Eurogam Project:
Specification of the VXI Readout Mechanism

J R Alexander

Version 1·0 April 1991

For phase 1 of the Eurogam project, the line specified as 'BLTACK*' by this specification is re-named 'Readout*' and is used slightly differently. The reason for this and the way in which it is used in phase 1 are explained in Appendix A.

1. Introduction

This document specifies the "REN-Readout cycle" which is used by Readout Controllers to read data from Eurogam VXI modules (currently the Ge, BGO and Master Trigger modules). The REN-Readout cycle is very similar to a normal VMEbus quad-byte block-read cycle, except that:

- A daisy-chain line is used to select slave modules in sequence, in the same manner as 'Readout Enable' (REN) does on the FERA[1]bus. (This signal is carried by VXI line Local-Bus4.)

- The slave modules determine the number of data words to be read during a cycle, and therefore the duration of a cycle, by use of a wired-OR 'Block Transfer Acknowledge' signal. (This signal is carried by VXI line TTL-Trig1).

- The REN-Readout cycle is identified by use of the 'user defined' VMEbus Address Modifier code 1B (hexadecimal), in order that standard VMEbus modules do not participate in the cycle.

1 'Fast Encoding and Readout of ADCs' bus, defined by LeCroy Corp.

2. *Signals used for the REN-Readout cycle*

Notes Signal source: M = VMEbus Master (i.e. Readout Controller)
S = VMEbus Slave (Ge, BGO, Trigger, . . . modules)
An asterisk (*) appended to a signal name indicates a signal which is true when at a TTL low level. All other signals are true when high.

Signal	Line used	Source	Function
A01	VME	M	= 0 for 32-bit data transfers
A02 - A05	VME	M	Event Number (of event being read out).
AM0 - AM5	VME	M	Address Modifier = 1B for REN-Readout cycles.
LWORD*	VME	M	Asserted by Master, to select 32-bit data transfers.
WRITE*	VME	M	*Not* asserted by Master, to select read transfers.
AS* DS0* DS1*	VME	M	Timing control (The Master asserts *both* Data Strobes, to select 32-bit data transfers)
D00 - D31	VME	S	Data lines (D00 is least significant bit, D31 is most significant bit)
DTACK*	VME	S	Timing Control
PASS*	LBUS 4 c	M or S	'Readout ENable' daisy chain (**out** from the module).
REN*	LBUS 4 a	Previous module's PASS*	'Readout ENable' daisy chain (**in** to the module).
BLTACK*	TTL Trig 1*	S	'Block Transfer Acknowledge'

The **PASS*** and **REN*** signals are used to select one slave at a time during the REN-Readout cycle. A slave is selected when its REN* input is asserted (true) and it is *not* asserting its PASS* output. The selected slave is the only one which may respond to data transfers (i.e. assert data and DTACK* in response to the data strobes).

The **BLTACK*** signal is used to control the length of a REN-Readout cycle, as described in section 2.2. It is a wired-OR line, asserted by all participating slaves in response to AS* and released by each slave as its data is read out. Its rising edge therefore indicates that there is no more data to be read.

2.1 Signals **not** *used for the REN-Readout cycle*

A06 - A31 These signals are not needed, and are not driven (by the Readout Controller) – so as to minimise the electrical noise generated during a REN-Readout cycle. Note that they will therefore be at logic 1 during the cycle, due to the VMEbus termination resistors. However, REN-Readout slaves do not need to monitor these lines; and any 'standard' VMEbus module which is accidently addressed should not respond to Address Modifier 1B.

