US DHCAL integration with CALICE DAQ

Paul Dauncey

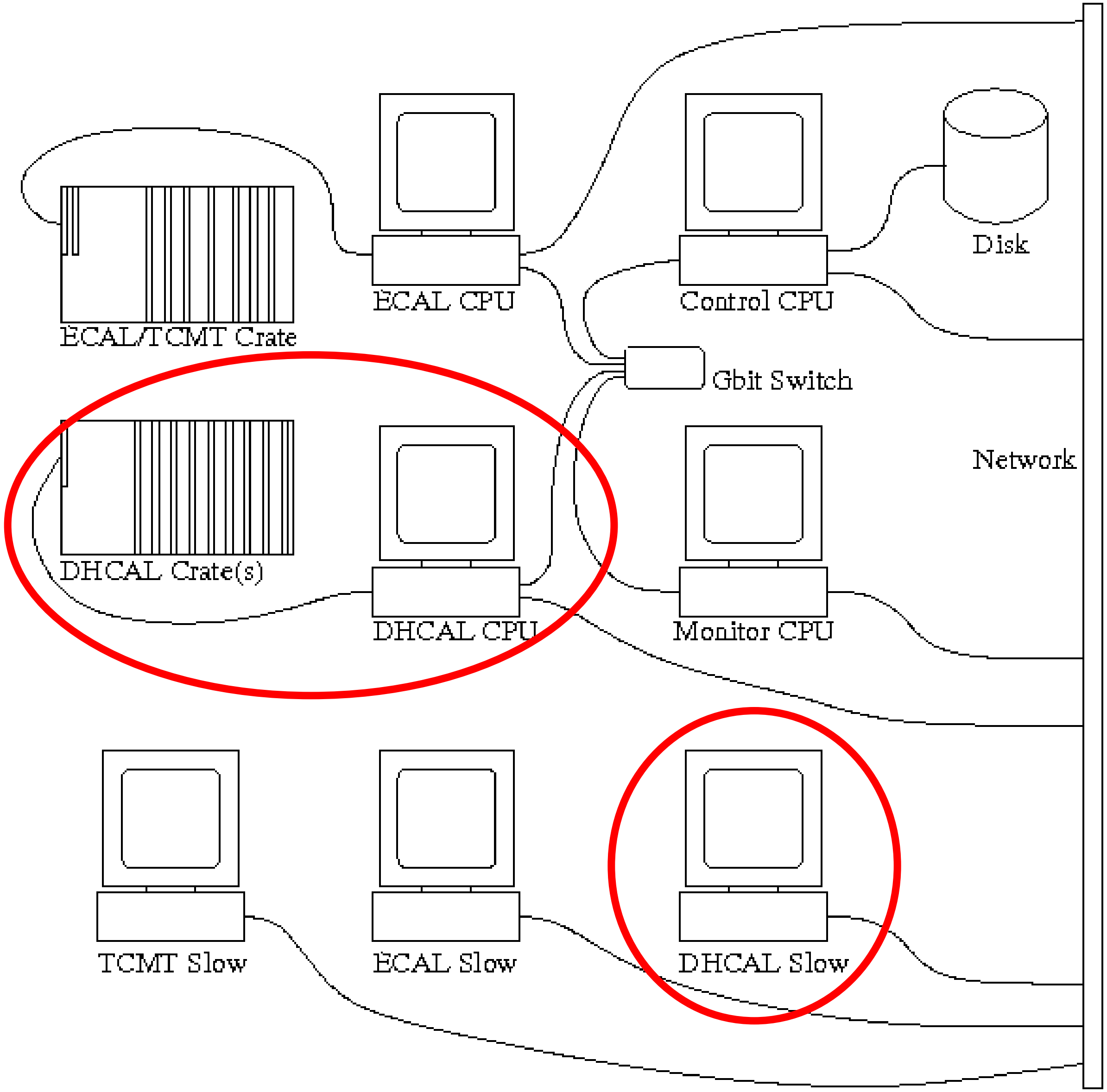
Overview

- Hope to run the US DHCAL together with the SiECAL and TCMT at FNAL in summer 2010
    - Requires integration of DHCAL into the CALICE system
- Basic goal of DAQ (online)
    - Get all useful data into the native binary data files
- LCIO converter (offline) then separates data
    - Event data into LCIO run files
    - Configuration, slow controls data into database
- Offline covered by Niels; just consider online here

