



CA91C042 App Note

Universe™ Performance Analysis and Performance Considerations

As a VME bus bridge, the Universe's most important function is passing data. This function resides in three of its channels: the PCI Slave Channel, VME Slave Channel, and the DMA Channel. Since each channel operates independently of the others and because each has its own unique characteristics, the following analysis reviews the data transfer performance for each channel. Where relevant, descriptions of some of the factors affecting performance and how they might be tuned in different environments are discussed.

The decoupled nature of the Universe can cause some confusion in discussing performance parameters. This is because, in a fully decoupled bus bridge the two opposing buses operate at their peak performance independent of the other. The Universe, however, because of the finite size of its FIFOs can never represent a 100% decoupled bridge. As the FIFOs fill or empty (depending on the direction of data movement) the two buses tend to migrate to matched performance where the higher performing bus is forced to slow down to match the lower performing one. This limits the sustained performance of the device. Some factors such as the PCI Aligned Burst Size and VME request/release modes can limit the effect of the finite FIFO size and boost performance.

Another aspect in considering the performance of a device is bandwidth consumption. The greater bandwidth consumed to transfer a given amount of data, the less is available for other bus masters. Decoupling significantly improves the Universe's bandwidth consumption, and on the PCI bus allows it to use the minimum permitted by the PCI specification.

To simplify the analysis and allow comparison with other devices, Universe performance has been calculated using the following assumptions:

As a PCI master : - one clock bus grant latency
- zero wait state PCI target

As a VME master: - ideal VME slave response (DS* to DTACK* = 30ns)

Assumed as part of any calculations on VME performance is the inclusion of VME transceivers with propagation delay of 4 ns.

PCI SLAVE CHANNEL

Coupled Cycles : Request of VMEbus

The Universe makes use of a "Coupled Window Timer" (CWT in the LMISC register), which permits the coupled channel to maintain ownership of the VMEbus for an extended period beyond the completion of a cycle. This permits subsequent coupled accesses to the VMEbus to occur back-to-back without requirement for re-arbitration.

The CWT should be set for the expected latency between sequential coupled accesses attempted by the CPU. In calculating the latency expected here, the designer needs to account for latency across their host PCI bridge as well as latency encountered in re-arbitration for the PCI bus between each coupled access. Care must be taken not to set the CWT greater than necessary as the Universe blocks all decoupled write transactions with target-retry, while the coupled channel owns the VMEbus. It is only when the CWT has expired that the PCI bus is permitted to enqueue trans-actions in the Tx FIFO.

When a coupled access is attempted to the VMEbus, the Universe generates a target-retry to the PCI initiator if the coupled path does not currently own the VMEbus. This occurs if the Universe is not currently VMEbus master, or if the DMA is currently VMEbus master or if entries exist in the Tx FIFO.

If the Universe does not have ownership of the VMEbus when a coupled access is attempted, the Universe generates a target-retry with a single wait state (see Figure x). The request for the VME bus occurs six Clk64 periods after the cycle is retried.

Coupled Cycles : Reads

Once ownership of the VME bus is attained by the coupled channel, the Universe propagates the cycle out to the VMEbus. Figure 1 shows such a coupled read cycle against an ideal VME slave. There are 24 wait states inserted by the Universe on the PCI bus before it responds with TRDY#. Further wait states are inserted for each extra 30ns in slave response.

Performing 32-bit PCI reads from VME, gives a sustained performance of approximately 4 MB/s. Figure 2 shows several of these access occurring consecutively.

