

A PMC Based Computing Core

PCI bus has been very successful in providing a standard interface for high performance I/O applications. On VME or CPCI single board computers modular I/O functions are usually implemented in PMC format. The "Multifunction Computing Core" (MFCC) described here has been developed by CES to provide a uniform hardware basis for all digital I/O applications that require processing power for reformatting data or handling complex protocols. The MFCC features a PowerPC based computing core (CPU, SDRAM, and FlashPROM), a PCI bridge and a large FPGA to implement a wide variety of custom specific protocols. About one quarter of the PMC surface is reserved for an electrical transition module adapting the FPGA I/O pins to the electrical signal characteristics required by the specific application. In addition, 64 I/O pins of the FPGA are wired to the VME P2 connector. A fast staging buffer (SSRAM) is directly attached to the FPGA.

INTRODUCTION

PCI Bus is widely used as a high performance local bus both in desktop workstations and Single Board Computers (SBCs). With the RIO2 806x [1] and the RIO2 406x[2] CES has built a family of PowerPC based SBCs both in VME and CPCI which use PCI as their local bus and provide two PCI Mezzanine Card (PMC) extension slots within a single VME or CPCI slot. Up to 6 PMCs can be attached to a single RIO2 by using the PCI extension boards PEB 6406 and PEB 6407[3].

Data acquisition and control functions that used to be implemented in complete VME crates can today be integrated into a few slots using SBCs that drive I/O functions implemented as PMCs. The performance of such a system depends critically on its capability to off-load I/O intensive tasks from the main processor. As any modern high performance microprocessor, the PowerPC works best on data in its own system memory where it can access them using a sophisticated two-level cache architecture. Frequent accesses to I/O registers break this performance. On the other hand, a

large number of I/O subsystems used in data acquisition require the implementation of complex protocols. The possibility to pre-process data already on the PMC level becomes thus a key element in the construction of PCI centered data acquisition and control systems. Providing a PowerPC computing kernel directly on the PMC solves this problem in a very flexible way and provides a maximum of software compatibility between the PMC and the host SBC.

With the Multifunction Computing Core (MFCC 8441) CES has created a PMC that provides a PowerPC (603e or Arthur) computing core together with a large FPGA suited to implement complex application specific I/O functions. A second FPGA implements a PowerPC to PCI Bridge that provides all features necessary for efficient communication with the host system. An electrical adapter carries the front panel connector and all circuitry needed to adapt the FPGA signals to the electrical levels required by a specific application. It is implemented as separate PCB occupying one quarter of the PMCs surface and attached by a 120-pin connector to the MFCCs main PCB.

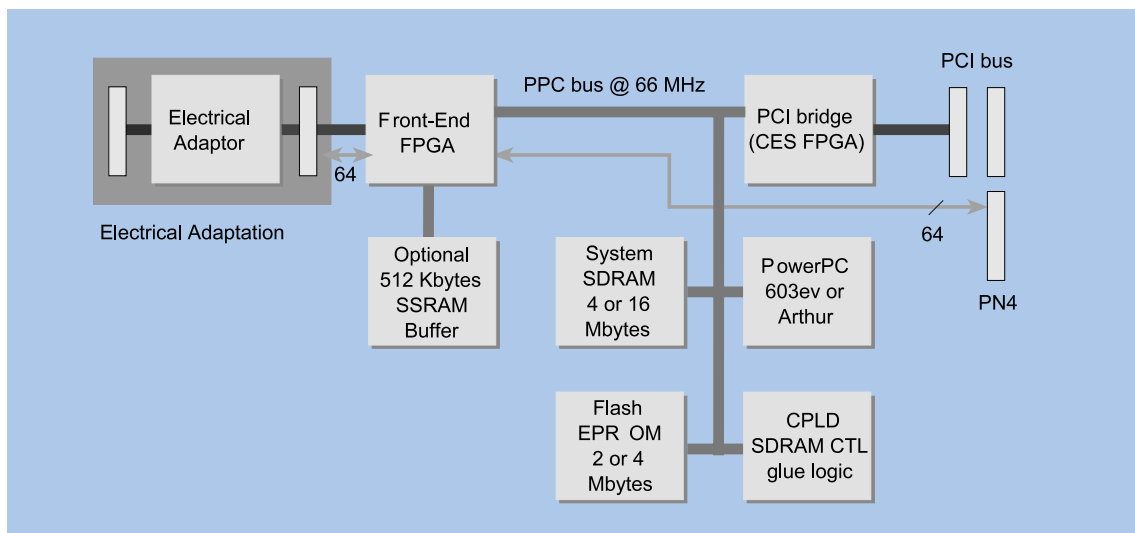


Figure 1. MFCC block diagram

A block diagram of the MFCC is given in Figure 1.

