ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S11LLC09

A CAMAC-Resident Microprocessor For The Monitoring Of Polarimeter Spin States.

David Reid, Derek DuPlantis, Neil Yoder
Indiana University Cyclotron Facility
2401 Milo B. Sampson Lane, Bloomington, IN 47405

Don Dale
TRIUMF
4004 Wesbrook Mall, Vancouver, B.C. Canada V6T 2A3

Abstract: A CAMAC module for the reporting of polarimeter spin states is being developed using a resident microcontroller. The module will allow experimenters at the Indiana University Cyclotron Facility to monitor spin states and correlate spin information with other experimental data. The use of a microprocessor allows for adaptation of the module as new requirements ensue without change to the printed circuit board layout.

I. Introduction

A custom CAMAC module has been developed to allow for the remote monitoring of polarimeter spin states in the Indiana University Cyclotron Facility . The module will provide experimentalists at the facility with data on spin states of the particle beam and allow them to correlate experimental data with this information. The cyclotron control computer sets the polarized ion source by means of a task which also acts as a network multi-node server, providing spin-state data to client tasks on one or more data-acquisition computers, each of which will copy current spin-state data to the module. The standard data-acquisition programs will access the module while reading other event data.

II. Module Functionality

To meet the experimental demands outlined above the module has two eight bit registers for holding the polarimeter states. One register holds the current state and the second register holds the latched state. Most data-taking events read the current state register. For certain CAMAC commands the latched state is read and the current state is moved into the latched state register.

The experimentalist's interface to the module is through seven output LEMO connectors on the front panel of the module. Three connections give the spin data. One connection gives information on what type of particle (proton/deuteron) is being accelerated through the cyclotron. A ready bit, an interrupt bit, and a valid bit are also brought out to the front panel.

A sixteen bit timer countdown register is used in the module. This register counts down in 0.1 second units. This register is loaded by a CAMAC write operation and begins counting down immediately upon being loaded. When the countdown register makes the 1 -> 0 transition the module, current state register, and the valid outputs are cleared.

A sixteen bit sequence number register holds polarization cycle sequence number to permit correlation of event streams from different data acquisition computers. The sequence number register is specified by the data acquisition computer and written to the module.

III. Module Components

i. Microcontroller

To meet the functionality requirements for the module it was decided to use a resident microcontroller. The use of the microcontroller will allow the module to meet the current specifications and give additional flexibility to meet future demands. By using a microcontroller the module can be readily adapted to future uses without the costly printed circuit board redesign that would result from the use of logic circuitry.

The microcontroller that was chosen for this project is the Intel 80C196KC. The chip has a sixteen bit wide internal data bus. Because the registers for the module are specified to be eight or sixteen bits wide the internal data bus for the chip allows for direct, full width register operations.[1]

The module can perform complete operations before the next CAMAC cycle due to a 16 MHz clock.[2]

ii. External Memory

The 196 can take advantage of external memory devices. This features makes the use of external ROM and RAM onboard the module possible.

The module uses three external memory devices. Two 8K x 8 EPROMs are used to hold the code for the microcontroller. The microcontroller accesses the code for its internal operations from the EPROMs. Two chips are used in parallel to allow for sixteen bit wide memory words to be used.

Two 8K x 8 RAM chips are used for external memory register space for the microcontroller. For the functionality of the module as now specified this additional

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S11LLC09

RAM space is not necessary. However, the decision to incorporate this memory space was made to provide for unforeseen modifications to the requirements for the module. By designing in this extra memory the module will be able to do more complex data and register operations in the future. Again, because of the RAM chips memory structure, two chips were used to allow for sixteen bit data words.

One Integrated Device Technology 7133 Dual Port RAM is used as register space for data that will be written onto the CAMAC data bus. The use of this is necessary to meet the CAMAC specifications for dataway operations. The microcontroller (even with a clock frequency of 16 MHz) cannot respond to a CAMAC write command fast enough to have the appropriate data present on the CAMAC dataway when the S2 line is asserted 700 ns into the CAMAC data cycle.[3] The IDT chip was chosen because of its 55 ns address access time.[4] This speed will allow for valid data to be present at the dataway for the read command.The IDT chip has a sixteen bit wide memory structure.

The usage of a dual port RAM chip is acceptable because of the functionality requirements of the module. No data manipulations must be done on the data present in the registers in the 700 ns from the beginning of a CAMAC data cycle to when valid data must be present on the dataway.

To load the appropriate register data onto the dataway within 700 ns an EPLD is used to decode the CAMAC Function and Address lines. This EPLD logically recognizes the CAMAC read commands and then sets the address to point to the correct memory location in the Dual Port Ram. When the S2 line is asserted the Dual Port then writes the requested register data onto the CAMAC dataway.

The EPLD also sets the specified values for the Q, X, and Look-At-Me (LAM) lines. Use of the EPLD insures that these signals are present on the dataway at the appropriate times in the CAMAC cycle. The Q line can be asserted or deasserted, or can be set to reflect the state of the ready bit or the LAM. The EPLD also sets X=1 for all valid functions.

iii. Data Latches

Because of the timing constraints of the CAMAC dataway, cycle valid data from a CAMAC write operation will not be present on the dataway by the time the microcontroller can respond.[5] To solve this problem, 74AS573 Transparent Latches are used to latch in the write data and hold it until the microcontroller is ready to process the information. The data is latched off the bus by the logical AND of the N and S1 lines. This insures that valid write data is held for the microcontroller.

