THE EVOLUTION OF THE FASTBUS GENERAL PURPOSE MASTER (GPM)

H. Müller
CERN, European Organization for Nuclear Research,
EP Division, Geneva, Switzerland

Abstract

The numerous programmable Fastbus masters which are integrated into LEP data acquisition systems are mainly based on the Motorola 68 K microprocessor family. In a Fastbus system, the coupling between the microprocessor bus and the IEEE Fastbus can be solved in different ways: memory mapped, co-processor driven, or list driven.

The GPM technique couples the 68 K data bus to Fastbus much like memory mapped I/O. A small set of addresses is provided to select all varieties of the Fastbus protocol. Since any operation in Fastbus can be written like a program, this technique was called "general purpose".

The commercial, single width GPM module [1] can directly execute 68 K instructions in Fastbus. Its master and slave ports handle 20 Mbytes/s data rates using a 128 kbyte data buffer which is mapped into the 68 K address space.

New projects are building up on the GPM technique and its well established software environment. The CERN CHI interface [2] and the DELPHI/LIP SGPM common project use a new standardized GPM programming model. Porting of the Los Alamos implementation of the IEEE Fastbus standard routines [3] to this new model is under way.

GPM architecture

The classical GPM is a 68 K system with a Fastbus master and slave ports on the crate segment, sharing a 128 K data memory (fig. 1). The slave and master as well as the 68 K system are 3 independently working units which allow a high degree of parallel activity. The 68 K system can address its own slave port via the master port, or the master port can execute high speed block transfers whilst the 68 K CPU is executing other activities.

Fig. 1

Although the possibilities provided by this architecture for passive and active I/O are manifold, additional mechanisms to load the 68 K system memory and to generate interrupts from Fastbus were implemented to enhance the possibilities.

Complete control of a GPM via Fastbus is available via the slave port, if required.

GPM applications

In the DELPHI experiment, GPM masters are used as event supervisors for local and global readout tasks which may be alternating with 3rd level trigger tasks. The high computing power required for the trigger tasks will be better matched by the new SGPM project, based on a 68030 processor with the same architecture as the GPM.

The L3 experiment uses GPMs to build events from distributed buffers into a downstream buffer. In both experiments, a Block Mover module (BM2) is under control of a GPM to perform very high speed inter-segment data moving. Smaller experiments use the GPM as interface to a host computer, to readout Camac branches or to build a multiprocessor farm for off-line analysis.

In test systems, the GPM is used with the CERN Monica debugger which has commands for compact Fastbus test or initialisation sequences. The debugger may be used together with the CERN cross languages via a host computer, or, with the GPM resident FILS interpreter language.

GPM technique

A direct correspondence between the 68 K data bus and Fastbus address or data is established, whenever the 68 K accesses one of the GPM's key addresses. The small set of addresses required to select the Fastbus control line combinations for the different protocols has been extended to 16 bit in the new programming model (fig. 2). The additional addresses allow selection of the arbitration vectors and protocols, together with the primary address protocol. Two sets of addresses for primary address and data cycles are available.

Fig. 2

A similarly mapped set of special addresses is required for GPM internal registers and functions, like explicit arbitration, single line access, or control of the DMA chip.

Software support

The Eprom-resident CERN Monica debugging monitor and its run-time libraries make the well supported CERN microprocessor software environment [4] available to GPM users. Stand-alone tests can be performed via the resident GPMTEST Fastbus tests which can be called from the Monica user menu. Using its resident line-by-line assembler and/or the optionally resident PILS language, test software can be written without the need for a host computer.

A connection to a VAX or a PC host computer is required to load the CERN cross software or to use PILS libraries and to save programs. The Los Alamos implementation of the Fastbus standard routines for the GPM is available as a library.

The CERN microprocessor support makes multi-tasking available using the RMS 68 K kernel software. Porting of both the commercial OS-9 and pSOS-68 K kernel systems is under way by other groups.

GPM embedded processors

Fast and low overhead message channels are desirable for the integration of a general purpose processor into specific Fastbus environments. The "Fastbus channel" is available as native, high bandwidth medium, interconnecting embedded processors and host computer. For the reception of interrupts, the slave port implements an interrupt sensitive CSR register, passing 68 K interrupt vector numbers to the CPU. A special CSR function allows loading of data into the 68 K address space.

These mechanisms can be exploited for inter-processor message exchange and for a complete control of a GPM from a host computer, connected via Fastbus.

The applicability of block transfers and selective broadcasts for down loading purposes greatly reduces loading time overhead.

GPM - family hardware

Originally intended to provide a memory mapped access to the CERN Fastbus Dual Slave Memory (DSM) [5], the GPM's auxiliary connection is now equally used to connect peripherals. Currently available are memory extension cards and an SCSI bus adapter. The GPM extended applications board (GEA) board which is under design in the DELPHI experiment contains these options and a cheapernet adapter as well as a second master port for the cable segment on a single extension board. The DSM-GPM connection allows combining of two CERN stand alone modules into one multiport, intelligent data buffer.

CHI project

The processor and the master port of the CERN CHI host interface project have been designed, building up on the experience gained with the GPM. The GPM programming model described above has been adapted, allowing for extensions on future implementations.

SGPM project

The need for an increase of computing power by a factor 5 to 10 in a very similar master/slave architecture as the GPM has been identified by the DELPHI experiment. This project builds up on the standardized GPM model developed for the CHI, replacing its processor by a 68030 microprocessor. The availability of complete design components from the CHI

project on an advanced CAD system should allow us to design the SGPM in rather short time.

Performance

Fastbus operations which are executed as individual CPU instructions perform at the speed specified for the CPU operation. Autonomous block transfers, which can be interleaved with CPU driven Fastbus operations, perform at transfer rates of up to 20 Mbytes/s. Performance tuning is possible, since special compound operations may be compiled as user defined subroutines. In this way, one can take advantage of the 68 K instruction set and use conditional branching within a Fastbus operation.

Conclusion

The GPM technique is well adapted to the large range of requirements imposed on a general purpose processor. Whilst the integration into specific, small and large scale environments is successfully progressing, the GPM technique was further developed for the CERN CHI project and the DELPHI SGPM. Both use the same programming model, such that standard Fastbus routines will be available on these processors as well as on future GPM-type processors.

References

- [1] H. Müller, On-line Fastbus Processor for LEP, Proc. International Conference on Computing in High-Energy Physics, Amsterdam (1985).
- [2] The CERN Host Interface, presentation by R. McLaren, this conference.
- [3] T. Kozlowski and W.M. Foreman, An implementation of the new IEEE standard routines for Fastbus, Fifth conference on real-time computer applications in nuclear particle and plasma physics, San Francisco (1987).
- [4] J. Blake, H.V. Eicken and D. Foster, Developing programs for the Motorola 68000 microprocessor at CERN, Computer physics communications 38 (1985) 315-318.
- [5] H. Müller, A general purpose Fastbus master (GPM) and memory module (DSM) for on-line applications, IEEE, Nucl. Sci. Symp. (1985).