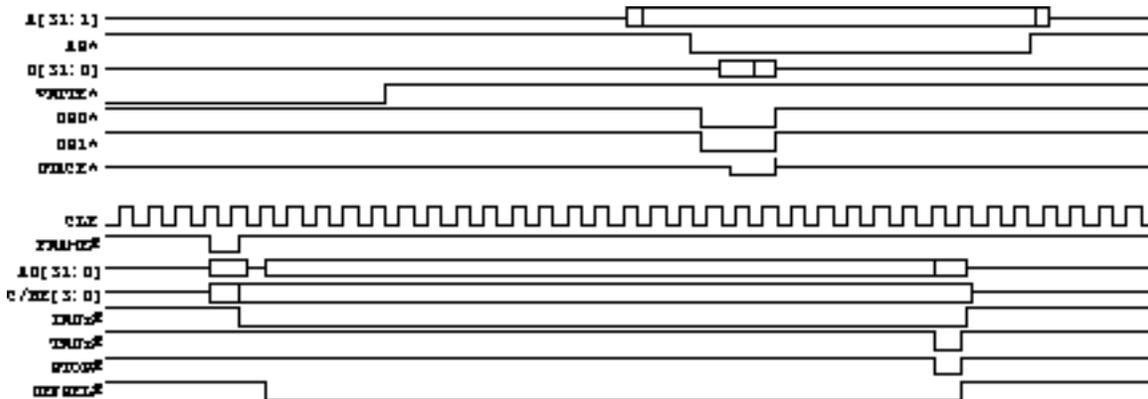


Figure 1 : Coupled Read Cycle - Universe as VME Master

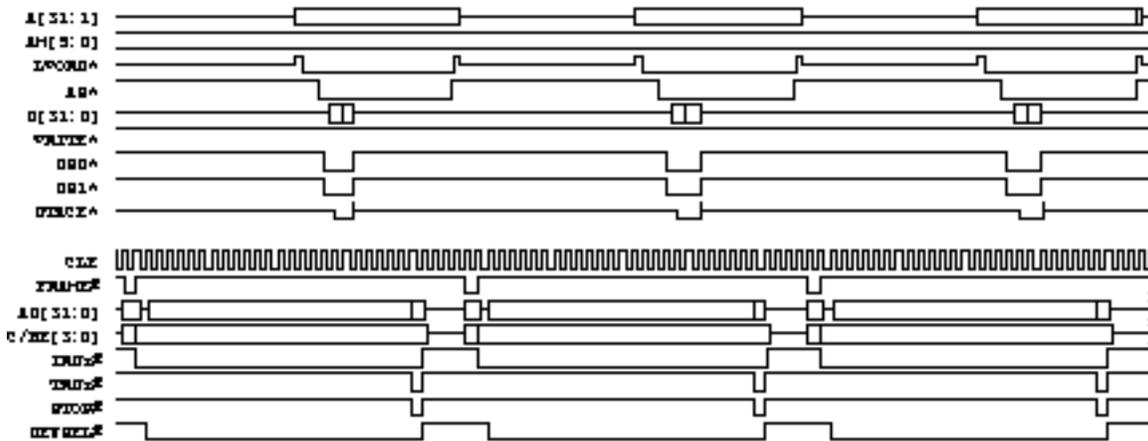


Figure 2 : Several Coupled Read Cycles - Universe as VME Master

Coupled Cycles : Writes

The performance of coupled write cycles is similar to that of coupled read cycles except that an extra wait state is inserted. Figure 3 shows a coupled write cycle against an ideal VME slave. Twenty-five wait states are inserted on the PCI bus by the Universe before it responds with TRDY#. A slower VME slave response translates directly to more wait states on the PCI bus.

The sustained performance, when generating write cycles from a 32 bit PCI bus against an ideal VME slave is approximately 4 MB/s.

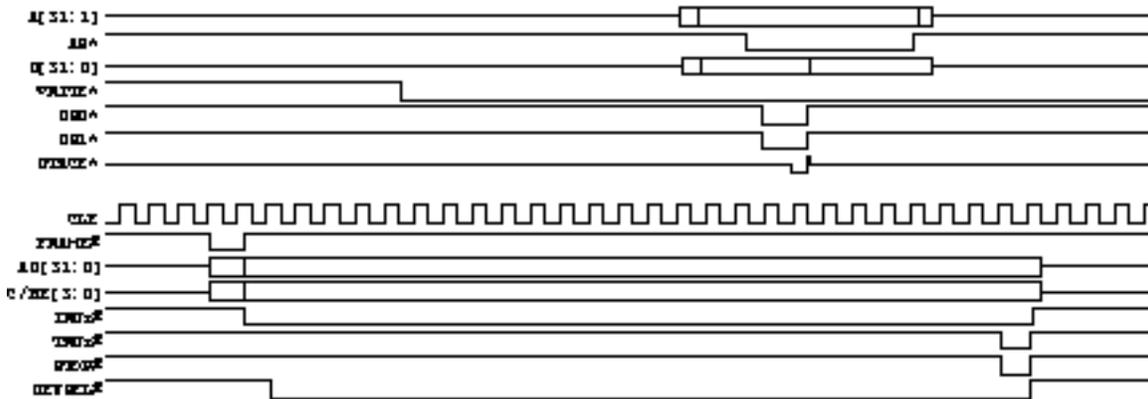


Figure 3 : Coupled Write Cycle - Universe as VME Master

Decoupled Write Cycles : Effect of the PCI Aligned Burst Size

The Universe permits enqueueing of PCI write transactions of length, up to the value indicated by the PABS bit in the MAST_CTL register. This bit forces the Universe to break up PCI transactions at a 32-byte or 64-byte aligned address. The value chosen by the designer will depend upon system specific parameters. The higher limit typically gives higher performance in those systems capable of generating burst of that size because it permits more data to be transferred with less addressing and arbitration overhead. On the VMEbus, longer block lengths are generated with the accompanying limit on the number of address phases being generated.

It is important to recognize the limits, however, of using a larger burst size. The Universe's Tx FIFO is capable of storing a maximum of 128 bytes with a maximum of four transactions. Attempts to enqueue data in the FIFO encounter a target-retry from the Universe if there is insufficient room available for a full burst size. This implies that the incidence of retries encountered by a PCI initiator will increase with the length of the programmed burst size. A side effect of this is that the FIFO may empty more often if the latency between retries excess the time taken by the Universe to dequeue a transaction to the VMEbus.