ARCHITECTURE

After passing this electrical adapter or a corresponding adapter in the rear of the crate (s. below), signals are processed by a Front-End-FPGA before they are stored in the MFCC's system memory which forms, together with the PowerPC CPU, FEPROM and timers the MFCC's CPU core. Finally, the filtered data are forwarded through the PowerPC-PCI Bridge that is also used to control the MFCC from the host system.

The Front-End FPGA

A "Front-End (FE)" FPGA (ALTERA 10k50, 10k100, 10k130) is dedicated to application specific logic. A total of 138 I/O pins are available to implement a wide variety of communication protocols. 64 pins are wired to the third PMC connector. This connector is connected to P2 on the RIO2 and to the CPCI I/O connector on the RIO2, leaving all the space available on rear-mounted transition modules for electrical adaptation and connectors. The remaining 74 lines are fed to the front panel adapter board that carries application specific electrical transceivers and connectors. Out of the signals available on the front panel, 32 lines are also routed to a fast SSRAM (up to 512 KB) that can be controlled by the FE-FPGA. This buffer memory can be used to implement an intermediate data buffer compensating for latencies due to arbitration for the PPC bus.

The PowerPC CPU core

A PowerPC (603e@166MHz. Arthur@200MHz) CPU provides the CPU power that is needed to process the data stream delivered by the front-end before passing it on to PCI. By optimizing the interface to the system memory (16, later 64 Mbytes of synchronous DRAM, 2 clock CAS latency) we obtained memory access speeds 30% better than on the RIO2 or RIO2. Firmware, application software and even complete OS kernels can be stored in 2 Mbytes of Flash EPROM (4 Mbytes foreseen) that can be programmed by the on-board CPU or through the PCI interface. It is complemented by a serial Flash EPROM that holds the FPGA logic, vital configuration parameters and a backup version of the boot firmware. As all programmable logic on the MFCC, the serial FEPROM can be programmed via the "CES In Situ Programming" (CISP) tool, allowing the card to be completely configured by simply connecting a PC to the corresponding 10 pin microconnector.

The PowerPC-PCI bridge

The PowerPC to PCI Bridge, implemented in an ALTERA 10k50 FPGA, provides everything needed for the efficient communication between the MFCC, the host CPU and other PCI devices. In addition to standard PCI configuration registers, it provides a PCI slave interface that maps the MFCC's internal resources to PCI, a PCI master interface that gives the on-board CPU access to the host PCI system and a DMA engine (foreseen) for efficient data transport between PCI and the MFCCs system memory. Two sets of FIFOs are provided: one is reserved for serial line emulation between the MFCC and the host CPU, the second is free to

implement message protocols. Each set contains two 8-bit * 256 FIFOs, one for each direction. The MFCC can also monitor the PCI INTA-D lines and act as an interrupt controller for these interrupts.

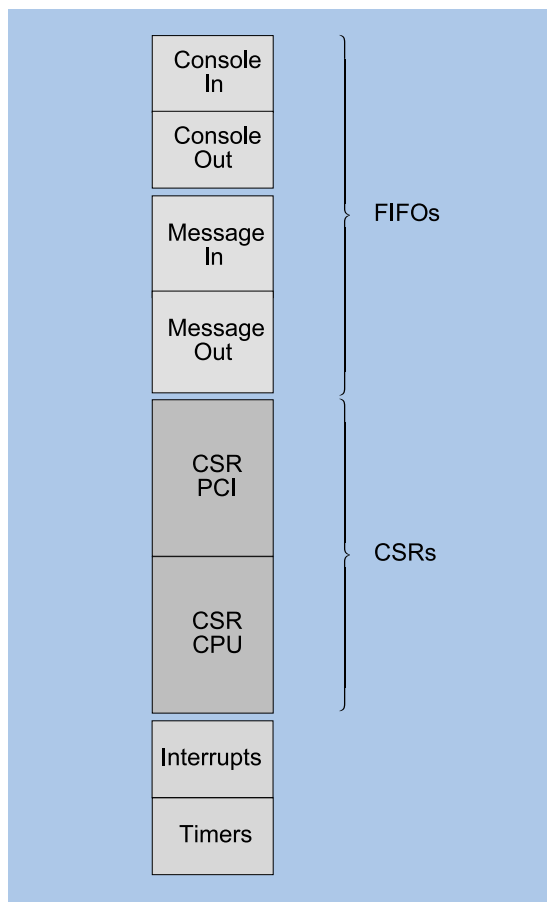


Figure 2. Register layout of the PowerPC-PCI Bridge

DEVELOPMENT ENVIRONMENT

Developing a specific I/O function based on the MFCC may involve both the development of logic for the FE-FPGA and the development of application software for the CPU core.

