

PVIC: a transparent full speed PCI to PCI Connection

The PCI bus is widely used as a local bus for both VME / CompactPCI™ processor boards and desktop workstations. The PVIC concept has been developed to allow PCI based processors to be interconnected into clusters on a large distance as well as to allow PCI based I/O cards to be transparently addressed by remote processors. The design concept offers two operating modes, which can be used concurrently: one is a completely transparent access via a memory-mapped architecture using a flexible source and destination page-based memory allocation system. The other one is a full set of advanced features aimed at reducing the transfer and handling time of data. These features are implemented in hardware and include: on-board chained DMA controller, global shared memory functions, intelligent broadcast and multicast cycles as well as global semaphores. These mechanisms will be described and application examples will be discussed.

INTRODUCTION

The PVIC [1] was designed to provide a transparent memory mapped interconnection between modern computer buses like PCI, CompactPCI™ and VME64x with the following requirements :

- Transparent bus to bus interconnection.
- Embedded multi-channel DMA controller for high bandwidth.
- Autonomous interrupt dispatching.
- Hardware semaphore for distributed processing.
- Multidrop up to 15 node.
- Nodes distributed from 1 up to 200 meters.
- Scaleable bus protocol (66 to 532 Mbytes/s).
- Multicast and broadcast.
- Bridging between different physical interfaces.
- Mirrored-Memory up to 2 Mbytes (local read and global write).

The PVIC Bus protocol has been defined to support these requirements.

PVIC INTERFACES

All PVIC interfaces are built of the same functional blocks as shown in the figure 1. The basic blocks are interconnected with dedicated bus protocols. The PIB bus is used to drive the PVIC physical interfaces (GTL+, DIFF or ODL) and the PCB bus synchronises the buffer management between the PVIC and the local bus.

The PVIC physical interfaces are implemented on daughter-boards.

The PVIC bridge capability is controlled by a specific "PVIC front-end" logic driving two independent PIB buses. This allows to interconnect transparently different PVIC physical interfaces like GTL+, DIFF and ODL. The PVIC bridge is built around fast CMOS switches and a bi-directional FIFO. Fast CPLD is controlling the data direction and the PVIC arbitration process. The PVIC bridge introduces 4 PVIC clock cycles latency on a PVIC transaction.

PVIC PHYSICAL INTERFACES

Three PVIC PIB daughter boards have been defined:

1. The PIB 6800 daughter card adapts the local TTL PVIC bus interface (PIB) to GTL+ PVIC. It can be used for simple point-to-point PVIC connection of up to 2 meters or for multi-point connections on a front-panel 0.025" ribbon flat cable bus of up to 1 meter length and 15 PVIC nodes.
2. The PIB 6801 daughter card adapts to DIFF PVIC. It can be used for multipoint PVIC connection of up to 20 meters and 15 PVIC loads.
3. Finally the PIB 6802 daughter card adapts to Optical PVIC. It can be used to interconnect PVIC subsystems at distances up to 200 meters. The bi-directional optical links includes special transactions for the PVIC arbitration process. The serialisation & deserialisation is handled by the proven HP GLINK chip-set.

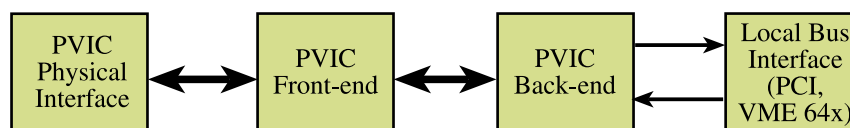


Figure 1.

PVIC PROTOCOL & TRANSACTIONS

The PVIC bus protocol uses a source synchronised packet transmission. The PVIC read is handled with a splitted transaction: read request and read acknowledge. The PVIC implements the following bus cycle types:

Write	Fully acknowledged single cast node to node write
Write_Broadcast	Fully acknowledged broadcast write
Read_Request	Read request to selected node
Read_Response	Read response back to node
Write_MM	Broadcast write with no handshake to mirrored Memory
Configuration	PVIC bus configuration

Table 1.