In general, the PABS should be programmed with the size of the burst the Universe will most typically encounter in the system. If the majority of access to the Tx FIFO will be short transactions, it is recommended that the PABS be programmed for a value of 32 bytes. Only in systems where transactions tend towards the 64 byte length, or where accesses to the Tx FIFO are infrequently made should the PABS be programmed to 64 bytes.

Decoupled Write Cycles : Effect of the Posted Write On Counter

The Posted Write On Counter (PWON in the MAST_CTL register) controls the maximum tenure that the PCI Slave Channel will have on the VMEbus. Once this channel has gained ownership of the VMEbus for use by the Tx FIFO, it only relinquishes it if the FIFO goes empty or if the number of bytes programmed in the counter expires. In most situations, the FIFO empties before the counter expires. However, if a great deal of data is being transferred by a PCI initiator to the VMEbus, then this counter ensures that only a fixed amount of VME bandwidth is consumed.

Limiting the size of the PWON counter imposes greater arbitration overhead on data being transferred out from the FIFO. This is true even when programmed for ROR mode since an internal arbitration cycle will still occur. The value for the PWON counter must be weighed from the system perspective with the impact of imposing greater latency on other channels (the DMA and Interrupt Channels) and other VME masters in gaining ownership of the VMEbus. On a Universe equipped card which is only performing system control functions, the counter would be set to minimum. On a card which is responsible for transferring considerable amounts of performance critical data the counter will be set much higher at the expense of system latency.

Decoupled Write Cycles : PCI Target Response

As the PCI target during decoupled write operations to the VMEbus, the Universe responds in one of two manners:

- 1) It immediately issues a target retry because the FIFO does not have sufficient room for a full PCI transaction, where a full transaction length is defined by PCI Aligned Burst Size (PABS) field.
- 2) It responds as a zero-wait state target receiving up to 32 or 64 bytes in a transaction. Once a full transaction has been enqueued, a target-disconnect is issued.

In either case, the Universe will consume the minimum possible PCI bandwidth, never inserting wait states.

Decoupled Write Cycles : VME Master Performance

As a VME master, the Universe waits until a full transaction has been enqueued in the FIFO before requesting the VMEbus and generating a VME cycle. If the VMEbus is already owned by the decoupled path (see Effect of the Posted Write On Counter, above), the Universe still waits until a full transaction is enqueued in the FIFO before processing it.

If configured to generate non-block transfers, the Universe can generate back-to-back VME transfers with cycle times of approximately 165ns (AS* to AS*) against an ideal VME slave (30-45 ns). A greater cycle time is required between the termination of one full enqueued transaction and the start of the next. This inter-transaction time is approximately 220ns. As such, the longer the PCI transaction, the greater the sustained performance on the VME bus. With 64 byte PCI transactions, the sustained rate is 23 MB/s. With 32 byte transactions, this drops to 22 bytes. Each of these numbers are calculated with no initial arbitration or re-arbitration for the bus. Figure 4 shows the Universe dequeuing a transaction with multiple non-block VME transfers.

Generation of block transfers significantly increases performance. The inter-transaction remains at approximately 240 ns for BLTs and 280ns for MBLTs, but the data beat cycle time (DS* to DS*) drops to about 105ns against the same ideal slave. Again the length of the burst size affects the sustained performance because of the inter-transaction time. For BLTs operating with a burst size of 64 bytes, the sustained performance is 33 MB/s, dropping to 29 MB/s for a burst size of 32 bytes. MBLTs operating with 64 byte bursts perform at a sustained rate of 51 MB/s, dropping to 39 MB/s for 32 bytes.



Figure 4 : Several Non-Block Decoupled Writes - Universe as VME Master

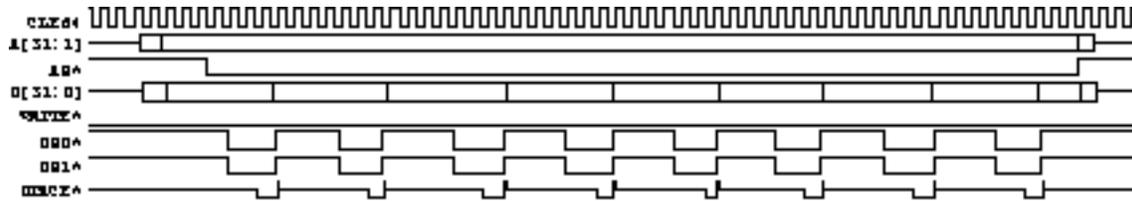


Figure 5 : BLT Decoupled Write - Universe as VME Master

VME SLAVE CHANNEL

Coupled Cycles : Block vs. non-Block Transfers

The Universe VME Slave Channel handles both block and non-block coupled accesses in similar manners. Each data beat is translated to a single PCI transaction. Once the transaction has been acknowledged on the PCI bus, the Universe asserts DTACK* to terminate the VME data beat.

