A Configurable RT OS Powered Fieldbus Controller for Distributed Accelerator Controls.

A.Matioushine, A.Sytin, P.Vetrov, S.Zelepoukine (IHEP),
M.Clausen, W.Ebenritter, B.Schoeneburg (DESY).

IHEP – Institute for High Energy Physics, 142284 Protvino, Russia.
DESY – Deutsches Elektronen Synchrotron, Notkestr.85, 22603 Hamburg, Germany.

Abstract

A General Purpose Fieldbus Controller (GPFC) is primarily designed as a compact and economical solution for introducing advanced programming and operating environment at the fieldbus level of accelerator control systems. Its scalable modular architecture enables a number of applications, ranging from a highly intelligent fieldbus node upto a powerful networked host driving several fiedlbuses. GPFC is implemented as a standalone $IndustryPack^R$ (IP) carrier, so that a wide range of commercially available plag-on IP modules (CAN, Profibus, Ethernet, digital and analog I/O, etc.) can be used to configure it to application needs. GPFC may have a single board configuration hosting 2 IP modules, or an extended (dual board) one, which hosts 4 IPs. GPFC is powered with the Motorola $ColdFire^{TM}$ CPU and runs $VxWorks^R$, the real time operating system from Windriver Systems. GPFC, configured with low cost and highly reliable real-time CAN communications, seems to be an efficient solution for interfacing SEDAC crates, which are widely used at DESY for equipment controls. The GPFC project is jointly run by DESY and IHEP. GPFC implementation for SEDAC control is described and, additionally, using GPFC in the EPICS based distributed control system is discussed.

1 Introduction

The idea of GPFC (= General Purpose Fieldbus Controller) is rather simple. When considering the use of modern fieldbuses, like CAN [1] or Profibus [2], in accelerator controls, it is VME that usually plays a role of a fieldbus controller. VME computer is a key architectural component for most of the control systems. It provides unified equipment access and runs time critical monitoring/control tasks, being a common solution for a "front-end computer". However, in many cases, especially those concerning the use of fieldbuses, one might want to look for a more specific, compact and less expensive solution, namely a standalone fieldbus controller. And this is what the GPFC basically is.

Both parties, DESY and IHEP, are interested in such controller for their control applications. The preliminary commercial market study led us to conclusion, there are no device available which have all desirable features. So then, the GPFC project was initiated, with the following design goals:

Device should provide at least two communication ports, one for a fieldbus and second for connecting to the host level. CAN and Profibus are must, Ethernet connection is highly desirable. These ports should be implemented with changeable modules to provide flexibility in connection styles.

Device should be reasonably powerful in terms of CPU and memory resources to support non-trivial applications running under VxWorks [3] or equivalent real-time operating system. It is highly desirable to provide software storage onboard to support autonomous operation.

Device should be provided with an advanced software development environment (i.e., VxWorks + Tornado [4]) to enable quick and efficient application development cycle.

Device should be small sized to allow it to be embedded into equipment or to be installed in space limited areas. In particular, it has to fit into the SEDAC module form factor (see more details in a separate section below).

Device should provide low power consumption to avoid the need in forced air cooling. This is to allow its installation in unmanned (radiation) areas with a strongly restricted human access (for troubleshooting and maintenance).

Device should be scalable in terms of I/O and memory resources. Its minimum configuration has to provide functionality of a simple single-branch fieldbus controller, while more advanced configurations should be able to support several fieldbus connections and Ethernet, and run big and complex "front end" applications (see more under the EPICS title below).

Device should be cheap to be in good accordance with its intended application for low cost fieldbuses, and designed in so way, that to enable its production in-home, mainly using DESY and IHEP electronic workshops.

The project is jointly run by DESY and IHEP. Other interested parties are welcome. Following sections present in more details the architecture and functionality of the GPFC. Also, the first and some foreseen applications are discussed.

2 Device architecture

The GPFC's architecture has been designed to be modular and scalable in order to provide an incremental range of configurations. First, there is a main board hosting CPU and memory and carrying two sets if IP plugs for I/O. These modules are assumed to be (but not limited to) fieldbus interface ports. It is the smallest but functionally complete configuration, able to store and run application under VxWorks operating system. Second, an extension board, carrying two more I/O modules and providing additional memory, can be added. Both boards are stacked via a separate connector, which translates the CPU bus and some auxiliary signals. The two-board configuration offers maximum I/O and memory resources. There may be also a third board, inserted in-between the two above mentioned ones (multi-board stackability is provided by an appropriate connector choice). This extra board is intended for possible future extensions of the system. Possible board configurations are shown on Fig.1.

