By Kim Vu,
Vice-President of Engineering,
Solflower Computer, Inc.

CP1006VME - A merging technology of CompactPCI & VMEbus in a single chassis supports Solaris 2.x and Solaris X86

This paper described a solution for application where both busses - CompactPCI and VMEbus - need to be merged to create a system with features to support 'old' VMEbus hardware and the newcomer CompactPCI peripherals.

The complete system is integrated in a single 19-inches rack-mount chassis and provides seven CompactPCI and six VME slots. Even VME64 specification does not support hot swap, users can remover or insert VME hardware without having to power down and reboot the CompactPCI system. This combination can be uses for CompactPCI SPARCengine, or standard Pentium CompactPCI single board CPU running Sun's Solaris 2.6 operating system.

NEW BUS MEETS THE OLD WORLD

CompactPCI is a modern, very high performance industrial bus based on the standard PCI electrical specification and built on a rugged 3U or 6U Eurocard packaging. CompactPCI, for all its advantages and benefits, still lags behind VMEbus because of its limited the number of peripherals available for many standard application. Many system integrators have difficult time to find the right CompactPCI interfaces for their CompactPC

In fact, system integrators don't have much choice when they select CompactPCI interfaces, and currently many of VME boards have features that aren't replicated in the CompactPCI world, In this case system integrators may go for a dual-bus architecture solution.

Combining the two buses can solve the problem: CompactPCI and VMEbus. Why not CompactPCI and Multibus? The answer is CompactPCI and VME buses have the same 6U form-factor. In fact, CompactPCI shares the VME mechanical attributes since CompactPCI was born. In addition, VMEbus has been proven to be an outstanding technology since its inception over 15 years ago, still with plenty of growth potential. A combination of CompactPCI and VMEbus promises a perfect marriage for industrial applications and provides system integrators a window to a large and diverse system market.

THE DUAL-BUS SOLUTION

Based on the same form-factor and mechanical component is a good way to start to combine these two busses. However there are many issues that need to be resolved such as: PCI-VME bridge hardware design, chassis with dual backplane construction and the software driver to support VME peripherals, and the challenge goes beyond just putting two buses together. Solflower has addressed all of these issues and developed a solution that is low cost, easy to use, and in most cases, plug and play. It is designed to be compatible with all current VME hardware and software peripherals.

PCI_VME bridge hardware:

Tundra Semiconductor has designed the Universe II and it seems to be the completed PCI-VME in a single silicon chip. Only few components such as buffers and oscillators need to be added to build the complete PCI-VME Bus Bridge.

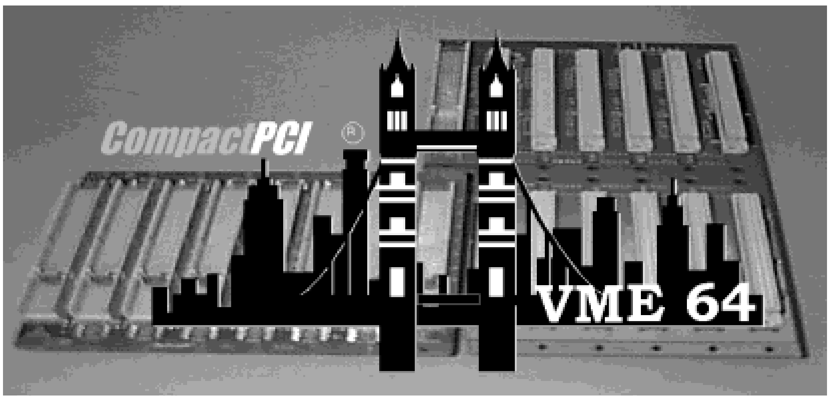

Figure 1. The Universe Bus-Bridging chip provides 64 bit data transfers between VME and PCI bus.