A non-block transfer and the first beat of a BLT transfer have identical timing. In each, the Universe decodes the access and then provides a response to the data beat. Subsequent data beats in the BLT transfer are shorter than the first due to the fact that no address decoding need be performed in these beats.

MBLT transfers behave somewhat differently. The first beat of an MBLT transfer is address only, and so the response is fairly fast. Subsequent data beats require acknowledgment from the PCI bus. With a 32 bit PCI bus, the MBLT data beat (64 bits of data) requires a two data beat PCI transaction. Because of this extra data beat required on the PCI bus, the slave response of the Universe during coupled MBLT cycles is at least one PCI clock greater (depending upon the response from the PCI target) than that during BLT cycles.

Coupled Cycles : Reads

During coupled cycles, the Universe does not acknowledge a VME transaction until it has been acknowledged on the PCI bus. Because of this the VME slave response during coupled reads is directly linked to the response time for the PCI target. Each clock of latency in the PCI target response translates directly to an extra clock of latency in the Universe's VME coupled slave response.

The address of an incoming VME transaction is decoded and translated to an equivalent PCI transaction. From initial assertion of AS* on the VMEbus to assertion of REQ# on the PCI bus is typically 7-8 PCI clock periods. During the data only portion of subsequent beats in block transfers, the time from DS* assertion to REQ# is about 6 clocks.

From assertion of REQ#, the Universe does not insert any extra wait states in its operations as an initiator on the PCI bus. Upon receiving GNT# asserted, the Universe asserts FRAME# in the next clock and after the required turn-around phase, asserts IRDY# to begin data transfer.

Once TRDY# is sampled asserted, the Universe responds back to the VMEbus by asserting DTACK*. If the initiating VME transaction is 64 bit and the PCI bus or PCI bus target are 32 bit, then two data transfers are required on PCI before the Universe can respond with DTACK*. No wait states are inserted by the Universe between these two data beats on PCI. The assertion of DTACK* from the assertion of TRDY# has a latency of 2 clocks. Figure 6 shows a typical non-block coupled read cycle.

When accessing a PCI target with a zero wait state response, the Universe VME response becomes approximately 14 PCI clock periods (about 420ns in a 33MHz system) during single cycles, and the first beat of a BLT. During pure data beats in both BLT and MBLTs, the slave response becomes 13 clocks.

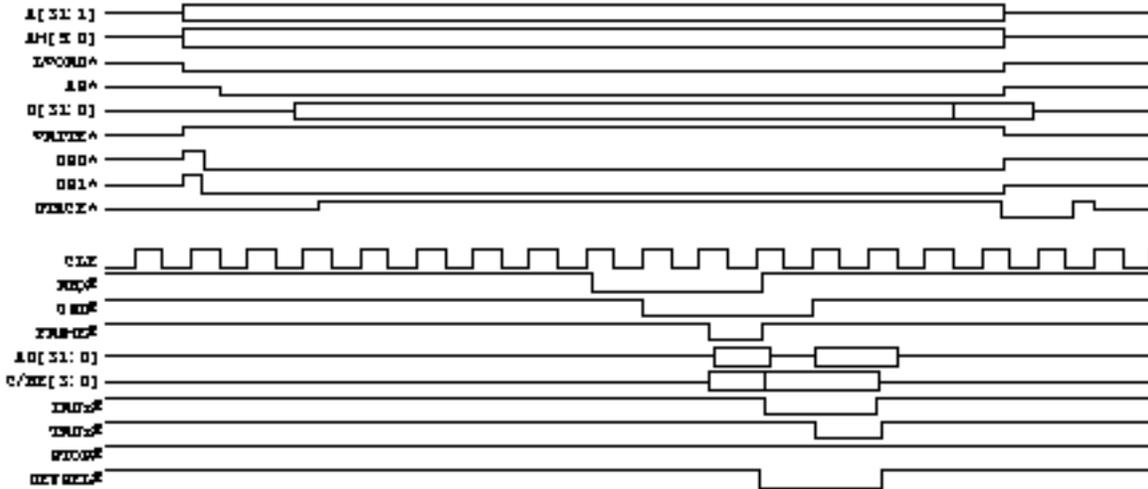


Figure 6 : Coupled Read Cycle - Universe as VME Slave

Coupled Cycles : Writes

Coupled writes in the VME Slave Channel operate in a similar fashion to the coupled reads. The VME slave response is directly linked to the response of the PCI target. In generating the request to the PCI bus, coupled write cycles require one further clock over reads. Hence, during single cycles, or the first beat of a BLT, the time from AS* to REQ# asserted is 8-9 PCI clocks, while DS* to REQ# is 7 clocks for the data beat portion of a block transfer.

