FERMILAB-Conf-91/290

Fastbus Readout Controller Card for High Speed Data Acquisition

S. Zimmerman (1,2), V. Areti (1), G. Foster (1) and K. Treptow (1)

(1) Fermi National Accelerator Laboratory
P.O. Box 500, Batavia, Illinois 60510

(2) Universidade Federal de Rio Grande de Sul, Electrical Engineering Department
Porto Alegre, RS 90210, Brazil

October 1991

* Presented at the *Nuclear Science Symposium*, Santa Fe, New Mexico, November 4-9, 1991.

FASTBUS READOUT CONTROLLER CARD FOR HIGH SPEED DATA ACQUISITION

S.Zimmermann[1,2],V.H.Areti[1], G.W.Foster[1], U.Joshi[1], K.Treptow[1]
[1]Fermi National Accelerator Laboratory*
Batavia, IL 60510, USA
[2]Universidade Federal do Rio Grande do Sul**
Electrical Engineering Department
Porto Alegre, RS 90210, Brazil

Abstract

This article describes a FASTBUS Readout Controller (FRC) for high speed data acquisition in FASTBUS based systems. The controller has two main interfaces: to FASTBUS and to a Readout Port. The FASTBUS interface performs FASTBUS master and slave operations at a maximum transfer rate exceeding 40 MBytes/s. The Readout Port can be adapted for a variety of protocols. Currently, it will be interfaced to a VME bus based processor with a VSB port. The on-board LR33000 embedded processor controls the readout, executing a list of operations downloaded into its memory. It scans the FASTBUS modules and stores the data in a triple port DRAM (TPDRAM), through one of the Serial Access Memory (SAM) ports of the TPDRAM. Later, it transfers this data to the readout port using the other SAM. The FRC also supports serial communication via RS232 and Ethernet interfaces. This device is intended for use in the data acquisition system at the Collider Detector at Fermilab.

I. INTRODUCTION

The FRC[1] is designed to scan a variety of FASTBUS modules and deliver this data through a readout port. Although it can be used in different FASTBUS based data acquisition systems, it is primary intended for the Collider Detector (CDF) data acquisition system at Fermilab. The current data acquisition system is being upgrade to handle the higher luminosity of the accelerator and modifications to the detector. Before the design of the FRC began, the FASTBUS Smart Crate Controller[2] was strongly considered. However, it does not have enough RAM to hold all events and its Readout Port would need major modifications. Below are some of the features of the FRC (see Figure 1):

a) A 32 bit RISC embedded processor (40 MHz LR33000) which implements the MIPS R3000 instruction set.
b) 4 Mbytes of triple ported video dynamic RAM (TPDRAM).
c) FASTBUS master and slave capabilities, with a peak block transfer rate on FASTBUS higher than 40 MBytes/sec.
d) Readout port capable of peak transfers of 20 MBytes/sec, using a 16 bit data path.
e) Interface to experiment
f) RS232 and Ethernet interface
g) Single width FASTBUS card.

For the application of this module at CDF, the readout port will be interfaced to a VMEbus based Scanner CPU with a VSB bus master interface. The VSB bus is used to implement a 16 bit wide address/data path, called Scanner Bus.

The main data path of the FRC passes through the TPDRAM. The TPDRAM has three asynchronous, independent, data access ports: the Serial Access Memories a and b (SAMa and SAMb) and the DRAM port. The FASTBUS controller of the FRC reads data from FASTBUS modules and stores it in SAMb. The SAMb controller, which is working in parallel, detects when the SAMb is full, and then writes its contents to the DRAM. Later, when the Scanner CPU requests, the SAMa controller reads this data to SAMa, and the Readout Port controller transfers it to the Scanner bus. The processor does not directly control the readout, which is done under the control of these four independent controllers. However, the LR33000 is used to configure the FASTBUS and Readout Port

*Work performed under the auspices of DOE.
** Work partially supported by CNPq, Brazil.

