ANL-HEP-CP--84-77

DE85 004817

DATA PREPROCESSOR AND COMPACTOR FOR THE SOUDAN 2 NUCLEON DECAY EXPERIMENT

J. W. Dawson, E. N. May, and N. Solomey
Argonne National Laboratory, Argonne, IL 60439

Abstract

This paper describes a prototype preprocessor data-compaction system for the Soudan 2 proton decay search experiment. The Soudan 2 experiment will have more than three million potential data words per event to examine, while less than one percent of these data words will have valid data for typical events. In an effort to reduce the amount of data to be stored and analyzed, a data preprocessor was developed which scans the data words. If a data word is valid (ADC count above a preset threshold), that data word is passed to the host computer for experiment monitoring and storage on magnetic tape. To obtain fast data compression, a hardware comparator is used. The hardware comparator places valid data into a FIFO (first in first out stack) where the host computer can acquire the data through CAMAC. The comparator and FIFO are controlled by a microprocessor (8086 CPU), and the microprocessor is programmed for decision-making and communication between the compactor, CAMAC, the host computer and a local terminal.

Introduction

The Soudan 2[1] nucleon decay experiment is a collaborative effort of Argonne National Laboratory, University of Minnesota, Tufts University, Oxford university, and Rutherford Appleton Laboratory. The purpose of the experiment is to measure the nucleon lifetime and to study the decay channels exhibited by candidate events.

The detector consists of Hytrel drift tubes sandwiched between sheets of steel. Mylar film with copper equipotential strips insulates the Hytrel from the steel, and drift potentials applied to the copper strips create a drift field inside the Hytrel tubes as a result of IR drops in the tube walls. Viewed from the end, a stacked assembly bears a resemblance to a honeycomb.

At the ends of the tubes are orthogonal sets of proportional wires and bussed cathode pads, yielding X-Y information about charge drifting down the Hytrel tubes. The proportional wire and cathode signals are bussed from 8 wire-cathode planes, amplified, and shaped. This effectively multiplexes the readout of a large detector for nucleon decay experiments. The signals are digitized by 6-bit CMOS flash encoders (RCA 3300) and written in CMOS static RAM (Hitachi 6116) using 9 bits of address information. The flash encoders and RAM's are clocked with a 150 ns clock, and since the drift velocity is approximately 1 cm/μsec, there is a spacial resolution (exclusive of diffusion) on the order of 2 mm.

For the 1200-ton detector currently envisioned at Soudan 2, there are 6144 signal busses coming to the electronics, and with 512 addresses per RAM there are something over 3 million data words with potentially valuable information in the event of a trigger. Since in a typical event only perhaps 1% of the data is valid (has a non-zero amplitude), there is a clear cut requirement to be able to compact the data. In the initial phases of this work, the data was compacted in the on-line software of a PDP 11/34. The compactor described in this paper shortened the compaction time by a factor of 500 from the time required by the PDP 11/34.

The data channels, including amplifiers, damping, flash encoders, and RAM's, are built on cards which conform dimensionally to the RABBIT[2] specification of the CDF group at Fermilab, although the backplane protocol is completely different. These cards are called Analog Cards and are addressed through the crate backplane by a crate controller which embodies the data preprocessor.

Description of Preprocessor

A block diagram of the major components of the data processor is shown in Fig. 1. The data processor functions as a crate controller using an Intel 8086 microprocessor running at 5 MHz, and an Intel 8259 interrupt processor supporting eight interrupts. The interrupts are used for such things as responding to FIFO full or empty, interrupt driven data transfers, sensing end of compaction, etc. The functions of the interrupt lines are given in Table 1. Two RS232 serial ports are provided using Intel 8251 UART chips; however, only one of the ports is tied to the interrupt processor to allow interrupt driven transfers. The non-interrupt driven port may be connected to a terminal for communications with a monitor program in PROM for hardware debugging and software testing. The baud rate for the two serial ports is 1200 baud. Decoding is provided for 48 read and write I/O instructions which are hard-wired to perform various tasks in the system. Representative tasks might be such things as clearing latches, reading registers, etc. Details of the I/O ports is given in Table 2.

