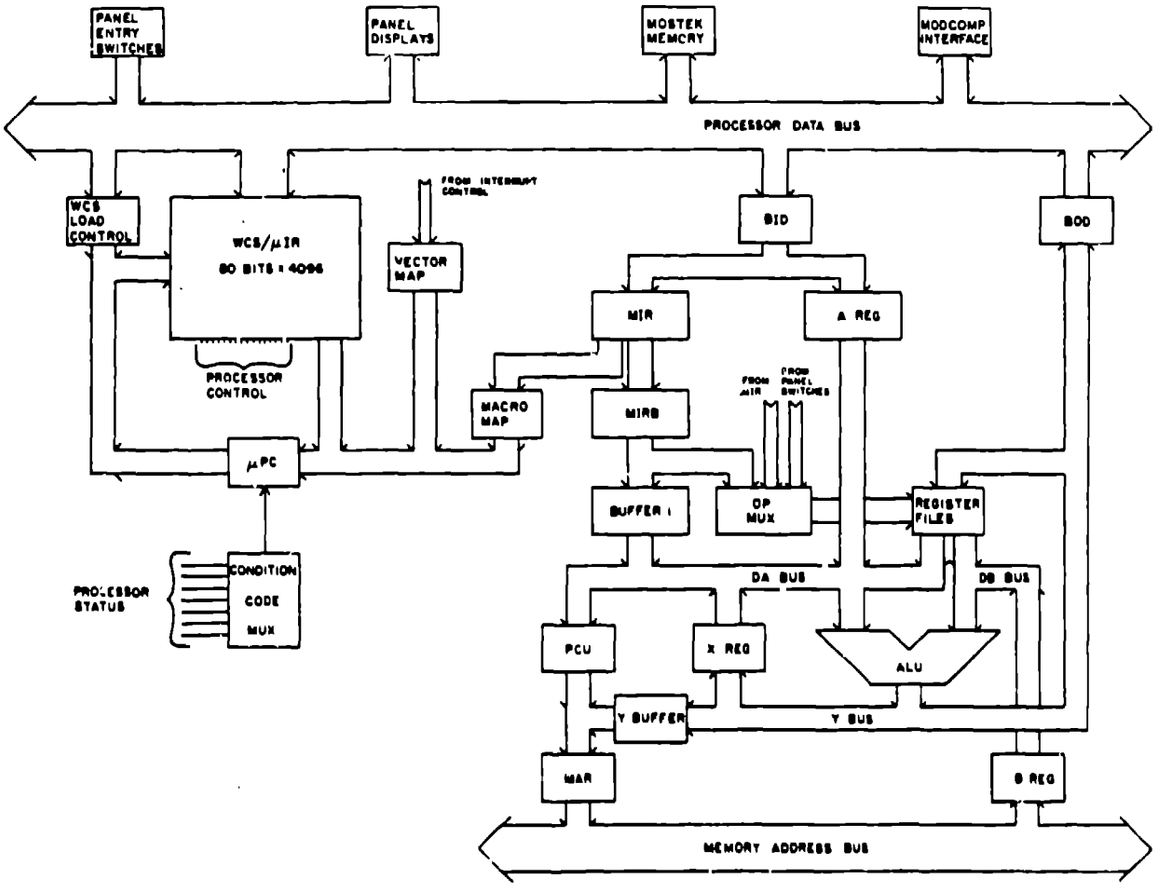

Fig. 1 BMP architecture. Only the datapaths are shown.

interface is controlled via microcode stored in writable control store (WCS) organized as 4096 words, 80 bits wide. The 80-bit format for the microcode supports as much parallel processing capability as practical to enhance the speed of the processor. The AMD 2910 is used for the microprocessor sequencer and, of course, is itself driven by the microcode. With other fields the microcode directly controls the flow on the main processor data bus and various internal buses, the source and destination for ALU operations, the ALU operation, the functions peripheral to the ALU performed by the AMD 2904 (condition code, carry-in and shift-in/shift-out management), program counter control, memory accesses and I/O. In addition, a three bit field specifies the time allocated for each microstep, thus enabling the microcode to be optimized for speed. The architecture for the BMP is shown in Figure 1 with emphasis on the various data paths. Table 1 indicates the bit fields of the microcode instruction used for control of processor functions.

