PowerPC and High Speed Interconnection Technologies Applied to Distributed Real-Time Data Acquisition and Computing

Joël Bovier, François-Henri Worm

Creative Electronic Systems
Route du Pont-Butin, 70
CH 1213 Petit-Lancy
Switzerland

Tel: +41-22-792-57-45 / Fax: +41-22-792-57-48
Email: ces@lancy.ces.ch
Internet: http://www.ces.ch/CES_info/Welcome.html

1. Introduction

Architectures used for controlling large real-time systems have always reflected the evolution of computer technology. Historically they have been based on:

- central computers (mainframes)
- minicomputers (with the first real-time kernels)
- microprocessors

When the concept of a central computer - or computer cluster - was abandoned the technology of interconnections to share data and commands between distributed control units became a key element of the architecture.

Data transport architectures evolved from:

DMA channels	point to point	single master	fast
through:			
GPIB	multidrop	multimaster	slow
up to:			
bus extensions	multidrop	multimaster	fast

Local area networks are now a standard feature in computer architecture and have provided well-established solutions to control the operation of different computers with:

- Ethernet (TCP-IP)

and

- FDDI (being well established)

and

- ATM (about to provide the next step in performance)

However these interconnections do not provide a natural control flow and therefore are not well adapted for systems requiring a deterministic behavior.

With the enormous growth of the processing power achieved by the PowerPC processors - an order of magnitude at the lowest end of the line compared with the 68040 generation - with the satisfactory experience acquired with real-time UNIX on VME platforms and with the emergence of PCI as a mezzanine carrier bus a new architecture can be planned.

2. Basic Elements of the Architecture

Detailed here is an architecture which is both up-to-date as to technology and appropriate for real time applications.

Hardware Elements Consist of:

• Input / Output Couplers	(VME Boards or PMC Mezzanines)
• Real-Time Processing Units,	2 types of platforms:
- Development Platforms	603, 604, 620 based with single PMC
- Target Platforms	603, 603E, 604E based multiple PMC
• Inter System Couplers (Data Movers)	(VIC, FDL)
• Network Couplers	(FDDI, ATM)

Software Elements Consist of:

- Lynx-OS (or VxWorks) development suite including:
  - self-hosted development systems
  - diskless development systems
  - target systems

- Complete set of:
  - VME
  - VSB
  - VIC
  - FDL
  - FDDI
  - ATM   libraries or drivers

- interprocessor synchronization mechanisms
- extensive set of low-level debugging tools
- advanced object oriented middleware, e.g. RTWorks

2.1 The Input / Output Couplers

The I/O couplers are connected either directly to VME in the case of couplers requiring no on-line dynamic data manipulation, such as ADCs and DACs, or to the PCI bus in the form of PMCs (PCI Mezzanine Cards) when either complex or dynamic data manipulation is required.

In the latter case they are housed on a PowerPC-based I/O controller (RIO2 8060).

2.2 The Real-Time Processing Units

Two PowerPC-based processors with different special enhancements for real time use have been designed as complementary platforms, because a single design could not meet all requirements.

The two platforms are:

- the development platforms (RTPC 8067)
- the target platforms (RIO2 8060)

2.2.1 The Development Platforms

The development platforms are complete self-hosted development systems providing the following facilities:

- full hardware and software development environment for the target platforms
- top of the line processing power for massive on-line mathematical analysis
- top of the line VME / VSB performance with an intelligent controller to guarantee a stable processing power even with high or random VME or VSB bus activities.

The development platforms have a system memory structure organized to run a full-featured real time UNIX (Lynx-OS) providing a comfortable and efficient software environment. Object oriented middleware (RTWorks) is also available at this level.

