ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S11LLC04

Driving Serial CAMAC Systems from VME Crates

W. Heinze
CERN 1211 Geneva 23

Abstract

Large control systems in the 80's were often based on Serial CAMAC loops driven by 16 bit minicomputers. These 16 bit computers, becoming obsolete in the 90's, are advantageously replaced by VME crates. To maintain the investment in Serial CAMAC hardware and software, an inexpensive Serial Highway Driver has been developed which operates in a VME crate as simple I/O module. With this system, both classical configurations, i.e. the Highway Driver on the I/O bus of the minicomputer and the Highway Driver in a so-called CAMAC mother crate, can be replaced with minimal costs and improved performance.

This paper presents a VME Serial CAMAC Driver and compares the performance of the VME driven Serial Highway to the ones driven by minicomputers. The comparison is based on the experience gained with the beginning of the replacement of Norsk Data minicomputers by VME crates in the CERN/PS control system as described in [1].

I. INTRODUCTION

The control system of the CERN Proton Synchrotron complex (PS) is at present based on Norsk Data 16 bit minicomputers which control 26 Serial CAMAC loops connecting approximately 240 CAMAC crates to drive the accelerator equipment. The rejuvenation of the PS control system starts with replacing the minicomputers by VME crates, called Device Stub Controllers (DSCs). The VME computing element is the MVME147 module, performing not only the tasks of the minicomputers but also executing the real-time tasks of the currently used Auxiliary Crate Controllers (ACCs).

The interface between the accelerator equipment and CAMAC has to be maintained for another some 10 years just because the investment in terms of money is so high. This was the reason to develop an inexpensive Serial CAMAC Highway driver as a VME module (abbreviated as SDVME). The first series of 20 SDVMEs have been assembled at CERN but the module is now also fabricated by a major CAMAC/ VME manufacturer (C.A.E.N., Viarregio, Italy).

II. DESIGN OF VME DRIVEN LOOPS

The actual Serial CAMAC loops will be rearranged into smaller loops. Every loop is then controlled by one DSC. The new loops consist of up to about 10 CAMAC crates, the number depends on the necessary computing power to control the associated equipment because the CAMAC crates do not house computing elements (the ACCs) anymore. They are not more than simple input/ output devices.

In the present control system, 2 types of CAMAC Serial drivers are used which are now replaced by the SDVME: one on the I/O bus of the minicomputer and one in a CAMAC crate which in turn is driven by a dedicated CAMAC Branch driver. Especially for the latter case which is used in 22 (out of the 26) loops, a considerable overhead of CAMAC equipment and cabling is avoided thus improving reliability, maintainability and equipment access times.

In general, the transmission speed on the Serial Highway of 2.5Mbit/s, bit serial, is maintained. Bit serial transfer permits to continue to use U-port adapters reducing the necessary number of twisted pairs in the Highway cable from 4 to 2 (one for the command, the other for the reply part) and allowing loop reconfiguration. Figure 1 shows the standard configuration. For special cases (instrumentation), the transmission speed is selected to 5Mbyte/s, i.e. byte serial. This is not a problem because in these cases, normally only one CAMAC crate is controlled by a DSC, and the distance is very short.