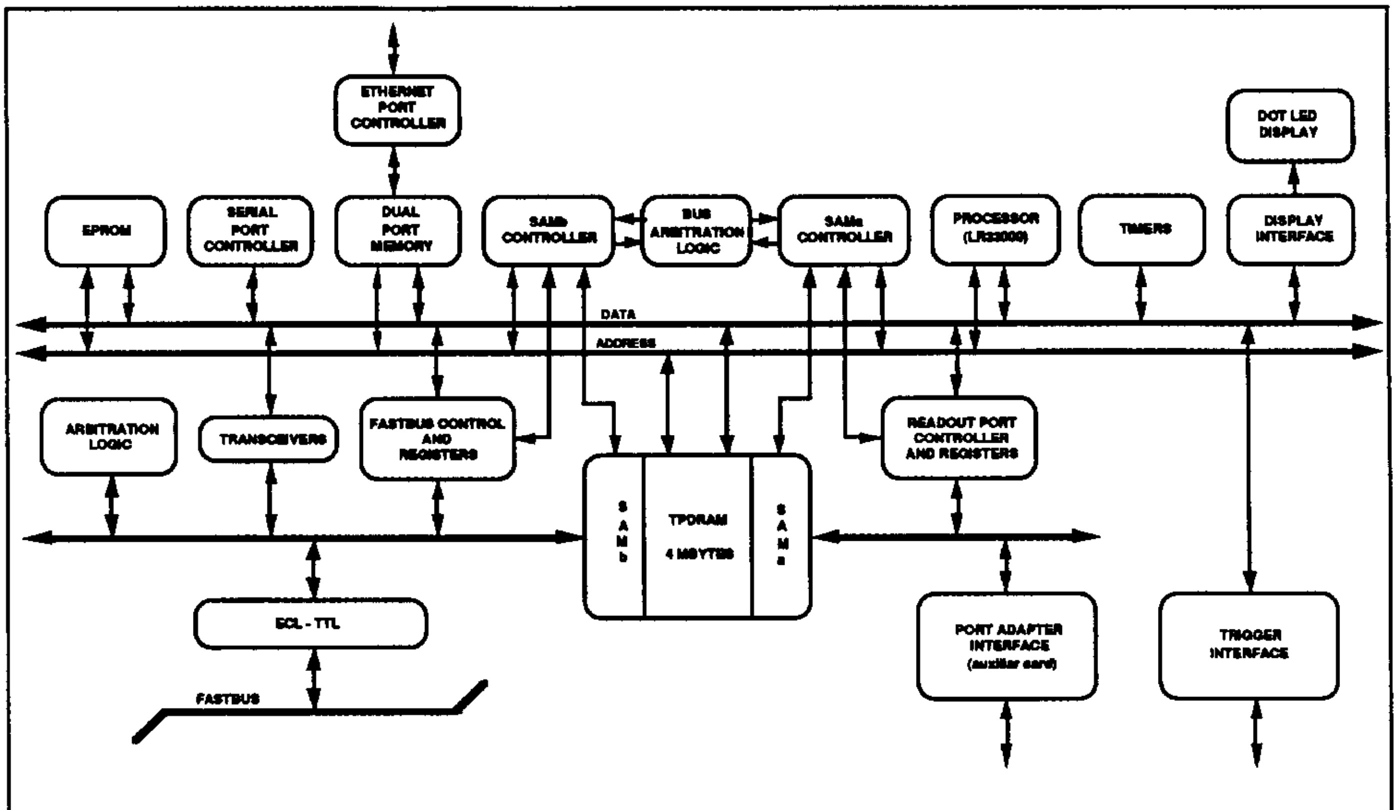


FIGURE 1. FRC BLOCK DIAGRAM

controllers, manage data buffer storage and provide header information.

