MiniBooNE Event Logic

Local application

Fri, Feb 21, 2003

For MiniBooNE, data streams are used for amassing the data from a series of 15 Hz cycles. It is desirable to be able to collect the data records from all `0x1D` event cycles following the last one of a series of such cycles. This note describes how a clock event can be manufactured to allow delivery of reply data following any sequence of a selected event.

The initial scheme of collecting such data used event `0x11` for delivery, counting on there being a `0x11` event that follows a sequence of `0x1D` events. But `0x11` cycles are any for which no Booster beam is accelerated, so there are many of them and thus replies can occur at 15 Hz. A problem with 15 Hz replies is that the Acnet Vax-based applications may miss some replies when they come at 15 Hz; *i.e.*, delivering 15 Hz replies to a Vax application is unreliable.

In addition to the 15 Hz delivery being unreliable, most of the `0x11` replies will be empty. The full reply will be sent, but its header will indicate that no data record is actually available within the reply buffer. When this was noticed, about 5K bytes were being collected in this way. Although the IRMs can certainly deliver 5K-byte replies at 15 Hz, it is better if they do so when the effort is actually fruitful.

A suggestion was made to improve this situation. Suppose we write a local application that "makes up" a clock event that is asserted only on the cycle immediately following a sequence of `0x1D` events. It would not have to be a real clock event that is imposed on the Tevatron clock signal; rather, it could be a software clock event only known internally to the node. A bit map is maintained in these front ends that marks all clock events that occurred during the previous 15 Hz cycle of front end activity. Event-based data requests are fulfilled based upon the contents of the bit map. A clock event can be manufactured for the current cycle by merely setting a bit in this bit map. Each 15 Hz cycle, local applications are executed before outstanding data requests are fulfilled.

This suggestion was followed, and the result is a new local application called `MBEV`. It has two parameters of interest. The first is an event for which sequences are monitored. The second is the target event that is to be asserted only on a cycle following a sequence of the first event. It was installed in `node0642`, which is the MiniBooNE target front end. The first parameter value was set to `0x1D`, and the second was set to `0xFD`, an unassigned Tevatron clock event.

For a `RETDAT` request that asks for 1000 bytes of data stream records, each of 96 bytes in length, based on clock event `0xFD`, the replies during a test showed that 9 records were delivered within each reply. That corresponded to the fact that sequences of 9 event `0x1D` events were occurring at that time.

This scheme may find application elsewhere to mark a cycle following a series of one or more cycles of a selected Booster reset clock event.