As with reads, the response from the PCI target's assertion of TRDY# to DTACK* assertion by the Universe adds two clocks to the transfer. Figure 7 shows a typical non-block coupled write cycle.

Because write cycles on the PCI bus require one less clock than reads due to the absence of the turn-around phase between address and data phases, the overall slave response during coupled writes works out to the same as coupled reads against an identical target. In accessing a zero-wait state PCI target, the Universe's coupled write slave response then is approximately 14 PCI clocks. During subsequent data beats of a block transfer (either BLT or MBLT), the slave response (DS* to DTACK*) is 13 clocks.

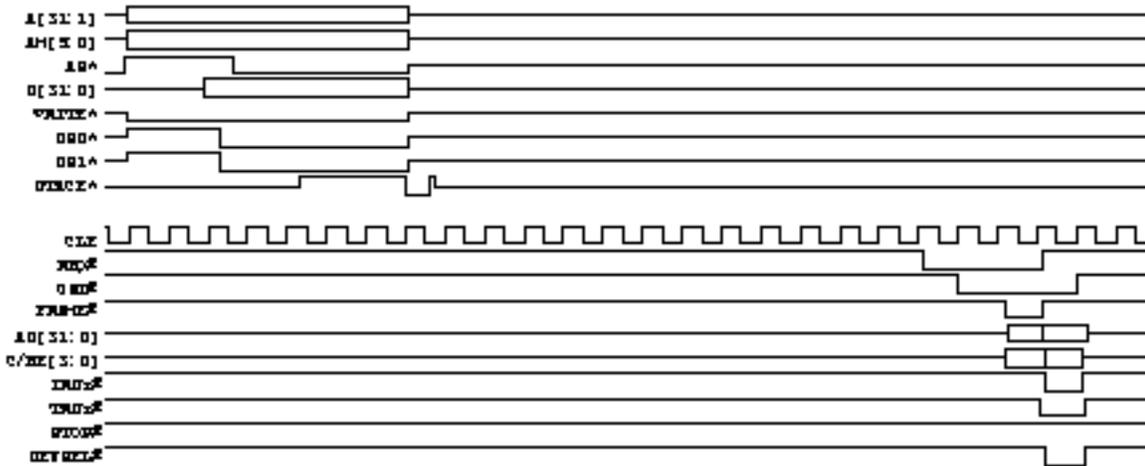


Figure 8 : Non-Block Decoupled Write Cycle - Universe as VME Slave

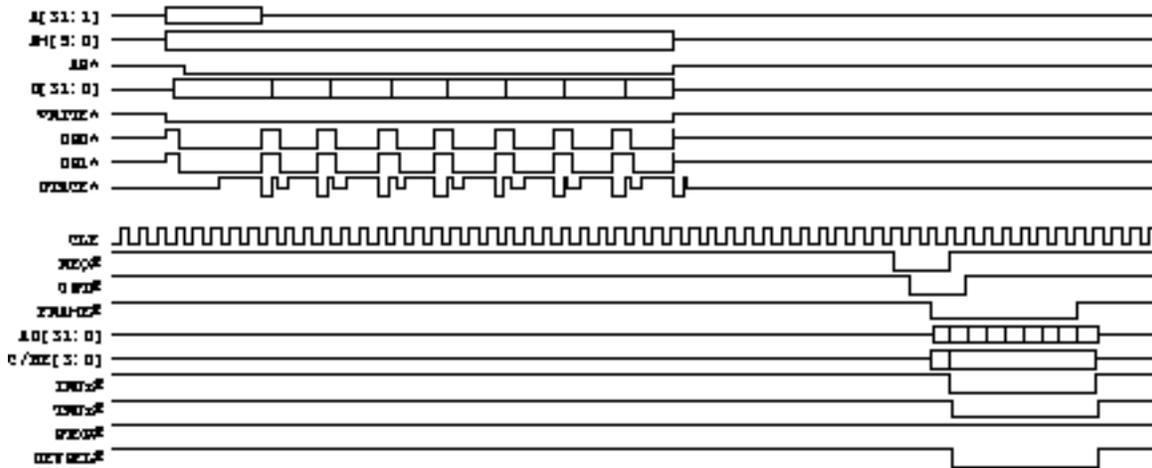


Figure 9 : BLT Decoupled Write Cycle - Universe as VME Slave

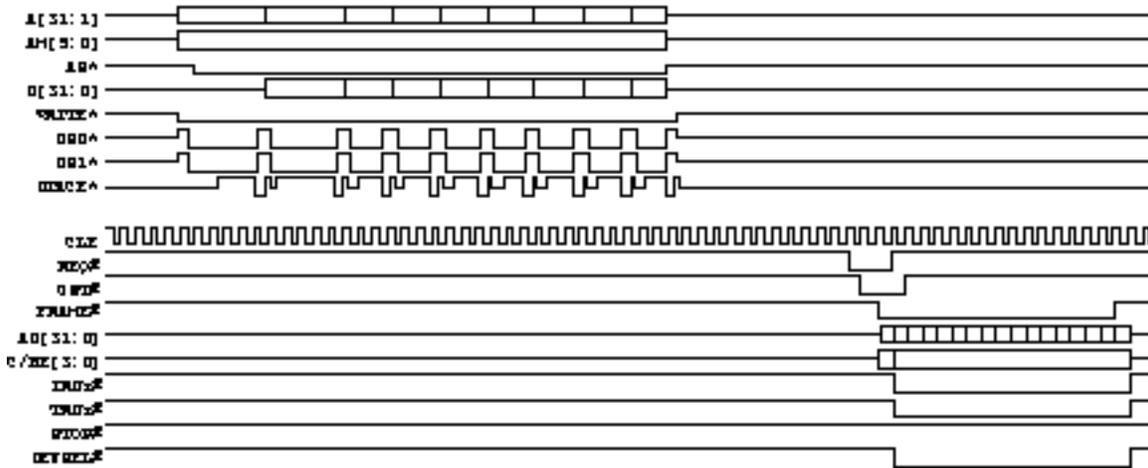


Figure 10 : MBLT Decoupled Write Cycle - Universe as VME Slave

Decoupled Write Cycles : PCI Master Performance

On the PCI bus, the Universe dequeues data from the Rx FIFO once a full VME transaction has been enqueued or once sufficient data has been enqueued to form a PCI transaction of length defined by the PABS field.

Since the Universe does not perform any address phase deletion, non-block transfers are dequeued from the FIFO as single data beat transactions. Only block transfers result in multi-data beat PCI transactions; typically 4 or 8 data beats. In either case, the Universe does not insert any wait states as a PCI master. The clock, after the bus has been granted the Universe, drives out FRAME# to generate the address phase. The data phases begin immediately on the next clock. If there is more than one data phase, each phase will immediately follow the acknowledgment of the previous phase.

In each case, because of the lack of any wait states as a PCI master, the Universe is consuming the minimum possible bandwidth on the PCI bus, and data will be written to the PCI bus at an average sustained rate equal to the rate at which the VME master is capable of writing it.

