SLAC-PUB-2413
October 1979
(I)

SNOOP MODULE CAMAC INTERFACE TO THE 168/E MICROPROCESSOR*

D. Bernstein,† J. T. Carroll, V. H. Mitnick, L. Paffrath, D. B. Parker
Stanford Linear Accelerator Center
Stanford University, Stanford, California 94305

Abstract

A pair of 168/E microprocessors will be used to meet the realtime computing requirements of the SLAC Hybrid Facility. A SNOOP module and 168/E Interface provide the link between the host computer and the microprocessors. By eavesdropping on normal CAMAC read operations, the SNOOP provides a direct data transfer from CAMAC to microprocessor memory. The host computer controls the processors using standard CAMAC programmed I/O to the SNOOP.

Introduction

The SLAC Hybrid Facility (SHF) selectively takes bubble chamber pictures by using data from external particle detectors, i.e., proportional wire chambers, Čerenkov counters, etc.[1] An electronic fast trigger initiates transfer of data from detectors to CAMAC modules, the host computer reads the CAMAC modules, and a software algorithm selects triggers of interest. This entire sequence, including the software decision, must be completed within the bubble chamber flash delay of 2.5 - 3.0 ms. A pair of 168/E microprocessors[2] will be used to execute more efficient algorithms and increase the quantity of data that can be processed within this time constraint.

The SNOOP Module and a 168/E interface provide a CAMAC link between the host computer (currently a NOVA 840) and the 168/E microprocessors. A program running on the host computer can control the processors using standard CAMAC I/O to the SNOOP, e.g., load program memory, start processor, check processor status, read results from 168/E data memory, etc. The SNOOP also provides a fast direct data transfer from CAMAC to 168/E memory by eavesdropping on I/O to other CAMAC modules in the same crate. When SNOOP is in the Listen Mode, CAMAC data can be transferred simultaneously to the host computer and microprocessor memory at a rate of 2 μs per 16 bit word. The CAMAC interface minimizes the data transfer time to the microprocessor.

A block diagram of the system is shown in Fig. 1. The host computer uses NC023C NOVA controllers to access any CAMAC module including the SNOOP. The SNOOP usually occupies slots 22 - 23 next to the controller,

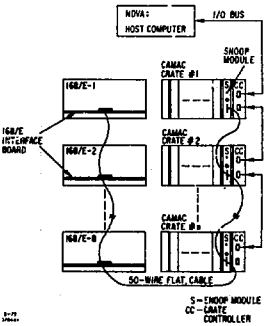

Figure 1.
System Diagram

but it can be placed at any normal station number. SNOOP modules and 168/E interface boards are all connected in a daisy chain using a 50 wire flat cable. Any one SNOOP module can control and transfer data to 1 - 8 microprocessors, and data from multiple CAMAC crates can be transferred directly by using a SNOOP in each crate. In the proposed SHF configuration two CAMAC crates are interfaced to a pair of 168/E microprocessors.

SNOOP Module

Figure 2 shows a block diagram of the SNOOP module, with the CAMAC bus on the left side and the I/O lines to the 168/E on the right side of the figure. CAMAC functions for controlling the SNOOP are defined in Table I. As shown in Fig. 2, the SNOOP can operate in the following three modes:

W = Write Mode,

R = Read Mode,

LM = Listen Mode.

Operation of the SNOOP in the Write or Read Modes follows usual CAMAC protocol, i.e., the station number (N) addresses the module, and the SNOOP intercepts the control and function lines. The S1 strobe is used primarily to transfer data between the host computer and 168/E, while the S2 strobe is used for internal control.

In the Write Mode, data is transferred from the host computer to 168/E processor via the CAMAC crate controller, SNOOP module and 168/E interface. The SNOOP decoder interprets write commands F(16,17,20,22), as defined in Table I, as the same function as far as the subaddress, function and write data lines are concerned. Only F(19) is separately decoded and in coincidence with S2 loads the SNOOP module word counter. Functions F(16) and F(17) load 168/E data and program memory, which are physically and logically separate on the processor memory boards.

In the Read Mode, data is transferred from the 168/E to the host computer via the 168/E interface, SNOOP module and crate controller. The SNOOP decoder interprets read functions F(0,1,4,6) as the same function as far as the subaddress, function and read data lines (in the direction toward the CAMAC bus) are concerned. With the exception of F(19), most of the logic to process read or write functions is on the interface board. The SNOOP decodes the CAMAC command as R-Mode or W-Mode and loads the function line drivers as shown in Fig. 2.

