Eurogam Project: Discussion Paper on Histogramming Module

J. Alexander

May 1990

The following points have already been discussed and, I believe, agreed upon:

- Only those parameters which are part of multi-parameter events are histogrammed—i.e. 'true' singles histogramming is not implemented.
- Histogramming is implemented at a *single* point in the sytem: the histogramming unit, which is connected to the 'DT32' cable-bus between the Readout Controllers and the Event Builder. This unit 'spies' on the event data, and does not take part in controlling the flow of data. (Data flow is controlled by the Trigger Unit, Readout Controllers and Event Builder.)
- The histogramming unit is controlled via the Eurogam Setup and Control Ethernet. Control functions include starting and stopping the collection of histograms, and the selection of which parameters are histogrammed. Parameters are identified by the 14-bit address field in each 32-bit parameter word.
- It is preferable that histograms can be read from the histogramming unit during histogram collection. They are read out via the Setup and Control Ethernet.

It has been suggested that the histogramming unit should be built in VMEbus mechanics, and a suggested layout is illustrated in Figure 1. Here, all units are commercial items, except the Histogramming Module. The number of dual-ported memory modules in the unit depends upon the memory capacity of the modules we select, and the number of concurrent histograms required.

It has also been suggested that the histogramming unit can share a VME crate with the Event Builder. John Cresswell believes this will not be possible, as the Event Builder will use all the bandwidth of the VMEbus backplane. However, the decision can be delayed until a late stage in the design phase.

Data Rates

In order to specify the histogramming unit, one needs to know the rate at which data will be collected. The 'burst' data rate on the DT32 bus is expected to be 10 MHz; but the average rate is not completely specified. At the meeting in Strasbourg (8-9 March) a minimum data rate of 200 K parameters per second was requested; with the possibility of 2 M parameters per second to be considered. These are the data rates required *on magnetic tape, after* any data rate reduction within the Event Builder. The factor by which the data rate will be reduced is not known, but John Cresswell expects it to be about 5. This would give average data rates of 1 and 10 M transfers per second, respectively, on the DT32 bus.

Histogramming Module

The structure illustrated in Figure 2 is proposed for the Histogramming Module. A FIFO ('First In, First Out' buffer) is used between the 10 MHz burst data rate on the DT32 bus and the average, slower, rate at which histogram updates can be made.

A software-programmable look-up table is used to examine the address and qualifier fields of each parameter on the DT32 bus, and inform the control logic whether the parameter is to be histogrammed or not. If it is to be histogrammed, the Base Address of the histogram (also stored in the look-up table) is appended to the data field, to form the address (within VSB memory) of the histogram channel to be incremented. (The format of the VSB address is discussed later.) This address is entered into the FIFO. All the above can be achieved within the 100 nS interval between parameter transfers on the DT32 bus.

When the address reaches the other end of the FIFO, the control logic there uses it to select a 32-bit VSB location, reads the value in that location, increments it by 1, and then writes the new value back. According to work already done at Daresbury for Gamic (the precursor of Euogam), it should be possible to increment a location in a commercial memory board, via the VSB, in 750 nS. Thus it should be possible to handle an average histogram update rate of $1 \cdot 3$ MHz. If every parameter on the DT32 bus is to be histogrammed, then this is also the average data rate which can be handled; but of course a higher data rate can be handled if only a subset of the parameters are to be histogrammed.

To provide a higher rate of histogram updates than this, several histogramming units can be connected to the DT32 bus in parallel. It is then neccessary for software to allocate a subset of the parameters to be histogrammed to each unit, in such a way that no unit exceeds the histogram update rate mentioned above. The histogramming units can all be accommodated in the same VME crate; but each obviously requires its own VSB backplane.

Questions

1 Effect of parameter qualifier bits

What effect do the two qualifier bits associated with each parameter have upon histogramming? The options I can foresee are:

1. No effect—i.e. histogram is updated as normal.

2. A parameter with either qualifier bit set is not histogrammed.
3. A parameter with either qualifier bit set is routed to an alternative histogram or histograms—up to four different histograms could be created, depending on the values of the qualifier bits.
4. The data field is forced to all 1's if either qualifier bit is set, so that 'qualified' parameters are counted in the top histogram channel (which is unused, for sliding-scale ADCs).

