Zero deadtime event readout for experiments at the Cyclotron Institute

K. Hagel, A. Hannaman, A.B. McIntosh, M. Sorensen, A. Abbott, A. Jedele,
M.K. Youngs, and A. Hood

Event readout in experiments at the Cyclotron Institute have historically suffered from deadtime issues resulting from the inability of the data acquisition system to acquire events while event readout is in progress. Many modern data acquisition modules, however, support multievent readout. Some modules allow the acquisition of a number of events. When the so called multievent capability is activated, the acquisition can benefit from fast streaming where larger readout lengths consisting of several events can be read out more efficiently than multiple readouts of single events with a shorter lengths. Other acquisition modules have multiple readout banks so that acquisition can occur at the same time as readout.

With experiments now exploiting the capabilities of waveform digitizers whose single event readout is relatively slow and the complexity of many experiments requiring significant numbers of waveform digitizers, the deadtime issue became untenable. Corrections for deadtime increased from the historically typical 10-20% to 95% or larger. In addition to the significant corrections required because of deadtime, statistics suffered as only a few percent of events could be accepted into the computer and the rest were ignored.

We therefore implemented a multievent readout into our data acquisition system. This required implementing new infrastructure into the data acquisition and analysis systems. Although the implementation of the multievent readout is designed to be general, the current focus is on the multievent readout of Struck 3316 waveform digitizers [1].

To implement multievent readout in the online data acquisition system, we changed the triggering from using the interrupt of the controller module to searching for the event threshold set on the waveform digitizers to be met on at least one channel of all digitizers used in the experiment. Once this threshold is reached, all events of all modules are read out. This is read into the data acquisition system as a traditional "event", but is tagged in the event header as a multievent. In that way event counters and rates are maintained. Events are written to disk as well as sampled to the online analysis monitor in the same way as was used for single event readout.

In the backend analysis, the multievent tag in the buffer header triggers a multievent decoding of the buffer. This decoding separates data from all of the channels into lists. The multievent flag then triggers an event building routine which builds events by comparing timestamp differences between the various events of the various channels. After the events are built, the traditional online analysis routines are called.

A similar operation occurs when preparing the raw data files for offline analysis. When the events are converted to root trees [2], the multievent flag triggers the same event building routine and the same infrastructure builds the events into the event data structure that is written into the root trees.

We have tested and then used the multievent readout in two experiments. The first was a NIMROD [3] experiment to test the new implementation. Employing 9 Struck digitizers [2] with single event readout limited our data rate to about 200 events per second with 90-95% deadtime. After implementing the multievent readout described above, we achieved 4000 events per second with no deadtime using 100 multievents per readout. Increasing the beam rate above that did not yield more events because that was the point where the events were accumulating faster than they could be readout due to the limit of our data transfer rate.

Another experiment employed 18 Struck digitizers and 3 Mesytec Madc32 [4] peak sensing ADCs in two VME crates. In this experiment, we employed an external clock to synchronize the digitizers in the different crates as well as exploiting the multievent and external clock capability of the Mesytec ADCs. A test experiment leading to the one discussed here employed four digitizers using single event readout. That test experiment suffered significant deadtime and a limit of a few hundred events per second. We were able to achieve 2000+ events per second with the multievent readout of 18 digitizers and 3 Mesytec ADCs. The limit of 2000 events per second was driven by the limit of beam that could be used with the forward detectors rather than the data acquisition limit.

We note also that multievent readout can also be useful in experiments where single event readout is used. One example is the neutron ball (NBL) [5] readout. That readout has typically been done counting triggers from the PMT signals with a 100us gate that results from the experimental trigger to count signals from the reaction followed by a 100us gate to count signals from the background. We have tested the NBL readout with a fission source using digitizers with multievents. We used a single event trigger from the fission fragment to open a 200us gate which allowed all signals to be counted and timestamped during that 200us. Using this method, signals the first 100us as determined from the time stamp can be used to establish the signals from the reaction and from the second 100us to establish the background. This method can also be used in multievent readout as described above and the timestamps are used to match the NBL signals to each event with a 200us gate instead of the much smaller gate when, for example, CsI signals are matched to build an event.

Another application using multievents with single event readout and achieving zero deadtime involves studying decaying nuclei. If, for example, events do not need to be matched, a multievent number of, say 20, can be used to detect with zero deadtime all events that occur and the results can be analyzed according to timestamps even with single event readout. This technique was exploited in a recent study of multinucleon transfer [6] where alpha particles emitted from heavy products implanted in an active catcher were detected.

The primary benefit of multievent readout is that much more efficient readout of large parameter experiments can be achieved. This increases both the statistics accumulated in the experiment in addition to increasing the efficiency of Cyclotron usage given the significant demand of beamtime. In the future, the multievent capability of more acquisition modules will be implemented into the software. As we progress to ever more complex experiments, it may be necessary to split the data acquisition into parts and acquire each part using multievent capabilities in parallel and then implement true event building. This development will occur as the need arises.

[1] Struck Innovative Systems, https://struck.de/sis3316.html.

[2] R. Brun, F. Rademakers, Nucl. Instrum. Methods Phys. Res. **A389**, 81 (1997).

[3] S. Wuenschel *et al*., Nucl. Instrum. Methods Phys. Res. **A604,** 578 (2009).

[4] Mesytec, http://mesytec.com/products/nuclear-physics/MADC-32.html

[5] R.P. Schmitt *et al*., Nucl. Instrum. Methods Phys. Res. **A354**, 487 (1995).
[6] A. Hood, *Progress in Research* , Cyclotron Institute, Texas A&M University (2020-2021), p. II-1.

.