The microcontroller memory maps the write data latches into its external memory structure. When the microcontroller is ready to process the write data, it asserts the correct address lines and processes the information.

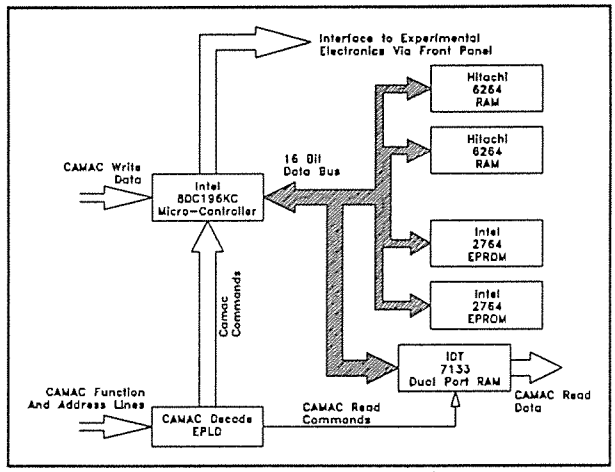

Figure 1 Module Block Diagram

These latches also store the CAMAC Function and Address lines. This information is held until the microcontroller can process and decode the commands.

All external memory decoding will be handled by a second EPLD. This EPLD decodes the address bus from the microcontroller and asserts chip enable lines to each external memory device. Addresses were specified to allow for use of the entire memory space of each external device in the future.

IV. Camac Functions

The module has been designed to respond to the following commands, while setting Q and LAM lines accordingly.

F(0)A(0)	Read entire Polarity Register, set Q to the value of the ready output pin.
F(2)A(0)	Read entire Polarity Register, latch the current state into the latched register, set the ready output pin to the value of the 5th bit of the latched register, clear the LAM and interrupt pin, and set Q to previous ready output pin.
F(0)A(4)	Read Timer register, assert Q line.
F(0)A(6)	Read Sequence register, assert Q line.
F(9)A(1)	Clear the current Polarity register, clear the timer register, clear ready bit, set LAM and interrupt output pin, set Q to LAM value.
F(16)A(1)	Write current state, clear the ready bit, assert LAM, set interrupt bit, assert Q.
F(16)A(4)	Write timer register, assert Q.
F(16)A(6)	Write sequence register, assert Q.
F(24)A(0)	Disable LAM, deassert Q.
F(26)A(0)	Enable LAM, deassert Q.
F(8)A(0)	Test LAM, set Q=LAM.
F(10)A(0)	Clear all registers, clear LAM, set Q=LAM, clear interrupt bit.

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S11LLC09

F(16)A(0) Write entire polarity register, assert Q.
F(27)A(1) Test interrupt bit, Q=interrupt output.

The module will enter appropriate reset states by the assertion of the Z, I, or C lines. The Z signal resets the module by clearing all registers, disabling the LAM, and stopping the countdown timer. The C line clears the registers and stops the timer. The I line inhibits the module and precludes any processing while the line is asserted.

V. Software

Programming, written in Assembly, for the microcontroller is stored in the EPROM chip. Software development for the module will be done using the Intel EV80C196KC Evaluation Board and 2500AD 8096 Assembler software on a PC. The EPROM chips will then be burned with the code for the module.

The microcontroller is in an idle state until the module is addressed. An interrupt to the microcontroller is generated by the N line of the CAMAC dataway. This interrupt causes the microcontroller to access the EPROM memory vector and begin executing the program. The microcontroller will then access the F()A() latches. The controller decodes these and proceeds with the proper operations. During all program executions the microcontroller uses a high speed output pin to assert the X line as a module busy line. This alerts the computer is still processing and avoids the possibility of the module receiving a second interrupt before the processing of the first command is complete.

VI. Status

As of this writing the module is ready to enter the testing phase. The design of the circuitry and printed circuit board layout has been completed and the module has been assembled. Software development for the microcontroller needs to be completed and extensive testing is required before the module is ready to be incorporated into experimental setups.

VII. Acknowledgements

The authors wish to acknowledge the contributions to this work by T. Capshew, J. Collins, D. W. Frame, J. Graham, and the Computer/Electronics group at Indiana University Cyclotron Facility.

This work was done under the support of National Science Foundation Grant #NSF PHY 90-15957.

VIII. References

[1] Intel, *16-Bit Embedded Controllers Handbook*, Mt. Prospect, IL, 1990, pg. 5-1 - pg. 5-126
[2] Peter Clout, *A Camac Primer*, LA-UR-82-2718, Los Alamos National Laboratory, 1982, pg. 72 - pg. 74
[3] IEEE Std 583-1975 pg. 50
[4] Integrated Device Technology, *High-Speed CMOS Databook*, Santa Clara, CA, 1988, pg. 5-61
[5] Peter Clout, *A Camac Primer*, LA-UR-82-2718, Los Alamos National Laboratory, 1982, pg. 33