To facilitate coding in the preprocessor, a macrocode capability is included in the design. It is believed that for low count rate experiments (typically complex coincidence experiments) considerble preprocessing might be required and microcode development for these applications would be tedious and needlessly difficult. Thus, the macrocode implementation gives the programmer the tools to which he is accustomed, the machine language instruction set, general purpose registers and a program counter. The program counter is provided through use of the AMD 2930, a device

Table I. Microde Word Format

Bits	Function
0-6	Source register select
7-13	Destination register select
14-15	Operand select
16-21	Program control unit (PCU) control
22-23	Register increment/decrement control
24-25	Y bus destination select
26-27	Bus input data (BID) destination select
28-29	DA Bus source select
30-35	Condition code control
36-37	Carry-in control
38-45	Other status/shift control for AMD 2904
46-54	Arithmetic/logic operation
55-56	Format option select
57-59	Reserved for MODCOMP interface
60	Request data from MODCOMP
61	Y bus source select
62-64	Processor data bus source select
65-67	Processor data bus destination select
68	Unused
69-71	Bulk memory interface control
72-74	Microstep clock period select
75-79	Microprogram sequencer control
0-11*	Microprogram branch address
16-19*	Shift control for AMD 2904
0-23*	Programmable literal
0-25**	Future CAMAC Operation control
46-49**	Priority interrupt control

*Format overlap context dependent
**Format overlap option dependent

ıle 2. Typical Instruction Execution Times

monic	Instruction	Execution Time (μs)			
		BMP		7830**	
		No Branch	Branch	No Branch	Branch
)R	Add register to register	0.33		0.3	
)M	Add memory to register	0.99		1.4	
)MM	Add register to memroy	1.23		1.4	
)X	Add memory (short-indexed) to register	0.66		1.4	
)XM	Add register to memory (short-indexed)	0.90		1.5	
)I	Add memory (immediate) to register	0.66		0.6	
)RB	Add register to register and branch if nonzero	0.66	1.32	0.7	1.4
)M	Load register from memory	0.99		1.4	
)M	Store register in memory	0.99		0.7	
)R*	Multiply register by register	4.61		5.2	
)R*	Divide register by register	6.65		7.4	
.S	Logical left shift by register	0.69+0.16N		0.9+0.2N	
.M	Branch and link		1.32		1.4
.NS	Branch and link if negative condition code set	0.66	1.32	0.7	1.4
.X	Branch and link (short-indexed)		1.23		1.2
.NS	Branch and link (short-indexed) if negative condition set	0.53	1.23	0.5	1.2
)P	Hop unconditionally		1.23		1.6
S	Hop if negative condition code set	0.33	1.32	0.9	1.6

roducts and dividends for the BMP and 7830 are 48 and 32 bits, respectively.
Times taken from MODCOMP Classic Central Processor Model 7830, Reference Manual, 110-142000-000.

icitly designed for the macro program control unit), and the instruction set through conventional of an instruction register and mapping PROM niques to obtain entry points into the microcode. se of microcode for specific data acquisition ions is reserved for only those applications which d speed and permit simplicity.

I. IMPLEMENTATION OF FIRMWARE AND SOFTWARE

microcode or firmware developed for the processor be divided into two categories, basic processor tions and macro instruction implementation. The t processor functions support front panel opera- s such as program counter load and increment, main ry load and display register management. In tion block transfers to and from main memory and k transfers of channel numbers for histogramming supported as basic functions.

ware for the macro instruction implementation is erned to emulate much of the MODCOMP instruction since the processor is envisioned for use pri- ly with MODCOMP computers. To date six operand essing modes are supported as indicated in Table 2 the various add instructions. Instructions sup- three basic data types: bit, word and file which ists of one to sixteen registers. Integer opera- s including multiplication and division are imple- ed, but floating point operations are not planned his time. With the flexibility afforded by the able control store for the microcode, additional ructions taylored to the peculiar demands of data isition are anticipated.

microcode is written in symbolic form using a -assembler available from MicroTec of Sunnyvale, fornia, which is essentially a Fortran implemen- on of AMD's meta-assembler, AMDASYM. The assembler lexible, convenient and enhances generation of a documented code.