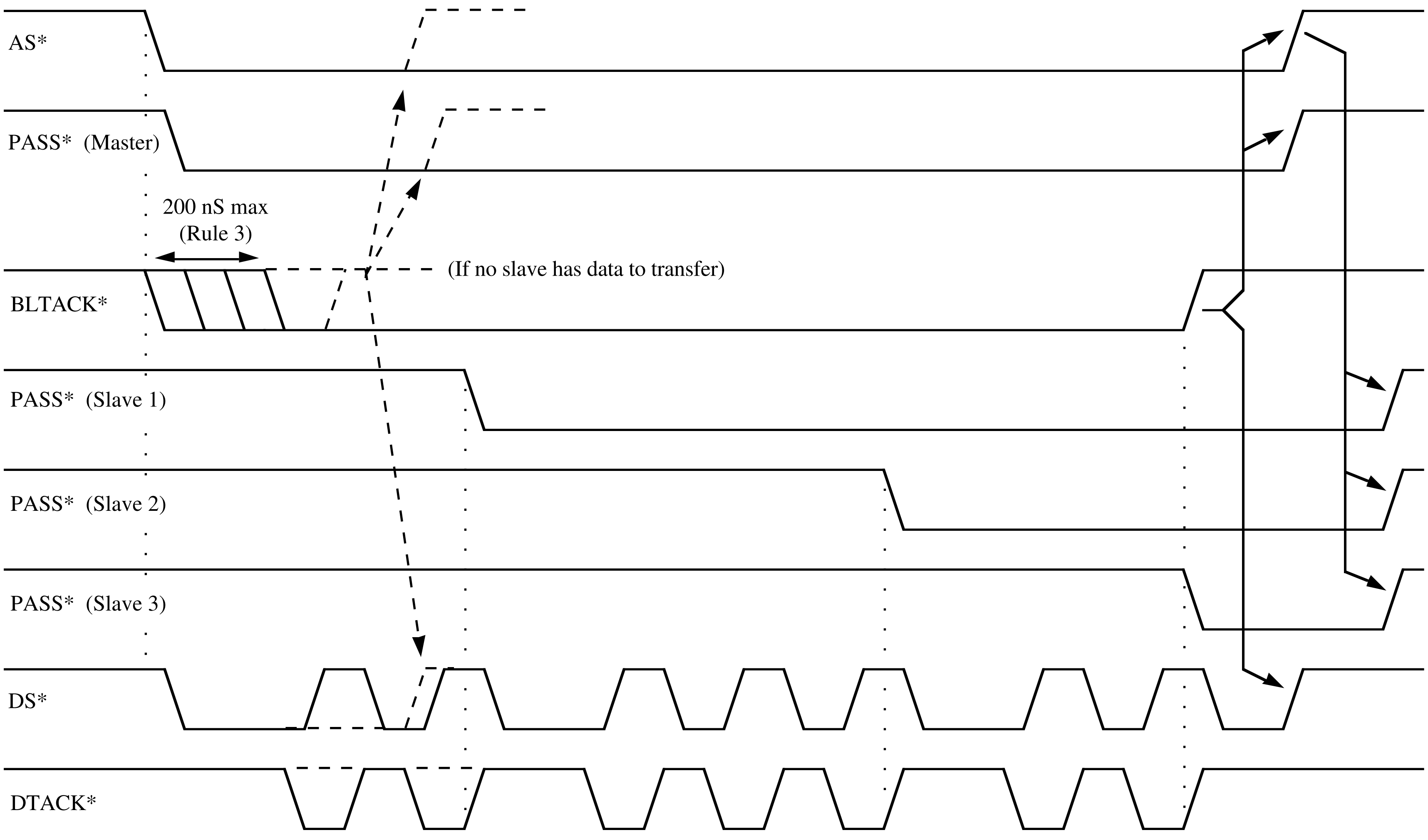
BERR* This signal is not driven by any *slave* during a REN-Readout cycle:

a) There are no error conditions for a participating slave to report during the cycle. (A slave without data to transmit simply asserts PASS*, and releases BLTACK* if it has not already done so, as soon as it receives REN*.)

b) A standard VMEbus module does not participate in a REN-Readout cycle at all, and therefore does not drive *any* of the VMEbus lines.