In the Listen Mode (LM) of operation, any data the host computer reads from a CAMAC module in the same crate as the SNOOP is also written into 168/E data memory. Set Listen Mode function F(28) activates the LM control line which takes over the function of the N-Line (station address). Consequently, the SNOOP is active regardless of which modules are addressed by succeeding CAMAC instructions. In this mode the SNOOP's 16 bi-directional R-lines are gated toward the 168/E interface and any data on the CAMAC R-lines which the host computer reads is also strobed into 168/E memory. In the LM-Mode the SNOOP does not pass the CAMAC function and subaddress lines to the 168/E interface, and the main function decoder is inhibited. Clear Listen Mode F(30) can be recognized and executed using the S2 strobe and a separate decoder activated by the LM control line. The condition S2·F(30) clears the LM-Mode and returns the SNOOP to a state which recognizes non-LM commands. Since the SNOOP in LM-Mode is sensitive to any CAMAC I/O, all

*Work supported by the Department of Energy under contract number DE-AC03-76SF00515.
†On leave from The Weizmann Institute, Rehovot, Israel.

Presented at the 1979 Nuclear Science Symposium, San Francisco, California, October 17-19, 1979.

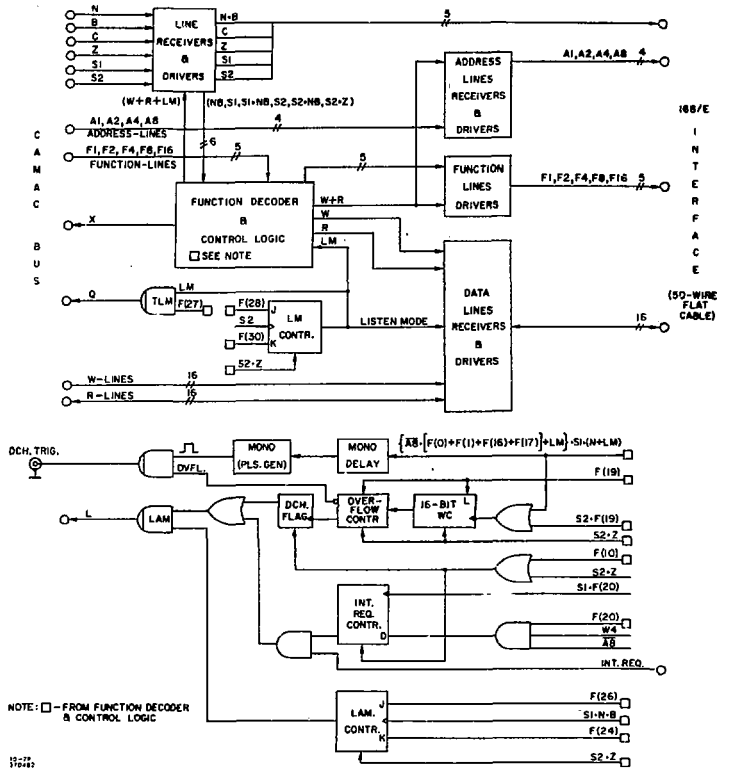

Figure 2. SNOOP Block Diagram

instructions to other modules which are required to set up the data transfer should be completed before setting LM. Similarly, a safe programming protocol is to clear the LM-Mode as soon as the data transfer is completed. LM data transfer can use either CAMAC programmed I/O or Direct Memory Access (DMA) in triggered or autonomous modes.

In any of the three modes of operation, (W-Mode, R-Mode and LM-Mode), the SNOOP Word Counter (WC) can be used to supervise a DMA data transfer. For this purpose, the function F(19) loads the WC with the two's complement of the number of 16-bit words to be transferred. After each transfer cycle, the WC is incremented, and an external DCH trigger is sent to the crate controller to initialize the next transfer cycle. When the WC overflows, the OVERFLOW control sets the DCH-flag, which in turn activates the L-line (if LAM has been enabled using F(26)), and the transfer stops. The WC is especially convenient for DMA transfers from the 168/E memory — it is not required for LM-Mode transfers to the 168/E data memory.

In addition to a WC overflow, the SNOOP L-line can also be activated by an Interrupt Request (INT.REQ) from the 168/E interface. The INT.REQ flip-flop is set true by a Write Status Register F(20) command with W4 (bit-3) in the status register. A SNOOP module with the INT.REQ set is expecting an interrupt and will set its LAM if an interrupt request is received from a 168/E interface.

The SNOOP is currently a double width CAMAC module with external connectors on the front panel for the 50 wire cable to the 168/E interface and the DCH trigger cable. It has a single LED which is gated on whenever the SNOOP is active. The Function Decoder uses a 32 × 8 bit PROM, which substantially reduced the number of IC's which would have been required with conventional line decoders.

