Chapter 11

2eVME Protocol

11.1 Introduction

The 2eVME (**2 edge VME**) transactions add four important features to the VME64 and VME64x architectures:

1. Theoretical doubling of the peak block data rate to 160 MB/sec
2. Master and slave terminated 2eVME transfers
3. Reduced number of address modes
4. Room to add more features in the future

This protocol will work with either the 96 pin connector on VME64 backplanes or the 160 pin connectors on VME64x backplanes. Just this feature (capability) can be added to VME64 boards and systems. The product can't be called VME64x, but can be marketed as VME64 with 2eVME capability.

11.1.1 2 Edge Handshakes

The method being used to double VME64 and VME64x's backplane performance is a two edge handshake for each data transfer. VME64 transfers use a four edge handshake for each data transfer. With a tightening of the timing parameters, and by making the timing more technology independent, the peak data transfer rate can be doubled to one transfer every 50 nsec, instead of 100 nsec. This translates to a peak block data transfer rate of 160 MB/sec (eight bytes per transfer on a 64 bit bus). For 3U boards, the peak rate is 80 MB/sec with 32 bit transfers.

11.1.2 Address Phases

In order to speed up the address phase, like the data phases, two edge handshakes are also employed. All 2eVME address broadcasts are split into three phases, address phase 1, address phase 2 and address phase 3. Three times as much information can be transferred during the address phase as compared to VME64 address phases.

11.1.3 Remapping the LWORD* Line

In the VME64 standard, the LWORD* signal line is used as a data transfer signal line, during 64 bit data transfers. The LWORD* line is redefined as address bit 0, A[0]. This effectively provides a full 32 bit address bus, labeled A[31:0] and reflects the usage of the line.

11.1.4 Extended AM Codes

Since there are only a few unassigned address modifier codes left, an extended address modifier (XAM) coding scheme is used. AM Code 0x20 is assigned for 6U 2eVME transfers and AM Code 0x21 is assigned for 3U 2eVME transfers. The eight LSB (least significant bits) of the address field A[7:0] are used to carry the extended address modifier code information during the first address phase. See Table 11-1, which maps the address lines to the extended AM code field. With 8 bits, 256 additional address modifier codes are available for each of the 6U and 3U 2eVME transaction sets.

The initial 2eVME definition defines two address modes for 6U boards (Table 11-2) and two address modes for 3U boards (Table 11-3). With 256 possible extended AM codes for each board size the extended address addition leaves plenty of room for future expansion.

Table 11-1 Extended Address Modifier Line Definitions

A7	A6	A5	A4	A3	A2	A1	A0
XAM7	XAM6	XAM5	XAM4	XAM3	XAM2	XAM1	XAM0

Table 11-2 6U 2eVME Extended Address Modifier Codes

XAM Code	Function
00	Reserved
01	A32/D64 2eVME Transfer
02	A64/D64 2eVME Transfer
03-FF	Reserved

Table 11-3 3U 2eVME Extended Address Modifier Codes

XAM Code	Function
00	Reserved
01	A32/D32 2eVME Transfer
02	A40/D32 2eVME Transfer
03-FF	Reserved

11.1.5 Address Modes

For 6U VME64x boards, the 2eVME block data transfers are sized at 64 bits, or 8 bytes per transfer. One address range is defined for A32 type transactions and another for A64 type transactions. Usage of the supervisory / non-privileged and the program / data sub-modes are no longer necessary, with the very large A64 address range. This will most likely provide enough space to map the memory into special functional groups, if that is required by the specific application.

For 3U VME64x boards or those VME64x boards with only a P1/J1 connector, two address ranges are defined: A32 and A40.

11.1.6 Known Length 2eVME Transfers

With many of the new microprocessors that employ caching architectures, as well as DMA controllers, the size of the data being written or requested via a read is known in advance. This size information is presented during address phase two. Address lines A[15:8] are used to carry the beat count. This effectively informs the slave in advance of the amount of data that it is requested to receive in a write transaction or the amount of data it is to supply in a read request. The beat count ranges from 0 to 256 data beats.

All 6U block data transfers are performed with 64 bit data words (eight bytes per transfer). The beat count field is a count of the number of 8 byte transfers, ranging from 1 to 256 transfers. For 6U 2eVME transactions, a maximum of 2 KB can be transferred in a single block. For 3U all 2eVME transactions, the maximum is 1 KB. As with the VME64 MBLT transfers, no 6U 2eVME transaction can cross a 2 KB boundary, and no 3U 2eVME transaction can cross a 1 KB boundary.

The Beat Count field represents a maximum transmission value for the transfer, not the absolute expected size. Hence, boards doing pre-fetching based upon this value have no guarantee that all pre-fetched data will be transferred.