There is a 16-bit register called the Crate Status Register (CSR) which is written and read from the 8086 CPU. The bits of the CSR serve to configure the hardware and to indicate the status of various important conditions in the crate controller. For example, the FIFO will take its input from the data bus, CAMAC, or the compactor depending on the status of two of the bits of the CSR. Another bit is used to indicate whether the crate is in a data-taking status. The meaning of each bit in the CSR is outlined in Table 3. The CAMAC/host system can read the CSR at any time and obtain a snapshot of the detailed crate status. This provides the mechanism for synchronizing the data transfer between the host/CAMAC and the 8086/FIFO systems.

The address space is 64K bytes, byte addressable. Each 16K-byte quadrant of memory may be either RAM or PROM, although typically the top 16K must be PROM since the microcode requires the boot routine to be at the address FFF0 hex. Since 64K constitutes one paragraph in the 8086's addressing logic, the segment register is not used, and the top four bits of the address bus are not decoded. Thus the maximum addressable memory space is 64K bytes.

The data format used is address x data, the layout of the bit fields in a 24-bit word is:

23 18	17 15	14 6	5 0
CARD	CHANNEL	RAM ADDR	ADC DATA
ADDR	ADDR	0 - 511	0 - 63
0-63	0 - 7		

Thus an 8086 crate controller can accommodate 64 eight-channel KDK analog cards, each with 512 address

Contributed paper to the IEEE 1984 Nuclear Science Symposium, Orlando, FL (October 31 - November 2, 1984)

JSW

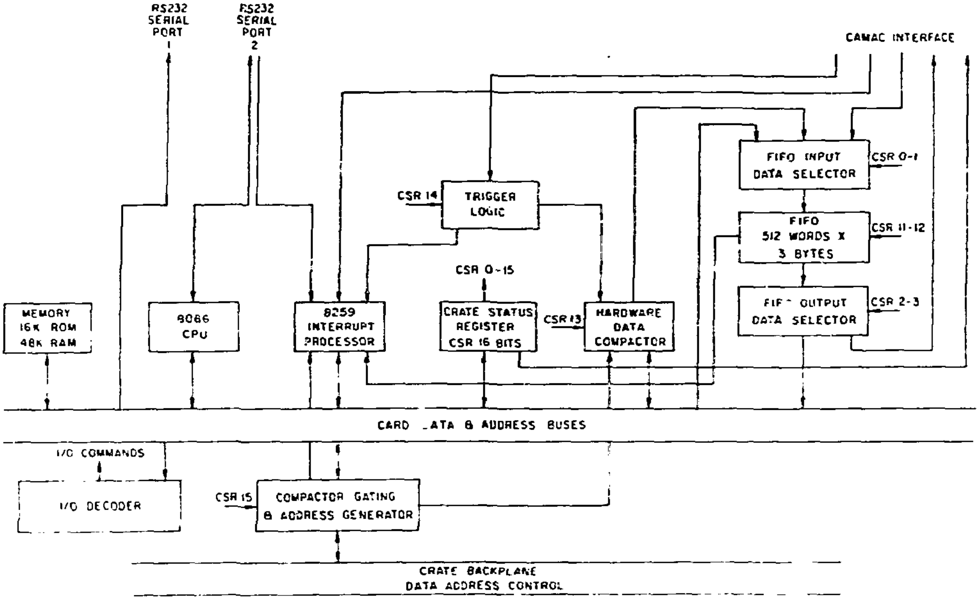

Fig. 1. *Block diagram of data preprocessor.*

memories.

