

VME CONTROLLER BASED ON POWERPC 440GX EMBEDDED PROCESSOR FOR USING IN KSTAR DATA ACQUISITION AND CONTROL SYSTEM

Andrey Chertovskikh, KBSI, Daejeon, Korea

Abstract

The KSTAR (Korea Superconducting Tokamak Advanced Research) project provides for dozens of different subsystems with thousands of informational and control channels. Many of subsystems have different data bandwidth and network connectivity. It makes constructing of Data Acquisition and Control system very complicated and difficult task. Developing of new PowerPC-based VME controller is aimed at using it in wide range of network environment with different requirements to data bandwidth and connectivity. This document proposes new design principles for building network Data Acquisition system based on advanced futures of VME controller with newest PowerPC 440GX embedded processor from IBM. The key futures of the proposed VME PowerPC-based controller are wide set of the standard high-speed programmable network interfaces and high performance PowerPC embedded processor. As a result of using IBM PowerPC 440GX microprocessor new VME controller will have cost effective design and wide variety of software tools including real time operating systems. This makes the controller universal and powerful tool for using in KSTAR Data Acquisition and Control System.

MAIN FUTURES OF THE VME POWERPC-BASED CONTROLLER

The main goal of design of the VME PowerPC-based controller is a creation of convenient and effective tool for control VME systems inside high-speed computer network. The controller has several prevalent network interfaces that make it universal for different Data Acquisition and Control systems. The industrial standard PowerPC 440GX embedded processor is selected as a main CPU for the controller. It means that the controller can use broad set of software tools including real time operating system OS Open. Large SDRAM (up to 2 GB) make the controller convenient for effective buffering of experimental data and running operating systems. The PCI Mezzanine Cards (PMC) slot gives opportunity to use additional standard modules with wide range of futures: DSP and other processors, special interfaces and communication links. The list of main futures includes following:

- 500-600 MHz PowerPC 440GX embedded processor from IBM.
- Up to 2 GB DDR-333 SDRAM.
- Gigabit Ethernet interface.
- 10/100 Mbps Ethernet interface (2 ports).
- FireWire (IEEE1394) 400 Mbps interface.
- RS232 port.

- GPIO interface.
- PCI Mezzanine Cards Slot.

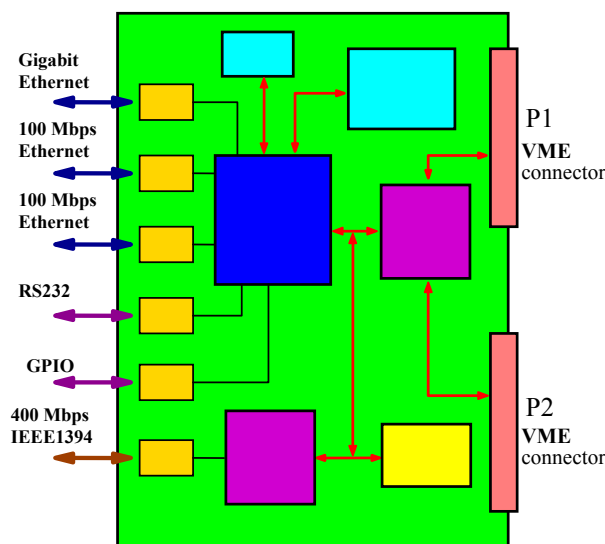


Figure 1: VME controller block diagram.

IBM POWERPC 440GX EMBEDDED PROCESSOR

The IBM PowerPC 440GX embedded processor is a newest system-on-a-chip (SOC) design that integrates PowerPC 440 embedded processor core with a rich mix of peripheral cores by implementing IBM's high speed CoreConnect™ technology. Built on 440GP infrastructure, using the Book E architecture it maintains signal compatibility while enhancing the future set and overall performance.

The main PPC440GX Features:

- PPC440 core
 - ✓ 466-600 MHz target frequencies, 2.0 DMIPS/MHz
- 32/64 – bit DDR200-DDR333 SDRAM controller
- PCI-X, 32/64 bit, 133 MHZ
- External bus control (EBC) interface
- Implemented peripheral cores
 - ✓ 10/100 Ethernet MAC (two ports)
 - ✓ 10/100/1G Ethernet MAC (two ports)
 - ✓ GPIO interface
 - ✓ Two IIC (inter-integrated circuit) bus interfaces
 - ✓ Two UART interfaces

DIFFERENT TYPES OF CONNECTION WITH VME PPC440GX-BASED CONTROLLER

Having applied PowerPC-based VME controller we provide a standard programming model for configuration network interfaces and reading VME modules. It means that the controller has operating system with network and VME specific drivers. This standard approach is similar with programming model for well-known VME controllers from Motorola MVME167 and others. It provides using of wide range of prevalent software including EPICS, VxWorks OS, Linux OS, OS Open RTOS and others. This conventional programming model is convenient and can be implemented very easily, but it does not satisfy the requirements of the high performance real time data acquisition systems. Using of operating systems, traditional network and VME drivers requires additional time for handling of network requests, data buffering and system process scheduling.