A block diagram of the RTPC 8067 is shown below:

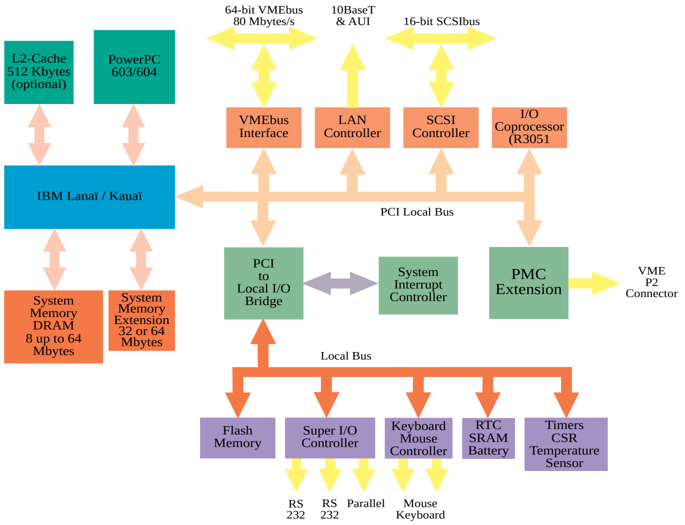
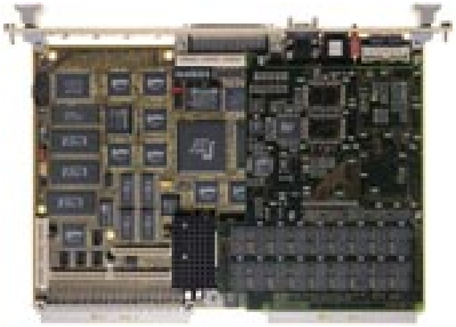


Fig. 1. RTPC 8067 block diagram

A detailed implementation of the RTPC 8067 is shown below:



The RTPC 8067 is based on the industry-leading IBM PowerPC chip; either the 603 clocked at 66 MHz or 603E / 604 chip clocked at 100 MHz. This delivers state-of-the-art performance and a guaranteed power escalation path in a single-slot VMEbus 6U form factor. A complete set of I/O functions, including VME, VSB, Ethernet and SCSI, are also provided, making the RTPC 8067 truly a single-board computer. The performance of the RTPC 8067 is balanced between raw CPU throughput and I/O bandwidth, offering solid processing speed without the I/O bottlenecks that often impede RISC CPUs' performance on other boards. The RTPC 8067 uses the industry standard PCI as a backbone bus. It also provides a PCI mezzanine for add-on off-the-shelf PCI interfaces as well as PCI extension bus.

Architecture and Operation

The RTPC 8067 consists of several subsystems, each of which provides a specific function. These subsystems are interconnected on one PCI bus.

CPU Subsystem

The CPU subsystem is contained on a daughter card and is built around the 64-bit PowerPC chip. The PowerPC processor is clocked at 66 or 100 MHz and has an on-chip 2 x 8 kB of first level cache. It is interfaced to an optional external 512 kB second level cache. It is also bridged to the PCI bus and interfaced to the DRAM (8 up to 128 MB) using the IBM Lanaï - Kauaï chipset. An unusual but vital feature is the support of the second level cache with a size of 512 kB. ECC is provided for the second level cache and the DRAM.

A multiple-level write buffer is provided to minimize CPU stall cycles during writes to the memory and PCI bus. Coupled with single-clock DRAM access during cache refills, this minimizes the number of CPU cycles wasted due to CPU stalls.

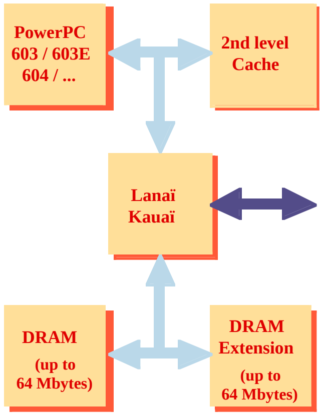


Fig. 2. CPU Subsystem

PCI Bus

The PCI bus is a high-speed synchronous parallel 32-bit bus driven by a 32 MHz synchronous protocol. The PCI bus provides a sustained 128 MB/s bandwidth. The following potential masters are connected to the PCI bus: the PowerPC CPU and the DRAM via the Lanaï - Kauaï chipset, the Ethernet controller, the SCSI controller, the SUB I/O bus, the PCI extension, the VME interface, and the optional VSB interface. These masters share the PCI bus through a central arbiter. A PCI extension connector is available to connect the PMC extension card (PEB 6406 and PEB 6407).

Global Memory

The global memory resides on the CPU daughter card and on an optional memory extension piggy-back, providing storage ranging from 8 to 128 MB. It supports ECC and communicates with the CPU via the Lanaï - Kauaï chipset. It is also interleaved on two alternate banks to provide the bandwidth required for maximum speed CPU cache refills.