II. MAIN MEMORY AND SAM CONTROLLERS

The main memory of the FRC consists of four banks of 256K x 32 bit wide memory. Each bank uses eight 256K x 4 TPDRAM integrated circuits (MT43C4257 from Micron). The main memory data is accessed via the DRAM port or via either of the two 512 x 32 bit SAM ports. The DRAM part of the memory needs refresh, while the SAMs are fully static memory.

All three ports (DRAM, SAMa and SAMb) may be operated asynchrounously and independently of one another, at the maximum allowable frequencies. Data may be transferred bidirectionally between the DRAM and either SAM, through two bidirectional internal paths. The only time the ports are synchronized is during transfers to or from the DRAM and SAM portions of the device, but this does not affect access from the other SAM port. The FRC uses just a restricted set of DRAM-SAM transfers: full and split read/write, pseudo write, full and split bit masked write and clear bit masked register.

Pseudo write transfers are used to initialize SAM write operations while full read/write to transfer the whole SAM. The split transfer is an important feature in this application, because it eliminates critical timing required to maintain a continuous serial data stream to the FASTBUS or to the Readout Port. In a split transfer, the SAM is divided into a upper and a lower half. While data is being serially read or written by a external device in one half of the SAM, the other half can be internally accessed using a split transfer. The QSF outputs of the SAMs inform which SAM half is being externally accessed. Therefore, the SAM controllers monitor these signals to know when it is necessary to transfer more data between either SAM and the DRAM.

A Bit Masked Register (BMR) can individually mask each bit transferred between either SAM and DRAM. Different methods exist to program the BMR, but the FRC uses one that clocks the mask in parallel with each word being written in SAMb. Later, during a write transfer, the BMR masks off the SAMb positions that were not stored with new data.

The SAM controllers control the transfers between either SAM and the DRAM. Figure 2 shows a block diagram of the SAM controllers and their associated logic. Each SAM controller has its own programmable pointer, consisting of two parts: a