As a result of this the real time data acquisition with data rates near VME bus bandwidth cannot be implemented in frame of the model. So in conjunction with standard programming model non-standard programming model is supported. In this model VME controller can be considered as a network bridge between different network segments with different network and physical interfaces. Some of the segments are ordinary Ethernet segments – Gigabit and 10/100 Megabit Ethernet network segments, but one of the segments is special – it is VME bus segment. According this approach each VME module is a network device with own network address and can be accessed from any network node. The PPC440GX-based controller has to translate any network access - IP packet addressed to VME module, into VME bus transaction with that module. This translation can be both very simple, like reading out or writing in single register on VME bus, and complicated procedure, like reading block of data from FIFO or memory on VME bus with previous checking flag of ready of accessed module. Also a special mode with hardware interruption from VME modules must be provided when VME modules generate interrupts to the VME controller for requesting specific service – reading filled memory buffer, sending error message and so on. In this case VME controller must have special interrupt handlers for each VME module that can generate interrupts.

Thus described programming model applies following main functions of the PPC440GX based VME controller:

1. Transferring data from external host computers to VME modules by network packets.
2. Transferring data from VME modules to external host computers by network packets.
3. Special service for VME bus including handling of interrupts from VME modules.
4. Configuration of the VME controller by network connection to host computer.
5. Passing network packets from one channel to another according lookup table.

First two functions seem as simple and ordinary network tasks for a network processor: the controller has to detect network addresses of VME modules; replace addresses by corresponding lookup table and pass data to/from VME bus. But third function – special service for VME bus implies special service program for each VME module. It operates like driver of some peripheral device – checks status of the device, serves requests from the device and organizes queues for services from external access to the device. Forth, configuration function, of the controller is obvious - the controller must be initiated before using. It has to get information about installed VME modules, theirs VME and network addresses. Moreover special service programs – drivers for installed VME modules must be loaded and initiated. Last listed above function of the controller means function of network bridge – transferring packets between different segments of the network.

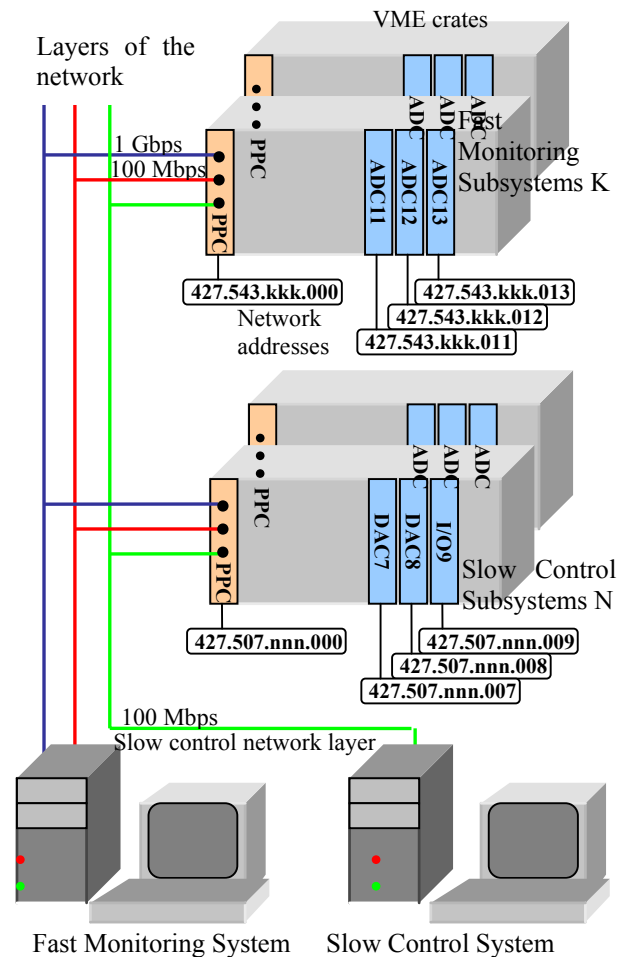


Figure 2: Network-Type Data Acquisition System.