VME and VSB Interfaces

CES has designed a special VME / VSB to PCI interface with a performance exceeding that of other commercially available products.

The VME interface works in master and slave modes; the optional VSB interface works in master mode only. Both have block transfer capabilities, 64-bit BLT for VME and 32-bit BLT for VSB. The respective bandwidth reaches 80 Mbytes/s for VME and 40 Mbytes/s for VSB. A 64 x 32 bit deep FIFO offers an optimal use of the PCI, VME and VSB busses by performing write posting and read prefetch. An independent DMA logic is associated with the bus interfaces and the 64 x 32 bit deep FIFO allowing very high speed linked-list data transfers. The VME slave interface allows the mapping of the internal resources (DRAM, FIFOs, SRAM, CSR, etc.) through size programmable windows (16 Mbytes to 128 Mbytes), addressable from the VME bus.

The master interfaces use a unique MMU architecture to allow mapping of the VME and VSB addresses as they are written to by the CPU or the DMA logic. Windows, 4096 in number, of 64 kB each are provided for a flexible mapping of the VME and VSB spaces. Each window contains information such as high address, AM code, and the VME and VSB access enables. This architecture allows the user to build all addressing modes on the VME and VSB and to map the address transferred to the external bus to any desired value. Each master interface has its own dedicated arbitration requester with all available options implemented. The VME interface also provides a full slot 1 arbiter including time-out generator and ROR, RWR and Fair and Hidden arbiter modes.

Communication FIFOs

The RTPC 8067 provides eight communication FIFOs (32-bit wide and 255 words deep) which can be accessed for READ and WRITE operations by the CPU and the VME slave. They can be used to implement a multiprocessor message passing capability. Each FIFO may be configured to interrupt the CPU when empty, not-empty or full, via a set of registers accessible from the I/O-bus control operation.

I/O Interfaces

Only the special features will be described, Ethernet, SCSI and other interfaces being standard.

Local Resources

The RTPC 8067 local resources include a Flash EPROM, an SRAM, a real time clock, timers and an interrupt controller. These resources are connected to the PCI bus via glue logic.

Flash EPROM

The Flash EPROM memories containing the bootstrap program, firmware and monitor program are built around 4 Mbits Flash EPROMs allowing capacities from 1.5 to 6 Mbytes. Special utilities are available in the PROM monitor to support management of the Flash EPROMs (down-loading programs, ROMable kernels, ...).

SRAM Memory

A 128-KB SRAM memory is provided on the RTPC 8067. This memory is used as a working space for the second processor (the R3051) and it can also be used as a data communication space.

Real-Time Clock / NVRAM

A real time clock is implemented on the RTPC 8067 using the M48T18 chip. It has a backup battery for saving the time-of-day value, and an 8 kB nonvolatile memory space.

Timer-Counters

Three additional 16-bit timers (1 µs resolution) are provided using a Zilog Z-CIO 8536 chip. They have programmable multi-function outputs and flexible re-trigger facilities for timer control and synchronization. Signals are available on the on-board connector.

Interrupt controller

A CES ASIC, the SIC 6351 chip, is used as a local interrupt controller and as a VME interrupt generator. The SIC chip can handle up to 48 vectorized or non-vectorized interrupt sources.

List Processor

A List and Boot processor based on the R3051 RISC chip is provided on the RTPC 8067. This processor is used during the boot process to down-load the PowerPC CPU operating parameters, and to program the RTPC 8067 configurable logic. It is interfaced to the PCI bus via glue logic and communicates with the PowerPC processor through control logic. Furthermore it can control all the VME / VSB I/O flow, thus relieving the PowerPC engine of this tedious task, and hence providing the highest computing power from that processor.

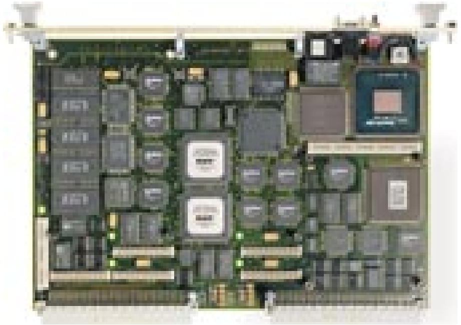
2.2.2 The Target Platforms

The target platforms are optimized for delivering the maximum power on:

- VME
- VSB
- PCI

from a low-power PowerPC processor equipped with application oriented software of the embedded type. This means the system memory architecture is specially adapted to this task with a much larger Flash EPROM memory than on the development machine. The fan-out for PMCs is also larger than on the development machine, most applications requiring two or more PMCs. It must be noted that both platforms support direct access between PCI and P2 with high speed bus backplane supported (VSB etc.).

