PROCEEDINGS OF SCIENCE

Experience with DAQ for ATLAS Semi-Conductor Tracker

Bruce Gallop[1]
Rutherford Appleton Laboratory
Chilton, Oxon, United Kingdom
E-mail: bruce.gallop@cern.ch

The Semi-Conductor Tracker is a silicon strip tracker that forms part of the ATLAS detector at the LHC. The DAQ system is responsible for reading data from the detector for analysis. Recent experiences of commissioning and integration of the DAQ are described including problems and solutions.

17th International Workshop on Vertex detectors
Utö Island, Sweden
July 28th - 1st August, 2008

[1] Speaker

PoS(VERTEX 2008)042

1. Introduction

ATLAS [1] is a general purpose detector at the LHC collider at CERN. It is formed from an inner tracker, calorimeters and a muon detector, for the characterisation of the products from LHC collisions. In order to reduce the data volume from the detector, an online hardware trigger, known as Level 1, uses signals from the calorimeters and muon detectors to read out data corresponding to particular bunch crossings. After a collision, data is stored within the detector until the Level 1 Accept (L1A) signals this readout.

The following sections describe the Semi-Conductor Tracker sub-detector and the DAQ software used for control and readout of the detector. Section 4 describes some recent issues arising from recent experience of the DAQ. Finally section 5 provides a conclusion.

2. Detector description

The SCT (Semi-Conductor Tracker) [2] is part of the inner tracking detector in ATLAS. It is formed of a barrel and two endcaps. Detector modules are arranged on four cylinders in the barrel and nine disks on each of the endcaps. The support structure is arranged so that a track from the interaction region passes through at least four modules.

The barrel portion is made up of 2112 detector modules. Each endcap contains 988 modules. A module comprises 1536 strips giving a total of 6.3 million channels distributed over 4088 modules.

2.1 The SCT module

There are two styles of module, one for the barrel [3] and one for the endcap [4]. Each barrel module is formed from four silicon strip sensors, connected together in pairs to form strips 12 cm long. Each pair of sensors is placed back to back and one side rotated by 40mrad in order to provide a stereo angle to increase the z resolution. The strips have a pitch of 80μm. A hybrid circuit wraps around the sensors and provides the electrical connection between the sensors and the read-out chips, and between the chips and the off-detector readout.

The sensors for the endcap modules are tapered to fit on a disk and come in different varieties depending on their radial position. The endcap hybrid is folded over to enable connection of the chips to sensors on both sides, and connected to the sensors at one end.

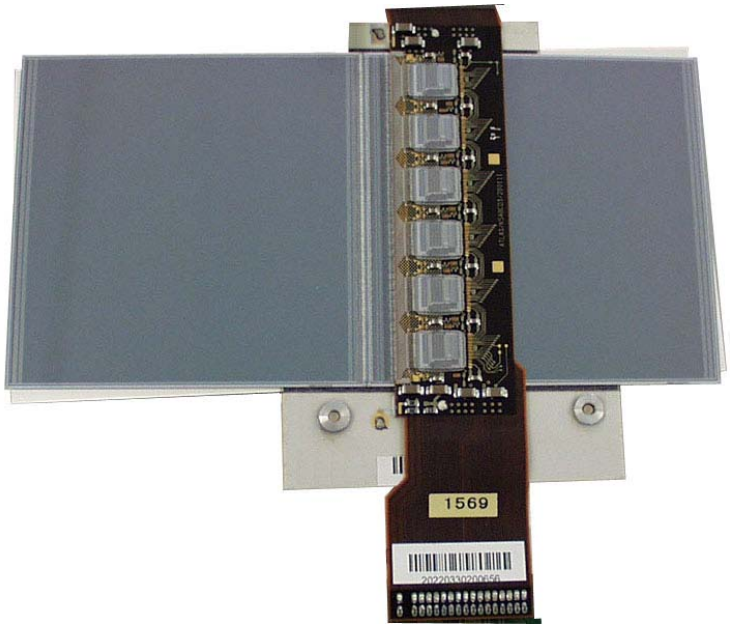

Figure 1: The SCT barrel module showing the top side of the hybrid with six ABCD chips above the sensor. Around the edge of the top sensor can be seen the angled bottom sensor.

The strips are read-out by 12 custom ASICs (ABCD) [5], each servicing 128 channels. The chip is clocked at the same rate as the LHC bunch crossings (40MHz). Charge collected from the detector is amplified and shaped before being fed to a discriminator which produces a binary hit signal for every bunch crossing. The binary hits are then stored in a pipeline for 132 clock cycles. When an L1A trigger arrives, the data for the three contiguous bunch crossings at the end of the pipeline are stored in an eight cell deep buffer for read out from the chip. Before being read out, the data is compressed by suppressing non-hit channels based on the pattern of the data in the three bunch crossings. This has the effect of making the data stream length a function of the hit occupancy of the detector.

