# 1 Data Aquisition (DAQ)

M. Zeier July/Aug. 99

$G_{en}$ was the first experiment in Hall C that used CODA2.x. Previous experiments used CODA1.4. Documentation on CODA can be found on `http://www.jlab.org/coda/`.
A paper from Dave Abbott which discusses coda2.0 → coda2.1 changes and also more recent after-$G_{en}$ developments gives a good overview and is available at `cdaqs2:~cdaq/gen98/Documents/daq/CODA_RT99.ps`.

- **Hardware**:
  The upgrade from CODA1.4 to CODA2.x required a change of the backend system from `cdaqh1` (HP) to `cdaqs1` (Sun). As frontends served VME single board computers (running under VxWorks) located in the various FASTBUS and VME crates, which were connected over a fast Ethernet network with `cdaqs1`. The CPU's in the FASTBUS crates (for ADC and TDC readout) were VME ppc's (Motorola MVME 2300) in a Struck Fastbus Interface, which is a mini VME crate that goes into a FASTBUS crate. The CPU's in the VME crates (scaler readout) were 68k's (Motorola MVME162). A complete list of the frontend systems is given in
  `cdaqs2:~cdaq/gen98/Documents/daq/roclist.txt`.

- **CODA Processes**:
  CODA is running Read Out Controllers (ROC) on the CPU boards in the FASTBUS and VME crates, which are responsible for programming and readout of ADC, TDC and Scaler modules.
  On `cdaqs1` different coda processes are running:
  *Run Control*: Main Control graphics interface. Allows the user to select run types and execute CODA transitions.
  *rcServer*: Does the connection between components and Run Control.
  *Event Builder* and *Event Recorder*: Build an event and write it to disk.

- **CODA usage, transitions and states**:
  How CODA was run during $G_{en}$ is documented in
  `~cdaqs2:~cdaq/gen98/Documents/daq/running_daq_gen.ps`

- **Readout Lists**:
  The user written readout lists, which determine what a ROC does during which CODA transition can be found in:
  `cdaqs2:~cdaq/gen98/coda2/crl/` CODA2.0 readoutlists, not using the SFI sequencer
  `cdaqs2:~cdaq/gen98/coda2/crlseq/` CODA2.0 readoutlists, using the SFI sequencer
  `cdaqs2:~cdaq/gen98/coda2/crl2.1/` CODA2.1 readoutlists, using the SFI sequencer

- **Trigger Activity and Data Flow**:
  In case of an event the Trigger Supervisor (see section about electronics) passes an event type (which is dependent on the trigger pattern) to the ROCs. The ROCs execute the trigger section of the user written readout lists, which specifies the readout for each event type. Each CPU buffers the data in allocated memory space. The event fragments from the various ROCs are transferred to the Event Builder (running on `cdaqs1`) via 100BaseT Ethernet, where the event is built and sent to a memory system (DD system). At this point nonphysics events are inserted into the data stream. The Event Recorder consumes the data from the DD system and writes it to the data disk. Available storage are four 17 GB data disks. A process in the background is constantly backing up the data to the silo.

- **Triggers**:
  Physiscs triggers were disabled during the first seconds of a run when pedestals were taken and during each period of potential helicity transition (PHT) for $\simeq 600\mu s$. The scaler readout trigger for the helicity gated scalers occured during each PHT, i.e. every second. The other scalers were read out every 5 seconds, which was initiated by a forced sync in the Trigger Supervisor. Details about the trigger scheme are available in:
  `cdaqs2:~cdaq/gen98/Documents/Timing/ts_setup.ps` and
  `cdaqs2:~cdaq/gen98/Documents/Timing/gen_ts7.ps`
  `cdaqs2:~cdaq/gen98/Documents/Helicity/scalers.ps` shows the scaler trigger scheme.

- **Run Types**:

CODA allows the user to specify different run types, defining the hardware which is involved and the readout lists which are downloaded. During $G_{en}$ we had three different run types, a *main* run type involving all detector systems, a *cosmics* run type to take cosmics data with the neutron detector and a *moller* run type to measure the beam polarization.

- **Event Types**:
  Physics event types were based on the trigger pattern in the Trigger Supervisor. For details refer to Electronics section. The following gives the basic idea.
  0: Asynchronous scaler readout
  1: HMS Single
  3: eB coincidence
  4: Pedestal
  5: Helicity gated scaler readout
  6: Laser
  7: Cosmics
  13: Illegal Trigger pattern

  Nonphysics events:
  129: Final scaler event (only there when run ended properly)
  130: Run info event (Run sheet that pops up at run start)
  131: Epics event
  133: Go_info event (Information about run mode)
  140: Polarized Target Event ("Labview Event")
  141: NDET HV event

- **Run Mode**
  All physics runs were taken in unbuffered mode. ADC sparsification was turned on for the HMS side but not for the NDET. Typically the HMS singles were prescaled by a factor 2 to 8. The run mode was determined by flags as specified in the following places:
  (1) flag file: `cdaqs2:~cdaq/gen98/coda2/gen.flags`
  (2) Header Files: `cdaqs2:~cdaq/gen98/coda2/crl2.1/*.h`
  (3) CODA run type configuration (keywords following the readout list).
  The flags can be specified at more than one place, but (1) overrules (2)

and (2) overrules (3).
The contents of (1), (2) and (3) go into the data stream with the Go_info event as well as the thresholds used for ADC sparsification.

- **Performance**:
  Some typical numbers for $q^2 = 0.5$:
  Event size: 2.3kB
  Event rate (100 nA): 390 Hz (for ps1=4), 300Hz (for ps1=8)
  Computer dead time: 10% (300 Hz)

- **Experiment History**:
  At the beginning of July Sequencer programming of the SFI's was introduced. This made the readout of the FASTBUS crates considerably faster.
  After run 21980 (9/4/98) CODA2.0 was upgraded to CODA2.1.

- **Outlook**:
  A new Trigger Supervisor (TS) is under construction, which provides improved possibilities for "scheduled triggers". This could be used for a better scaler trigger scheme. During $G_{en}$ the PHT blank period was increased to 600 $\mu$s and the scaler (helicity gated) trigger occurred towards the end of this period to increase the probability that the TS is not busy and the trigger will be accepted. On the other hand the TS forced sync was abused for the asynchronous scaler readout. Later we realized that in rare cases the asynchronous scaler readout blocked the helicity gated scaler readout trigger.
  The new TS will have an external input for a "scheduled trigger", which is accepted even when the TS is busy. Using this input for the scaler trigger (helicity gated) will guarantee that no scaler read gets lost.

  For the next running of $G_{en}$ we should make sure that the DAQ can run in buffered mode. In this mode scalers have to be read out at each sync point for later corrections in case of sync errors. Ideally the sync check would occur during PHT when the scalers are read out anyway.

  Make sure that all frontends are ppc boards.

Nothing speaks against 30 Hz helicity flip rate, except perhaps the additional deadtime introduced by the more frequent PHT blank period.