A detailed implementation of the RIO2 8060 is shown below.



The PCB accommodates either the 603 chip clocked at 66 MHz (RIO2 8060-603) or the 604 chip clocked at 100 MHz (RIO2 8060-604) and delivers state-of-the-art performance and a guaranteed power escalation path in a single-slot VMEbus 6U form factor. A complete set of I/O functions, including VME, VSB and Ethernet are also provided. The performance of the RIO2 8060 is balanced between raw CPU throughput and I/O bandwidth, offering solid processing speed without the I/O bottlenecks that often impede the CPU performance in other boards.

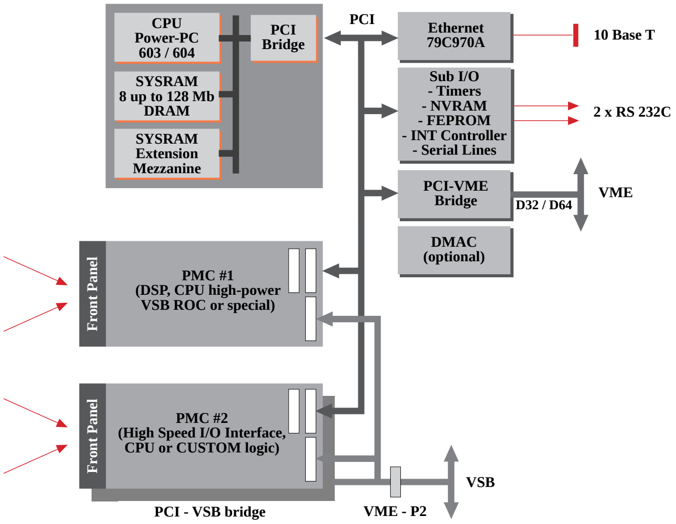


Fig. 3. RIO2 8060 Block diagram

CPU Subsystem

The CPU subsystem is built around the 32-bit Power-PC 603 / 604 chip from IBM. The Power-PC 603 processor is clocked at 66 MHz whereas the Power-PC 604 is clocked at 100 MHz and has an on-chip 2 x 8 kB cache. It is bridged to the PCI bus and interfaced to the DRAM (32 / 64 / 128 Mbytes), using the IBM Lanaï/Kauaï (IBM 660 Bridge) chipset.

A Multiple level write buffer is provided in order to minimize CPU stall cycles during writes to the memory and PCI bus. Coupled with single-clock DRAM access during cache refills, this minimizes the number of CPU cycles wasted due to stalls.

PCI bus

The PCI bus is a high-speed 32-bit bus driven by a 32 MHz synchronous protocol. The PCI bus provides a sustained 128 Mbytes/s bandwidth. The following potential masters are connected to the PCI bus: the Power-PC 603 / 604 and the DRAM via the Lanaï / Kauaï (IBM 660 Bridge) chipset, the Ethernet controller, the Sub I/O bus, the PCI extension, the VME interface and the optional VSB interface. These masters share the PCI bus through a central arbiter.

PCI mezzanines

The RIO2 8060 supports two independent PCI mezzanines on the PMC form factor. These PCI interface mezzanines can use the full PCI bus bandwidth (128 Mbytes/s). The RIO2 8060 has a front panel cut at the PCI interface board level to allow the connection of their I/O signals. A PCI extension connector is provided at the bottom of the card to extend the mezzanine number to 4 or 6 PMCs.

Global Memory

The on-board global memory provides storage ranging from 32 to 96 Mbytes and supports data parity. An extension piggy-back of 16 to 64 Mbytes can be added to the motherboard. The global memory communicates with the CPU via the Lanaï / Kauaï (IBM 660 Bridge) chipset. It is also interleaved on two alternate banks to provide the bandwidth required for maximum speed Power-PC 603 / 604 cache refills.