Boundary conditions

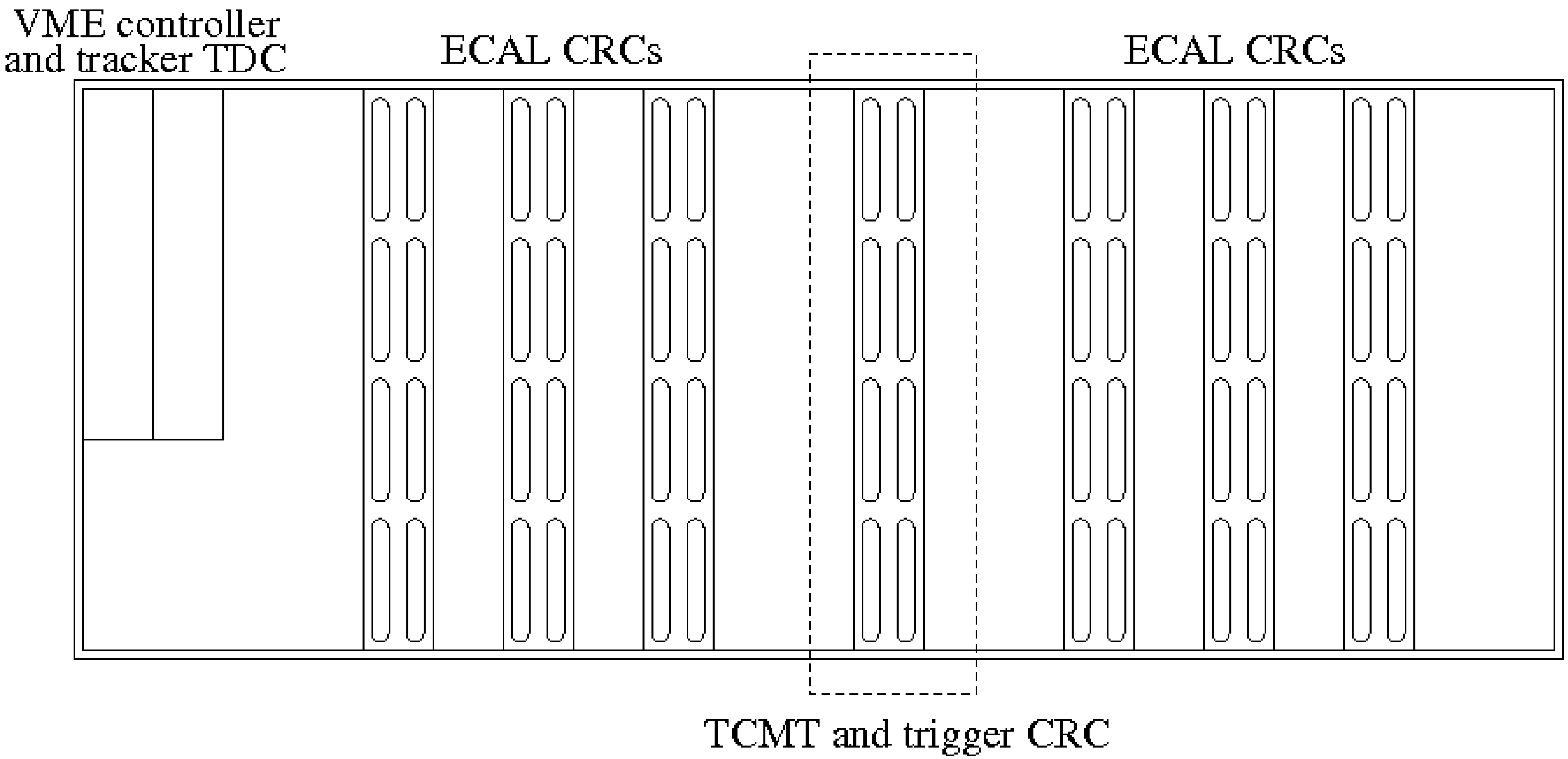
- Output data format has to be native raw data
  - Compatible with other beam test data
  - Must not change the native data structure in a non-backwards compatible way
- DHCAL can run stand-alone or combined
  - Stand-alone will be for early/cosmics tests at ANL
  - Combined is with ECAL and TCMT at FNAL, but initial FNAL runs are likely to be stand-alone also
  - Must ensure the DAQ readout code and data structure from these two modes are identical (or as similar as possible)

System overview



Standalone system

ECAL and TCMT



- Assume will combine CRCs into a single crate
  - Allows "other" VME crate to be used by AHCAL at DESY
  - A simple trick in software ("split crate cludge") allows both systems to appear to have a crate to themselves
  - N.B. cannot run the two systems genuinely independently without two physical crates

DHCAL and triggers

- Potentially more complicated than previous CALICE cases
  - DHCAL is built to be able to free-run, with zero dead time
  - Other CALICE systems have been triggered
  - Getting these two to work together will require some preparation
- DHCAL will not "trigger" on the CALICE central triggers
  - But can timestamp the triggers sent to it
  - All DHCAL hits also timestamped
  - Allows hits to be associated to triggers later
- Must be able to combine ECAL and TCMT event data with DHCAL data
  - Central DAQ has no timestamping or hardware trigger number
  - Lost or extra triggers cannot be identified except by counting
- Need to make data association robust
  - Ensure redundancy to catch missing/spurious triggers