The *FIFO* is organized as 24 bits wide by 512 words deep (Monolithic Memories C67401). Since the data bus of the 8086 is 16 bits wide, it is necessary when writing or reading to FIFO from the data bus to treat the bottom 16 bits of the FIFO as a word and the top 8 bits of the FIFO as a byte which is written/read from the low byte of the data bus. Transfers between CAMAC or the compactor and FIFO are 24-bit transfers, although transfers to CAMAC may be 16 bits. The FIFO provides signals called IDERR and RDERR indicating that the FIFO is either full or empty. These signals are available both to the 8086 and through CAMAC. During compaction, if the FIFO becomes full, IDERR will halt compaction and notify the 8086.

The compactor is a hard-wired micro-sequencer which is under software control of the 8086. The compactor has an address register which may be cleared or initialized by the *8086. This address register* provides address to the backplane bus which is used to address the analog cards, the RAM's on the analog cards, and the address in RAM. When compaction is to take place, the 8086 first clears the address register, and then gates it to a four-phase clock with a period of 600 ns. One phase is used to advance address. When the address register addresses a card, a RAM on a card, and an address in that RAM, a six-bit data word is returned via the data bus. The magnitude of this six-bit word is compared with the magnitude of a one-byte data word which has previously been written by the 8086 in the compactor's threshold data register. This is done on the next phase of the clock. If the magnitude of the data word exceeds that of the word in the register, a level called WRCOMPD is raised and latched. On the next phase of the clock, if WRCOMPD is high, the content of the address register is concatenated with the data word and pushed to the FIFO. The next phase of the clock is not used, and on the subsequent phase address is advanced and the cycle continues. Compaction is ended when the card address in the address register becomes equal to the value of a byte which has previously been written in the compactor's card-count register. At the conclusion of compaction, a level called CDONE is raised and it may be used to interrupt the 8086.

There is a single width CAMAC interface which supports 24-bit DMA's in either direction. In addition to data transfer, there are two CAMAC commands which may be used to interrupt the 8086. Commands to enable, disable and generate LAM's are available. Commands to generate external NIM level signals are available. The two error conditions from the FIFO indicating full and empty are also carried to the CAMAC interface, and are used to cause the CAMAC Q to go to zero in order to terminate a DMA. At the CAMAC interface, there is a NIM level input for an external trigger signal, which is *translated* to TTL and carried to the crate controller. This interface module allows triggers to be generated either by the host or by external logic. A complete description of *the available CAMAC commands is given in Table 4.*

At the crate controller, the receipt of a trigger initiates a run-down timer which counts 320 cycles of the 150 ns clock and then stops data taking, at which time compaction can begin. *This allows 29 μsec of* history of the ionization to be stored before the trigger and 48 μsec of drift history to be stored after the trigger.

Software for 8086 CPU

A comprehensive set of software tools has been developed to test and make use of the 8086 controlled preprocessor and compactor hardware. These include a stand-alone debugging monitor, a virtual terminal communications link to a VAX 11 *series computer for* code development (via cross software tools), and *downloading* of code to the 8086 memory, a data acquisition control program, etc. There is a system monitor, which is a modifed version of the Intel 86/12 monitor reconfigured for our memory layout and port numbers, and to support downloading of code from a VAX

11/730 through the RS232 port. The monitor supports memory reads and writes, and comparisons between two sections of memory. Input and output to various ports are also supported. The monitor permits one to input code in machine language and execute the code with setable breakpoints or single step through the code using the trap bit interrupt. The PROM version of the monitor program resides in high memory at address E000 to F500 hex.

A virtual t.rminal communications program was written that supports data transfer to the 8086 from the VAX 11/730. The 8086 monitor port is attached directly to the VAX 11/730 terminal port with a null modem connector. To communicate with the 8086 monitor, it is necessary to execute this communications program. There are two modes one could be in while running this communication program. First, the user's terminal could be attached as the command terminal for the monitor, and second, the user can be in the command mode where the VAX generates command strings to be sent to the monitor program in the 8086. With this system implemented, it became possible to prepare code in 8086 assembly language on the VAX, execute a cross assembler that would produce 8086 machine code in a HEX file format; this HEX file could then be downloaded via the virtual terminal program to the 8086 and executed under control of the 8086 monitor.

