VME DAQ – Rates and Restrictions

Data transfer using VME related hardware is expected to be the primary method by which front-end modules are configured and read out. These include both commercial VME modules as well as custom-designed TDC and FADC boards from JLAB and other collaborating institutions. Recent extensions to the VME specifications (e.g. VME64x) have defined a number of block data transfer protocols that can be implemented.

In general there are now 5 access protocols that can be used to transfer data from VME slaves. They are listed below along with their theoretical limits.

Protocol	Description	Transfer speed limit (MB/s)	Minimum transfer size
SCP	Single Cycle I/O	10 (write) 5 (read)	1 byte/module
BLT32	Block Mode (32 bit)	40	4 bytes/mod
MBLT	Block Mode (64 bit)	80	8 bytes/mod
2eVME	Dual Edge (64 bit)	160	16 bytes/mod
2eSST	Dual Edge – source synchronous (64 bit)	320	16 bytes/mod

In practice most of the above rates are not achievable, however we have some experience from commercial implementations that give a general idea of what we can expect for our applications. First, there is a commercial VME to PCIX bridge chip (Tundra Tsi148) that is now commonly found on many VME-based single board computers that implements all of the above transfer protocols. This chip provides two semi-independent DMA engines that can be programmed to initiate transfers between CPU memory and the VME bus. The speed of transfer is primarily independent of operating system (i.e. VxWorks or Linux), but can be highly dependent on the VME slave design and block transfer sizes.

For example, and internal Motorola application note measured DMA transfer performance using 2eSST transfer modes between two MVME6100 CPUs using the Tsi148 chip. Below the results are summarized:

Buffer Size	Write (MB/s)	Read (MB/s)
64KB	239	238
128KB	248	246
256KB	252	250
1MB and >	256	254

These results represent a practical upper limit on what we might expect to achieve with VME transfers.

There are a limited number of commercial boards that have implemented all the available block transfer modes. One board that has shown the highest performance so far is the SIS3320 FADC from Struck. Below are results of tests done at JLAB using an MV6100 as the master:

SIS 3320 block transfer speeds	
BLK32	27MB/s
MBLK	55 MB/s
2eVME	110 MB/s
2eSST	132 MB/s

An important consideration for making the DMA block transfers from a crate full of modules as efficient as possible is to minimize the overhead of the setup and execution of the transfer itself. In general there are two ways to do this. First is to use a feature of the DMA engine of the Tsi148 to set up a linked list of multiple DMAs to perform – essentially one for each board in the crate. This can be complex to set up as it requires knowledge of source and destination addresses and byte counts for each transfer. If slave terminated transfers are used (i.e. bus rrrors or 2eSST), then one cannot ensure a contiguous block of data at the destination (It would require a subsequent copy to a final buffer). The second method is to provide a token-passing scheme between modules in the crate to make the transfer from multiple modules look as though it comes from one. This is also called "chained" block transfer. The last module in the chain then terminates the transfer. This is the most efficient, but it requires that all modules to be placed in the crate appropriately and the required hardware in place to support it.

Gluex DAQ Requirements

One issue that must be considered is if the DAQ requirements of the GlueX detector can be handled by the VME backplane. If we consider the 200 kHz trigger, the event processing time is only about 5μs which is on the same order as the interrupt latency for the CPU. Therefore we must block up events and readout data only periodically. A workable practical limit for readout frequency is about 1kHz which translates to 200 events/block and less than 1 ms for readout.

To be conservative even at 200 MB/s a practical limit on the amount of data that can be extracted from an entire crate within the .001 second readout interval is probably only about 160kB. For ease of calculation we assume 16 modules per crate you get 10 kB/module for a 200 event block. So

10 kB/module / 200 events = 50 bytes/module-event ~ 12 words/mod-event

(where 1 word = 4 bytes). Care should be taken as to the use of headers and trailers for events generated in the slave modules. VME deals in primarily 32 and/or 64 bit quantities, so event data should be packed with this in mind.
(Note: The 2e transfer modes are restricted to a 4 word/module granularity, meaning that each module if it generates any data must pass at least 4, 8, 12 words, etc… even if some of the data is "padding" or dummies.)

One concern is the number of available channels per crate, and the expected channel occupancies. For example, the JLAB Flash ADC is 16 channels, so a full crate has 16x16 = 256 channels. If we have a minimum of 2 words/channel for the data (both time and energy) then at 200 kHz we can handle 5 channels hit per module or about a 30% occupancy. This should not be

an issue. However, for the 100 Mhz Flash at 72 channels per module has up to 18 modules x 72 = 1296 channels/crate. If one allows only 5 channels/module hit in this case then the maximum occupancy that can be handled is only 6-7%. This is potentially problematic, and it may require some module firmware support of flushing data for a given event if it crosses a programmable limit.

As a final note one cannot completely ignore OS related issues. The vast majority of our experience to date with VME access and data transfer has been with vxWorks. This environment certainly provides the most efficient interaction at the hardware level. All VME access is directly memory mapped within the OS. For linux there is the necessary interface of the virtual memory of user space and the kernel that involves some overhead (in memory copies). But, the Tsi148 is now available on VME SBCs with powerful muli-core Intel processors that can run the latest versions of the linux kernel with real-time extensions. There is plenty of post transfer processing power on these platforms to handle the data. Of course a single gigabit Ethernet link will only offload the processed data from the crate at 100-120 MB/s but the faster we can get it into the CPU from the VME bus the more time we have to handle it.

David Abbott
DAQ Group
F268 Cebaf Center
x7190