11.1.7 Slave Terminated 2eVME Transfers

In some applications, I/O boards collect a random amount of data, ranging from zero bytes to many kilobytes. Masters reading this data from the slave have no prior knowledge as to the amount of data being retrieved. Slaves can terminate the block transfer at any time. A Master which expects a Slave termination is advised to put the maximum block size that it can receive in the Beat Count bits in the second address phase.

The 6U Slaves indicates this termination by asserting RETRY* and then asserting BERR*. The 3U Slaves indicates this termination by asserting RESP* and then asserting BERR*.

This termination method tells the Master that the Slave has no more data and is not expecting a resumption of that transfer.

11.1.8 Slave Suspended 2eVME Transfers

Large 2eVME transfers can be divided into smaller blocks by the slave, simply by terminating the block transfer after a few transfers. Masters can resume the block transfer at the next logical address. Slaves issuing the suspension indicate to the Master that more data can be transferred, just not at this time.

At times a slave board becomes busy and is unable to service a read or write request. This could be due to the resource, such as a dual port memory, being busy, or possibly its input buffer is still full, or whatever. The slave board can signal the requesting master to retry the transaction a little later. In some applications, the master might not rerun a slave terminated transaction, but just go on to the next task.

This same mechanism also works for deadlock resolution. If a board's master is in the process of trying to access another board's resource through a bridge, and at the same time the other board's master is trying to access the first board's resource via the same bridge, one of the masters must back off. By using the suspend protocol, the bridge between the two buses can request one of the two masters to back off and try again, thereby resolving a possible deadlock situation.

6U Slaves requests that a master suspend a transaction by asserting RETRY* and then toggling DTACK*. 3U Slaves requests that a master suspend a transaction by asserting RESP* and then toggling DTACK*. The suspend termination tells the Master that the Slave is stopping this transaction but is expecting a resumption of that transfer at a later time. Suspended transactions always occur on an even beat count

11.1.9 Slave Error State

Slaves which detect an error can signal this condition to the Master by asserting BERR* in response to a transition of DS0* or DS1*. This is similar to the current BERR* usage in VME.

11.1.10 Master Terminated 2eVME Transfers

By definition, the master controls the size of each block transfer, by specifying the beat count. The 2eVME protocol is designed to allow for early master termination, even before the beat count is reached (decremented to zero). This can be used for a variety of special type applications and situations. When the Master terminates it is always on an even beat count.

11.1.11 2eBTO Bus Time Out Timer

The 2eVME protocol holds one of the data strobes low during the entire data transfer cycle. A block transfer could consist of up to 256 data beats plus the address broadcast phase. If each data beat takes 100 nanoseconds then the total data transfer cycle will be over 25.6 microseconds. This requires the VMEbus BTO delay be set to over 25.6 microseconds.

In many systems, this delay is too long for all bus transactions. A specific bus time out timer is defined for 2eVME block transfers, called 2eBTO(x). The 2eBTO(x) bus timer will time out when the bus transaction takes too long.

11.2 Requirements

11.2.1 Transceivers and Connectors

Rule 11.1:

2eVME boards shall use VITA 2-199x Enhanced Transceiver Logic (ETL) bus transceivers, or equivalent for both high and low going edges on all address, data and control bus signal lines.

Rule 11.2:

System integrators using the 2eVME protocol shall ensure that the proper VME64 or VME64x backplane and boards loaded into the system will allow for monotonic rising and falling bus signals, when driven with VITA 2-199x Enhanced Transceiver Logic (ETL) bus transceivers.

Observation 11.1:
The RESP* signal is in the z row of the 160 pin connector, therefore the 160 pin connector needs to be used to run the 2eVME protocol on 3U boards if the RESP* signal features is used. 6U boards running the 2eVME protocol can use either the 96 pin or the 160 pin connectors.

11.2.2 Extended AM Codes

Rule 11.3:
All 6U VME64x boards that perform 2eVME transactions shall use AM Code 0x20 and shall use the extended AM Codes as defined in Table 11-2, 6U 2eVME Extended AM Codes.

Rule 11.4:
All 3U VME64x boards that perform 2eVME transactions shall use AM Code 0x21 and shall use the extended AM Codes as defined in Table 11-3, 3U 2eVME Extended AM Codes.

Permission 11.1:
6U VME64x boards may also participate in 3U 2eVME transfers using the applicable AM and XAM Codes defined for 3U 2eVME transfers.

11.2.3 Data Size

Rule 11.5:
The maximum number of data beats shall be 256.

Observation 11.2:
Rule 11.6 limits the data transferred to 2 KB, i.e. 256 beats x 8 bytes in a 6U VME64x board and 1 KB in a 3U VME64x board.