With the ability to write programs in assembler and download them into the 8086 for execution, an extensive series of debugging programs were written. A program was written to test each subsystem of the electronics. One program tested the addressing of memory and the dependability of each bit of each memory word. A second program tested the compactor electronics. There were programs written to test the CAMAC data transfer in both directions. To execute these programs, it is necessary to download the HEX file into memory and then, while communicating with the monitor, execute the program by giving the entry point into the program as well as the last executable statement for the break point. Errors found are reported by the virtual console terminal.

An assembly language code was written to handle data collection. This was a simple single buffered compaction routine; it handles an interrupt and before the 8086 CPU starts data collection again, all data transfer operations to the host/CAMAC must be completed. This compactor software was designed for a CAMAC system interfaced to a VAX 11/730 or a PDP 11/34, using the Argonne MIDAS or the Fermilab MULTI data acquisition software package, respectively. The compactor machine code is stored in PROM memory, CAMAC commands are available to boot the processor, send initial parameters, and begin execution. Then upon receiving an event interrupt, the 8086 starts the hardware compactor and waits for either a FIFO full interrupt or a compactor done interrupt. After it receives either of these interrupts, it notifies CAMAC of an event. When the 8086 receives the FIFO empty interrupt, it either returns to waiting for another event if compaction was done, or for the next FIFO of data to be transferred to the host computer. To provide synchronization between 8086 and host computer, two bits in the CSR can be read by the CAMAC system to determine if the 8086 needs to place more data into the FIFO to complete the readout of the current event.

A general facility to pass commands and data from the host/CAMAC system to the 8086 microprocessor has been implemented. It has a message header format to allow the 8086 to identify an incoming message of variable length. The 24-bit CAMAC field is divided as 8 bits of command and 16 bits of data. The message command interpreter is a useful software device that is part of the compactor software. It uses the FIFO and the two CAMAC interrupt lines to the 8086 to pass data to the 8086. The message command interpreter is executed on an interrupt-0, and the 8086 configures the FIFO so that CAMAC may write into the FIFO and the 8086 may read from the FIFO. Next, the host computer writes a four-word pattern into the FIFO with the last word being the number of commands to execute. Then commands are written into the FIFO, such as set compactor threshold, set number of cards to scan, and routine to execute next. After the host is finished writing data, CAMAC sends an interrupt-1 for the 8086. When the 8086 receives this interrupt-1, it reads the FIFO and executes the commands. The 8086 then returns to the data acquisition routine and waits for an event interrupt.

Conclusion

For the Soudan 2 detector, a number of improvements are planned.[3] The crates will be MULTIBUS instead of the pseudo-rabbit which will make the cards smaller and much easier to fabricate. The crate controller will be implemented using a commercially available 86/12A or 86/05 single-board computer plus a Compactor Card (COM) of our own design. The COM will incorporate the functions which are not included in the CPU, such as the compactor, crate status register, FIFO, etc. The FIFO will be implemented in CMOS static RAM with an 8X60 FIFO RAM manager chip. The RAM will be part of the CPU's address space so that the CPU has immediate access to compacted data. The FIFO will be 32 bits wide by 4K words deep, and 16-, 24-, and 32-bit tranfers will be supported. The compactor will operate with pipelined address and data so that 150 ns per address will be required. Compaction will be more completely under the control of the 8086 and only one RAM will be compacted at a time. RAM's with no valid data will be identified by an active channel latch and compaction may bypass them completely. There will be a transfer byte count register, so that the number of bytes transferred in a DMA will be under software control of the 8086. A crate will compact sixteen 16-channel analog input cards, and when all channels contain data, approximately 40 msec will be required to process a full crate. Twenty-four such crates operating in parallel will be used in the Soudan 2 experiment.