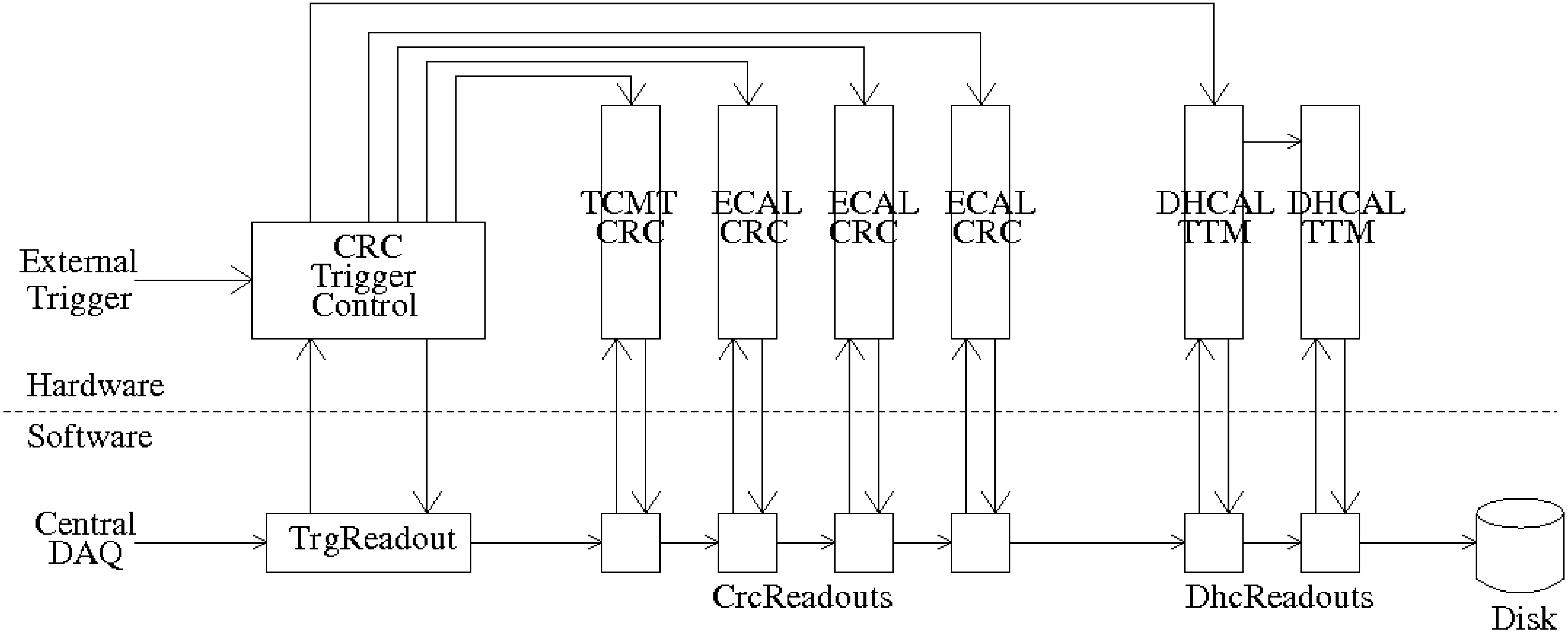
Trigger handling

- DAQ software sends two records for each event
  - Trigger: sent during spill at up to 1kHz
  - Event: sent after spill at up to 100Hz
- These have different uses
  - Trigger records are used to capture data which cannot wait until the end of the spill; must be fast << 1ms
  - Event records capture the bulk of the data; must finish before next spill to have no efficiency impact
- Up to now, DAQ ensures a one-to-one correspondence between a hardware trigger, a software trigger record and a software event record
  - Blocks any second trigger until first trigger record has been completed
  - Ensures trigger counter does not increment during VME crate readout
  - Makes trigger-event record association straightforward
- Ability to do trigger control is in CRC firmware
  - Only activated for CRC in crate slot 12

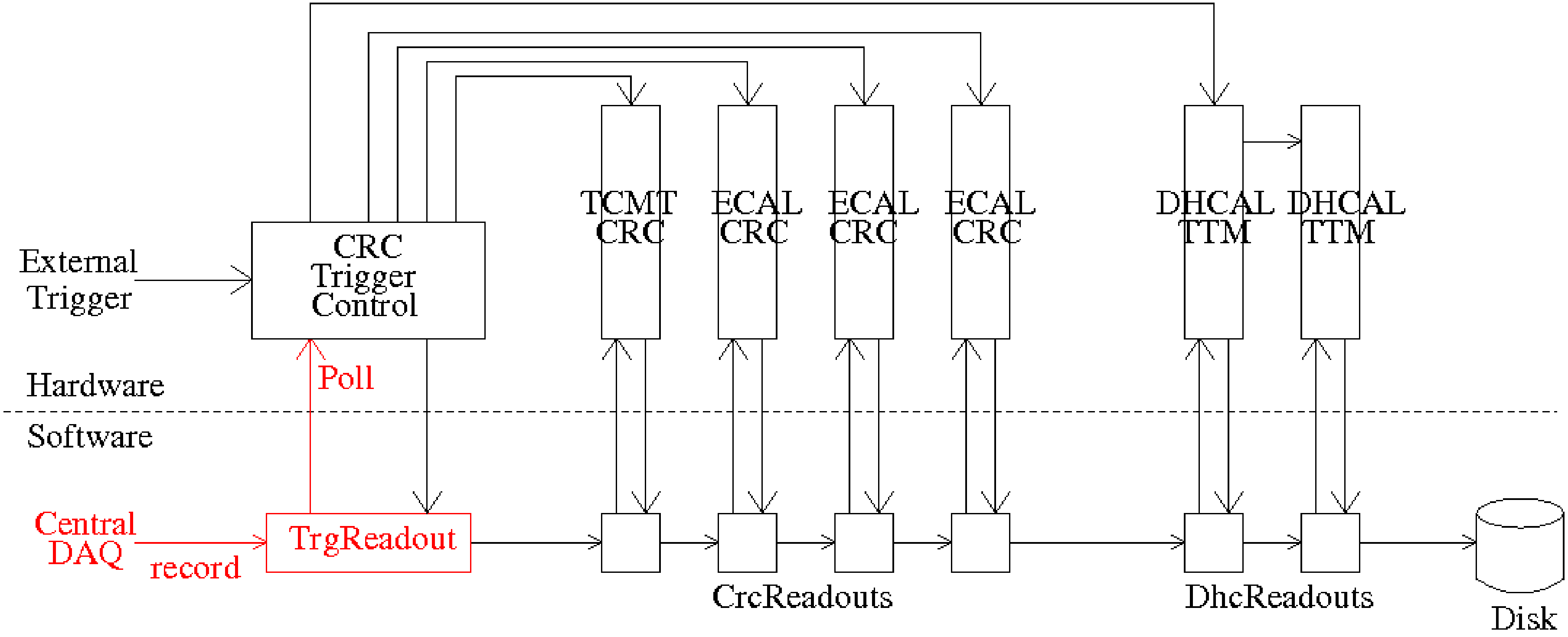
Trigger handling

- For DHCAL, agreed solution for trigger records for combined running
  - Record the number of triggers and timestamp of the trigger from DHCAL
  - If no trigger or more than one trigger, then error. Doing this locates exactly the first trigger with a problem
  - Data from whole spill do not need to be discarded
  - CRCs do something similar already
- To keep system as close as possible
  - Do similar system for stand-alone runs