However, the VMEbus 'bus timer' (implemented in the Resource Manager module) *will* drive BERR* if a slave module fails to respond to the data strobes within the timeout period. The Readout Controller should treat this as a normal module failure, terminate the cycle, and inform the controlling software that an error has occurred.

Fig 1 Illustration of the REN-Readout cycle protocol



2.2 Description of the REN-Readout cycle

Figure 1 illustrates the cycle protocol. It shows a cycle in which data is read from three slave modules: the first and third providing two data words each, and the second providing three data words. (The cycle sequence is only *described* here: see section 3 for the specification.)

1) The master initiates the cycle by placing the appropriate Address and Address Modifier code on the bus (not illustrated), and then asserting AS*, PASS* and the Data Strobes. In response, all participating slaves assert BLTACK* within a certain time.

2) The first slave transfers its data, validating each word with DTACK* – which handshakes with the Data Strobes as in a normal VMEbus block transfer.
   Note that it is permissible (but unlikely) for DTACK* to be asserted *before* BLTACK*.

3) When it has transmitted all its data, the first slave asserts *its* PASS* signal, and releases the BLTACK* line (but the line remains asserted due to slaves 2 and 3).

4) The second and third slaves transmit their data, as in steps 2 and 3.

5) When the BLTACK* line finally rises (after the last slave has released it) the master knows that there is no more data to be read, and it terminates the cycle by deasserting all its signals.

6) In response to the deassertion of AS*, all the slaves deassert their PASS* signals.

The master uses the address lines A02 - A05 to indicate which event it is reading during a particular REN-Readout cycle[2]. If none of the slaves has data for this event then BLTACK* might not be asserted at all, or it might be asserted and then released again when the slaves have determined that they have no data for the event. This is illustrated by dashed lines in Figure 1.

---

[2] In Eurogam, the Master Trigger module broadcasts a 4-bit event number at the same time as the Validation signal, thus 'labelling' the event. The readout controller broadcasts this 'label' on A02 - A05 during event readout.

Note that in phase 1 of Eurogam, Ge and BGO cards neither store event labels nor take notice of A02 - A05 during event readout. For this reason the Master Trigger module ensures that an event has been read out before it validates the next one.

3. *Specification*

Rule 1
A REN-Readout cycle shall conform to the VMEbus specification for Quad-Byte Block-Read transfers, except where modified by the following rules.

Observation 1 VMEbus Rule 2-12 prohibits block transfers which cross any 256-byte addressing boundary. This limits a block transfer to 64 longwords *if* the transfer accesses bytes which are normally accessible at consecutive VMEbus addresses. This is not the case for the REN-Readout cycle which, therefore, is not limited in the number of transfers which can be made.

To aid clear presentation, the remaining rules are split into 4 sections:

Cycle initiation
Assertion of PASS*, and release of BLTACK*
Data transfers
Cycle termination.

3.1 *Cycle Initiation*

Rule 2
The Readout Controller shall transmit the signals whose source is ‘M’ in Table 1. Their values shall be as shown. It shall *not* drive Address lines A06 - A31. It shall assert its PASS* signal with the same timing as AS*, ± 10 nS.

Observation 2 The Readout Controller transmits the Event Number, on A02 to A05, to specify which event it is reading. In Eurogam it must read the oldest event not yet read out, because Eurogam VXI modules store events in time sequence, in FIFOs.

Rule 3
When a slave module which is capable of participating in a REN-Readout cycle receives AS* asserted together with Address Modifier code 1B, it shall assert the BLTACK* signal within 200 nS, unless it is certain that it will not have data for the event being read.

Observation 3 All slave modules with data for the event being read will assert BLTACK* when the cycle commences; and each will release the signal when it has transmitted that data, as specified in the following section. (A module with no data for the event will either not assert BLTACK* at all, or will release it when it has had time to determine that it has none.) Thus the rising edge of BLTACK* indicates to the Readout Controller that there is no more data to be read, and that it can terminate the cycle.

