VMEbus FAQs
Frequently Asked Questions for the Beginning VMEbus User

John Rynearson, VITA Technical Director

Question: Every VMEbus system must contain a system controller. What is the system controller and what does it do?

VME - A Multimaster Bus— VME provides the flexibility of being a multimaster bus. A VME system may have one, two, or up to 21 masters in a full 21 slot backplane. Each master has the ability to request the bus and carry out a data transfer activity. Modules that do not have master capability are called slaves. Each module that plugs into the backplane can be designed to be a master, a slave, or both. Masters have the ability to request the bus and put addresses on the bus. Slaves do not. They just “watch” the address bus and respond by either reading or writing data if an address on the bus matches their address.

Because the VMEbus is a shared resource, however, a mechanism must exist to allow boards to arbitrate for and be granted the bus. This mechanism is called arbitration and is carried out by the VMEbus system controller. The system controller always resides in slot one and provides functionality independent of the module which is plugged into slot one. That is, a master or a slave may reside in slot one along with the system controller functions.

Other Functions - SYSCLK and IACK Daisy Chain Driver— Besides bus arbitration the system controller provides a 16 MHz system clock (SYSCLK) and the interrupt acknowledge (IACK) daisy chain driver. The 16 MHz system clock is provided as a common utility. Since the VMEbus is an asynchronous bus there is no guaranteed relationship between the system clock and any other signal on the bus. Designers are free to use the 16 MHz system clock in lieu of using an on-board oscillator. However, while the system clock will always remain a part of the VMEbus specification, future extensions may restrict its use. It is recommended that designers not use the system clock for future designs.

The IACK daisy chain driver is a simple circuit which insures the proper timing relationship between IACK and the IACK IN/OUT signal during interrupt acknowledge cycles.

The DTB Arbitration Bus— The VMEbus is made up of four sub buses. One of the sub buses is the data transfer bus (DTB) arbitration bus. This bus contains four bus request signals (BR3* through BR0*), four sets of bus grant in/out signals (BG3IN*/BG3OUT* through BG0IN*/BG0OUT*), a bus clear (BCLR*) line, and a bus busy (BBSY*) line. The BBSY* signal is asserted by any master that has the bus. Masters give up the bus by deasserting BBSY*. (Note that the asterisk after the signal name indicates that the asserted level of the signal is low.)

The arbitrator on the system controller can be set up in one of three arbitration modes: priority (PRI), round-robin (RR), or single level (SGL). The arbitration mode selected is part of the system initialization process and is static during the life of the system. Priority mode gives highest priority to bus request level 3, then 2, then 1, and finally 0. Round robin mode grants the bus in sequential fashion to the next bus request in line with round robin ordering being - 3, 2, 1, 0, 3, 2, 1, 0, etc. Single level arbitration only honors bus request level 3. Bus request levels 2, 1, and 0 are ignored. SGL is provided for simple systems that don’t require a sophisticated arbitration mechanism.

The VMEbus specification allows the arbiter to operate with bus request level 3 set to PRI mode which bus request levels 2,1, and 0 operation in round robin mode. Similarly, bus request 3 and 2 can operate in PRI mode with 1 and 0 set to RR mode.

Acquiring the Bus— If a master wants to use the bus to transfer data, it must first request the bus. It does this by asserting one of the four available bus request lines - BR0* through BR3*. Depending on the arbitration mode and current bus activity, the arbiter may or may not take action. For example, if the arbiter is operating in PRI mode and the current bus master has acquired the bus at BR3*, the arbiter will wait until the current master gives up the bus by deasserting BBSY* before granting the bus to another master. On the other hand if the current master has acquired the bus are BR2* and a master requests the bus on BR3*, then an arbiter running in PRI mode will assert the BCLR* signal indicating to the current master that a master of higher priority wants the bus. The VMEbus specification does not mandate how long a master can keep the bus even after it receives a BCLR* signal. This decision is left to the system designer. The VMEbus specification only offers the advice that for prompt real-time response the current master should give up the bus as soon as possible.

Priority or Round Robin— The choice between priority mode and round robin mode is application dependent and is left to the discretion of the system designer. PRI mode is usually best for a system that must service events in a hierarchical fashion and where the service of specific events must be completed before bus mastership is given up. RR mode is usually best for a system where all masters perform tasks that are of equal priority and the servicing of events can be arbitrarily started and stopped without a problem.

Conclusion— The VMEbus provides a flexible scheme for bus arbitration that allows the system designer to meet many diverse application requirements.