Rule 11.6:
A 6U 2eVME transfer shall not cross a 2 KB boundary and a 3U 2eVME transfer shall not cross a 1 KB boundary.

Rule 11.7:
All address and data lines shall be a logical zero for features which are not implemented or are part of Reserved fields.

Rule 11.8:
The beat count shall be sent in A[15:8] during the second address phase. The value is the number of beats divided by two. For example 0x00 = none, 0x01 = 2 beats, ... 0x80 = 256 beats.

Rule 11.9:
Beat counts 0x81 through 0xFE shall be reserved.

Observation 11.3:
Since the master can only do even beats for master terminated block transfers the modulo 2 number for the beat count is compatible with the specified operation.

Rule 11.10:
The beat count shall be set to 0xFF as a default value when slave termination is expected and the beat count is unknown.

Permission 11.2:
The slave may use the beat count in a read operation to prefetch the required words and thereby increase performance.

Rule 11.11:
The beat count represents a maximum value only. Slaves shall always be prepared for an early termination by the master during 2eVME transfers.

11.2.4 Protocols - General

Permission 11.3:
Usage of both the RESP* and RETRY* signals is optional by both the master and the slave.

Observation 11.4:
RESP* and RETRY* are used as a qualifier on the meaning of a timing response line.

Observation 11.5:
RESP* is used by 3U boards and RETRY* by 6U boards. 6U master boards must use RESP* when communicating with 3U boards while performing 2eVME transactions.

Rule 11.12:
In 2eVME protocols only one timing (DS*) line shall transition in any one bus cycle except when the transaction is ended.

Rule 11.13:
In 2eVME protocols only one timing response (DTACK* or BERR*) line shall transition in any bus cycle except when the transaction is ended.

Observation 11.6:
The support of slave terminated and slave suspended operations are not mandatory for either masters of slaves.

Recommendation 11.1:
Designs for masters conforming to the VME64x standard should implement slave terminated and slave suspend operations.

Rule 11.14:
The master timeout timer shall be set to "T" (as defined in the VME64x Standard).

Observation 11.7:
The time-out value is one half the bus monitor BTO value. See VME64 Standard for full explanation.

11.2.5 Address Phase Protocol and Timing

Rule 11.15:
Starting addresses shall be aligned on 16 byte boundaries (i.e. 128 bit aligned).

Rule 11.16:
2eVME address timing shall be as defined in Tables 11-6 and 11-7, as measured at connector on the backplane.

Rule 11.17:
2eVME address protocols shall be as defined in Figures 11-1 through 11-4.

Rule 11.18:
BERR* shall only be asserted on the first or third address phase instead of toggling DTACK*. See Figures 11-3 and 11-4.

Rule 11.19:
DTACK* shall be the only valid responses during address phase 2.

Rule 11.20:
Masters shall ignore the RETRY*/RESP* and BERR* during address phase 2.

Observation 11.8:
Slave terminated and slave suspend operations are not supported during address phase 2. Anomalous behavior can result if these responses are sent.

Observation 11.9:
RETRY* and RESP* are not asserted in the first and second address phases.

Observation 11.10:
If RETRY* or RESP* is asserted before DTACK* is toggled on the third address phase the

Master interprets that response as indicating that the Slave is suspending the operation but the Master can expect data when it tries again. This can also be interpreted as a Slave suspend response. See Figure 11-2.

Observation 11.11:
If RETRY* or RESP* is asserted before BERR* is asserted in the third address phase the Master interprets that response as indicating that the Slave has no data and the transfer attempt is to be terminated. See Figure 11-3.

Observation 11.12:
Since the target address is in two parts the target device generates DTACK* during the first phase of the address cycle if that portion of the address is recognized. The lowest byte of the address comes in the second address phase and therefore cannot be used for the address of the target device. The low byte only specifies an internal address.

Rule 11.21:
Address lines A[20:16] in the second address phase shall provide the GA of the Master as defined in Chapter 3 of this standard. If not implemented the value shall be 0x00.

Observation 11.13:
The GA in the second address phase provides a mechanism to determine the bus master for a particular transaction.

Rule 11.22:
Address lines A[31:24] in the second address phase shall provide a subunit number of the master. If not implemented the value shall be 0x00.

Observation 11.14:
The subunit number provides a mechanism for identifying which part of a master initiated a transaction. In the case of a VME module with several processors it could be the processor number.

11.2.6 Data Phase Protocol and Timing

Rule 11.23:
2eVME data timing shall be as defined in Tables 11-6 and 11-7, as measured at connector on the backplane.

Rule 11.24:
2eVME data protocols shall be as defined in Figures 11-5 through 11-12.