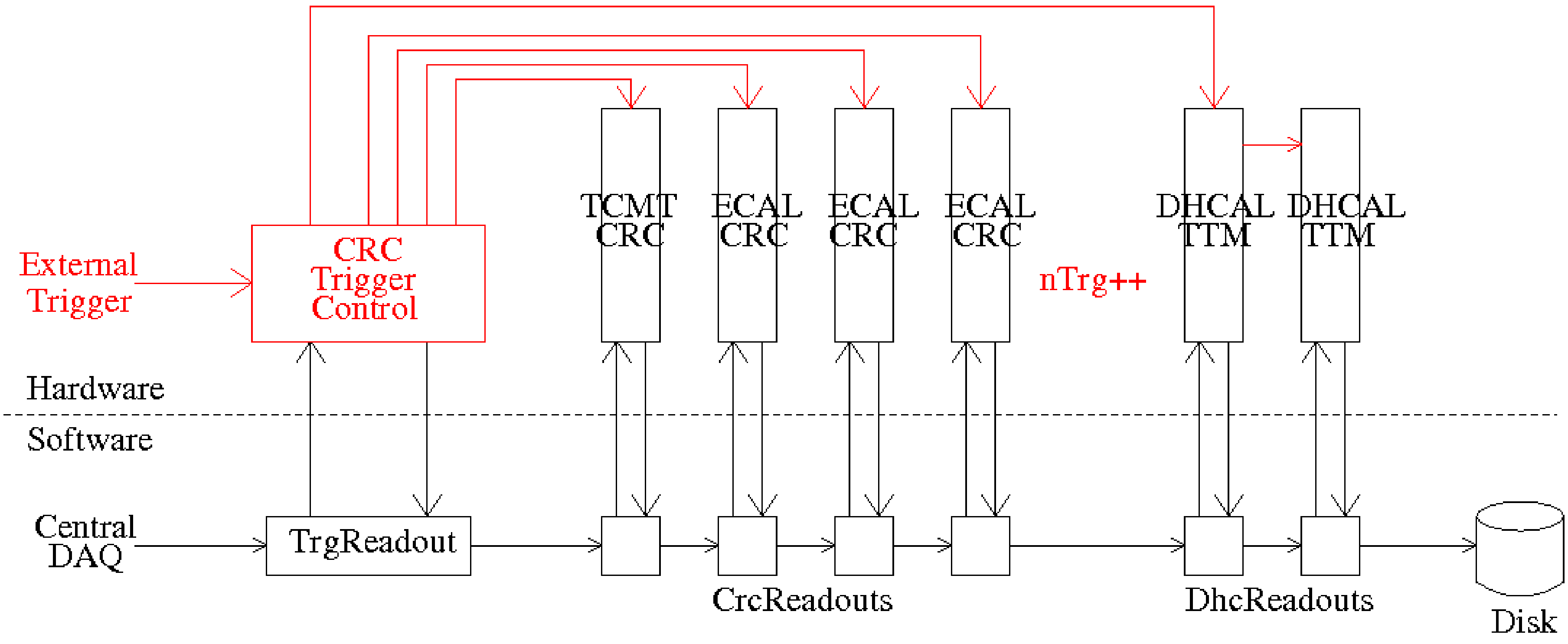
Combined system operation



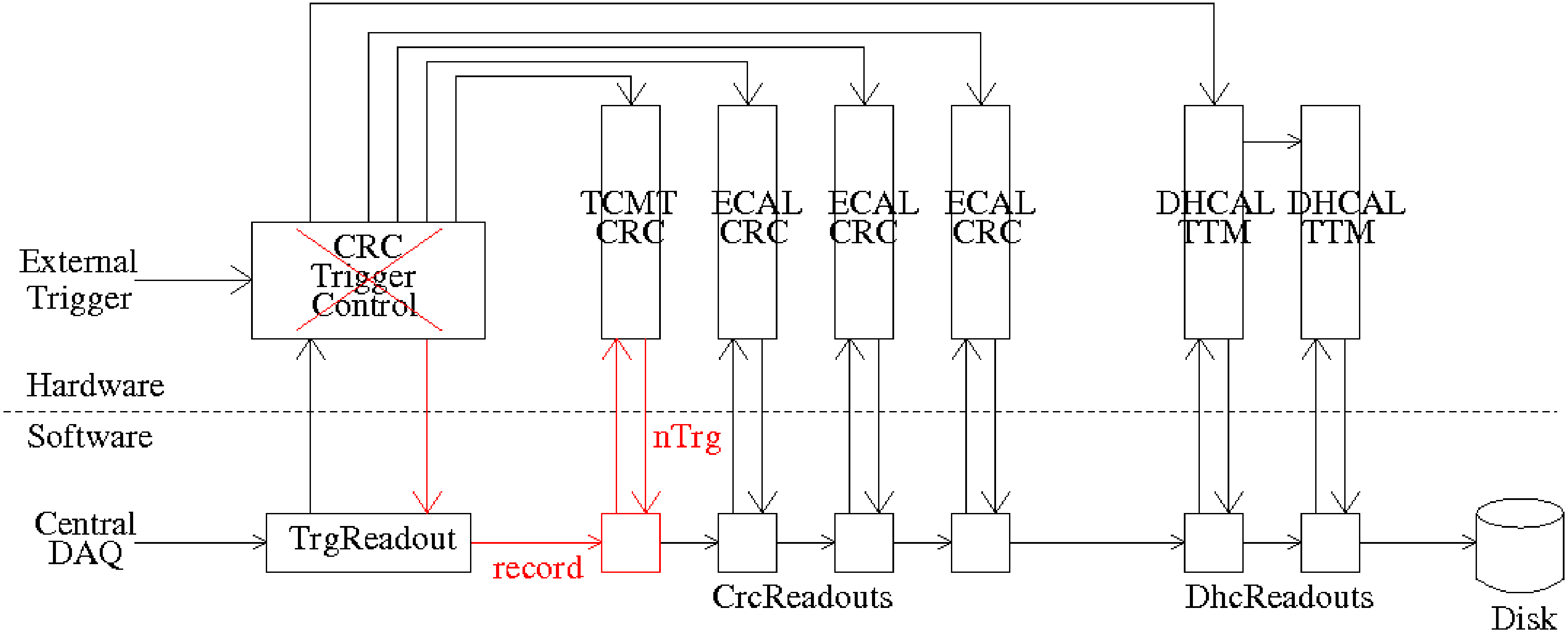
Combined system operation



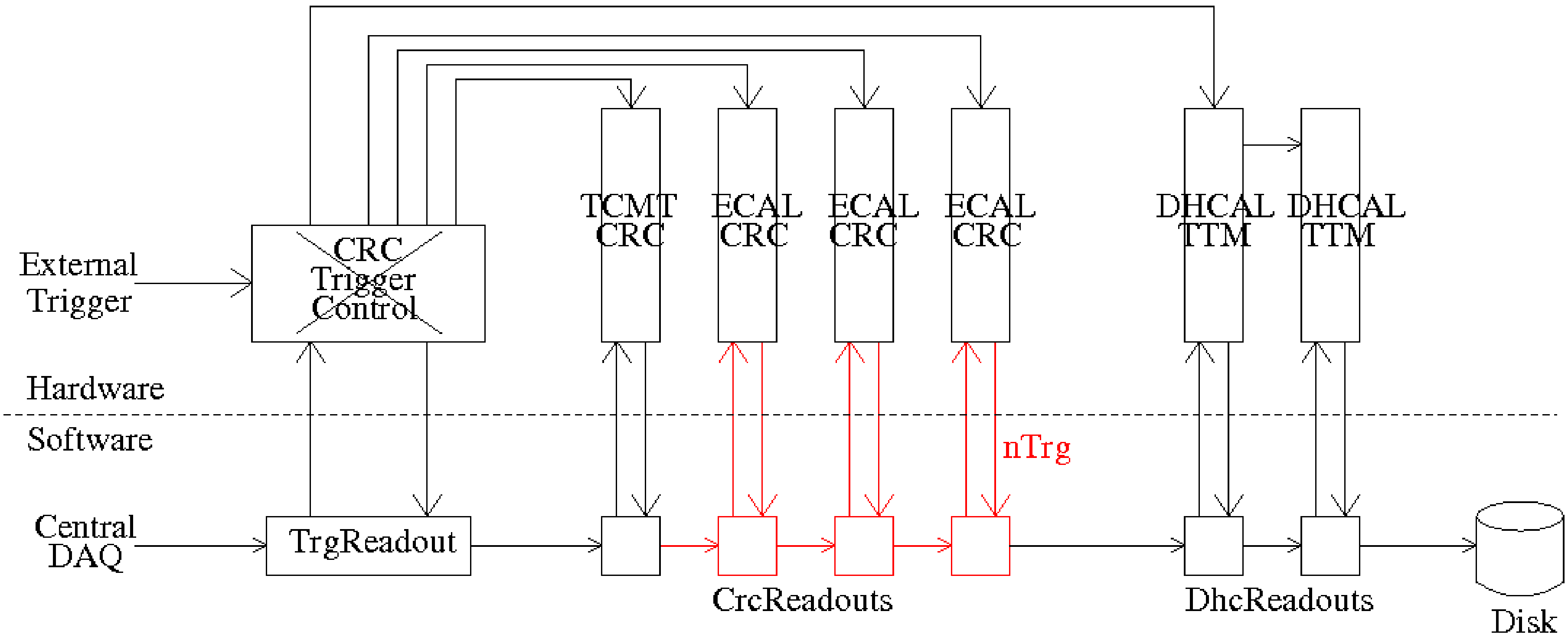
Combined system operation



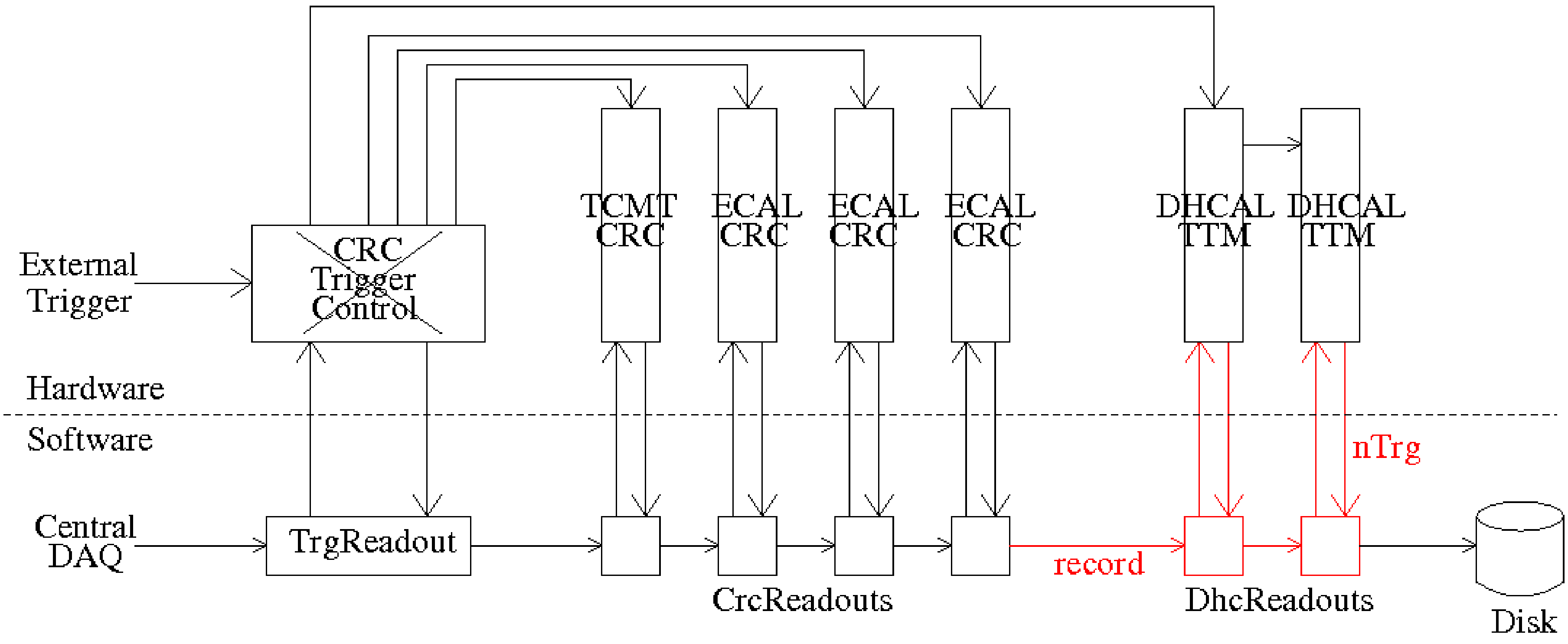
Combined system operation



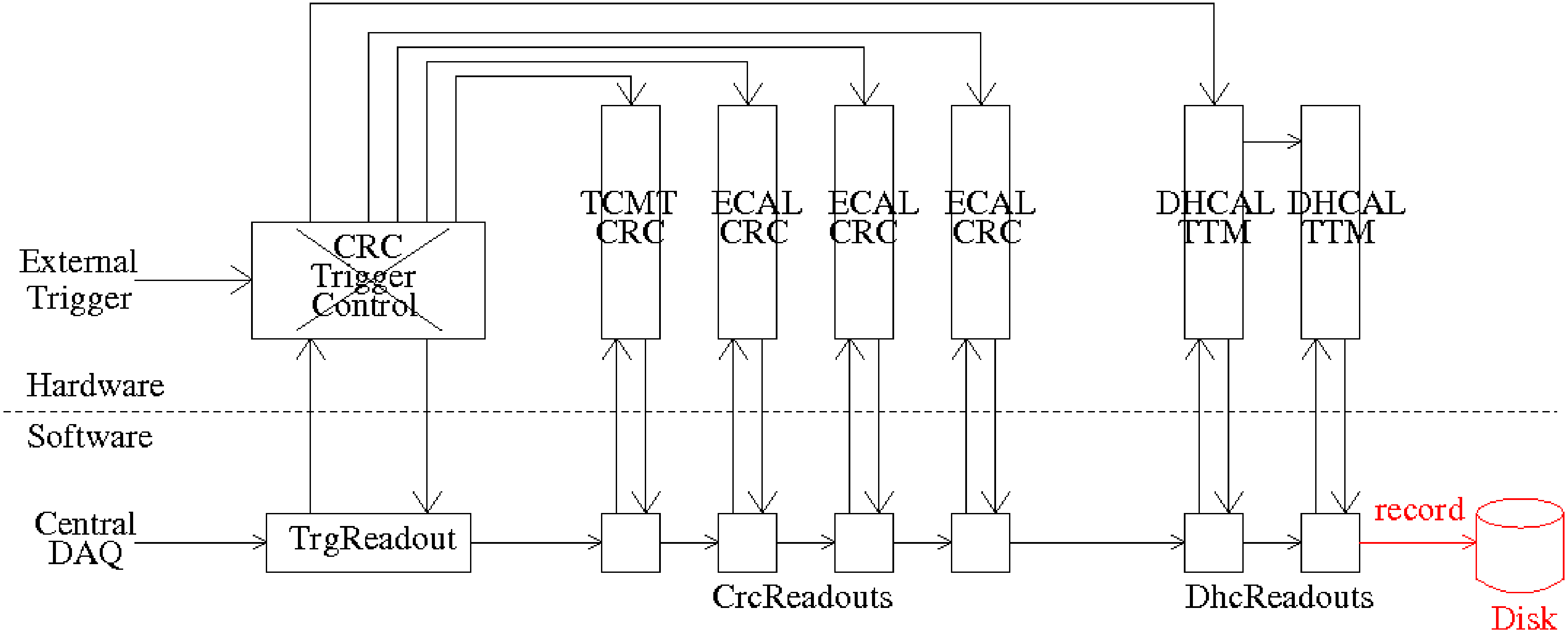
Combined system operation