Access to the BMP is obtained from the MODCOMP computer through the basic I/O services of the operating system. At the system level the only custom software required was the I/O handler. By treating the processor as an I/O device, multiple user tasks may queue prioritized requests for service using standard subroutine calls available for either assembly language or Fortran. Histogramming or other functions unique to the processor can be selected with device dependent option bits available in calls to these routines. With this approach it is easily possible, for example, to have in a multitasking environment concurrent requests to the BMP for clearing data, for histogramming several channels of data acquisition, for read out for display purposes and for storage to magnetic tape. Although not yet in use, the queue could include requests to load custom microcode or application macrocode as well.

The generalized data acquisition code "Z" has been extensively modified to support data acquisition with the BMP in a manner transparent to the user. Functionally the commands of "Z" treat data in main memory and in bulk memory identically, but operationally are very different since bulk memory data areas, which may be megawords in length, are successively processed as segments. The event analysis language, EVAL [2], operating under "Z" is soon to be extended to a version which will compile macro instructions for the BMP. This will be a convenient tool to harness the flexibility and power offered by the BMP for a wide variety of problems.

IV. RESULTS AND PERFORMANCE

The first two BMPs were installed on MODCOMP IV/25 computers at WNR and began routine data collection in March, 1981. A third unit was installed on a MODCOMP 7860 in a mobil data acquisition system most recently taking data at the Los Alamos Meson Physics Facility (LAMPF). All units have been kept in service with only minor difficulties due mainly to chip infant mortality. As expected the simple histogramming data

rate where the BMP is executing only microcode depends entirely on the speed of the I/O bus and CPU overhead for initializing the data transfer. If only the BMP microsteps are considered, a data rate of 800 KHz is possible. Measurements where the CPU is sending out random output channel numbers to be incremented, maximum rates of 215 KHz and 300 KHz are achieved for the MODCOMP IV/25 and 7860, respectively. When used in conjunction with the differential branch driver[3], actual data rates drop due to the speed of the CAMAC branch cycle and to conflict within the I/O processors. The resulting maximum data rate for the MODCOMP IV/25 is 150KHz. At 125KHz the associated composite deadtime is 0.6%.

When operating with macro instructions in the BMP, some degradation of data rates may be expected. Macro instructions execution times depend largely upon the speed of the main memory whereas microcode runs from the WCS with 50ns access and microstep cycle "tuned" for the shortest execution times possible. However, the BMP design optimizes macro instruction performance through use of interleaved memory and pipelined memory fetches. The resulting typical execution time for a register to register instruction reduces to the effective access time of the main memory, 330ns. Other simple instructions are generally multiples of 330ns depending upon how many memory accesses are required for the instruction. A complex instruction, like multiply register by register, for example, takes more microcode and correspondingly greater time, in this case 4.6us. Table 2 shows that BMP macro instruction times follow closely those for the MODCOMP 7830. While the BMP may not be much faster when functioning as a macro machine, it does so while leaving the more powerful minicomputer free to do other useful work.

V. FUTURE BMP EXTENSIONS

In the present implementation the BMP exists as a shared resource on the MODCOMP I/O bus serving data acquisition requests from several experiments. The configuration is cost effective since only one BMP is required for these experiments. To the extent that this configuration averages the memory needs of several experiments, a savings in memory may result. The chief disadvantage results from user interactions, both in deadtime effects and possible induced crashes. An auxilary crate controller version of the BMP is planned which would support only a single user and would bring data directly from CAMAC into the processor eliminating both the I/O bus congestion and the user interaction. Anticipated data rates should be limited only by the dataway cycle resulting in 1 MHz event rates. However, with the present TTL logic, little processing (if any) would be processed at that rate. Perhaps in yet another version, ECL logic could be applied to boost the processing capability.

*Supported by the U.S. Department of Energy

[1] R. V. Poore, D. E. McMillan, R. O. Nelson and J. W. Sunier, "Data Acquisition System at Los Alamos P-9 Accelerator Facility", IEEE Transactions on Nuclear Science, NS-26, 708 (1979).

[2] R. A. Hardekopf, R. V. Poore, J. W. Sunier, M. Neiman, and A. Holm, "Event Analysis Language for High Speed Data Acquisition", IEEE Transactions on Nuclear Science, to be published.

[3] D. E. McMillan, R. O. Nelson, R. V. Poore, J. W. Sunier, and J. J. Rose, "A High Speed CAMAC Differential Branch Highway Driver", IEEE Transactions on Nuclear Science, NS-26, No. 4, 4450 (1979).