LOCAL-BUS TO PVIC TRANSACTION

The local-bus to PVIC bridge is controlled through an out-going scatter-gather (see figure 2). This mechanism similar to a MMU allows to remap completely the local addressing scheme to the PVIC addressing. The PVIC interface uses a local-bus PCI memory space window of 32 to 256 Mbytes. This window is tailored in 64 Kbyte pages (4096 * 64K = 256M). Each individual page is associated to a page descriptor defining the PVIC out-going parameters. The out-going scatter-gather page descriptors are stored in the first 16 Kbytes of the SSRAM.

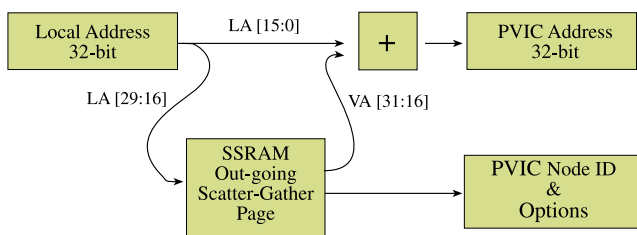


Figure 2.

PVIC TO LOCAL-BUS TRANSACTION

The PVIC slave interface decodes and receives the PVIC transactions (see figure 3). The decoding is handled only through the node-id information. Each PVIC slave owns a unique node-id. PVIC broadcast is also decoded if node-id = 0. The PVIC slave logic is using only the PVIC clock which could be asynchronous related to the local time domain. The time domain translation is done through a Dual Port SRAM (DPRAM). The PVIC slave bus-to-bus MAPPED transaction is identical to the previous one except for the local-bus addressing. To simplify the user interface on UNIX like OS, an in-going scatter-gather was added to remap the PVIC address in 4 Kbyte pages. This allows the fast and direct fill of a user buffer consisting of a chain of memory blocklets.

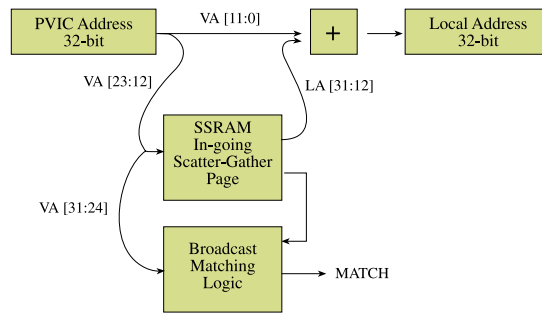


Figure 3.

The PVIC slave bus-to-bus MAPPED interface is also incorporating a filtering mechanism to support "Selective Broadcast" or "Multicast". This filtering mechanism is built with an AND-OR match logic on an 8-bit field:

Match IN field	PVIC Address [31:24]
Match LOCAL field	In-going scatter-gather [7:0]

Table 2.

The PVIC slave bus-to-bus MAPPED write transaction must have successful MATCH to be processed on the local PCI bus. In case of unMATCH the cycle is rejected. The match logic is applied only for write. The 4096 page descriptors of the in-going scatter-gather are located in the second 16 Kbytes of the SSRAM.

PVIC EMBEDDED DMA CONTROLLER

The PVIC incorporates a DMA controller (DMAC) which is integrated into the back-end FPGA (see figure 4). It provides the following features:

- Local to PVIC DMA.
- Support for only READ local, WRITE PVIC.
- DMA chaining in the SSRAM (128 Kbytes).
- 4x 32-bit word chaining-block.
- 32-bit local address, 32-bit PVIC address.
- DMA can be controlled remotely through the PVIC bus.
- DMA interrupt dispatching over PVIC.

The local DMAC supports only local read and PVIC write but a DMA in the reversed direction can be performed by programming and controlling the remote PVIC through the PVIC bus. The DMA initialisation is done via DMA chaining-blocks located in the local SSRAM. Many chaining-block lists (DMA chains) can be built. The DMA process is started by writing the SSRAM address of the first chaining-block into the DMA Request FIFO and setting the DMA-start control bit. The DMA Request FIFO allows to post up to 64 DMA chains by loading the FIFO with further chain start addresses.