Flash EPROM

The Flash EPROM memories containing the bootstrap program, firmware and monitor program are composed of 8 Mbit Flash EPROMs allowing a capacity from 4 to 8 MB. The Flash EPROMs can be programmed on-board through 4 kB pages.

VME and VSB interface

The VME and the optional VSB external bus interfaces (VSB interface is implemented on one of the PMC mezzanines) both work in master and slave modes and have block transfer capabilities, 64-bit BLT for VME and 32-bit for VSB. The respective bandwidth reaches 80 Mbytes/s for VME and 40 Mbytes/s for VSB. A 64 x 32 bit deep FIFO offers optimal use of the PCI, VME and VSB bus by performing write posting and read prefetch. An independent DMA logic is associated with the bus interfaces and the 64 x 32 bit deep FIFO allowing very high-speed linked-list data transfers. The slave interfaces allow the mapping of the internal resources (DRAM, FIFOs, CSR, etc.) through size-programmable windows (16 MB to 128 MB), addressable from the VME and VSB buses.

VME and VSB interfaces are identical with those of the RTPC 8067

VME Slot 1

The Slot 1 controller can be either enabled or disabled, as a single VME Crate may contain several RIO2s, each one running a different embedded application. It supports ROR, RWR and Fair and Hidden requester modes.

Remote External Reset

This is a mandatory requirement for distributed systems; it has been implemented on the front panel of the board (LEMO 00).

2.3 Inter System Couplers (Data Movers)

In distributed real time architectures data is first reduced at the local level, and then a common set of global parameters has to be maintained through the entire system. Furthermore this data has to be refreshed at the right frequency to provide a globally coherent image.

Several solutions have been tried up to now, the most popular ones being:

- transparent memory-mapped access to remote resources (VME to VME repeater, VIC bus, ...)
- reflective memory architectures

They are offered either as alternatives or additions to memory-mapped connections (VIC bus).

These solutions provide an excellent access time which is close to the data transportation time, as none of the data encapsulation needed in network solutions is required. They provide an acceptable bandwidth.

Reflective memories however are not well adapted to the power of the real time processors, as they are not able to present data which has not been preselected, and modification of the data base, if possible, is a lengthy process.

To benefit from the new processing power of RISC processors, a new concept including more possibilities and setting a new standard had to include:

- data transportation should occur at the same speed as on the local backplanes (50 to 100 MB/s)
- data transportation should not interfere with the acquisition and processing of the local units.

i.e.: Max. Data acquisition
AND
Max. Data reduction
AND
Max. Data transportation

capabilities available at the same time.

- data transportation should combine the flexibility of memory mapped connections with the features of reflective memory and dynamic data base reconfiguration. The Fast Data Link provides all these features. The table below gives an overview of the FDL relative to other solutions.

	Speed (Mbytes/s)	CPU occupancy (%)	Event Trigger	Global time
VME to VME	5-10	10-60	Yes	No
Reflective Memory	1-15	10-60	No	Variable
FDL	15-40	0-5	Yes	Yes
FDL +	40-100	0-5	Yes	Yes

The speeds of data collection, local data processing, data transmission and data distribution need to be **matched** and achieved by an **independent** agent which delivers the data to the processing units.

The Fast Data Link concept embeds 3 different agents:

- a data gathering agent
- a data transfer agent
- a data scattering agent

The FDL is available on 2 different busses: VME (FDL 8050) and PCI (FDL 7217). Explanations will be given with reference to the VME implementation.

The data gathering and distribution methods and the software aspects related to system architecture are described here. For a more complete view see, “FDL, a Deterministic 100 Mbytes/sec Data Link” by J. Bovier and J-F. Gilot, paper M3A-b of these proceedings.

The Data Gathering Mechanism

The FDL 8050 is equipped with a fully block-capable D64 VME Master Interface which operates as both a high speed VME Master and a VME Slave interface. It is used to download the acquisition list as well as to read the status of the Fast Data Link.

Data gathering can be done as a succession of single shot transfers, a chained list of block transfers or any combination thereof.

A complete VME crate can be read in a single operation in the sparse data scan mode. The FDL 8050 is equipped with a detection mechanism for an end of block, with automatic skip to the next block of data. Sparse Data Scan (SDS) operations are therefore possible at very high speed. The complete crate can thus be read in a single operation at a rate exceeding 50 MB/s.