The CISP (CES in Situ Programming) tool

In order to develop code for the FE-FPGA, a user needs logic design environment capable to generate code for the ALTERA 10k FPGA family. CES can provide assistance in form of training and VHDL or AHDL descriptions of functional blocks (such as the PPC bus interface). The implementation of a complex I/O function by CES according to a customer's specification can be discussed.

Once the FPGA code is generated, it must be transferred into the MFCCs serial FEPROM to be loaded into the FPGA during the next boot. This can be done in three ways.

- By the on board CPU
- By a PC attached to a 10-pin micro connector via the CISP tool.
- Using the PMC JTAG pins.

The first way is flexible and can download the FPGA code from any resource that is accessible to the host RIO2/RIOC system (disk, network...) However, if due to a programming or configuration error the MFCC fails to boot, this way of re-loading the FPGA is blocked. For a development environment in which the FPGA code is frequently changed and the risk of faulty intermediate versions exists, the second method is therefore better adapted. The third method is both flexible and safe but requires a host system that can manipulate the JTAG lines on the PCI bus. This will be supported in future versions of CES PowerPC boards.

When using the CISP tool, the MFCC is connected to the parallel port of a PC via a small adapter box. CES provides the software to download and verify all programmable logic of the MFCC, including the serial FEPROM that hold the FPGA equations and the electronic signature of the board's configuration. This programming procedure does not depend on the prior state of the MFCC and can be used even if no valid FPGA code has yet been loaded.

SOFTWARE

The MFCC is equipped with the "MFCC_Mon" boot

monitor that provides a subset of the PPC_Mon [4] functions available on the RIO2/RIOC. Commands for diagnostics, loading and starting of application programs, loading of FEPROM and FPGAs and setting of configuration parameters are available.

Figure 3 shows a screen copy of the MFCC_Mons startup messages.

Boot Process

Depending on the state of a hardware switch, the MFCC can either boot in automatic mode or rest in the reset state until the host system releases the reset by writing to a register. During this write cycle, a pattern can be passed to the MFCC, which can be interpreted by the MFCC's boot firmware to define the boot mode. In automatic mode or once the host system releases the reset, the MFCC proceeds through its boot phases according to configuration parameters stored in the MFCC's serial FEPROM. Early during the boot process the communication channel to the host system is initialized to be used for MFCC_Mon's console I/O. The boot can then stop in MFCC_Mon or it can proceed to start an application program or boot an Operating system.

```

MFCC_Mon Boot:
PPC_Mon>mfcc.1 connect
MFCC8441 rev: 0.0 serial: 000
CPU : PPC603e ver: 2.01 speed: 133 Mhz
Memory size : 16 Mbytes
FEPROM :
PCI bridge :
I/O interface :
Adaptator :
Entering boot diagnostics
0 Check System Memory (0x00000000 - 0x00effffc) 0 OK
1 Check Interrupt Handler 0 OK
2 Check PCI Bridge 0 OK
3 Check Cache and MMU 0 OK
4 Check Interrupt Controller 0 OK
5 Check TIC Timer and WatchDog 0 OK
6 Check FIFO's (0 - 4) 0 OK
*****
* MFCC_Mon MFCC8441 monitor - version 0.1 *
* CES SA Copyright 1997 *
*****
MFCC_Mon>help<CR>
command list:
boot cache cm config
diag dm dr dp
ds fm fp fs
go help lm load
lp mm nvram pm
pp ps set show
sm smon sp ss
status tm tp version

```

Figure 3. Screen copy of the MFCC_Mons startup messages

Ad ZF Microsystems

Communication with the host system

The MFCC communicates with the host CPU via its PCI interface. In the simplest form, a terminal emulation protocol is implemented using one of the MFCC's two FIFO sets. This communication channel can either be used in handshake mode so that writing to the console is blocked once the corresponding FIFO is full, or in free running mode removing the first character in the FIFO each time a new character is added to its end. In the first case the boot process is synchronized with a terminal emulation utility on the host system, in the second case the MFCC boot proceeds independently (some of the console output may be lost).

An application program can be compiled and linked on the host system (RIO2 or RIOC) using the development environment provided by CES (MFCC low-level libraries, Makefiles.). Once an application program has been built it can be downloaded in the MFCC's system memory or FEPROM using the corresponding MFCC_Mon commands. The connection to MFCC_Mon is established using a terminal emulation facility on the host system.

