A GENERAL PURPOSE FASTBUS MASTER (GPM) AND MEMORY MODULE (DSM) FOR ONLINE APPLICATIONS

H. Müller

CERN, European Organization for Nuclear Research, CH-1211 Geneva 23, Switzerland

Abstract

The GPM [1] is a high performance Fastbus master/slave, driven by 68 000 (or optionally 68 020/68 881 processors). Developed for general purpose applications it will perform control, trigger and diagnostic functions in three different experiments [2] at CERN (DELPHI, L3, VIRTUS). Commercially available as single-board Fastbus master/slave, the GPM is a cheap processor component which can be connected to the DSM Fastbus dual-slave memory [3] to perform as a low-level trigger processor, event formatter or data-spy. The GPM supports interprocessor interrupts via a Fastbus CSR register, interrupts on events in the Fastbus (like SR), and external interrupts. The GPM (fig. 1) is equipped with 1/2 Mbyte of RAM, 1/4 Mbyte of ROM and a 32 Kbyte Fastbus I/O buffer. Any Fastbus operation can be generated as mixture of autonomous block transfers and assembly-language instructions which are directly executed in Fastbus. Powerful diagnostics can be performed by the possibility to address its own slave-port, via the Fastbus and the applicability of the debugging monitor commands to individual Fastbus/68 000 cycles. A terminal and host connection is available. The DMA supported parallel port can be used for interface applications. For external interrupts and coaxial pulse I/O, a NIM pulse interface is available. The 2 Mbyte Dual Slave Memory (DSM) can be used either as a stand-alone Fastbus memory, or as a memory extension module of a GPM, with independent I/O on crate and cable segments (fig. 2). It supports linear or circular buffer concepts and can be used as a data-spy on cable segments.

Fig. 1

1. THE GPM GENERAL PURPOSE MASTER

Sixteen functional units are accommodated on one Fastbus multiwire board. The slave and the master ports can communicate via their home segment. High speed I/O and CPU activities are decoupled by using a separate Fastbus I/O buffer (32 Kbytes).

GPM CONNECTIONS

Fig. 2

The slave port provides access to six CSR registers and to the data space buffer, via geographical and logical addressing. Class-N and T-pin broadcasts are implemented. Special functions (CSR bits) provide access to the microprocessor's address space or disable the processor's access to the slave's resources (NTA or buffer). Separate NTA pointers are used in CSR and data space.

By using a bipolar DMA controller which in its default mode functions as Fastbus NTA, software controlled buffer concepts can be used (word counter mode, address comparison, wraparound, up/down counting etc.).

The master can execute 32 bit instructions of the microprocessor as Fastbus cycles, or it can execute autonomous block transfers from or to the data space buffer with parallel processor activities in other parts of the address space. Block transfers and microprocessor/Fastbus instructions can be mixed to generate Fastbus operations of any complexity. In particular, operations can be rationalized by using microprocessor operations like ADD, TEST, CLR, etc. as indivisible operations in Fastbus.

Interrupt controllers are used for Fastbus events like SR and for protections, like generating data cycles without connection.

Peripheral interrupt and timer chips are used for a programmable, parallel I/O connection on the GPM's front panel. An optional DMA controller allows to use this port for data transfers and interface applications (max. 4 Mbytes/s).

The Fastbus CSR(1) register is used to generate interrupts to the microprocessor for any 68 000 vector within normal Fastbus write cycles.

The two serial I/O ports can be used for a direct connection to user terminals and host computers. A multi-drop protocol on these ports allows to connect GPM's to an independent serial bus with duplex-asynchronous protocols (max. 1 MB/s).

The two-port data buffer (32 Kbytes) is used by both the slave, or the master for high-speed input or output, independent of the processor's activity. The microprocessor has, via an internal second port, access to the buffer which can be competitive or exclusive with Fastbus I/O. The sharing of the buffer between the slave, master and microprocessor provides many I/O possibilities without the need for internal transfers.

The resident software in EPROM's (max. 320 Kbytes) allows to chose between two different debugging monitors which contain initialization and run time libraries. Optionally, the PILS language [5] can be added to provide a transportable interpreter environment, in particular for tests and diagnostics. For file storage, personal computers like the Macintosh or IBM-PC are used as intelligent terminals, directly connected to the GPM.

The cross languages available at CERN (F77, C, PASCAL, MODULA-2 etc.) provide a high degree of flexibility in the choice of software for the GPM in different environments [4]. The libraries (Fastbus, multitasking) are available from general support groups at CERN.

The GPM is available commercially as CERN propriety multiwire board, dissipating about 50 W. Optionally, a 68 20/68881 processor plug-in board can be used to boost the GPM's processing power (gain 2-5), depending on software). Future GPM's may integrate 32-bits processors in its default configuration. The modularity of the GPM and the multiwire routing facility at CERN allow for upgrades in the future without major difficulties.

2. THE DSM DUAL SLAVE MEMORY

The DSM [2] is a 2 Mbyte buffer memory, with Fastbus slave ports on both the crate and the cable segment. A third port on the auxiliary connector provides Fastbus-independent access via 68 000, or 68 020 chip protocols. The GPM can thus map one or several DSM's into its address space, providing the possibility to access several independently filled DSM's as one logical buffer.

In its default configuration, the DSM provides memory blocks in the Fastbus CSR and data space, which are addressable in random or block transfer mode from one or both ports. The two-port access is resolved on a cycle-by-cycle basis if no synchronization via software bits is used.

Different modes can be initialized to use the DSM as FIFO-like wraparound buffer or as partitioned memory with software pointers in the DSM's directory-memory in the CSR space (fig. 3).

Fig. 3

The structure of input data blocks is retained at the output port on two levels: block transfer boundaries and event boundaries are marked in invisible bits of the memory to generate corresponding SS responses during readout.

The DSM may be used as spy module which copies Fastbus data transfers from its connected cable segment into its buffer, which in this application is organized like a FIFO between the cable input and the crate segment output (fig. 4). The spying is controlled by two bits to synchronize data frames with the demand of a spying master. Only data information is copied.

The DSM, used as data-spy

Fig. 4

The DSM uses separate internal address pointers for both ports, and for CSR and data space (total 4). The auxiliary port can access all DSM resources via the microprocessor's address mechanism.

The CSR(1) register can be used as backup and as address comparator for input and output in FIFO applications. Overwrite warnings can be generated via SS responses and transfers can be repeated by reloading the start address in the FIFO from the DSM's backup register.

The DSM uses two alternating asynchronously-refreshed RAM memories in parallel to be able to use the DSM as synchronous memory in pipeline mode, at higher speed. The design goal is 20 Mbyte/s with commercially available RAM memories.

Photo 1: GPM Module

References

- [1] H. Müller and R. Vachon, The GPM General Purpose Fastbus Master/Slaved, USER MANUAL, CERN/EP/Electronics Note 85-02.
- [2] H. Müller, Online Fastbus Processor for LEP, Proc. International Conf. on Computing in High-Energy Physics, Amsterdam 1985, CERN/EP 85-6.
- [3] H. Müller, C.M. Story and S. Falciano, The DSM, A Fastbus Dual Slave Memory for Data Acquisition, CERN/EP/Electronics Note 85-03.
- [4] H. Müller, General Purpose Masters GPMs in the DELPHI On-line System, Proc. Fastbus Software Workshop, CERN, Geneva, 23-24 September 1985.
- [5] R.D. Russel, D.O. Williams and F.L. Tremblet, PILS, A Portable Interactive Language System, Proc. 4th Biennial Conference on Real Time Computing, Chicago (1985).