A column of six chips forms a serial readout chain on each side of a module, the first of which sends data from all the chips to one of a pair of VCSELs (Vertical Cavity Surface Emitting Laser) which transmits the data from the module over an optical link to the off-detector electronics [6]. Similarly a single optical link transmits clock and command from the off-detector to the module. The command signal encodes the chip configuration and resets as well as the L1A signal.

2.2 DAQ System

The off-detector DAQ system resides in the large service cavern, outside the experimental hall and in a low radiation environment. It consists of modules in nine VME crates and nine rack mounted PCs. One VME crate contains module common across ATLAS that communicate with the TTC (Trigger, Timing and Clock) system. The other crates contain modules implemented by the SCT and Pixel sub-detectors. One VME module is the ROD (Read-Out Driver), which generates and parses signals to and from the modules. Connected to each ROD via a custom backplane is the BOC (Back-of-Crate), which converts the 40MHz optical signals between the detector and counting room into electrical signals for the ROD, and also sends

events to the rest of ATLAS via the S-link. The eight crates contain a total of 90 ROD/BOC pairs and a TIM (Timing Interface Module) VME module which receives timing and control signals from ATLAS. Finally each crate is connected to the network via a single-board-computer (SBC) to provide control of these modules.

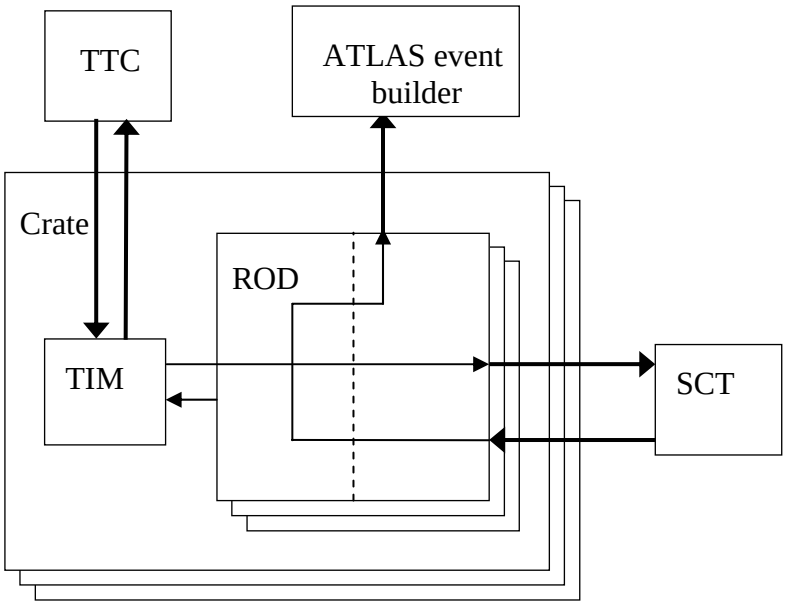

Figure 2: Interaction within a ROD crate. The ATLAS TTC system distributes trigger and clock to the TIM and receives busy from the ROD via the TIM. The ATLAS event builder receives event data from the ROD via the BOC. The BOC also sends data to and from the detector.

Connections from detector to counting room are via bundles of optical fibres. A single fibre carries command and trigger to a module, a pair of fibres carries event data back from the module. Fibres are connected in ribbons of 12 at the BOC and to sets of six modules at the detector end. Each pair of BOC and ROD services 48 modules.

The TIM distributes clock, trigger and reset signals from ATLAS trigger system (TTC) to the RODs using the backplane. For every trigger it also sends counters for event synchronisation. In the other direction it transmits a BUSY signal from RODs and generates a BUSY signal internally to veto fixed frequency triggers in order to avoid bond-wire resonances [7].

The BOC provides the optical interface to front-end modules and data flow. It is also responsible for distribution of the clock from the TIM to the ROD and the modules. The BOC itself is not accessible via the VME bus, but is configured via the ROD. Transmit (tx) fibres encode clock and command to the detector modules using a BPM (bi-phase mark) encoding (see Figure 3), the timing can be adjusted for individual channels, as can the parameters of the BPM encoding chip. The receive (rx) fibres deliver data from each side of a module. Binary data is extracted by a programmable discriminator and strobe, with reference to the same clock signal sent to the modules. The S-link connection to the ATLAS event builder is also on board.