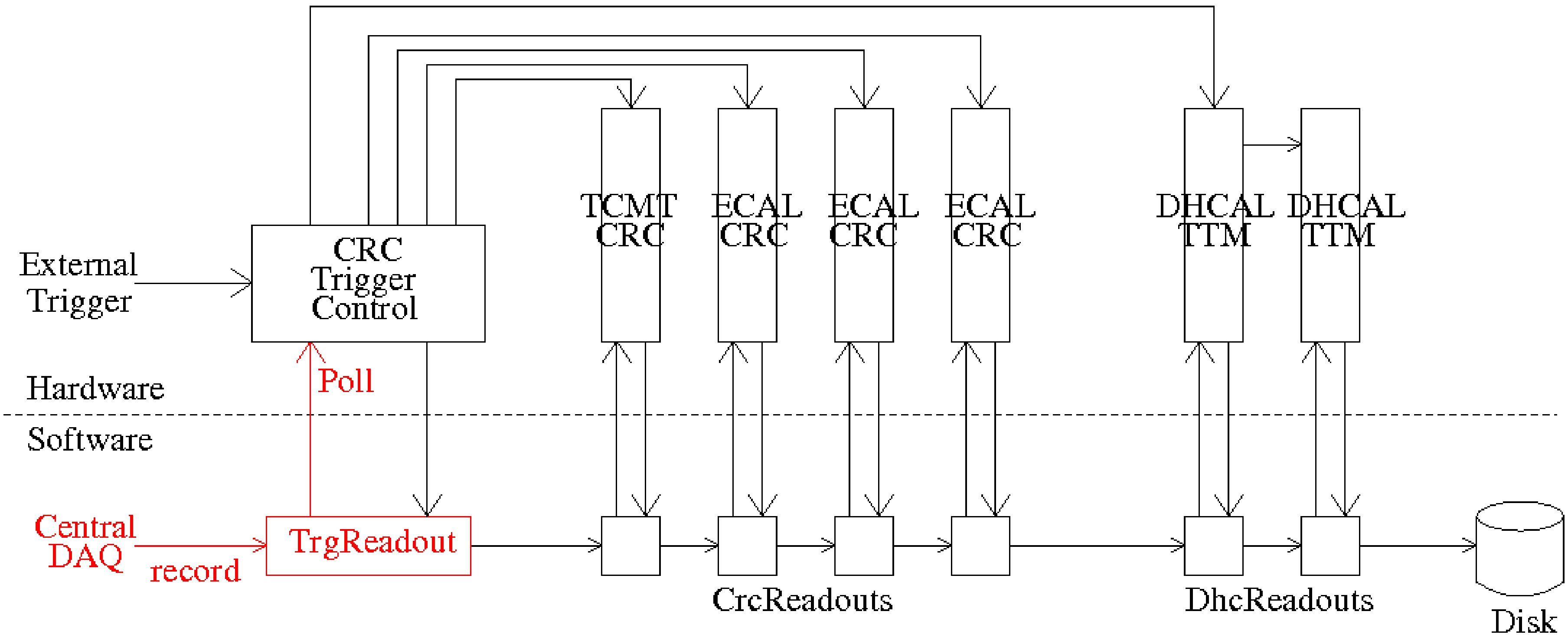
Combined system operation



Combined system operation



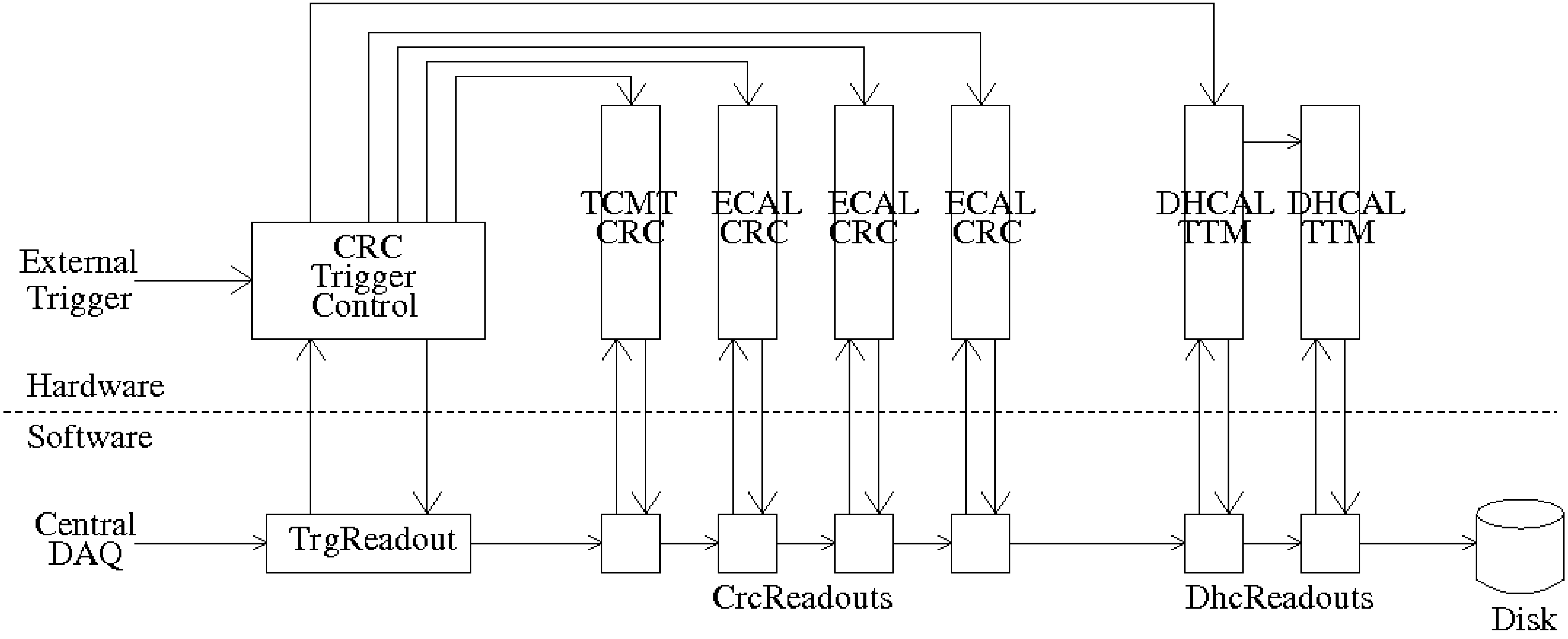
Combined system operation



Triggers in stand-alone running

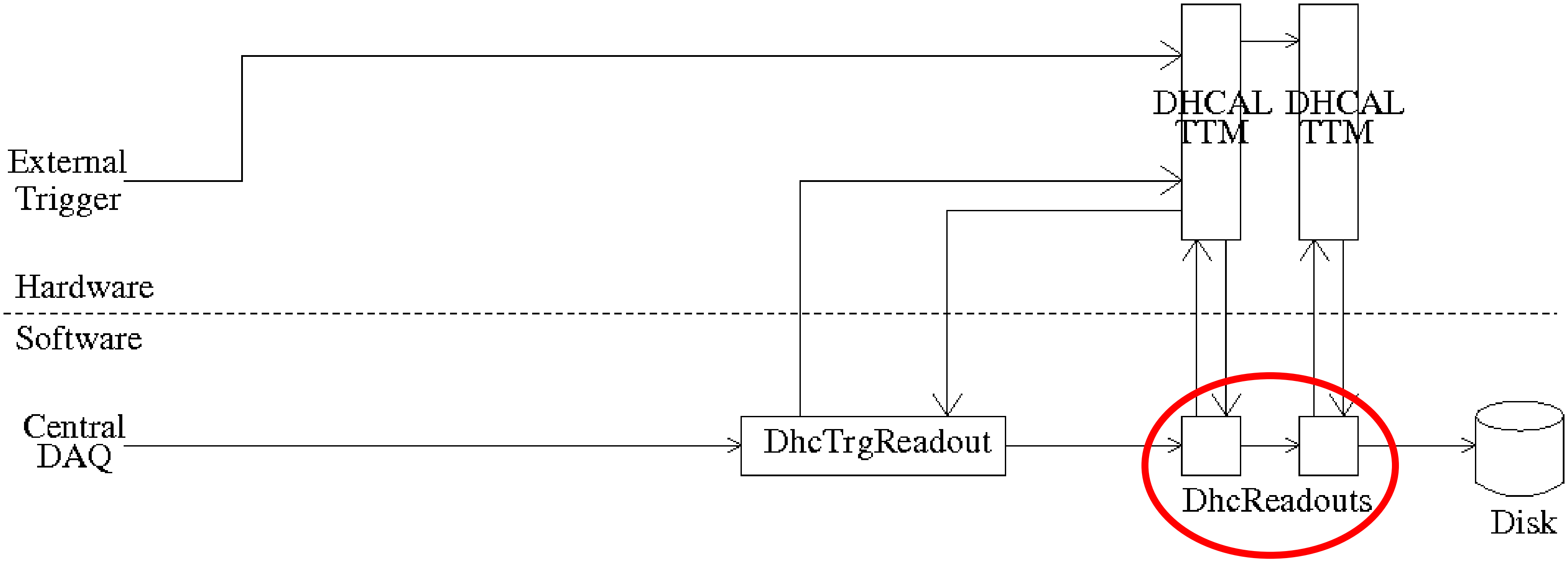
- In principle, no need to block triggers for stand-alone runs
  - System can handle triggers at any time
- But running like this will change data structure
  - No guarantee that only one hardware trigger since last trigger record; trigger number won't increment by unity
  - Could have more physical triggers and events than trigger records and event records; DAQ code different
- Cleanest to keep stand-alone mode as close as possible to combined mode
  - Introduce trigger control into DHCAL stand-alone running
  - Alternative would be to allow more triggers but discard "extra" trigger data and event data associated with them

Combined system operation



Stand-alone DHCAL operation

- Possible solution; control trigger through TTM module
  - Only software change so could be done immediately
  - Unclear if required functionality supported by hardware