MFCC_Mon contains a test facility that allows executing user specific tests of the card. As part of the MFCC's support environment, templates and Makefiles for such tests are provided.

Operating System Support

Alternatively, an operating system kernel can be booted on the MFCC. The MFCC system is then connected to the host system using a "backplane driver" implemented on the PCI bus. In this case the MFCC appears just as another node in a cluster of processors interconnected by a network. The port of LynxOS,

VxWorks/Tornado and Chorus to the MFCC is foreseen.

TYPICAL APPLICATIONS

The MFCC may be used to reduce the size of a multi-processor-based system by replacing traditional 6U Eurocard size SBCs with PMCs. It can also be used to maintain the I/O performance of a given I/O channel when additional channels are added to the system. This is possible because with the MFCC the CPU power for handling the protocol is increased in step with the number of I/O channels added. The application example shown below is taken from an aerospace bus controller system. ■

REFERENCES

- [1] CES, RIO2 8062 User's Manual. DOC 8062/UM ver. 1.0, Oct 1997.
- [2] CES, RIOC 406x User's Manual. DOC 406x/UM ver. 0.1 Oct 1997
- [3] CES, PEB 6406 and PEB 6407 CES VME PMC extension boards, User's Manual, DOC 6406/UM ver. 1.0 Dec.1997
- [4] CES, AWX 3317C CES PPC_Mon, Monitor for CES PowerPC based boards, User Interface, DOC 6406/UM ver. 4.0 Oct.1997

Mr. Weymann, PHD in Physics, participated in the DAQ design for experiments (OPAL, OMEGA) in high energy physics at CERN, involving both hardware and software developments. Since 1992, he is responsible for systems integration and technical at Creative Electronic Systems in Geneva, Switzerland.

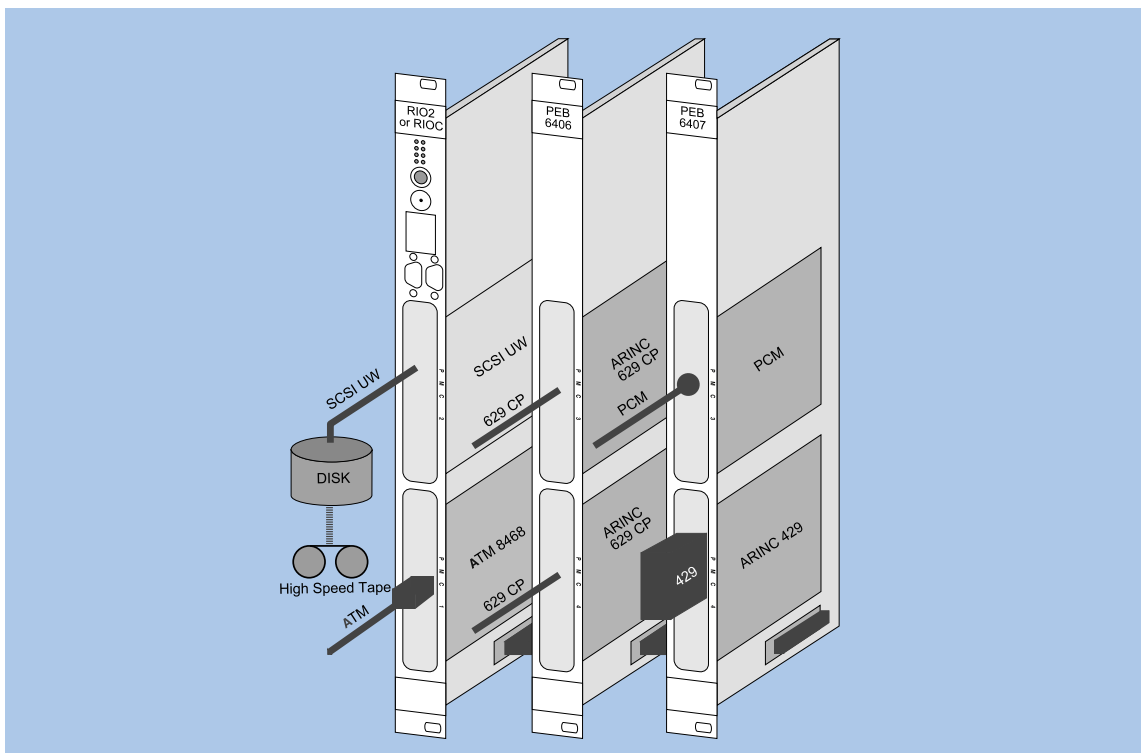


Figure 3. MFCC Application example