TABLE I: SNOOP / INTERFACE FUNCTIONS

FUNCTION	A8	TYPE*	SYMBOL	DESCRIPTION
0	1	S/I	RDM	Read Data Memory
0	0	S/I	RDM	Read Data Memory and Increment Address Reg.
1	1	S/I	RPM	Read Program Memory
1	0	S/I	RPM	Read Program Memory and Increment Address Reg.
4	1	S/I	RPC	Read Program Counter
4	0	S/I	RSR	Read Status Register
6	1	S/I	RAR	Read Address Register
6	0	S/I	RDSR	Read Device Select Reg.
10		S		Clear Look-At-Me (LAM)
16	1	S/I	WDM	Write Data Memory
16	0	S/I	WDM	Write Data Memory and Increment Address Reg.
17	1	S/I	WPM	Write Program Memory
17	0	S/I	WPM	Write Program Memory and Increment Address Reg.
19		S		Load Word Counter
20	1	S/I	WPC	Write Program Counter
20	0	S/I	WSR	Write Status Register
22	1	S/I	WAR	Write Address Register
22	0	S/I	WDSR	Write Device Select Reg.
24		S		Disable LAM
26		S		Enable LAM
27		S	TLM	Test Listen Mode (Q-Line)
28		S	SLM	Set Listen Mode
30		S	CLM	Clear Listen Mode
31		I	LM	Listen Mode Data Transfer

*S = SNOOP Module / I = Interface Board

168/E Interface

For the SNOOP module, it is sufficient to decode functions into three categories: READ, WRITE and CONTROL. However, the Interface Board must decode and process 17 of the 24 commands in Table I. Function processing by the 168/E Interface can be classified in four categories:

1) Device Selection,
2) Address Register,
3) Status Register,
4) Data I/O (Memory and Program Counter).

In the Listen Mode, F(31) is generated by the SNOOP module and executed on the Interface Board to write data memory and increment the Address Register (the subaddress and function lines are inhibited in this mode). The rest of the Interface functions in Table I are part of the CAMAC instruction set for the SNOOP module. A block diagram of the 7 /E Interface is shown in Fig. 3.

The SNOOP and Interface can control up to eight 168/E processors, and Device Selection refers to identification of the processor(s) referenced by a CAMAC command. Although all the 168/E Interface Boards have identical construction and are exchangeable, only one Master Board is loaded with the IC's necessary to receive and decode the device selected. The 50-wire flat cable includes eight device selection lines which connect the Master Board with other Interface Boards in a Daisy chain (See Fig. 1). Each Interface Board has a DIP-switch which selects one of these eight lines and defines the interface-processor number (0-7). There are two procedures for selecting a 168/E Interface Board:

i) The CAMAC instruction selects the board using subaddress bits A1, A2 and A4. Functions F(0, 1, 4, 6, 16, 17, 20, 22) can use this type of selection.

ii) The Interface is preselected by function F(22)·$\overline{A8}$ which loads the 8-bit Device Select Register (DSR) with its corresponding interface number (DSR = 1 selects Interface # 0).

When the DSR is non-zero the CAMAC subaddress lines (A1, A2, A4) are inhibited. The DSR must be used for Listen Mode data transfers since the subaddress lines are inhibited on the SNOOP module in this mode of operation. By using the DSR data can be transferred simultaneously to any combination of processors.

Data transfers to/from the 168/E program or data memory are executed under supervision of the Address Register. This register has two modes of operation:

i) Static Address. Function F(22)·A8 loads the Register and one of the commands A8·(F(0) + F(1) + F(16) + F(17)) is used to transfer data to/from the corresponding location in the processor memory.

ii) Incrementing Mode. The Address Register is loaded with the initial memory location and the commands $\overline{A8}$·(F(0) + F(1) + F(16) + F(17)) increment the Address Register after each I/O transfer.

For reading or writing the Address Register, the memory location is defined in halfword units. The 168/E processor has a 32-bit word length (for data memory) while the CAMAC SNOOP module and consequently the 168/E Interface use a 16-bit word length. For user's convenience and system versatility, each word in memory is divided into upper and lower halves (most significant and least significant half), with the selection made by the least significant bit-0 (= W1) of the 16-bit address. Bits 1 - 12 of the address select one of the 4K program or data memory addresses on each memory board, while bits 13 - 15 select one of the eight possible memory boards. On the 168/E Interface the 16-bit Address Register has the least significant bit duplicated in a flip-flop which is clocked at the same time as the Address Register. This extra bit controls alternatively the transfer to/from the upper and lower halves of a word, while the Address Register is incremented after two 16-bit transfers.