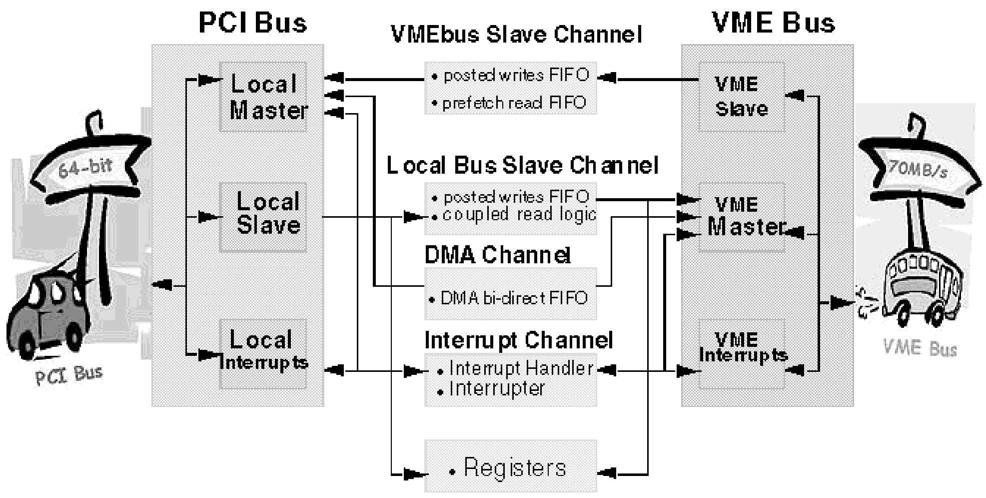

Figure 2. Tundra's Universe block diagram

The original architecture and functionality of the Universe is based on Tundra's SCV64 controller, therefore we can consider that the Universe can be used without headache to develop a dual-bus system. Figure 2 shows the block diagram of the Universe II chip.

The Universe Bus-Bridging chip provides 64 bit data transfers between VME and PCI bus. It includes onboard DMA engine, which can move data across the buses up to 50MB/sec. Complete Bus-Bridge with the associated buffer logic are laid out in a single 6U VME card without any cabling or daughter card attached.

Chassis for integration:

The dual bus system consists of a 3U Compact PCI backplane; a 6U VME backplane and the Solflower single 6U Bus-Bridge card. All of these components are mounted in a 19" standard rack mount and provide seven CompactPCI and six 6U-VME slots.

The 64 bit CompactPCI bus J1/J2 are placed on the bottom of the board, versus VMEbus J1 being positioned at the top and J2 at the bottom. To merge these two buses, the only one choice is to use the 2mm connector at the position J4 and J5 on the top of CompactPCI to map the VME signals. The specially designed VME backplane provides six slots for standard 6U format and one 220 pin 2mm connector, located on the left, to adapt the J4/J5 signals and remap the VME signals on P1 and P2. Finally the two buses can be mounted together in a same 19" rackmount chassis, providing seven slots CompactPCI on the left side, and six VME bus on the right side.

Figure 3 shows the combination solution of dual bus system. On the right side is a standard CompactPCI backplane, which has slot 1 on the left and the last slot on the right, where the CP1006 Bus Bridge resides. VME bus continues after the Bus Bridge board with slot 1 as system slot and the last as slot 6. This dual-bus architecture offers greater flexibility in mixing and matching the latest CompactPCI technology and the popular VME peripherals.

Software support

The Bus Bridge doesn't bring a great deal of application without software support. Any system built with the Bus Bridge must consider a degradation of performance, or the application software needs to be changed, unless the Bus Bridge is supported from a device driver, which makes transparent to the applications.

The CP1006 is designed to meet these goals: no degradation of performance and no application software change. It is capable of supporting Solaris's applications by providing plug-and-play for all existing VME Sun4 device drivers.

Many technical issues arise and need to be discussed when an integrator wants to have a Sun4 VME application run on the new Sun PCI-based SPARCengine CP1200, CP1500 or an Intel x86 Single Board Computer. Fundamental differences are address