We would like to thank R. Laird for board level design and construction, and D. Wood for the VAX virtual terminal software.

Table 1

Function of the Interrupt Lines on the 8259

IR0	CAMAC generated interrupt by F24.
IR1	CAMAC generated interrupt by F30.
IR2	8251A UART serial port #2 has received a character and then can interrupt 8086 using this line so that the character can be processed.
IR3	8251A UART serial port #2 is readv to transmit another character.
IR4	Event interrupt for 8086 can be generated by: 1. External trigger in CAMAC module. 2. CAMAC F28. 3. 8086 write to port 47.
IR5	IDERR A write error has occured in the

		FIFO. FIFO is full, compactor is automatically stopped so no data is lost, and 8086 is given this interrupt so that it may process FIFO data or interrupt the host computer.
IR6	CDONE	Compactor is done. FIFO is automatically closed.
IR7	RDERR	Read error in FIFO. FIFO is now empty, computer may restart compactor to fill FIFO again.

Table 2

8086 I/O Ports

I/O C	8-bit write to eight LSB of CSR. 16-bit read of CSR.
I/O 2	16-bit write and read to 16 LSB of FIFO.
I/O 3	6-bit read of the data from RAM pointed by Address, Chan, Card.
I/O 4	12-bit read and write of Address and Chan ∅ 3 bit 9 bit chan Address
I/O 5	6-bit read and write of card ∅ in selecting address.
I/O 6	6-bit write to set threshold for compactor. If (ADC (I).GT.(I/06)), then put in FIFO.
I/O 7	6-bit write to set the number of cards to scan. If (card (1).EQ.(1/O 7)) compactor done scanning cards.
I/O 10	8-bit read and write to 8 MSB of 24-bit FIFO.
I/O 12	RS232 Port ∅1 on connector JF programming and status port with 8-bit read and write.
I/O 14	RS232 Port ∅1 on connector JF data port with 8-bit read and write.
I/O 16	Port A for the 8258 interrupt controller. This port is the programming port.
I/O 18	Port B for the 8259 interrupt controller. This port is used for setting the mask on the 8 interrupts line.
I/O 17	(Wnd) set CSR 8 to a one.
I/O 19	(Wnd) set CSR 9 to a one.
I/O 20	(Wnd) clear CSR 10 by setting it to a zero.
I/O 21	(Wnd) set CSR 10 to a one.
I/O 22	(Wnd) clear CSR 11 by setting it to a zero.
I/O 23	(Wnd) set CSR 11 to a one.
I/O 24	(Wnd) clear CSR 12 by setting it to a zero.
I/O 25	(Wnd) set CSR 12 to a one.
I/O 24	9-bit read of the trigger RAM address (TRA).
I/O 26	(Wnd) clear CSR 13 set to a zero.
I/O 27	(Wnd) set CSR 13 set to a one.
I/O 28	(Wnd) clear CSR 14 set to a zero.
I/O 29	(Wnd) set CSR 14 set to a one.
I/O 30	(Wnd) clear CRS 15 set to a zero.
I/O 31	(Wnd) set CRS 15 set to a one.
I/O 32	(Wnd) Master reset line to all FIFO I.C. A pulse on this line clears the data bits in the FIFO.
I/O 33	(Wnd) Clear the FIFO full interrupt latch.
I/O 34	(Wnd) Clear the FIFO empty interrupt latch.
I/O 35	(Wnd) Clear the computer done interrupt latch.
I/O 36	(Wnd) Generator for compactor clear sequence.
I/O 37	(Wnd) Clear interrupt-1 la·ch.
I/O 38	(Wnd) Clear interrupt-0 latch.
I/O 39	(Wnd) Clear address, card ∅, chan ∅.
I/O 40	RS232 Port ∅2 on connector J2 programming and status port with 8-bi· read and write.
I/O 42	RS232 Port ∅2 on connector J2 data port with 8-bit read and write.
I/O 41	(Wnd) Clear CSR 8-bit set to a zero.
I/O 43	(Wnd) Clear CSR 9-bit set to a zero.
I/O 45	(Wnd) Set LAM on CAMAC and Interrupt A output.
I/O 46	(Wnd) Set Interrupt B output.
I/O 47	(Wnd) 8086 generated event interrupt.