The sustained performance on the PCI bus performing single data beat write transactions to a 32 bit PCI bus is 12 MB/s; double this for a 64 bit bus. When performing 32 byte transactions the sustained performance increases to 76 MB/s; 97 MB/s with 64 byte transactions. Again, these can be doubled for a 64 bit PCI bus. Bear in mind that the PCI bus can only dequeue data as fast as it is being enqueued on the VMEbus. Hence, as the FIFO empties, the sustained performance on the PCI will drop down to match the lower performance on the VME side. However, even with the decreased sustained performance, the consumed bandwidth will remain constant (no extra wait states are inserted while the Universe is master of the PCI bus.)

These numbers assume the PCI bus is granted to the Universe immediately and that the writes are to a zero-wait state PCI target capable of accepting the full burst length. Figures 8 through 10 show the Universe responding to non-block, BLT and MBLT write transactions to a 32 bit PCI bus.

Prefetched Read Cycles

To minimize its slave response, the Universe generates prefetched reads to the PCI bus in response to BLT and MBLT reads coming in from the VMEbus. This option must first be enabled on a per image basis.

When enabled, the Universe will respond to a block read by performing burst reads on the PCI bus of length defined by the PCI Aligned Burst Size (PABS in the MAST_CTL register). These burst reads continue while the block transfer is still active on the VMEbus (AS* not negated) and there is room in the FIFO. If there is insufficient room in the FIFO to continue (a common occurrence since the Universe is capable of fetching data from the PCI bus at a much faster rate than a VME master is capable of receiving it), then pre-fetching stops and only continues once enough room exists in the FIFO for another full burst size.

The first data beat of a block transfer must wait for the first data beat to be retrieved from the PCI bus -- essentially a coupled transfer. See the section on coupled transfers for details on coupled performance. However, once the pre-fetching begins, data is provided by the Universe in subsequent data beats with a slave response of 65ns. This continues while there is data in the FIFO. If the FIFO empties because data is being fetched from the PCI bus too slowly, wait states are inserted on the VMEbus awaiting the enqueueing of more data.

On the PCI bus, the Universe fetches data at 66 MB/s with PABS set to 32 byte transactions; 89 when set to 64 byte transactions. Once the FIFO fills, pre-fetching slows to match the rate at which it is being read from the VMEbus. Bandwidth consumption, however, remains constant, only the idle time between transaction increases.