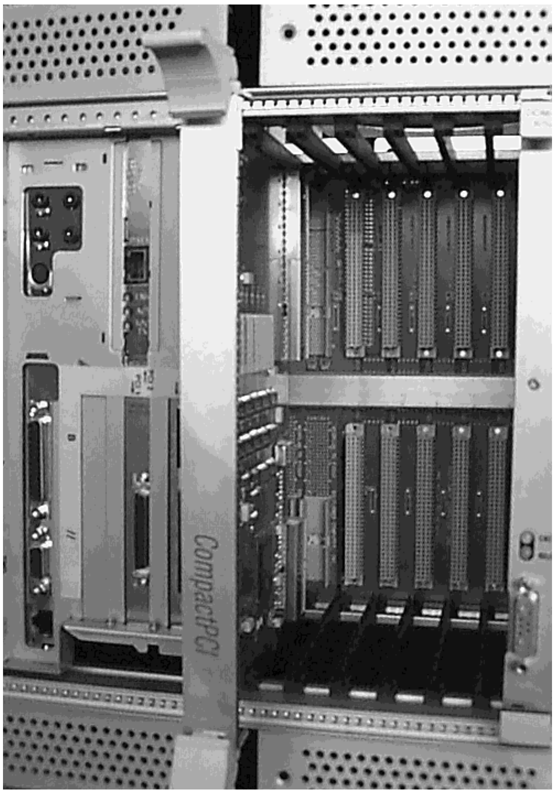

Figure 3. Dual Bus system without cable connections

spaces mapping; interrupt handling and DMA structures and the ENDIAN of the Intel x86 and the SPARC architectures:

In PCI, there are three types of address spaces: I/O, Configuration and Memory spaces. VME bus, on other hand, has address spaces different from each other by decoding AM codes. All of these can be covered by hardware - the Universe, but to manage the size of mapping space, it needs further consideration. According to the PCI specification, each PCI device should have an ID, base address and the length of occupied spaces. If the Universe wants to cover all the VME address spaces, then no more address space is available for other devices on the PCI bus. Furthermore, if the VME bus has already been mapped to the PCI space, but doesn't use all those spaces, it will be big waste resource for the whole system. PCI space is big (in the range of 1Gig to 2Gig), but it's always being shared with other devices, including the on board devices, and the space assignment is dynamically controlled from the Solaris kernel. If the space was not allocated correctly, the system could unpredictably fall into a panic mode.

The CP1006 nexus driver is designed flexible enough to map all the spaces when (and only when) the application needs, and adjusts the size to fit the empty space on the PCI bus. The following example shows how the CP1006 driver works:

When the CP1006 driver is installed, it examines the PCI configuration space to find out which addresses are already occupied and thus how much memory is available for accessing VME bus devices. If other PCI devices are installed in the same PCI space, the amount of PCI space available for VME bus access is reduced. However, the CP1006 driver can, in many cases, work around holes in the PCI memory space caused by other devices. The CP1006 driver accomplishes this by using up to 8 slave image windows for mapping address ranges on the VME bus.

Fig. 4 shows the CP1006 maps to the free PCI spaces. If VME device call the un-map function, the PCI space becomes free again. In this example, the CP1006 PCI-VME Bridge shares PCI bus with a PCI video card and on board Ethernet/SCSI interfaces.

The CPU I/O assigns the PCI physical addresses for accessing VME bus address space by using hardware base and bound registers. These base and bound registers permit CP1006 to define space on the PCI bus for its use that did not show up when the device was initialized by the Sun Open Boot PROM. The Solflower's CP1006 Bus Bridge occupies 4KBytes of space in the PCI memory space, and after initialization, the CP1006, in essence, becomes "bigger" than it used to be. There is no way to tell the Open Boot PROM about this changed condition.

For this circumstance, the CP1006 driver can exclude a certain range of PCI addresses for mapping, because it may happen that another CompactPCI card which resides in a certain PCI memory range, but doesn't tell the Open Boot PROM via its PCI configuration base register.