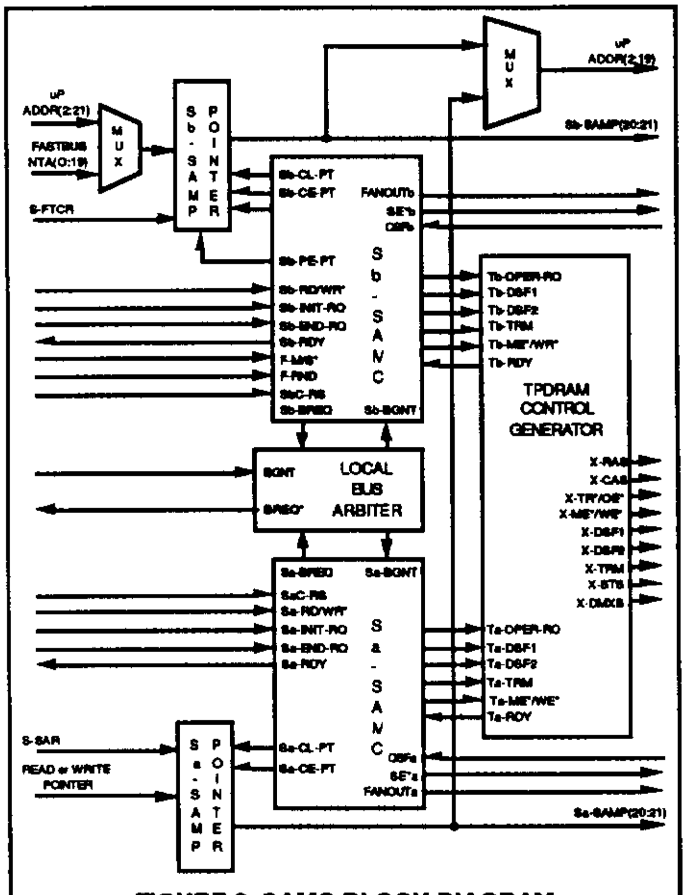


FIGURE 2. SAMC BLOCK DIAGRAM

latch for bits 0:7 and a counter for bits 8:19. It is not necessary to increment the first eight bits, but just clear them, because after the SAM is initialized, just a minimum of a SAM half (256 x 32 bit) is to be transferred. The TPDRAM Control Generator generates the necessary waveforms on the TPDRAM control lines, in order to perform the correct DRAM-SAM transfer. When one of the SAM controllers needs a TPDRAM transfer operation, it sets the control signals (that are forwarded by the TPDRAM Control Generator) and requests the operation (Tx-OPER-RQ = 1). Once it is done, the TPDRAM Control Generator answers with Tx-RDY = 1. To avoid conflict among the two SAM controllers and the processor, the Local Bus Arbiter controls the bus access, giving highest priority to SAMb and lowest priority to the processor.

The SAM controllers accept two different requests from FASTBUS and readout controllers: initialization and end request. At initialization (Sx-INIT-RQ = 1), the SAM controllers program the SAM pointers, initialize the desired SAM to read or write and enable the SAM port. At end request (Sx-END-RQ = 1), they save data if it was a write operation (Sx-RD/WR* = 0) and disable the SAM port. Sx-RDY informs the requestor that either operation is completed. The SAM controllers also detect when a SAM half is empty or full, and perform the required SAM-DRAM transfer, in order to keep the continuous serial data stream. The SAM controllers do not count the number of words transferred, which is done by the FASTBUS and Readout Port controllers. In case of a FASTBUS or Readout Port error, the processor can abort any transfer, resetting each SAM control logic thru signal SxC-RS. For a detailed example of the state diagram to initialize SAMb port, see Figure 3.

This memory system with the TPDRAM is very efficient for sequential transfer operations, both on FASTBUS and Readout Port side.

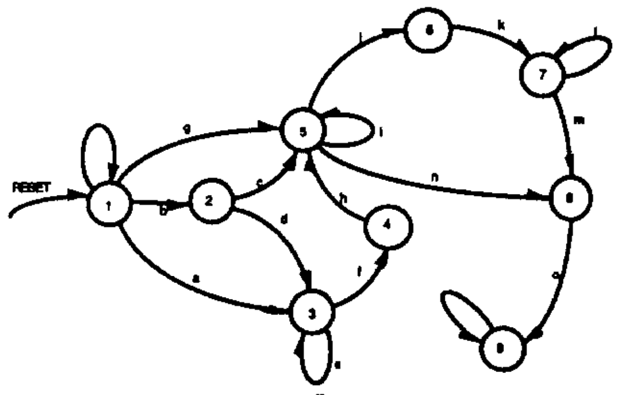
III. FASTBUS INTERFACE

The FRC is equipped with a FASTBUS[3] interface, capable of performing master and slave operations. Both master and slave interfaces execute single, block and pipeline transfers. When the FRC is to be a FASTBUS slave as well as a master, the FASTBUS Port Controller avoids this deadlock conflict. The FASTBUS interface supports geographic addressing and broadcast operations. In addition, the master interface can also perform logical addressing.

The FASTBUS master operations are done under the control of the processor, accessing five memory mapped registers inside the FASTBUS Port Controller:

a) Primary Address Register
b) Secondary Address Register
c) Transfer Control Register
d) Supplemental Transfer Control Register
e) Release Register

As an example, here is a detailed description of the Primary Address Register, addressable at BE02 MN00: Data to this write only register is the address of the intended FASTBUS slave. The "MN" part of the register address selects control bits for the Primary Address cycle. A FASTBUS Primary Address cycle begins when the processor writes to this register. Bits 15:13 of the address are placed on the FASTBUS MS lines and bit 11 on EG line. When bit 12 is set to 1, the FASTBUS mastership is retained at the end of the cycle. If the FRC is not already a FASTBUS master, an arbitration cycle is performed using the arbitration level set in the FASTBUS CSR#8 of the FRC. Bit 10 of the address indicates that the FASTBUS operations should be pipelined.



