TRIUMF/ISAC EPICS IOCS USING A PC104 PLATFORM

G. Waters, R.Nussbaumer TRIUMF, 4004 Wesbrook Mall, Vancouver, B.C., V6T 2A3, Canada

Abstract

The EPICS control system for the TRIUMF/ISAC Radioactive Beam Facility uses VME based IOCs (input/output controllers). In cases where the IOC only communicates with non-VME subsystems, notably PLC and GPIB device clusters, this implementation is very costly. For these systems we are presently installing more economical IOCs with Intel x86 CPUs in a PC104 form factor. We use an embedded controller based upon the ZF x86 "PC on a chip". This processor is designed specifically for embedded applications and uses industry standard PC architecture. It includes dual 10/100 BaseT Ethernet and 32-bit PCI bus plus a BIOS and failsafe Boot ROM. Details of the necessary integration work and initial operation experience will be discussed.

1. INTRODUCTION

The control system for the TRIUMF/ISAC Radioactive Beam Facility is implemented in VME. An embedded processor running VxWorks in each VME crate functions as an IOC for EPICS. In addition to handling VME IO some of the IOCs also control PLCs via TCP/IP and CANbus devices with an industry-pack interface. In some instances the IOCs control network devices only, leaving the VME crate essentially empty. To keep the cost per IOC down and to free up valuable VME processors and crates we decided to investigate the possibility of using processor cards based upon the pc104 standard [1]

2. PROCESSOR

One of the main considerations when choosing a suitable card and processor was its ability to run as an embedded system without the need of a local mechanical disk system for the boot process. We chose the MZ104+ from Tri-M Systems [2] because a board support package for VxWorks was available and the purchase price included a VxWorks target license. The board has the following features:

- Dual 10/100 BaseT Ethernet
- ZFx86 System-0n-a-chip (486 equivalent).
- Dual RS-232 serial
- EIDE and floppy interface
- Up to 64MB of RAM
- DiskOnChip Socket
- Phoenix BIOS and FailSafe ROM

3. CONFIGURATION

3.1 Development System

For initial development work and bootrom customization we used a floppy disk system for booting VxWorks. The MZ104+ was connected to a development kit which included a video card and floppy disk drive. We formatted a floppy disk with the VxWorks boot loader (vxld) in the boot sector and copied the VxWorks bootrom image (bootrom.sys) to the disk. Booting proceeds as follows:

1. The BIOS code loads and executes the bootstrap loader (vxld).
2. Vxld finds the bootrom.sys file, loads it into memory and transfers control to the bootrom code.
3. The bootrom loads the VxWorks image (from disk or via the network) into the target ram and passes control to its transfer address.

3.2 Operational

In an operational environment a diskless boot process is desirable. We used a BIOS editing tool to attach customized network boot code to the bios (option-ROM). The MZ104+ has a built in 12Kbyte BIOS update ROM (BUR). This allows fail-safe flash BIOS and option-rom installs and upgrades. A proprietary high speed port (Z-Tag) is used to download the new code. A special “dongle” plugged into the Z-Tag port handles the data transfer from a host PC’s parallel port. A Z-tag programming tool handles file selection and download. The act of plugging in the dongle disables the BIOS and starts the BUR. The dongle can operate in two modes: ‘pass through’ and ‘dongle’. In pass through mode data is transferred continuously through the dongle into flash. In dongle mode data is loaded into the dongle’s internal memory. It has sufficient memory to hold the vxloader and bootrom. The dongle can then be carried to a remote location and plugged into an MZ104+ for field upgrades. The BUR downloads directly from the dongle into flash.

3.3 Extension ROM

During the Power-On and Self-Test sequence the customized BIOS looks at the beginning of every 2Kbyte block from address C8000h to F0000h searching for a 55AA pattern. After validating the correctness of this extension ROM, it calls the routine at offset 03. This code can be anything, but for our purposes it is the VxWorks loader (VL). This is a modified version of the original source code freely available from ZF Micro [3]. Its function is to copy the bootrom stored elsewhere in flash to a RAM address specified in the bootrom header. For this sequence to work the BIOS settings must be modified to disable disk boots and enable the extension ROM scan routine. The BIOS editing tool allowed us to create a custom bios with the required settings as the default. Since the current configuration does not include any NVRAM the boot parameters are embedded within the bootrom image. To make a permanent change a new bootrom image must be created and loaded using the dongle.