After the CP1006 driver is loaded, VME driver can be

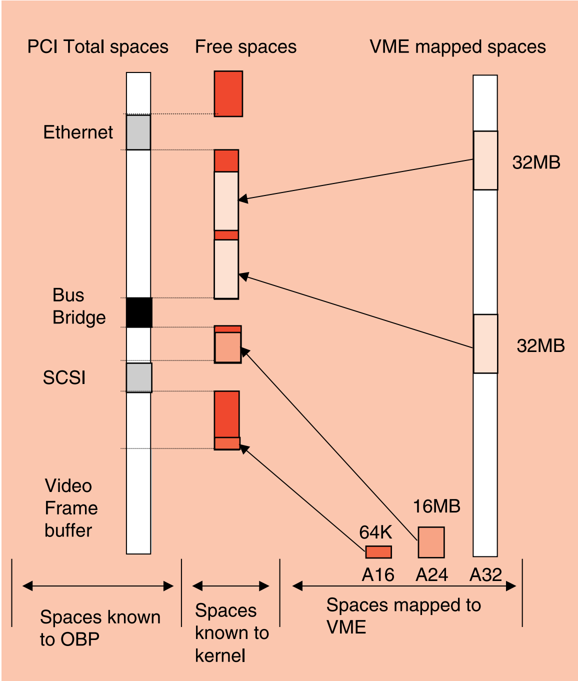

Figure 4. CP1006 maps 3 VME spaces to the free PCI spaces

attached to the nexus CP1006 as a leaf driver, by specifying in the configuration file and setting the parameter "parent=pvme" (pvme is the device driver name of the CP1006 Bus Bridge)

The CP1006 features two types of DMA transfer called slave DMA, which dma transfer is controlled from dma engine on the VME board. This mode allows the most common written Sun4's VME driver to work with the new PCI-base machine, no change, period. The second DMA mode is initialized from on board DMA engine, which resides in side of the Universe chip. The features such as linked list DMA and programmable transfer length provide the most performance can be archived through the Bus Bridge.

The byte-ordering issue exists due to the differences between the VMEbus and the Intel x86 processor architectures. It requires special hardware to avoid byte-ordering conflicts because the Universe doesn't covers this endian conversion. For example, if a Pentium X86 is on the CompactPCI system slot, and a Motorola CPU on the VME side, during data transfer across the VME bus, the data at the lowest byte address would be assumed to be the least significant, while it is actually the most significant. However this poses no problem if the 32bit data written and read by the same type of processor architecture, since the swapped data gets swapped again. For this reason, byte swap logic is available as an optional to correct this endian conversion issue.

HOT-SWAP - HOT TOPIC TO DISCUSS

Compact PCI specification R2.1 allows hot-swap for CompactPCI interface, It involves both hardware and software to maintain the continuing system operation. VME doesn't have that feature, but the CP1006 allows VME cards in the VME sub-bus be inserted or removed without shutting down the CPU system. Even though this is not a full hot swap feature as defined in the CompactPCI spec, but it allows the system integrator to replace the VME interfaces, while Solaris is still running on the CompactPCI bus. This saves a great deal of time since no CPU power down is required or Solaris reboot is necessary. This hot swap process is quite simple and requires only that the driver to be unloaded. The power of the VMEbus is turned off in an orderly fashion before it can be removed from the system. When the VMEbus is powered back on, the driver can be reloaded with a single command and the application can run again and the best part of this: the user does not pay extra for this hot feature. It is a standard feature and ready to use.

CONCLUSION

CompactPCI is seen as a compliment to VME bus. Application such as industrial control and/or medical instrument can be solved with both, CompactPCI or VME components. Integrators prefer to use CompactPCI solution where the hardware and software is available, but also need VME interfaces where CompactPCI device is not yet available. Bringing the best of the two worlds together is a perfect solution and to create a solution that economical and easy to use is every Engineer's dream. Solflower did it with the CP 1006 and definitely answers the questions that every integrator asks:

" How can I integrate my VME interfaces on the new CompactPCI system without any expensive engineering work?"

The solution we've just discussed can be applied as well to other platforms such as Window NT or VxWorks. The same hardware bus bridge and chassis construction can be used without change. Bus bridge driver software, which is available from several vendors, needs to be obtained to support the complete system. ■

REFERENCES

- VME64x/2mm on CompactPCI PICMG 2.2, Sept, 9 98
- CompactPCI Specification PICMG 2.5 R1 April 1998
- SPARCengine CP1200/1500 data sheet, Sun Microelectronics 1998
- PCI_VME User's and Installation Guide, Solflower Computer, Inc.1998.

Kim Vu has been with Solflower since 1990. Previously he was staff design engineer for Sun Microsystems, where he was responsible for the development of VME boards. Until 1986 he work for System Industries and Data System Design, designed storage interface compatible to Qbus and Unibus. He has a Diplom degree in Electronic from the Polytechnic in Germany.