3.2 Assertion of PASS, and release of BLTACK**

The following two rules specify conditions which must be met before a module may assert its PASS* output and release the BLTACK* line. If the module has data to transmit for the event being read, Rule 8 specifies a further condition which must also be met.

> **Rule 4**
> A module shall not assert PASS* until both the following are true:
> a) It has received the assertion of REN*
> b) It has no data to transmit for the event being read.

> **Rule 5**
> A module shall not release the BLTACK* line until it has no data to transmit for the event being read.

Observation 4 A module has no data for the event being read when either:

1) It is not involved in the event, or
2) It has transmitted its last data word for the event.

Permission If a module is not involved in the event being read, then it need not assert BLTACK* at all. However if it *does* assert BLTACK*, then it may release it again at any time prior to asserting PASS*.

Observation 5 It may be that a module receives REN* while some or all its ADCs are converting data for the event being read. In this case, it must wait until all its ADCs have converted *and* been read out before asserting PASS*. It may, of course, transmit the data from converted ADCs while other ADCs are still converting (providing that it maintains the order of ADC data words in those systems which require this, – such as Eurogam).

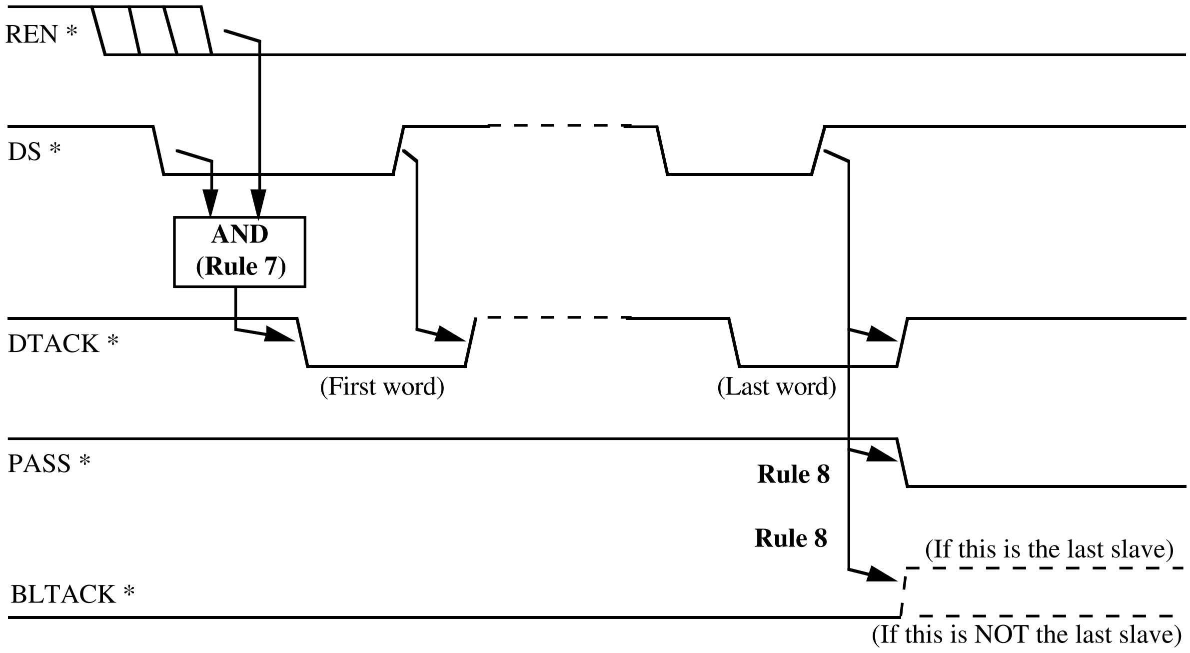
> **Rule 6**
> Once a module has asserted PASS*, it shall maintain it asserted until AS* is deasserted.