1) Wait for Sb-INIT-RQ = 1. Request local bus at Sb-INIT-RQ = 1.
2) Store Secondary Address in Sb-SAM pointer.
3) Peform a TPDRAM clear bit mask register, enabling the serial input of the mask.
4) Wait for completion of TPDRAM transfer request.
5) Perform a TPDRAM read or pseudo write transfer.
6) Clear Sb-SAMP latch and increment Sb-SAMP if necessary.
7) Perform the required split read or write.
8) Enable SE*b (SE*b =0), release local bus and set Sb-RDY.
9) Wait for Sb-END-RQ = 1 or transition on OSFb line.

a) Sb-INIT-RQ=1 and Sb-BGNT=1 and F-M/S*=1 and SB-RD/WR*=0
b) Sb-INIT-RQ=1 and SB-BGNT=1 and F-M/S*=0
c) Sb-RD/WR*=1
d) Sb-RD/WR*=0
e) Tb-RDY=0
f) Tb-RDY=1
g) Sb-INIT-RQ=1 and Sb-BGNT=1 and F-M/S*=1 and Sb-RD/WR*=1
h) Tb-RDY=0
i) Tb-RDY=0
j) Tb-RDY=1 and F-RND=0
k) none
l) Tb-RDY=0
m) Tb-RDY=1
n) Tb-RDY=1 and F-RND=1
o) Sb-BGNT=0 and Sb-INIT-RQ=0

FIGURE 3. PARTIAL SAMb CONTROLLER STATE DIAGRAM

Bit 10 of the Primary Address Register has an important meaning to increase the efficiency of FASTBUS cycles. The FRC, as a master, normally performs the following sequence of operations to carry out a data transaction: acquire FASTBUS mastership, execute a Primary Address, a Secondary Address and then the data cycles. When the Primary Address Register is write accessed with this bit set to one, the arbitration for FASTBUS begins. However, the onboard logic generates a DRDY* (Data Ready) to the processor, even before the completion of the Primary Address cycle. When the FRC gains FASTBUS mastership, the FASTBUS Port Controller completes the Primary Address. Meanwhile, the LR33000 proceeds and initiate a Secondary Address. The Secondary Address cycle is held up until the Primary Address cycle is done. Should there be a FASTBUS error during the Primary Address, the Secondary Address is not performed, and a local bus error is generated. The same scheme happens between the secondary address and data cycles.

Once the FRC properly addresses a FASTBUS slave, block and pipeline transfers will proceed under the control of the FASTBUS Port and SAMb controllers, while single data cycles are performed using the Secondary Address Register. Pipeline transfers allow programmable speeds of 100ns, 150ns and 200ns.

When the FRC is a FASTBUS slave, all the FASTBUS operations are done solely under the control of the FASTBUS Port and the SAMb controller. There are two means to synchronize the processor at completion of the data cycles: a maskable interrupt or a flag written in DRAM.

The FRC functions as a FASTBUS slave with three FASTBUS CSRs: CSR#0 (module ID and miscellaneous control), CSR#7 (Broadcast Class) and CSR#8 (Arbitration Level) and the TPDRAM may be accessed as data or CSR space. It is the user's responsibility to protect any specially dedicated memory locations.

The processor handles FASTBUS errors when the FRC is a FASTBUS master. When the FRC receives a non-zero SS response, the FASTBUS port controller generates a bus error signal (BERR*) to LR33000. The LR33000 can read the SS response thru the Transfer Control Register. When the FRC is a slave, invalid operations generate non-zero SS responses.

IV. READOUT PORT

The Readout Port is interfaced to the FASTBUS auxiliary connector, that has associated logic and level adapters. In principle, the Readout Port is a general port, because the FRC uses configurable logic components. However, the complexity of the protocol is limited by the logic available.