Figure 1 shows the memory map assumed for the boot sequence which proceeds as follows.

1. BIOS performs a ROM-scan sequence and the VxWorks loader (VL) is invoked.

2. VL relocates itself to main memory 0200:0000.

3. VL gains access to the full memory space and copies the VxWorks bootrom image from flash to low RAM.

4. VL transfers control to the bootrom which performs a network boot of VxWorks as specified by the boot parameters.

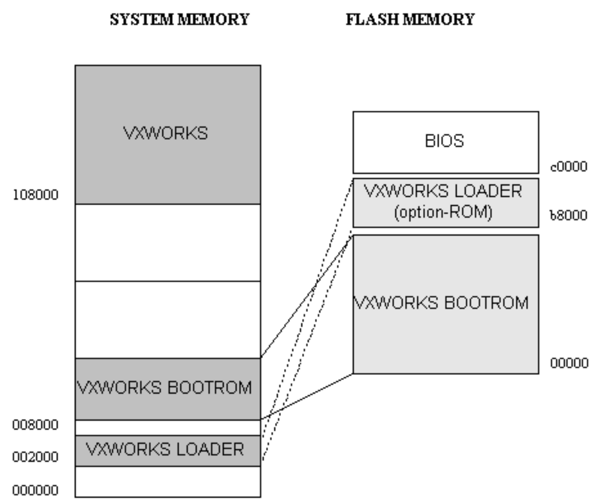

Figure 1 MZ104+ VxWorks Memory Map

4. PACKAGING

The MZ104+ card has been integrated with an embedded serial server [4], to allow a remote connection to the serial port. Figure 2 shows the complete unit configured to mount on DIN rails inside a standard TRIUMF/ISAC break out panel (BOP). Power is provided by the 24 volts available in all BOPs. A remote reset input has also been provided. This is controlled by a PLC driven contact within the same BOP.

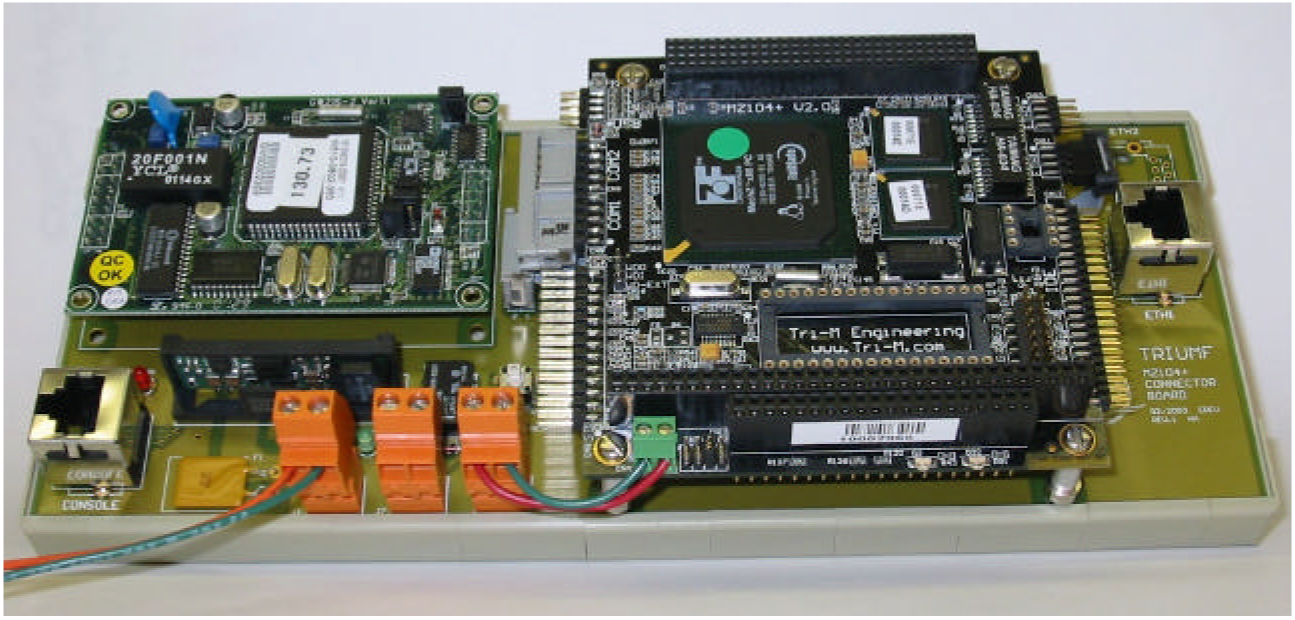

Figure 2 MZ104+ and serial interface (9 in x 4 in)