Should the option be selectable by software? Should it be selectable on a per histogram basis, or for all histograms simultaneously? (See also next question.)

2 Look-up table size

If the qualifier bits are to be examined by the look-up table (as is neccesary if the answer to the last question is that the effect of qualifier bits should be programmable for each histogram), then the number of address bits for the table is 16 (2 qualifier, 14 address field). State-of-the-art RAM chips for this application are 32 K by 8 bits—i.e. have 15 address lines. Is it therefore permissable (if qualifier bits are to be examined) to ignore the most significant bit of the address field—so as to halve the number of chips required for the look-up table?

3 Generation of Histogram Channel Addresses

The simplest way to generate the VSB address, i.e. the histogram channel address, is to concatenate the 16-bit data field from the DT32 bus with a 14-bit Base Address from the look-up table, as follows:

```
 31                             18 17                              2  1  0
+------------------------------+--------------------------------+------+
|         Base Address         |              Data              | 0  0 |
+------------------------------+--------------------------------+------+
```

(Note that the two least significant address bits are set to zero, as the channel value is a 32-bit, i.e. 4-byte, number.)

This scheme leads to a rather large memory requirement: if we assume 70 Ge histograms and 350 BGO histograms then $27 \cdot 5$ M channels, or 110 M bytes, are required! However, most of these channels will be empty, since most experimental parameters actually occupy fewer than 16 bits.

I therefore suggest a scheme in which the Base Address overlaps the data field, and the overlapping bits are logically ORed with the corresponding bits in the data field:

```
 31                                        X
+-------------------------------------------+
|              Base Address                 |
+-------------------------------------------+
                                 <---------> (OR these bits)
                                +--------------------------------+------+
                                |              Data              | 0  0 |
                                +--------------------------------+------+
                                 17                             2  1  0
```

This allows software to adapt the base address to be compatible with the number of significant bits in the data field—by putting 0's in the base address where it overlaps significant data bits. **It also requires that data sources, such as Ge and BGO cards, put 0's in the upper, unused, bits of their data fields.**

If 'X' is chosen to be 12, then

- the base address is 20 bits (but see below)
- 6 bits are OR'ed
- the smallest histogram is 1 K channels, which is about right for BGO histograms.

Using the numbers of histograms mentioned above, and assuming 8 K channels per Ge histogram, then the memory requirement becomes 910 K channels, which is less than 4 M bytes.

20 bits for the Base Address width may seem unneccessarily large, but by using state-of-the-art RAM chips (32-K by 8) can be achieved with only 3 chips—leaving 4 bits for control outputs. If we reduce this to 2 chips, and allow 2 bits for control outputs ("histogram this parameter", and "force data to all 1's"), then 14 bits remain for the base address. This will allow the module to generate 16,384 1-K histograms, 2,048 8-K histograms, or 256 maximum size (64-K) histograms. While this may be sufficient for our immediate needs, it seems unneccessarily restrictive—*if* there is room to accomodate the third RAM chip in the module.

4 Ge Energy ADCs

In theory one could, within the Histogramming Module, combine the parameters from the two (or three) ADCs associated with a Germanium detector's energy channel into a single histogram. In practise, I believe this would be prohibitively difficult, unless the channel gains are related by an exact factor of 2. I therefore suggest that this function is best performed in software, whenever a histogram is read out to be displayed.

5 FIFO overflows

How should overflows of the Histogramming Module's FIFO be treated? In view of the fact that histograms are statistical information, I suggest that it is sufficient to keep a count of the number of parameters that have been lost due to the FIFO being full—this will allow the system's performance to be monitored.

Is there also a need to implement a VMEbus interrupt to indicate that an overflow has occurred?

Setup, Control and Testing

The Histogramming Module is set up, controlled and tested from the VMEbus. Thus it is a VMEbus slave, but a VSB master. The only setting up required is to load the look-up table; unless control of the module's response to the qualifier bits is to be global—in which case an additional control register is required for this purpose. The module can be enabled (for histogram collection) or disabled, by software command. Access to the look-up table is only possible when the module is disabled, and includes writing to the table and reading it back for checking. Other operations on the module, for hardware testing, will be included where these are felt to be desirable and feasible. An example is the ability to read the FIFO output by a VMEbus operation.