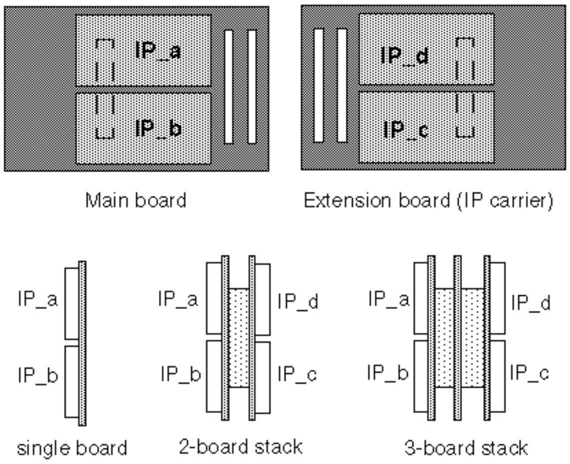

Figure. 1. GPFC board configurations.

At the early design stage a number of standard architectures were considered, and many key GPFC solutions came from there. In particular, stackable board configuration and initial board form factor were taken from PC104 [5]. However, we failed to fit the main board functionality into PC104's limited space and changed the form factor to 3U (100x160 mm).

2.1 CPU

The CPU choice was also done early because of its strong relation with the overall system architecture. A number of candidates were considered under the following selection criteria:

- low chip cost, low power consumption/dissipation.
- simple programming model, VxWorks support or the possibility to provide it in-home.
- simple surrounding system architecture, simple interfacing peripherals to the CPU bus.
- reasonable performance (to support VxWorks, CAN, Ethernet, ...).
- wide address/data bus (minimum 64 MB x 32 bit directly addressable).
- additional on-chip functionality (CPU bus controller, DRAM controller, UART, timer, ...).

By these criteria Motorola's ColdFire MCF5206 [7] has been found a best fit. MCF5206 is intended for low power and low cost embedded applications. Interfacing peripherals is simple and supported by on-chip logic. DRAM controller, UARTs and timers are embedded on-chip. Directly adressable space is 256 MB x 32 bit (8/16/32 bit transfers are programmable), performance is 17 MIPS at 33 MHz (clock frequency can be choosen in the [0-max] range). Its programming model is very similar to the one of its predecessors, the well known M68K family. And there exists VxWorks port for it, supported with Tornado development tools. Learning ColdFire is expected to be straightforward and easy for M68K application developers.

2.2 Memory

The main board is equipped with DRAM, EPROM/FLASH and SEEPROM memories (see Fig.2). This enable the board to be completely functional, using EPROM/FLASH for VxWorks+application code storage and SEEPROM for application critical data storage.

Because of space (and power) limitations only a single 2MB DRAM chip and a single EPROM/FLASH socket (max 1MB/512KB) are placed on the main board. Extension board delivers additional DRAM and FLASH memories of larger sizes - for advanced memory demanding applications.

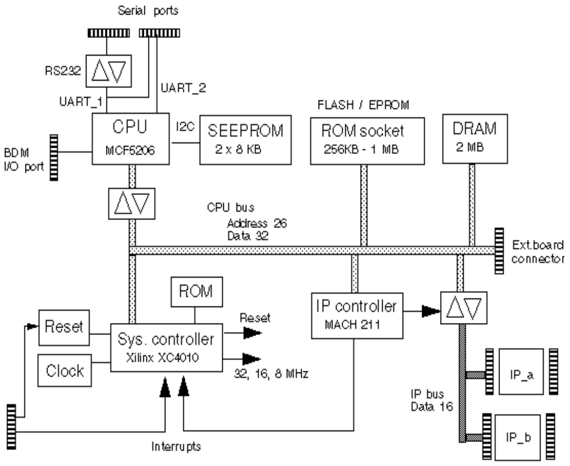

Figure. 2. The main board block diagram.

2.3 IP interfacing

I/O interface idea has been taken from the IndustryPack (IP) standard [6]. This was a very important decision, since it defined many hardware and software solutions that immediately followed. And this enabled the use of a wide range of commercially available IP modules (instead of a few ones to be developed in-house).

Of course, the IP market offers many IP carrier boards. So, why to develop one more? It has been found however, that commercial IP carriers are either too "heavy weight", expensive, big sized and too power consuming, or too "light weight", unable to run advanced kernels (VxWorks) and complex applications. As for GPFC, it occupies the niche having been found empty, combining some extracted features from both categories.