5. SOFTWARE

5.1 VxWorks 5.4

A VxWorks 5.4 board support package for the processor was available from ZF Micro Solutions. Under this release we experienced serious network connectivity problems. Usually within a few minutes of startup all network activity on the MZ104 would cease. This would usually happen simultaneously among all development MZ104s running at that time. It was as if the network controller (Intel 82559) simply stopped responding. However, within the ISAC controls subnet, which is protected by a firewall, all MZ104+ IOCs ran reliably. Despite frequent attempts to correlate the effect with site network events no definitive cause was found.

5.2 VxWorks 5.5

Concurrently with our work on pc104 devices we were proceeding with evaluation of a VME Pentium III processor module from General Micro Systems [5] namely the GMS V158. For this device we had Pentium support for VxWorks 5.5. The board uses the same network chip as the MZ104+ and we experienced no network problems of any kind. Wind River [6] has recently added support for the 486 under this version. We modified the MZ104+ board support package accordingly and have experienced no further network difficulties.

5.3 Future Development

The MZ104+ has on board a socket for a DiskOnChip [7]. This is a solid state disk which can be formatted as a bootable disk. However, this requires the MZ104+ card to be placed in the development backplane and attached to a disk system and video card in order to use the DiskOnChip dos utilities to configure it. Although the BIOS can "see" this disk enabling us to boot VxWorks as well as Linux from it, our current version of the board support package does not have full True Flash File System (tFFS) support. We plan to add such support and use the flash disk for the storage of boot parameters and information needed for bumpless reboots. We will also investigate the practicality of direct access to the BIOS extension ROM as an alternative to using the dongle for bootrom updates.

6. STATUS

At the present time, four MZ104+ units are operational and have been running reliably with EPICS R3.13.5 under VxWorks 5.4 behind a firewall for several months. We are currently updating them to VxWorks 5.5 and EPICS R3.13.9.

6.1 ISAC RF controls

Remote control of all RF cavities is provided by a simple UDP messaging system. The original drivers and EPICS device support were designed for the mvme162; a 68k based VME IOC. Since the messages are character based, network byte ordering was not an issue and the code was ported without modification.

6.2 ISAC Test Stand Vacuum

This system is controlled by a PLC using modbus protocol inside a TCP/IP message. Again, drivers and device support currently in use on mvme162 IOCs were successfully ported after adding extra code for network byte ordering. A second MZ104+ is used in a similar manner as part of an ion source expansion program.

6.3 GPIB

Drivers have been written and tested for

- GPIB/ETHERNET interface.
- PC104 GPIB IO card

Applications using these are the subject of a separate paper at this conference.

6.4 CANbus

There is, at this time, at least one suitable pc104 CANbus controller available. It uses the same hardware as an Industry Pack device and also a PMC card currently in use at TRIUMF. Adapting the driver and EPICS device support to pc104 will be straight forward.

7. CONCLUSION

We have completed extensive testing and evaluation of the MZ104+. It has proved to be robust, versatile and reliable under all operational conditions and well suited to our requirements. The pc104 standard is well established and we will continue to take advantage of the growing number of pc104 IO cards available well into the future.

REFERENCES

[1] PC104 Consortium, http://www.pc104.org/
[2] Tri-M Systems http://www.tri-m.com/
[3] ZF Micro Solutions http://www.zfmicro.com/
[4] GW21S-256, ATOP technologies http://www.atoptechnologies.com/GW21S-256.htm
[5] General Micro Systems, http://www.gms4vme.com/gmshome/index.html
[6] Wind River http://www.windriver.com/
[7] M-Systems http://www.m-sys.com/
[8] R. Nussbaumer, G. Waters, Inexpensive IOCs for GPIB support in the TRIUMF/ISAC Control System, these proceedings.