An 8-bit Status Register on the 168/E Interface Board provides the communication path between the host computer and the processor. Using this register as defined in Table II, a program on the host computer controls the 168/E processor and determines its status. Function F(4)·A8 Reads the Status Register (RSR), and F(20)·A8 Writes the Status Register (WSR). Only bits 2 and 3 are important for understanding the SNOOP and Interface operation. By setting bit-2 = 0, the host computer takes control of the 168/E backplane — this must be done before memory I/O instructions. Bit-3 enables/disables the Interrupt Request (INT.REQ) flip-flop on the SNOOP module which issued the WSR command. If Bit-3 is set in the Status Register the 168/E Interface will set the INT.REQ line in Fig. 3 when the processor executes a HALT instruction.

The block diagram in Fig. 3 shows the 16-line data path between the 50-wire flat cable and the 168/E processor bus. The Interface Board is located in the crate with the processor and communicates with it through the backplane of the crate.

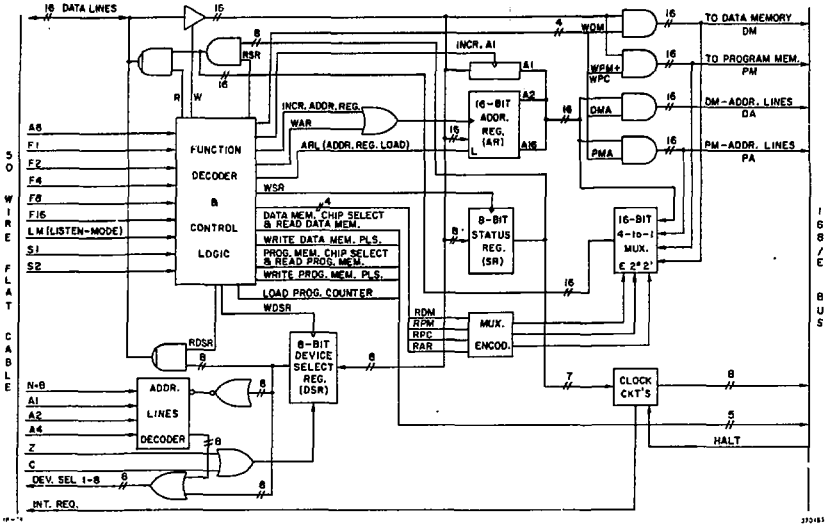

Figure 3. 168/E Interface Block Diagram

Data which originates in the Program Counter, Program Memory, Data Memory, and Address Register, is selected by eight Dual 4-line to 1-line multiplexers (MUX.). The 16 output lines from the MUX. are OR-ed with the 8 Status Register data lines and sent to the flat cable through 8T38 line receivers/drivers.

The 168/E Interface is constructed on a 32-DE-HEX wire-wrap board. Decoding the F-lines on the Interface instead of on the SNOOP simplifies the construction by reducing the number of lines required in the flat cable and the number of SNOOP drivers and Interface receivers. To further minimize the number of IC's required, the Interface uses three 32 × 8-bit PROM's instead of the more common Line Decoders. The 3-phase 168/E clock circuit, which is driven by a 20 MHz crystal oscillator, is built on the Interface Board. This solution has some advantages as far as the interconnection between the Interface and Processor is concerned.

TABLE II: 168/E STATUS REGISTER

BIT	TYPE	DESCRIPTION	STATE
0	WRITE ONLY	Start 168/E Program execution	1 = Start
1	READ/ WRITE	Single Step or Continuous Run Mode	0 = Single Step 1 = Continuous
2	READ/ WRITE	Control of 168/E backplane - source of address information	0 = Interface 1 = 168/E CPU
3	READ/ WRITE	Enable/disable interrupts when 168/E executes HALT instruction	0 = Disabled 1 = Enabled
4	READ ONLY	Set when 168/E executes a HALT instruction	0 = No Halt 1 = Halt
5	READ ONLY	Indicates when 168/E is executing instructions	0 = Stopped 1 = Running
6	READ ONLY	Indicates when 168/E DC power is on	0 = DC off 1 = DC on

Summary

The SNOOP module and 168/E Interface have been built and tested using 168/E diagnostic routines which execute on a NOVA 840. The system operates as designed in both the conventional Read/Write Modes and in the Listen Mode.

Acknowledgments

We would like to thank R. N. Fall and P. F. Kunz for their assistance in implementing this interface to the 168/E Processor.

References

1. C. V. Cautis et al., SLAC-PUB-2233 (1978), (to be published in Nuclear Physics B).
2. F. F. Kunz et al., SLAC-PUB-2198 (1978), paper presented at the 11th Annual Microprogramming Workshop, 1978.