Having failed to find a commercially available chipset for IP controller we have developed our own, using CPLD/FPGA techniques. The first GPFC prototype is made with two AMD MACH211s as an IP bus cycle translators (one per IP carrier board) and Xilinx XC4010E-PC84 as an interrupt controller (see Fig.2). The next version (under CAD design) use a single XC4010E-PQ160 for both functions.

Reduced IP specification is only supported (see [6] for details):

- 8 MHz IP modules only.
- 16-bit data path only.
- IP DMA mode is not supported.

This simplifies the design and seems to be adequate to intended (low cost, low power) applications.

IP modules are provided with power (+5V, -+12V) from the carrier, either main or extension, board. The connection of IP modules to the external world is done via 50-pin headers, placed on the carrier. So, many commercially available IP module packages with flat cabled accessories can be used as is, without any modification.

2.4 *Interrupt/system controller*

Xilinx chip actually supports more than just IP interrupt processing. It processes all interrupts, external to ColdFire:

- 8 IP interrupts (two per IP module).
- ABORT interrupt (ABORT button initiated).
- 4 external interrupts (from a separate connector on the main board).

Besides, Xilinx is responsible for system clocks (32/16/8 MHz), powerup and reset sequences, and some other system functions.

2.5 *Auxiliary features*

There are also two serial ports, provided by ColdFire's embedded UARTs. One of these has an RS232 interface, thus allowing the GPFC to be directly connected to a terminal or, via a serial link, to some host. RS232 and both UART ports are wired to auxiliary connectors placed on the main board.

A parallel 8-bit port, again provided by the ColdFire, is also wired to an auxiliary connector. In a SEDAC controller configuration this port is used for driving LED indicators displaying system activities and status.

Also, the BDM (background debug monitor) interface is presented via auxiliary connector, enabling the use of low level ColdFire debugging facilities (the popular SingleStep debugger from SDS, or other equivalent tools, can be used for testing and debugging of critical low level software or tricky hardware features).

Last, but not least, the interboard stackable connector translates all CPU bus and some intra-system signals. Besides its main role to attach the GPFC extension board, it can be used for additional system extensions. For example, those where one need to provide I/O interface different from IP.

3 SEDAC crate controller

SEDAC system [8] has been developed at DESY in the '70s and till now hundreds of SEDAC crates reliably work in many parts of the DESY accelerator control systems. These crates are connected to VME or PC front-ends via dedicated SEDAC serial bus. Widely using modern fieldbuses in newly developed and rejuvenating control systems, DESY aims to unify all equipment connections and so wants to replace home-made SEDAC serial bus with CAN.

Another demand is for introducing computing power into the non-intelligent SEDAC crate. VME computers or PCs currently driving SEDAC can be unloaded from low level communication and I/O transactions, if a small computer is installed in the crate. It can take much of this load, while providing efficient high level access protocols to VME and PCs. Moreover, this computer can run some applications (monitoring tasks) locally.

So, a SEDAC crate controller with CAN connection, able to run VxWorks (in order to easily port application software currently running under VxWorks on VME) has to be designed. From the other side, the need for a small standalone fieldbus controller is realized as well (see Introduction). Next step is evident - to put SEDAC specific functions into the feature list for that controller. That's the story of SEDAC crate controller in the GPFC context.

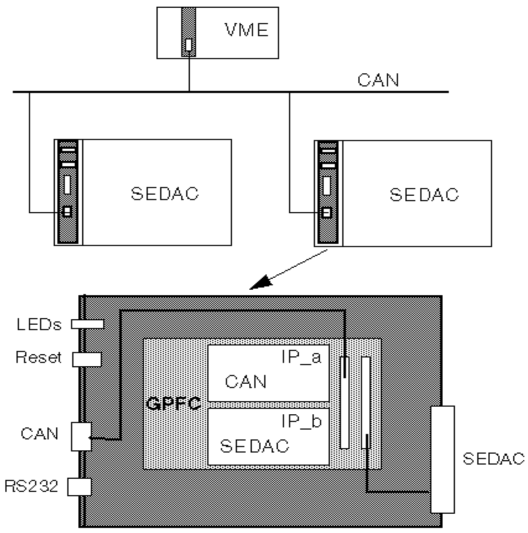

Figure. 3. SEDAC crate controller.