For the CDF data acquisition system, a specific bus protocol is implemented. Two types of readout devices will be interfaced to this bus: FASTBUS Readout Controllers and Rabbit Readout Controllers (RRC):

a) The Scanner Bus is a single master/multi slave bus, with the VSB[4] master interface acting as a Scanner Bus master and the FRC and RRC acting as slaves.

b) It accommodates a maximum of 16 slaves, each with its unique selected address. It is a multiplexed 16 bit address and data path.

c) Data transactions on the Scanner bus always consists of an integral number of 32 bit words. Therefore, a single data access requires two data cycles on the bus.

d) The Scanner bus implements broadcast and broadcall operations. A broadcall is a single word read transfer in which the Scanner bus master reads data from all the readout controllers attached to it. The purpose of this operation is to give a means for the Scanner to poll all readout controllers, in an efficient way, to find whether there is data to be read. The readout cards answer by asserting a specific data bit.

e) The length of the Scanner bus makes individual data acknowledge signals impractical. Thus, it is the responsibility of the data receiver to accept data at the rate sent by the transmitter. When the Scanner bus master writes data, the FRCs or RRCs are the receivers and when it reads data, the FRCs or RRCs are the senders.

f) The maximum cable length is 250 ft, and the electrical protocol is RS-485.

The Readout Port controller of the FRC is built to implement the protocol. Again, the transfers are done with minimum intervention of the processor and synchronization is accomplished by interrupt or by flags.

V. SYSTEM PROCESSOR, HARDWARE AND SOFTWARE DETAILS

The LSI Logic LR33000 RISC embedded controller contains 8 KBytes of instruction cache, 1 KByte of Data Cache, DRAM controller and counter/timers on chip. It provides initialization and supervision of FASTBUS and Readout Ports.

All state machines, counters, registers and miscellaneous logic are implemented in a XC3090 and a XC3020, speed grade -125, Programmable Gate Array from XILINX. These programmable devices were chosen due to the high number of flip-flops, which enable the design of multiple registers, and the number of I/O pins available, both important features in this application. The XC3090 holds the logic of the SAM, FASTBUS and Readout Port controllers and the XC3020, the Ethernet and dot matrix display controller and timers. The state machines are designed using mainly the One-Hot approach[5], very suitable for the XILINX part architecture. The hardware design is done so that the LR33000 can work with the EPROM, DRAM and RS232 interface, even without the XILINX part.

The Ethernet port is implemented using the AMD Am79C900 ILACC controller. It communicates with the LR33000 by means of a direct path with the processor data bus and a dual port memory. The direct path has just 16 bits and is used only for initialization, status information and diagnostic purpose. All data messages are exchanged with the processor via the dual port memory, to avoid the processor giving up its bus to the ILACC.

The FRC is equipped with 512K Bytes of EPROM, which will store comprehensive diagnostic routines, Ethernet drivers and a monitor. It also has a real time clock and a minimum of 50 bytes of RAM (Dallas DS1286 or DS1386), both backed up by a built-in battery. The board design allows the porting of real time operational system, such as VxWorks from Wind River Systems.

VI. REFERENCES

[1] H. Areti, G.W. Foster, U. Joshi, "CDF Data Acquisition Upgrade with a Commercial Switch" Fermilab note CDF/online/DOC1463, 1990.

[2] G. Cancelo, M. Bowden, R. Kwarciany, J. Urish, "An Intelligent Readout Controller for FASTBUS, The Fermilab FSCC", Fermilab note DAE/0012, 1989.

[3] "IEEE Standard FASTBUS Modular High-Speed Data Acquisition and Control System", New York, IEEE Press, 1985.

[4] "VSB Specification Manual, Revision C", Motorola Inc., 1986.

[5] S.K. Knapp, "Accelerate FPGA Macros with One-Hot Approach", Electronic Design, Sept. 1990.