- Critical: DchReadout software is unchanged between two modes
  - Doesn’t know or care which mode it is in

Event handling

- For event data several options
  - Ideally only put hits associated with trigger timestamp into corresponding event record, but data format makes this non-trivial
  - Effectively get large volumes of data; up to O(10MBytes) per VME module which need to be interpreted offline
  - Either divide up the large amount of data into chunks and put part into each event record
  - Or dump data from whole spill into acquisitionEnd record
- Constraints from keeping DAQ backwards-compatible
  - Any single data chunk (“subrecord”) must be < 64kBytes
  - Might be possible to increase this as (almost) all data subrecords so far are < 32kBytes (but messy for DECAL)
  - No limit on total record size in principle but incrementing memory to allow for hundreds of Mbytes in one record might break existing code
- Most likely to work/most conservative is breaking into small chunks and put into each event record

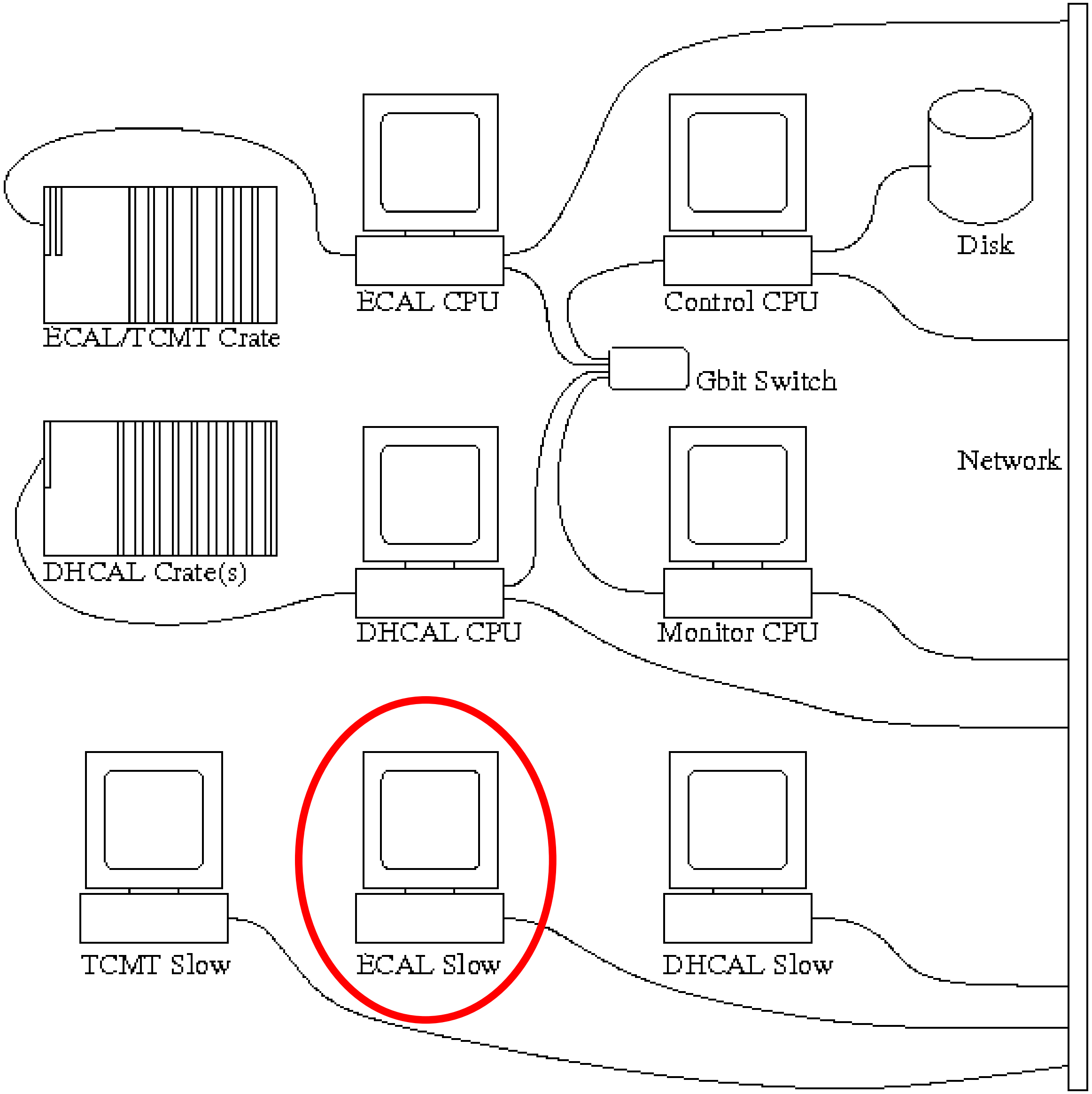
Testing before full combined running

- Critical item is trigger
  - Provided by TCMT CRC; don’t need ECAL to test this
- Stand-alone mode can be developed now, as system is assembled at ANL
  - If no possibility of using TTM, then in principle could bring VME crate and CRC from FNAL to ANL
  - Would require someone with experience of setting up trigger to visit ANL and help assemble CRC system
  - Timescale for this is within a month or so
- One DHCAL plane together with TCMT in the beam at FNAL would test almost all potentially problematic issues
  - Time offsets of two systems will not be measured in stand-alone running
  - Recognised by TechBoard as essential first step; “strongly recommended”
  - Aim for this around Feb 2010
  - Later addition of ECAL then straightforward

Some other items

- Slow controls data
  - Sven's system used for AHCAL and TCMT provides both a nice immediate display of values as well as an interface to the DAQ for recording the values
  - DHCAL probably will use a different system; will need to have DAQ interface
- Software coordination
  - No DHCAL modifications to central software have been returned to the main repository; at least runner.cc and SubRecordType.hh must have been changed
  - Worry that an inconsistent split of code will develop; it should be merged back together soon
- Tracking, scintillator and Cherenkov readout
  - Comes for free with combined system
  - Stand-alone at FNAL would need new solutions/code for all these if they are needed

One final item...



- ECAL slow controls data have never been interfaced to DAQ readout
- Should this be done before next run?