EDOC001

EUROGAM PROJECT

NSF DATA ACQUISITION SYSTEM

Proposed scheme for readout of VXI modules over VMEbus

Edition 1.0
March 1990

NSF Electronics Development Group
UK Science and Engineering Research Council
Daresbury Laboratory

Eurogam Project: Proposed Scheme for Readout of VXI Modules over the VMEbus

J R Alexander

March 1990

1 Introduction

2 Overview of the Scheme

The scheme essentially implements a 'FERA bus' style of readout, with the following features:

- The data is transferred over the VMEbus data lines.

  (Either one or two 32-bit Eurogam parameters could be transferred at a time. The transfer of two parameters would use the D64 transfer protocol described in the new Revision D of the VMEbus specification; and would require some means of indicating whether both parameters contained valid data — e.g. by use of a unique channel address to indicate an invalid parameter. This is not considered further here.)
- The timing of the transfer of each data word is controlled by the VMEbus Data Strobe and DTACK lines, as defined in the VMEbus specification.
- The Readout Controller (ROC) is VMEbus master during the readout; and the data acquisition modules (Ge, BGO, Trigger, etc.) are all slaves.
- A daisy-chain line is used to determine which slave module should be supplying data to the ROC, in the same manner as 'Readout Enable' (REN) does on the FERA bus. (This does not use a VMEbus line.)
- To other modules on the VMEbus, the readout appears identical to a normal quad-byte block read cycle; except that it may exceed 64 transfers — the maximum allowed for 'standard' block transfers.

The reason for making the ROC master of the VMEbus during the readout is that the data acquisition modules will already have slave logic on them for setup and diagnostic purposes, and the necessary additional logic will therefore be a minimum.

In order to implement this scheme, the following must be defined for what I will call the 'REN-readout cycle':

- The data format. As this has already been defined by the Eurogam software group; here I will only note that it consists of 16 bits of data, 14 bits of address uniquely specifying the channel and ADC within that channel, plus 2 qualifier bits.
- The VMEbus Address and Address Modifier codes used to define that a REN-readout cycle is to take place.
- The way in which the REN line is used:
    1. to select which slave should respond to each individual data transfer within the REN-readout cycle;
    2. the timing relationship between REN and the VMEbus signals.
- The physical implementation of the 'REN' daisy-chain.
- The test facilities which must be provided, within modules, for verification of the REN-readout mode.

3 Readout Initiation

The Readout Controller must become master of the VMEbus (by arbitration, if necessary) before it initiates a REN-readout cycle. It must then broadcast Address and Address Modifier codes which indicate that an event readout is to take place. For this I suggest that we use a 'User Defined' Address Modifier code, as this scheme uses a protocol that is not strictly a VMEbus protocol; and it is important that no standard VME or VXI module responds. I suggest AM code '1B' (c.f. '0B' and '3B' for extended and standard non-privileged block transfers).

For the initial, common dead-time, Eurogam system I cannot see any requirement for decoding the VMEbus Address — the unique AM code indicates that a REN-readout cycle is to take place. In a pipelined system the Address may be needed to identify which event is being read out; in which case a 16-bit address should easily be sufficient.

The ROC must ensure that the VMEbus WRITE* and IACK* lines are high (false) throughout the cycle.

The ROC must maintain control of the VMEbus, by maintaining AS* asserted, until readout is complete (as required during a normal VMEbus block transfer).

4 Use of the REN line

As illustrated in Figure 1, the REN daisy chain must form a loop starting at the ROC, passing through all the data acquisition modules, and returning to the ROC.

The ROC asserts its REN output at the beginning of a REN-readout cycle. If the first module in the chain has data to be read, it does not pass on the REN signal, and it responds to the individual VMEbus transfers by returning 32-bit (or 64-bit) data words. When it has no more data to be read it asserts its REN output to enable the next module in the chain. Any module that has no data to return simply asserts its REN output as soon as possible after its REN input is asserted.

Once a module has asserted its REN output, it must maintain it asserted until the end of the cycle (signalled by the removal of AS*).

Any module which receives a REN input while any of its ADCs are still converting data must not assert its REN output (at that time), and must delay its response to the VMEbus data transfer until it has data available.

When the last data acquisition module in the chain asserts its REN output the Readout Controller will know that readout is complete, and will terminate the cycle.

5 REN Timing

(See example timing diagram, Figure 2.)

When a module has data to transfer during a REN-readout cycle, it must use the logical AND of its REN input and the VMEbus Data Strobes to gate data onto the bus and to assert DTACK* (but of course, DTACK* must not preceed the data — to meet the normal VMEbus timing requirements).

As a consequence of the above, a module that has transmitted its last data word must not assert its REN output until the ROC removes *both* Data Strobes (in response to that module's assertion of DTACK*). This ensures that the REN input to the next module in the daisy-chain is not activated until the Data Strobes from the preceeding transfer are removed.

When the last data word has been transmitted from the last module, the ROC will receive the REN signal at its input, as described earlier. At this time the ROC will probably already have asserted the Data Strobes for the next transfer; but as no module will respond, the ROC can simply terminate the transfer, and the complete readout cycle, as though the transfer had timed out.

When the ROC terminates the cycle (by removing AS*), all participating modules must remove their REN output signals immediately.

All modules must meet the VMEbus timing specifications for quad-byte block transfers, in addition to the above requirements. Note that the VMEbus specification requires that the Address Modifier code remain valid throughout a block transfer, whereas the Address and LWORD* need not. Module designs must take this into account.

6 Physical Implementation of the REN Daisy-Chain

I suggest that we use two of the VXI Local Bus lines to carry the REN signals: one for the daisy chain, and one for the REN return from the last module to the ROC. The latter can be achieved by bussing the return line between the A and C connector pins in each data acquisition module. (Alternatively, the REN return could use a TTL Trigger or ECL Trigger line, if Local Bus lines are at a premium.) This has the following advantages:

- No front panel cabling is required.
- By making the connection to the return line an open collector output from each module, testing can be achieved by programming which module drives the return line — thus temporarily eliminating downstream modules from the readout.

The only restriction imposed by using the Local Bus lines is that the ROC and data acquisition modules must be situated in adjacent positions, or else jumpers must be inserted in unoccupied positions.

I suggest that the daisy-chain should run from left to right, as viewed from the front of the crate, as:

1. This is the same as VMEbus daisy-chain signals — thus saving possible confusion!
2. It ensures that the ROC is nearest to the VMEbus arbiter.

7 Test Facilities

1. As mentioned in the last section, a data acquisition module must have a control bit to select whether it routes its REN output signal onto its REN output pin (control bit = 1), or onto the REN return bus (control bit = 0). This bit could be part of a general Control/Status register within the module. It should be cleared to 0 by SYSRESET*.
2. It must be possible to set each data acquisition module into a test mode, in which it always outputs every possible parameter in response to a REN-readout cycle. Each parameter's 14-bit address field must reflect the value currently programmed, but the other fields may contain random data. SYSRESET* must put the modules into 'normal' mode.

   It must also be possible to trigger the Readout Controller to initiate a single REN-readout cycle, by writing to a test address within it. It is preferable that the data so obtained can then be read out of the ROC over the VMEbus, using standard VMEbus cycles (rather than having to go via the Event Builder).

With the above facilities, it should be possible for software to completely check the readout functions within a VXI crate.

To allow software to check the readout mechanism of the complete system (several VXI crates, the Event Builder, and beyond) all that is required, in addition to the above, is that the Master Trigger unit can generate test triggers under software control.