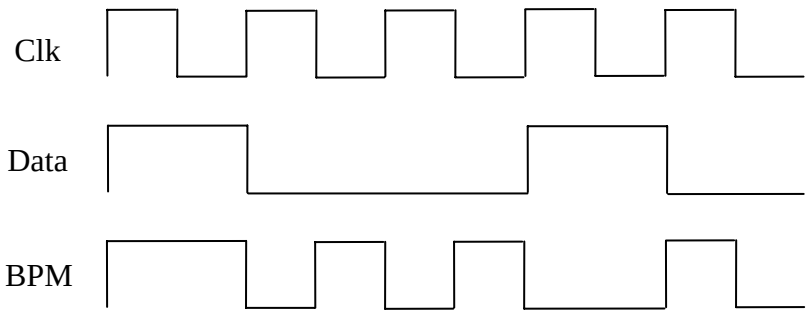

Figure 3:BPM encoding

The ROD is responsible for parsing the event data from the detector modules into a form understandable by the higher level triggers in ATLAS. It is also responsible for generating the control signals to the detector modules; sending the L1A trigger command, reset signals and DAC configuration. A BUSY signal is generated from the ROD when it is not ready for triggers.

The ROD is made up of five DSPs and five varieties of FPGAs. The design is shared with the Pixel sub-detector and the major differences are an increased RAM size in the Pixel system and the parsing of the bit stream. The FPGAs implement the hard real-time data path, sending event data to the ROS via the S-link. DSPs are used only for setup and monitoring. The FPGAs are programmed as Formatter, Event Fragment Builder (EFB), Router, Controller and Program Controller. The Formatters derandomise and format the bit-stream from each module link into blocks of 16-bit words. The EFB gathers the blocks of data for each link and marks formatting inconsistencies and counter errors. The Router sends events from the EFB to the S-link and to observation channels on the slave DSPs. The master DSP is directly accessible from the SBC and is used to set up the FPGAs and also controls configuration and calibration of the modules. A set of four slave DSPs are used to process event data in calibration mode and can be made to spy on data in physics mode.

3. DAQ Software

The rack-mounted PCs are used in combination to run the distributed calibration and analysis system, written in a combination of C++ and Java. These hosts are connected to each other, to the single board computers and to the ATLAS control room via ethernet.

The system was optimised for data-taking in physics mode. In this mode, the SBC is used to configure modules. Once this has been done and the run starts, the FPGAs do all the work of building events in response to triggers and sending to the S-Link. The job of the SBC then becomes purely one of monitoring.

3.1 Calibration software

The DAQ calibration software [8] is made up of software on the ROD crates and that on the monitoring hosts. The on-crate software provides the interface to the ROD, BOC and TIM hardware. It configures the VME cards and SCT modules and generates histograms (by controlling the ROD) for calibration and monitoring. The remainder of the calibration

infrastructure controls the generation of histograms on the crates, and the analysis of the histograms in order to generate results which suggest changes to the configuration.

As the front-end ABCD produces only binary hit data, choosing the correct threshold for the discriminator is very important. In theory, the threshold should be changed until no hits are generated for a particular input charge. In practice, data from a series of triggers can be collected to produce a smoothly varying occupancy value due to the noise from the sensor.

Calibration scans are produced using the DSPs on the ROD by varying module parameters (for instance, the discriminator threshold) and collecting hit data into occupancy histograms. For each scan, parameters are setup and one is scanned over a particular range. For each setting of this parameter, a sequence of triggers, typically 500, is sent to read out hit data from the detector module. The hits in the resulting events are histogrammed.

A calibration test is formed from a series of scans, each with different parameter settings. This series of scans are analysed together to produce a calibration result. For instance a pipeline test recognises defects in the chip pipeline; gain and noise tests produce bad channel maps, response curve (mV response for input charge) and noise estimation.

During a scan, if there are errors from one module, it is suppressed from the scan so that others can continue undisturbed and it can be examined in more detail later.

The optical links are also highly configurable. Finding the correct settings is complicated by the fact that a signal needs to go both to and from the module before it can be observed by the BOC. This is helped by examining the current through the on-detector PiN receiver diode, which is read out by the detector slow control system. This indicates reception of the clock signal, and the currents drawn by the chips are used to indicate reception of the command signal when this is used to send a configuration that draws more current.

To start with, the rx (from the modules) needs a correct threshold. This assumes valid, though not necessarily optimised, tx (command) settings. Most scans are carried out using the ABCD clock pass-through feature, which results in a half frequency clock at its data output (clock/2), as this provides a very regular signal which can be automatically monitored.