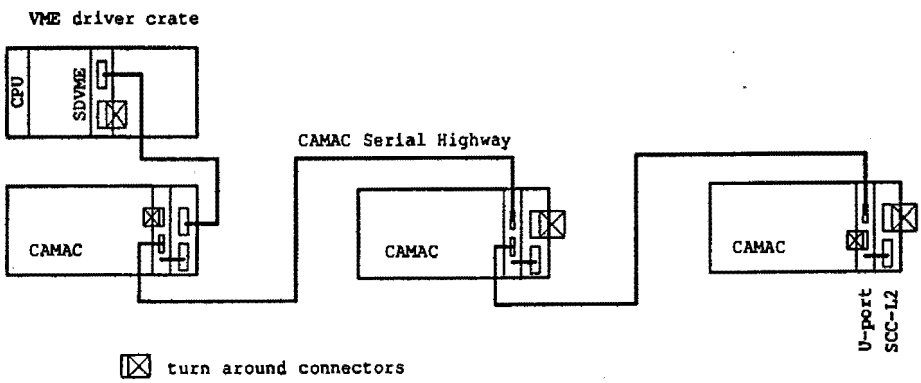

Figure 1. Standard loop configuration

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S11LLC04

III. CHARACTERISTICS OF THE SDVME

Communication between the SDVME as VME slave and the processor as VME master is governed by the Data Transfer protocol described in the VME specification. It is based on programmed I/O operation, short addressing and 16 bit data width (A16/D16). Driving the Serial Highway is done in conservative mode which simplifies error treatment and recovery.

The transmission speed is switchable to 0.5, 1, 2.5 and 5 MHz for byte and bit serial operation. Conform to the EUR 6100e specification, the appropriate number of space bytes is inserted into the Command messages to guarantee the completion of the Dataway cycle. The Serial Driver generates the necessary parity check patterns for the outgoing Command messages and checks them for the incoming Reply messages.

The Serial Driver resides on a single-width 6U VME board. A CAMAC access consists of writing to and reading from certain registers in the Serial Driver as shown in Figure 2. For a CAMAC write, one first loads the Command registers with SC, SN, SA and SF, and then loads the data into the Data Write registers W24-W17 and W16-W1 which generates the Command message. After a certain time which depends on the speed of the Serial Highway (e.g. about 55us in 2.5MHz bit serial systems), the VME processor can read the status (relative address $OC) which, after the Serial Driver having received the Reply message of the addressed Serial Crate Controller, contains the Reply. The Reply can be checked; if no error occurred, the next CAMAC function can be executed.

Reading of the Reply message can be done by polling a status bit NBUSY (which is reset at the start of the Command message and set with the arrival of the Reply message) or by connecting the read to an interrupt generated by the Reply message. If after a certain time (corresponding to 320 bytes in the message stream) the Reply message has not arrived, the NBUSY bit and the interrupt are set together with the corresponding error bits in the status register.

CAMAC read and control operations are generated by loading the command register. After having read the Reply message, a control operation is terminated. For a read operation, the data word has to be read from the data read registers.

The software driver must assure that the sequence of commands necessary for a CAMAC access cannot be interrupted by another CAMAC access to the same Serial Driver.

Arriving Demand messages are stored in a FIFO (first-in-first-out) memory. This assures that all possible LAMs can be served properly. The arrival of a Demand message is signalized by an interrupt.

So, one has 2 internal interrupt sources. They are treated by the VME compatible interrupter chip MC68153. Interrupt levels, interrupt vectors and masking bits are freely programmable. The Reply message generates an INT0, the Demand message an INT1.

RELATIVE ADDRESS	D15							D8	D7							D0	ACCESS
00	SA							SF							SN		R/W
02										SC							R/W
04										W24-W17							R/W
06	W16-W1																R/W
08										R24-R17							R
0A	R16-R1																R
0C	NBUSY	CERR	FWE	ERR CPL	ERR RED	ERR CP	ERR PB	NO SYNC						DERR	SQ	SX	ERR	R
0E	ERR DM		SC										SGL					R
10	CONTROL REG. REPLY INT0																	R/W
12	CONTROL REG. DEMAND INT1																	R/W
14	CONTROL REG. INT2 (NOT USED)																	R/W
16	CONTROL REG. INT3 (NOT USED)																	R/W
18	VECTOR REG. REPLY INT0																	R/W
1A	VECTOR REG. DEMAND INT1																	R/W
1C	VECTOR REG. INT2 (NOT USED)																	R/W
1E	VECTOR REG. INT3 (NOT USED)																	R/W
20																		R