*(Wnd) stands for the function is only a write port and no data is read by the devices.

Table 3

CSR Interpretation

This table gives the mean of each bit in the crate status register (CSR).

Bit Field Position	Interpretation
CSR0-CSR3	Used to control FIFO dataway input and output.
CSR0-CSR1	2-bit number decoded to determine input device: 0 = in from CAMAC. 1 = in from 8086 data bus. 2 = in from compactor.
CSR2-CSR	2-bit number decoded to determine output devices: 0 = out to CAMAC. 1 = out ot 8086 data bus.
CSR4-CSR7	(Not used)
CSR8	Error in message interpreting routine: 1 = yes error. 0 = no no error.
CSR9	Message interpreting routine running: 1 = yes 8086 still executing code in message Interrupting routine.

0 = no 8086 message interrupting routine ready to receive next CAMAC command.

CSR10 Ready bit-this indicates whether CSR bit 11 is valid:

1 = CSR 11 valid.
0 = CSR 11 not valid.

CSR11 This bit is used to indicate that data in the FIFO that has not been read out is the last data for this event.

1 = end of event.
0 = not last FIFO in event.

CSR12 FIFO gate. The FIFO input and output can be enabled and disabled by bit 12 in the CSR.

0 = FIFO close no other may be read or written.
1 = FIFO open read and writes enabled.

CSR13 Hardware compaction done indicator. This latch controls the starting and stopping of compactor.

1 = compactor on.
0 = compactor off.
Compactor turns itself off when compaction is complete.

CSR14 TRON determines if a trigger can be accepted.

1 = accept trigger.
0 = ignore trigger.

CSR15 DAT indicates if the analog cards are clocking and digitizing or if there has been a trigger so:

1 = analog cards are digitizing and writing data into RAM.
0 = analog cards have stopped writing data into RAM because there was an event.

Table 4

Available CAMAC Commands

F0 24-bit read of data from FIFO, Q-bit response becomes a zero when FIFO is empty.

F2 16-bit read of the 8086 Crate Status Register (CSR).

F16 24-bit write of data into the FIFO, Q-bit response becomes a zero when FIFO is full.

F18 Enable LAM. The LAM line on the crate will be masked off if this latch is not enabled.

F20 Disable LAM, and clear LAM and interrupt A output. This command disables LAM for crate but does not disable Interrupt A output.

F22 Clear Interrupt B output. Interrupt B output is reset to its 0 state.

F24 Interrupt line to 8259 interrupt controller on the 8086 system F24 = IR0. This command also resets the IDERR and RDERR latches, and clear the FIFO.

F26 Boot the 8086.

F28 CAMAC generated event interrupt controller on the 8086 system F30 = IR1.

Note all above commands are with A = 0.

References

1. "The Soudan 2 Nucleon Decay Experime.t: Description and Status Report," Soudan 2 Collaboration (University of Minnesota, Argonne National Laboratory, University of Oxford, Rutherford-Appleton Laboratory, Tufts University), ANL-HEP-PR-84-30 (September 1984).

2. T. F. Droege, K.J. Turner, and T. K. Ohska, CDF-119, PIN52F, Fermilab Internal Report (June 16, 1982).

3. "Conceptual Design of Soudan 2 Electronics," J. Dawson, Argonne HEP internal document PDK-115 (April 2, 1984); Addendum I, PDK-122 (April 24, 1984).

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of 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. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.