The list of VME transfers can be downloaded via the VME Slave port for dynamic operation, or can be stored in a local EPROM.

Trigger Conditions

Data gathering is extremely flexible, as the most common trigger modes have been implemented:

1. External Trigger

The FDL 8050 is equipped with 4 external trigger lines to start the data gathering upon reception of external stimuli coming from the equipment.

2. Synchronous Acquisition

Data gathering can also be started by a time base which is common to all FDL interfaces on the network and can be programmed with a resolution of 1 µs. The FDL maintains a central clock so that all data acquired within the system are correctly aligned in time.

3. VME Request

Data gathering can also start upon receipt of a local VME request which can be either an interrupt or a write to a special register.

Data Distribution Mechanism

In the same way that a data acquisition list can be introduced in the Fast Data Link, a data distribution list can be programmed according to the client / server concept. Data can be stored into any VME resource which is located in the receiving crate and described by an address and address modifier combination. Again this process takes place under control of the VME Master port of the FDL. The local distribution speed exceeds 50 MB/s if the Slave resource is equipped with an optimized VME Slave interface. An interesting possibility which results is the on-line selective distribution of data. The on-board processor in charge of the VME distribution can be programmed to associate the arrival of given data with a distribution mechanism.

User Interface Library

The Fast Data Link concept is a set of hardware and software tools. The hardware vehicle in the VME environment is the FDL 8050. Now we will describe the software tools. They can be divided into 2 groups:

- on-board firmware
- user level library

The on-board firmware is fixed and cannot be modified by the user. The firmware of the VME side CPU (VCPU) can eventually be finely tuned for a special VME readout or distribution sequence. The firmware of the Link side CPU (RCPU) cannot be accessed as it controls all the network functions. The on-board firmware delivers all the entry points on the VME and on the Link side to control the Fast Data Link operation through a user interface library. This library is written in C and can be linked to user programs developed on the most popular UNIX and real time operating systems.

FDL System Applications

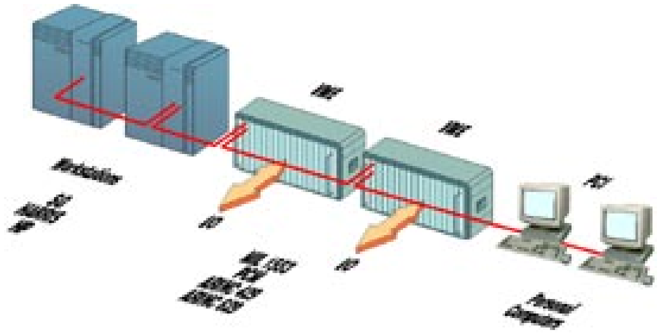


Fig. 4. Typical Aerospace Application

2.4 The Network Couplers

Three types of network connections are supported:

- Ethernet (TCP-IP)
- FDDI (TCP-IP)
- ATM (under test)

TCP-IP Networks have several functions. They are used to transfer the programs from the development stations to the target platforms (TCP-IP on Ethernet). They are also used as a protocol layer under NFS and FTP to connect any X-Window terminal to one of the development stations and to provide access to a unique development disk shared by all the development stations.

Ethernet is provided as a basic function to all the processors - either development or target. FDDI is used as an optional interface in the form of a PMC on both platforms. It provides not only a faster throughput than Ethernet but also a more stable one as collision avoidance mechanisms are included in the protocol. ATM is being tested as a the next step after FDDI but with extensive modifications to include a flow-control mechanism, a mandatory requirement for real time applications where two basic criteria must be respected:

- no loss of data
- a guaranteed worst case performance

CES has developed a PCI ATM interface corresponding to these requirements. It takes benefit from the latest ATM chip set available and incorporates not only a fiber optic standard SONET connection but also a parallel copper connection equipped with a flow control mechanism (Xon-Xoff type). The flow control connection can be used either in direct connections or in switch based connections with an ATT Phoenix switch.

3. CONCLUSION

This real-time system architecture has been successfully tested in different large control applications. It provides a significant improvement in the overall performance of the systems together with a much simpler operation. The stability of the system has also been significantly improved as all key elements are pre-packaged to work together. Future extensions are by the scalability of every component of the system.