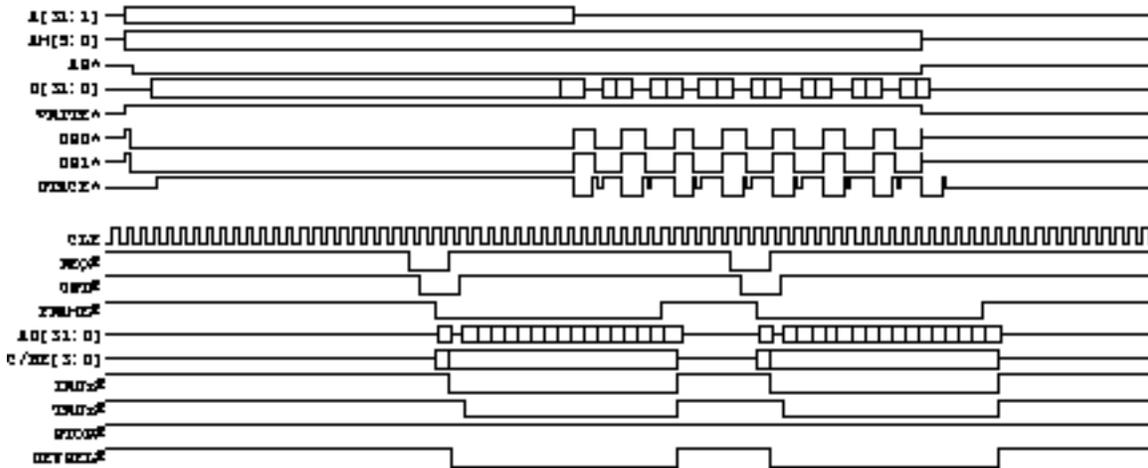


Figure 11 : BLT Pre-fetched Read Cycle - Universe as VME Slave

DMA CHANNEL

VMEbus Ownership Modes

The DMA has two timers that control its access to the VMEbus: the VON (VMEbus On) counter and the VOFF (VMEbus Off) timer. The VON counter controls the number of bytes that are transferred by the DMA during any VMEbus tenure, while the VOFF timer controls the period before the next request after a VON time-out.

While the bus is more optimally shared between various masters in the system, and average latency drops as the value programmed for the VON counter drops, the sustained performance of the DMA also drops. The DMA is typically limited by its performance on the VMEbus. As this drops off with greater re-arbitration cycles, the average VMEbus throughput will drop. Even if the Universe is programmed for ROR mode, and no other channels or masters are requesting the bus, there will be a period of time that the DMA will pause its transfers on the bus.

An important point to consider when programming these timers is the more often the DMA relinquishes its ownership of the bus, the more frequently the PCI Slave Channel will have access to the VMEbus. If DMA tenure is too long, the Tx FIFO may fill up causing any further accesses to the bus to be retried. In the same fashion, all coupled accesses will be retried while the DMA has tenure on the bus. This can significantly affect transfer latency and should be considered when calculating the overall system latency.

VME Transfers

On the VMEbus, the Universe can perform D08 through D64 transactions in either block or non-block mode. The time to perform a single beat, however, is independent of the bus width being used. Hence, a D08 transaction will transfer data at 25% the rate of a D32, which in turn is half that for D64.

There is a significant difference between the performance for block vs. non-block operations. Because of the extra addressing required for each data transfer in non-block operations, the DMA performance is about half that compared to operating in block mode. When considering that most VME slaves also have considerably slower slave response in non-block mode, the overall performance may drop to one-quarter of that achievable in block mode.

VME Transfers : Reads

When performing non-block reads on the VMEbus, the Universe cycle time (AS* to next AS*) is approximately 160ns, which translates to about 25 MB/s when performing D32 transfers. For block transfers the cycle time (DS* to next DS*) falls to about 130ns, or 30 MB/s for D32 transfers. For multiplexed block transfers (MBLT's) the cycle time remains the same, but because the data width doubles, the transfer rate increases to about 60MB/s.

VME Transfers : Writes

Non-block writes to the VMEbus occur at the same rate as non-block reads : 160ns cycle time (AS* to next AS*), or 25MB/s during D32 transfers. Block writes, however, are significantly faster with a 95ns cycle time (DS* to next DS*), or 42 MB/s. Multiplexed block transfers have slightly longer cycle times at about 110ns (DS* to next DS*), or 72 MB/s with D64 MBLT's.

PCI Transfers

As a master on the PCI bus, the Universe DMA follows the same general set of rules as the VME Slave channel does: it never inserts any wait states into the transfer (i.e. it never negates IRDY# until the transaction is complete) and will always, where possible, generate full aligned bursts as set in the PABS field of the MAST_CTL register.

Between transactions on the PCI bus, the Universe DMA typically sits idle for 10 clocks. Hence, minimizing the number of idle periods and re-arbitration times by setting PABS to its maximum value of 64 bytes increases the performance of the DMA on this bus. However, due to limitations in the DMA FIFO size, this does not always result in higher DMA throughput.