A DMA sequence is decomposed in two steps: First

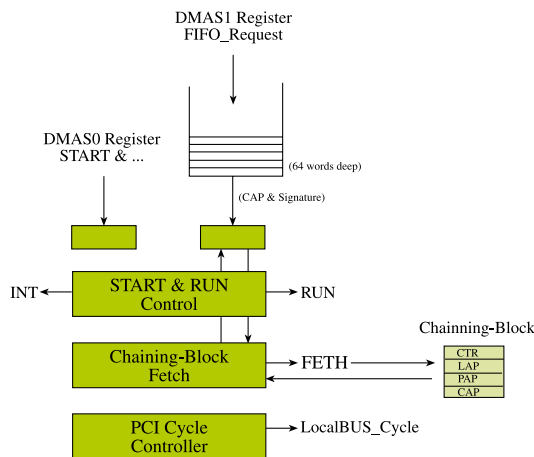


Figure 4.

the DMA chaining-block list must be build in the SSRAM and then the DMA chain execution can be started by writing to the DMA start register. These two steps are represented in the figure 5:

1. SSRAM chaining block(s) initialisation (Write).
2. Trigger the DMA controller by setting the DMA start bit and loading the DMA request FIFO with the first chaining-block address.

After the DMA controller has started, the DMAC will fetch the first chaining-block and begins its execution. The DMAC can be operated in two modes:

1. Single: the DMAC is executing a single DMA request stored in the FIFO.
2. Continuous: the DMAC is executing DMA chains as long as the DMA request FIFO is not empty.

If everything is correct (write-buffer free, word-count > 0, etc.) the local-bus interface is triggered to issue a local PCI read. The data read are immediately loaded in a local-write-buffer (LWB). At the end of the read transaction the DMAC is writing the header and the PVIC address in the current LWB. This sequence is rerun until the word counter reaches 0 or an error occurs. When the word counter reaches 0, the DMAC checks if the chaining should be continued or stopped. If the dmaCONT bit is set, the DMAC fetches the next chaining block and processes it. An end, suspend or error interrupt can be trigger individually on a per chaining-

block base.

The local bus occupancy is controlled by the PCI latency timer. This timer monitors the local bus used by the DMA and forces a disconnects when expiring. The DMA controller is using the same write-buffers (LWB) as for direct bus-to-bus transactions. An additional logic controls that at any time not more than 4 write-buffers are used by the DMAC. A special suspend logic allows to suspend running DMA transactions atomically. This guaranties that the source which has issued the suspend can also do the restart.

PVIC INTERRUPT CONTROL

The PVIC provides a mechanism to export transparently local interrupts or events to external PVIC nodes. On event detection (if unmasked) the local controller issues a High Priority Mirror Memory (RFMH) PVIC write transaction. This transaction can directly target the MM message FIFO. The lower 11 bits of the MM address accessed are written into the MM message FIFO. This allows the destination node to process all the pending interrupts efficiently by reading the interrupt status word at the MM address stored in the FIFO. Broadcast is supported as well. The following interrupts can be exported:

- Local PCI INTA#
- Local PCI INTB#
- Local PCI INTC#
- Local PCI INTD#
- Local PCI PERR#
- Local PCI SERR#
- Remote Write Error (RWERR)
- DMA Controller Interrupt

On PVIC8426 INTB, INTC and INTD are hardwired to the three front panel LEMO connectors. A 4 bit DMA signature allows to dispatch interrupts of up to 16 different DMA transactions at the same time.

The remote interrupt dispatching is defined with specific control-blocks located in the SSRAM. When one of these asynchronous events occurs, the interrupt controller executes immediately the associated control-block if a High Priority MM buffer is available. The interrupt dispatching has the highest internal priority.

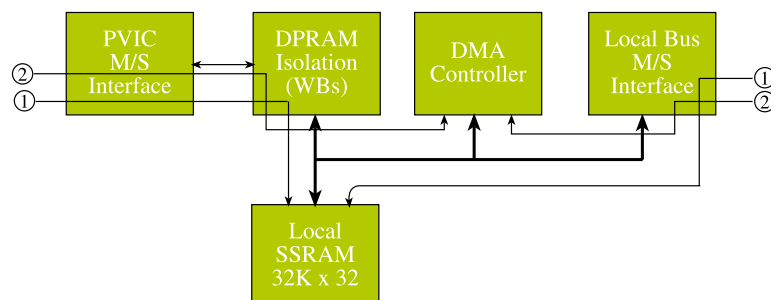


Figure 5.