Note that there are no specified maximum timings for the assertion of PASS*, or for the release of BLTACK*; but any unnecessary delays in performing these actions will of course reduce the rate at which data can be collected and, for Phase 1 of Eurogam, will increase the system dead-time.

3.3 Data Transfers

Figure 2 illustrates the data-transfer protocol which slaves must observe. The protocol is specified by Rules 7 and 8 together with the normal data-transfer rules in the VMEbus specification.

Fig 2 Illustration of the REN-Readout data-transfer protocol (at a slave)



Rule 7
A slave module which has data to transfer shall use the logical AND of its REN* signal and the VMEbus Data Strobes to gate data onto the bus and to assert DTACK*.

Observation 7.1 Rules 7 and 8 ensure an orderly transfer of operation from one slave to the next.

Observation 7.2 A slave module should only assert DTACK* *after* establishing data on D00 - D31, in order to meet the normal VMEbus timing rules.

Rule 8
A slave module which is transmitting its last data word (for the event being read), shall wait until the Readout Controller deasserts *both* Data Strobes before:

a) asserting PASS*

b) releasing BLTACK*

Observation 8 Rule 8 ensures that:

a) REN* to the next module in the daisy-chain is not asserted while the Data Strobes for the preceding transfer are asserted, and

b) BLTACK* is not deasserted until the Readout Controller has accepted the last data word to be transferred.

3.4 *Cycle Termination*

The Readout Controller will terminate a REN-Readout cycle when it detects the deassertion of BLTACK* (or when BERR* is asserted by the bus timer). It may have already asserted the data strobes, to initiate another data transfer, when BLTACK* is deasserted; but since no module will respond to the strobes it can deassert them, together with all its other signals, without any adverse effects.

Observation 8 The Readout Controller should ensure that it does not terminate the cycle in response to a glitch on the BLTACK* line. (This is a wired-OR line, and is therefore prone to wire-OR glitches, as well as normal noise pick-up.)

> **Rule 9**
> The Readout Controller shall not terminate the cycle until AS* has been asserted for at least 220 nS.

Observation 9 The Readout Controller cannot rely on the state of the BLTACK* signal until this time. (Rule 3 ensures that, if BLTACK* is going to be asserted at all, it will be asserted by a slave within 200 nS of the assertion of AS*.)

> **Rule 10**
> When the Readout Controller deasserts AS*, all modules shall release the BLTACK* signal, and shall deassert their PASS* outputs, within 50 nS.

Observation 10 It is important that when a new REN-Readout cycle is initiated the BLTACK* signal and all PASS* signals have been deasserted: rules 10 and 11 ensure this.

> **Rule 11**
> The Readout Controller shall maintain AS* deasserted for a minimum of 100 nS between REN-Readout cycles.

Observation 11 The VMEbus specification allows AS* to be deasserted for a minimum of 40 ns between cycles.

Observation 12 It is also important, following a timed-out cycle (and this includes normal VMEbus cycles), that the data lines are deasserted in response to the deassertion of AS* – but this is not mentioned in the VMEbus specification!

Appendix A

In phase 1 of the Eurogam project the BLTACK* signal is re-named Readout*, and is asserted earlier than in this specification: it is asserted in response to the Validation signal from the Master Trigger module. Otherwise, its timing and use are exactly as in this specification.

The reason for this is that the Master Trigger module monitors the ‘OR’ of the Readout* signals from all VXI crates to determine when event readout is complete since, in phase 1, it cannot accept a new event until readout of the previous one is complete. Readout* must be asserted in response to the Validation signal, rather than at the initiation of readout, because readout controllers are independent of one another and it is therefore possible for one to complete its readout before another has even begun. The ‘OR’ of the BLTACK* signals would, in this case, be a signal which was asserted and deasserted more than once during readout (of a *single* event); and the Master Trigger module would interpret the first deassertion as indicating completion of readout, which is incorrect.