Figure 2 shows example of using PPC440GX-based VME controllers for building large Data Acquisition and Control system. As example we used two different subsystems with different network data rate requirements: “Fast Monitoring subsystem” and “Slow Control subsystem”. As it may be clear from names, “Slow Control subsystem” has not high network data rate

requirements and 10 Mbps Ethernet can satisfy these requirements. In the example mainframe number N contains two modules of Data-Analog Converters (boards 17 and 18) and I/O module (board 9). Network addresses for these modules can be created by very simple way – 427.507.nnn.017, 427.507.nnn.018, 427.507.nnn.009, where “427” is a some number of the network, “507” – a number of subsystems, “nnn” – a number of VME mainframe and “017”, “018” and “009” – numbers of boards in the mainframe.

The main data flow to the subsystem from some external host computer will take one of 10Mbps Ethernet channels and will create “Slow Control network”. Of course, an access to this subsystem from other channels with other data rate will be also possible. For instance, “Fast Monitoring subsystem” can get some statistic information from modules of “Slow Control subsystem” via fast Gigabit Ethernet channel. In general, any host computer in network can get access to any modules in any subsystem in Data Acquisition and Control system via any possible channel. But system designers can limit the access to some subsystems (mainframes, modules) for safety purposes. Topologies of each subnet in common DAQ network can be different and depends on connectivity and data rate requirements. It can be star, monochannel or tree network.

Thus the programmable nature of the PowerPC-based controller interfaces gives us opportunity to implement some kind of “DNA” (direct network access) controller inside PPC440GX-based controller to organize “direct” connection between network and VME modules. Such kind of connections will not require software drivers and operating system running on CPU to pass packets of data between network and VME. Like usual DMA (direct memory access) controller “DNA” controller can be programmed by writing starting VME addresses, network address of the receiver and length of transfer in its control registers. Such way of using the VME controller can dramatically increase transfer rate and make the device as fast as “non-intellectual” type VME interfaces like SBS Model 620 and others.

The asynchronous and packet oriented nature of the Ethernet protocol makes “DNA-mode” unstable for this type interfaces – unpredictable transmission time will require special methods for retransmitting packets. However using of the FireWire interface will not meet this problem – isochronous mode of the IEEE1394 standard will help to avoid such kind of errors and makes the interface most effective and reliable for readout systems. Moreover the essential data rate for this interface (400 Mbps) is more balanced with VME bandwidth – 15-20 MB/s (up to 160 Mbps). One of the serious

disadvantages of this interface at present time is a short maximum length of the connections – 4.5m. But future specifications (IEEE1394b) will provide changing this limit for 50, 70 and 100m options. Implementing IEEE1394 external interface makes the connections VME and host computers very easy because of wide popularity of this type of interface. The interface can be used also to connect VME framework with external mass-storage devices, video devices and other equipment.

SUMMARY

This document proposes a project of the VME controller based on IBM PowerPC 440GX embedded microprocessor. Using industrial standard PowerPC microprocessor core gives opportunity to use a wide range of software including GNU C compilers, EPICS, Linux and real time operating systems. Implementing network controllers and peripheral interfaces inside one chip together with microprocessor core increases device integrity, allows applying broad set of network interfaces and makes device very universal component for different types of Data Acquisition and Control Systems. The PCI Mezzanine Cards (PMC) slot allows using additional standard modules. It can be DSP and other processors, special interfaces and communication links. In addition to standard programming model for the existing intellectual VME controllers, proposed “network-type” programming model for the VME interface has following new important futures:

1. *Every VME system in DAQ can be considered as a segment of local network.*
2. *Every VME module gets own network address and can be considered as a network device.*
3. *Every VME module is accessible for any computer in local network by standard network software.*

The controller has following external interfaces: Gigabit Ethernet, 10/100 Mbps Ethernet, FireWire (IEEE1394), GPIO and RS232 serial interfaces. Gigabit and 10/100 Mbps Ethernet interfaces make the device compatible with prevalent network standards. The FireWire interface adds new opportunities in connection the VME framework with broad set of devices: mass-storage devices, video devices, PC and so on. Due to isochronous mode this interface became ideal for real time readout systems with transmission time limits.

REFERENCES

- [1] PowerPC 440GX Embedded Processor Datasheet, IBM, <http://www-306.ibm.com>.