Figure 2. Address allocation of Serial Driver

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S11LLC04

The address space needed by the Serial Driver is only 17 word addresses (from $XX00 to $XX20, $ meaning hexadecimal), so short addressing mode has been chosen. The base address is set by a switch in steps of $800 (permitting 32 different address settings). The address modifier codes accepted are $2D (SDVME is only accessible in supervisor mode) or $29 and $2D (accessible in user and supervisor mode), the configuration is selectable by a switch. Figure 2 shows the address allocation of the SDVME and the location of the command and error bits. SA, SF, SN, SC are the serial subaddress, function, station and crate numbers. W1-24 are the write, R1-24 the read registers. Address $XX0C contains the status register.

The meaning of the status bits is as follows. The 4 LS bits (ERR, SX, SQ, DERR) are inserted by the Serial Crate Controller into the Reply message. The other bits are generated by the SDVME:

ERR	Command message was found by the Serial Crate Controller to be non correct,
SX	X-response of CAMAC access,
SQ	Q-response of CAMAC access,
DERR	delayed error, see Chapter 12.2 in EUR 6100e,
NBUSY	Reply message has arrived which means a Command - Reply transaction has been completed,
CERR	error was detected, it is the OR signal of the error bits ERR PB, ERR CP, ERR HED, ERR CPL and NOSYNC,
FNE	FIFO for Demand messages not empty,
ERR CPL	Reply message not complete, i.e. number of bytes wrong,
ERR HED	header in Reply message wrong,
ERR CP	column parity error,
ERR PB	byte parity error,
NOSYNC	synchronization lost, i.e. clock is no longer detected at the input port.

The register at relative address $0E contains the Demand message and an error bit: SGL is the Serial graded LAM pattern, ERR DM indicates an error in the Demand message, i.e. byte parity or column parity error. This register can only be read.

The following 8 registers at $10...$1E serve to program the MC68153 interrupter chip. For the exact meaning of the bits refer to the MC68153 data sheet. Only the first two control and vector registers are used.

The last address (at read access) does a complete reset of the Serial Driver, like SYSRESET on the VME bus.

An exhaustive description of the hardware can be found in [2].

IV. Specification Summary

- 6U single VME slave board with P1 connector
- register generated single CAMAC access
- conservative mode and PIO only
- bit and byte serial mode, 5, 2.5, 1 and 0.5 MHz selectable
- appropriate insertion of space bytes
- stacking of Serial Demand messages in a FIFO
- Reply message generates maskable interrupt, polling possible
- Demand message generates maskable interrupt, polling possible
- VME characteristic A16/ D16, I(x), x programmable
- low power consumption (6.5W) due to utilization of CMOS

V. Driver Software

The driver for single CAMAC instructions and block transfer runs under LYNX OS and is written in C. Originally, it was written under OS-9, but then transported to the new PS standard operating system which is the POSIX compliant operating system LYNX OS. The execution speed per CAMAC instruction for a list of CAMAC functions is 100Kwords/s (for 24 or 16 bit words). The speed for block transfer (executing the same CAMAC read or write function) is 200Kwords/s. Both values hold for 5Mbyte/s transmission speed on the Serial Highway and a 68030 CPU running at 25MHz. For 2.5Mbit/s transmission speed, 45us have to be added per word transfer.

VI. Acknowledgments

The original design of the SDVME was done by L. Antonov and V. Dimitrov, CLANP, Sofia. The software driver under LYNX OS was written by A. Gagnaire, CERN.

VII. References

[1] A. Gagnaire, N. de Metz-Noblat, Ch. Serre, C.H. Sicard, "Replacing PS Controls front end minicomputers by VME based 32-bit processors", November 1991, these Proceedings, (ICALEPCS, Tsukuba, Japan, 1991).

[2] L. Antonov, V. Dimitrov, W. Heinze, "A CAMAC Serial Highway Driver in VME", PS/CO/Note 91- 22, 1991.