Observation 11.15:
If RETRY* or RESP* is asserted before DTACK* is toggled during the data phase then the Master interprets that response as indicating that the Slave is suspending the operation but the Master can expect data when it tries again. This can also be interpreted as a Slave busy response. Data was not accepted or sent for that DS1* transition. See Figures 11-6 and 11-10.

Observation 11.16:
If RETRY* or RESP* is asserted before BERR* is asserted during the data phase then the Master interprets that response as indicating that the Slave has no data at that time and is finished. Data was not accepted or sent for that DS1* transition. See Figurès 11-7, 11-8, 11-11 and 11-12.

Observation 11.17:
The toggling of DTACK* without the assertion of RETRY* or RESP* during the data phase indicates that the Slave has accepted the read or write data. See Figures 11-5 and 11-9.

Observation 11.18:
The assertion of BERR* without RETRY* or RESP* during the data phase indicates that the Slave has had an error and the data was not accepted or sent for that DS1* transition. See Figures 11-7, 11-8, 11-11 and 11-12.

Recommendation 11.2:

Some bus timers are designed to assert BERR* when DS0* or DS1* has been asserted for greater than a set period, without monitoring the state of DTACK*. In such a case, the bus timer could time-out a 2eVME transfer since DS0* is held low for the duration of the transfer. To avoid this, the bus timer should be set to a value greater than the longest expected 2eVME transfer.

Rule 11.25:

Master and Slave 2eVME state machines shall be robust enough to recover gracefully if BERR* is asserted by the bus timer during a 2eVME transfer.

11.2.7 2eBTO(x) Bus Time Out Timer

Recommendation 11.3:

It is recommended that the 2eBTO(x) bus time out timer be used in conjunction with the 2eVME protocol.

Rule 11.26:

If implemented, the 2eBTO(x) shall monitor the following signal lines: DS0*, DS1*, and DTACK* lines. The BERR* line shall be driven low after x microseconds when either data strobe is low for longer than x microseconds AND no low to high or high to low transition has occurred on either data strobe or DTACK* for longer than x microseconds. The time out time is set by the system integrator.

Table 11-4 6U VME64x Signal Field Definitions

	Address Phase 1	Address Phase 2	Address Phase 3	Data Phase
AM[5:0]	0x20	0x20	0x20	0x20
A[7:0]	XAM Code (Table 11-2)	Internal Address A[7:0]	Reserved	D[39:32]
A[15:8]	Device Address A[15:8]	Beat Count	Reserved	D[47:40]
A[23:16]	Device Address A[23:16]	A[23:21] = 0 A[20:16] = GA of Master	Reserved	D[55:48]
A[31:24]	Device Address A[31:24]	A[31:24] = Subunit number in Master	Reserved	D[63:56]
D[31:0]	Device Address A[63:32] (= 0 for A32)	Reserved	Reserved	D[31:0]

Recommendation 11.2:

Some bus timers are designed to assert BERR* when DS0* or DS1* has been asserted for greater than a set period, without monitoring the state of DTACK*. In such a case, the bus timer could time-out a 2eVME transfer since DS0* is held low for the duration of the transfer. To avoid this, the bus timer should be set to a value greater than the longest expected 2eVME transfer.

Rule 11.25:

Master and Slave 2eVME state machines shall be robust enough to recover gracefully if BERR* is asserted by the bus timer during a 2eVME transfer.

11.2.7 2eBTO(x) Bus Time Out Timer

Recommendation 11.3:

It is recommended that the 2eBTO(x) bus time out timer be used in conjunction with the 2eVME protocol.

Rule 11.26:

If implemented, the 2eBTO(x) shall monitor the following signal lines: DS0*, DS1*, and DTACK* lines. The BERR* line shall be driven low after x microseconds when either data strobe is low for longer than x microseconds AND no low to high or high to low transition has occurred on either data strobe or DTACK* for longer than x microseconds. The time out time is set by the system integrator.

Table 11-4 6U VME64x Signal Field Definitions

	Address Phase 1	Address Phase 2	Address Phase 3	Data Phase
AM[5:0]	0x20	0x20	0x20	0x20
A[7:0]	XAM Code (Table 11-2)	Internal Address A[7:0]	Reserved	D[39:32]
A[15:8]	Device Address A[15:8]	Beat Count	Reserved	D[47:40]
A[23:16]	Device Address A[23:16]	A[23:21] = 0 A[20:16] = GA of Master	Reserved	D[55:48]
A[31:24]	Device Address A[31:24]	A[31:24] = Subunit number in Master	Reserved	D[63:56]
D[31:0]	Device Address A[63:32] (= 0 for A32)	Reserved	Reserved	D[31:0]