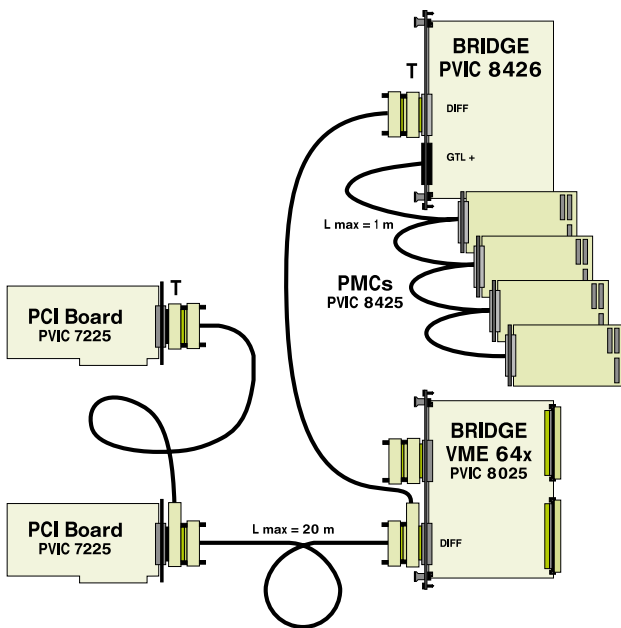


Figure 6.

PVIC MIRRORED MEMORY (MM)

The PVIC implements a local read and global write Mirrored Memory concept. The first part of the MM is reserved for the write and read buffers of the PVIC system. The rest is available for the user. Apart from the already mentioned PVIC message FIFO for efficient interrupt dispatching it also implements a global reservation mechanism. This mechanism allows the implementation of hardware semaphores, test-and-set operations and other atomic operations in a flexible and efficient way. This feature was especially added for distributed processing, farming etc.

PVIC IMPLEMENTATIONS

The following PVIC products are available:

- PVIC8426 CES RIO8062 [2] - PVIC Interface with PVIC Bridge.
- PVIC8425 PMC-PVIC single slot PMC with embedded GTL+ physical interface.

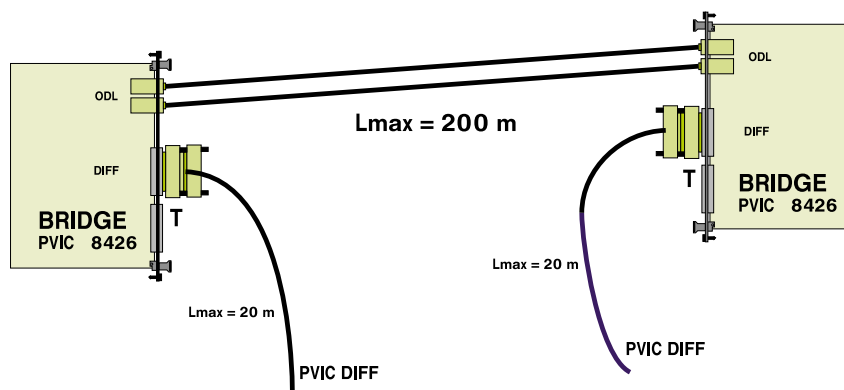


Figure 7.

- PVIC4025 CompactPCI - PVIC System Slot or Peripheral Slot (3U and 6U versions).
- PVIC7225 PCI - PVIC 32-bit 33 MHz.
- PVIC8025 VME64x - PVIC interface with PVIC Bridge.
- PIB6800 GTL+ 0.025" ribbon flat cable, up to 2 meters at 66 MHz.
- PIB6801 Differential Interface, 68pin SCSI cable type, up to 15m at 66MHz or 30m at 33 MHz.
- PIB6802 Optical Link up to 200 meters, 1.4 Gbit/s.

PVIC SYSTEM EXAMPLE

Figures 6 and 7 show two typical PVIC system examples. It is possible to mix PVIC DIFF with PVIC GTL+ through bridge elements. The PVIC bridge is only translating the electrical levels. The PVIC protocol remains identical and the system remains completely transparent.

REFERENCES

- [1] CES, PVIC 8426 Technical Specification.
- [2] CES, RIO2 8062 User's Manual. DOC 8062/UM ver 1.0, Oct 1997. ■

Joel Bovier has been Director of R & D at CES for 16 years. He graduated from the Ecole d'Ingenieur de Genève in Switzerland.