The DMA FIFO is only sufficiently large enough to contain two 64 byte PCI transactions. Also, the DMA only generates a read transaction on the PCI bus if there is sufficient room for a full transaction, and a write transaction only if there is a full transaction in the FIFO. A situation could arise during DMA reads (writes to the PCI bus) that the DMA has enqueued a full transaction into the FIFO. At this point, DMA will make an attempt to dequeue that transaction to the PCI bus. If, before the DMA is given an opportunity to dequeue that transaction another is enqueued from the VMEbus, the DMA will give up ownership of the VMEbus. This sort of operation will result in an oscillating tendency for the Universe to perform short transfers on the VMEbus and then release ownership. The effect may be to severely limit the throughput with excessive arbitration overhead.

The same situation could occur during DMA writes (reads from the PCI bus). The DMA begins writes to the VMEbus as soon as a full transaction has been enqueued in the FIFO and releases the bus if the FIFO goes empty, or only a partial transaction has been enqueued. It is possible that the DMA may write the full transaction to the VMEbus before another transaction has been completely enqueued from the PCI side, resulting in a loss of VMEbus ownership by the DMA since it gives up the bus if the FIFO does not contain a full transaction to dequeue.

Encountering either of the above situations depends upon factors such as the VME slave response, the PCI target response and PCI arbitration latency. It may typically be rectified by changing the PABS field. A smaller burst size makes it more likely that a full transaction will always be in the FIFO during writes to the VMEbus. A larger burst size allows the FIFO to empty more quickly during writes to the PCI bus.

Although both read and write transactions occur on the PCI bus with zero wait states, there is a period of eight PCI clocks during which the Universe remains idle before re-requesting the bus for the next transaction. With PABS set for 32 byte transactions on a 32 bit PCI bus, this translates to a peak transfer rate of 51 MB/s for reads, 53 MB/s for writes, doubling to 102 and 106 for a 64 bit PCI bus. With PABS set for 64 byte transactions, the peak transfer rate increases to 74 MB/s for reads, 77 MB/s for writes on a 32 bit PCI bus, doubling to 148 MB/s and 154 MB/s respectively for 64 bit PCI buses.

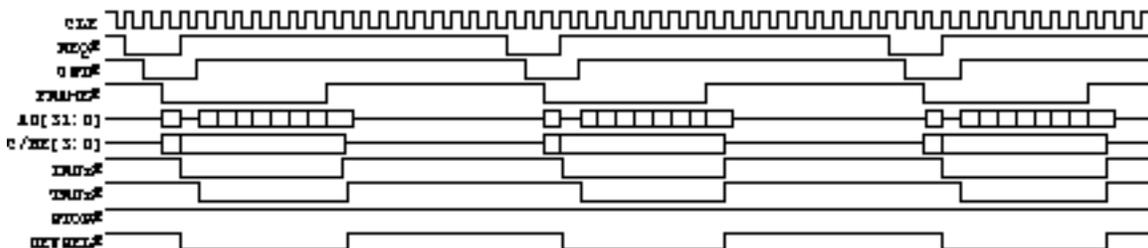


Figure 12 : PCI Read Transactions During DMA Operation

SUMMARY

Table 1 : PCI Slave Channel Performance

Cycle Type	Performance
Coupled Read - PCI target response	24 PCI clocks
Coupled Write - PCI target response	25 PCI clocks
Decoupled Write - non-block D32 - VME cycle time - sustained perf (PABS =0) - sustained perf (PABS =1) - D32 BLT - VME cycle time - sustained perf (PABS =0) - sustained perf (PABS =1) - D64 MBLT - VME cycle time - sustained perf (PABS =0) - sustained perf (PABS =1)	165 ns 22 MB/s 23 MB/s 105 ns 29 MB/s 33 MB/s 105 ns 39 MB/s 51 MB/s
Register access	8 wait states

Table 2 : VME Slave Channel Performance

Cycle Type	Performance
Coupled Read - non-block - block	420 ns 390 ns
Coupled Write - non-block - block	420 ns 390 ns
Pre-fetched Read - VME slave response (1st data beat) - VME slave response (other data beats)	(see coupled table) 65 ns
Decoupled Write - non-block slave response - block slave response	130 ns 65 ns
Register access - reads - writes	360 ns 240 ns

Table 3 : DMA Channel Performance

Cycle Type	Performance
PCI Reads - PABS = 32 bytes - PABS = 64 bytes	51 MB/s 74 MB/s
PCI Writes - PABS = 32 bytes - PABS = 64 bytes	53 MB/s 77 MB/s
VME Reads - non-block D32 - D32 BLT - D64 MBLT	25 MB/s 30 MB/s 60 MB/s
VME Writes - non-block D32 - D32 BLT - D64 MBLT	25 MB/s 42 MB/s 72 MB/s

Table 4 : Daisy Chains

Daisy Chain	Performance
IACK Daisy Chain - active Universe interrupt - no active Universe interrupt	21 - 36 ns 6 - 21 ns
Bus Grant Daisy Chain	32 ns