In the GPFC framework, interfacing a SEDAC crate was done with a specially developed IP module. CAN interfacing is via commercial module(s). A single board GPFC configuration (with 2 IP modules plugged on the main board) perfectly fits into the SEDAC module form factor. The GPFC board is fixed on a dummy carrier board having the normal SEDAC form factor. That dummy carrier provides connectors and wiring between the GPFC board and the SEDAC crate's bus connector (see Fig.3).

4 GPFC in the EPICS framework.

The initial idea of a General Purpose Fieldbus Controller was targeted to develop an intelligent frontend micro controller rather than a full function controller board. Having specified that VxWorks should be supported on this controller it should be possible to directly run the existing VxWorks dependent applications. These are for instance the drivers for CAN and other serial buses and the higher level driver support application called the Common Driver Interface (CDI). This way a whole set of developed drivers can be reused and only additional IP modules like the one for the SEDAC crate controller require new support routines.

Once we have reached the point of running the underlaying driver support routines which also directly interface to the EPICS fron-end software it was natural to ask whether EPICS [8] itself could be supported. Besides the necessary memory for EPICS and the IP/Ethernet support all problems should be already solved once the GPFC runs as a SEDAC/CAN crate controller.

This was the driving force to implement as much memory as possible (limited by a reasonable power consumption) and to start the work on the Ethernet support from the very beginning. This opens the door to run the full flavour of EPICS core functionality on a sigle board micro controller only limited by CPU power and memory. The latter will even be increased by implementing the second board with additional memory.

5 Summary.

While keeping a good piece of the VME computer's functionality, GPFC is a pocket book sized low power device, which does not generally require forced air cooling. It can be installed in unmanned areas and can be embedded into equipment if needed.

The GPFC's modular architecture provides flexibility for connecting I/O peripherals via plug-on IP modules. It is also scalable in terms of memory and I/O resources, and keeps the door open for the possible future system extensions and add-on features.

The GPFC runs VxWorks operating system from WindRiver Systems. This provides an advanced operating environment that is typically offered at VME level, and also allows to easily port applications, which formerly ran in VME. Although, other real time kernels can be installed as well.

Being designed primarily for fieldbus control applications, GPFC may actually have a wider application range. It can be configured as an autonomous device, as a slave driven from some host, or as a fully functional Ethernet network node.

The design of GPFC was not easy and straightforward but highly challenging and painful. Trying to get as much functionality as possible under conflicting requirements has resulted in several board layout versions. Selecting and obtaining components, many of which were just recently announced by manufacturers, was also a multistep process. Design of an IP controller part added its own specific difficulties. A separate task was development of the SEDAC interface controller.

The current status is finishing with a first prototype of the main board. It is operating at 16 MHz and VxWorks runs on it. The second (32 MHz) version of the board is under CAD design, having corrections and some important improvements from the experience of testing the prototype. Two SEDAC IP modules have been separately tested and are operable.

An extension board currently has a somewhat suspended status, having been designed at the functional level, but now waiting for the final updates from the experience obtained with the main board.

The first GPFC application (SEDAC crate controllers) is expected to be running in the early 1998. The series GPFC production is under discussion between DESY and IHEP.

6 Acknowledgements

We would like to thank G.Csuka who consulted us on VxWorks/Tornado intrinsics and provided a user friendly software development environment, S.Rettig-Labusga and P.Urbschat who helped us in solving many everyday problems, S.Makonine and A.Machnachev who much contributed to the SEDAC interface IP module. We are grateful to G.Bialek, K.Mess, G.Hochweller for their consultations and comments, and to P.Gasiorek for his help with the FPGA/PLD programming. H.Ouw, W.Adamek, O.Niemetz did their best providing us with components and connectors. We also are grateful to R.Apel, D.Jahnke and R.Saust for their contribution at the PCB production stage and to W.Rode for the quick and faultless making the prototype boards.

7 References

[1] CAN Specification V2.0. Bosh, 1991.
[2] DIN, Der Profibus, Norm 19245 Teil 1 und 2, Beuth Verlag.
[3] VxWorks Programmers's Guide. WindRiver Systems, 1994.
[4] Tornado User's Guide. WindRiver Systems, 1996.
[5] PC104 Specification, V2.3. PC/104 Consortium, 1996.
[6] VITA 4-1995. IP Modules Draft Standard. VITA, 1995.
[7] MCF5206 User's Manual. Motorola, 1997.
[8] H.Frese and G.Hochweller. The serial data acquisition system at PETRA. DESY, 1980.
[9] R.Dalesio, A.Kozubal, and A.Kraimer. EPICS Architecture. Proc. of the ICALEPCS 1991, pp. 278-82.