The rx delay and threshold are set using the 2D Rx Scan. This is run in clock/2 mode. For each setting of the threshold and delay, we count the number of ones within a time window using counters added to the ROD firmware for this purpose. This is one of the shortest tests. As no trigger is sent (or read out), we don’t worry about fixed frequency, and the counting is done in hardware so the majority of the time spent is due to switching the parameters. Due to using the balanced clock/2, it fails to find correct values for lasers that turn on slowly, as the data signal uses a return to zero protocol.

A problem optimising the tx is that reducing the laser amplitude too far can remove the clock from chips, which causes a hysteresis effect when it doesn't recover immediately. So far, the best procedure is to start with a high value, to lower it until doesn’t respond correctly and then choose a value sufficiently higher than this.

4. DAQ Experiences

The following indicate recent experiences as a result of integrating the DAQ with large numbers of detector modules.

4.1.1 Mark-Space Ratio Test

One parameter that needs setting for the tx fibre connection in the BOC is the mark-space ratio of the BPM output. It compensates for the difference in speed between the 0->1 and 1->0 transitions. This affects the jitter on the clock signal reconstructed on the module, which should be minimised for correct timing of the clock (which is used as the data collection strobe).

The initial procedure was to scan across all mark-space ratio (MSR) values and for each carry out a scan of receiver delay. This was used to find the width of the mark and space in the returned clock/2 for each value. Next a single bit is sent to flip the BPM decoder so that it uses the rising edge for the clock instead of the falling edge for the clock (or vice-versa). The ratio for each setting is then measured again and the MSR setting at which the difference is minimised is calculated.

This has produced inconsistent results due to a variety of reasons. As with the tx laser current, a bad MSR setting can result in an invalid clock. This causes problems when the MSR is restored and the module continues to return bad data. Glitches from changing the MSR may also cause the BPM decoder to flip before the intended flip.

These problems have led to the development of a new procedure. For each setting of the MSR we first examine the ratio of the returned clock/2 by scanning the receiver delay as before. Next, a command with an odd number of bits is sent to flip the BPM decoder and the changed ratio is measured. This allows the two measurements to be linked by only one polarity flip. The MSR setting is then changed and a configuration stream is sent to the modules. This has the effect of recovering from problems due to loss of clock or glitches. The process is repeated for each MSR value.

This procedure has produced more consistent results. The major remaining issue is that it takes a long time, mainly because of the amount of data that is read out over the relatively slow VME bus. A future implementation might implement the same procedure using the master DSP on the ROD.

4.1.2 ROD Busy

The ROD has several buffers in the dataflow path, which are optimised for the expected rate and distribution of data. If one of these buffers fills up unexpectedly a ROD BUSY is asserted which inhibits triggers sent by the central trigger processor. This can happen if the settings of the optical link are not properly optimised and data is thus corrupted. The loss of bias across a module also produces a situation where more data are generated, due to an increased number of hits (due to increased noise).

It is possible for the ROD to fill the buffer storing trigger information if data takes too long to read out. In this case a reset is needed and data stored in the FIFOs is lost. If data to a single link is corrupted then it is not helpful to lose data from all the other modules on the ROD. It is therefore important to trap this situation and turn off or recover the link before this happens.

Code on the DSP has been written to monitor flags on the ROD and mask off links that are producing too much data i.e. the front-end FIFO is almost full. Work is in progress to identify causes at the module link level to prevent a ROD level BUSY.

4.1.3 ABCD Simulation

The FPGAs in the RODs have been expanded in the past few years to support larger FIFOs needed by the Pixel sub-detector. This means there is some space for additional features, beyond those originally planned. One feature added recently allows the simulation of a module in the front-end (formatter) of the ROD. When a trigger arrives at the Formatter, a bit-stream is generated which corresponds to a data packet from a module with some number of hits in it. This allows the whole data chain to be exercised without the modules working (which requires cooling and power). This has been useful during detector down time. The main caveat is that reset signals are not received by the formatter so that the front-end synchronisation counters are not reset.

4.1.4 BCID Counters

The BCID (Bunch Crossing ID) is used to identify an event based on the bunch crossing counter. This allows synchronisation of events within ATLAS, between sub-detectors, and within the SCT between the front-end and the DAQ. For instance, if a trigger for some reason doesn't arrive at a detector module, then that module will not send back data. Data that is sent when the next trigger arrives might become associated with the previous event. This will be flagged by checking for a BCID mismatch.

The BCIDs in ATLAS are synchronised by sending a reset (BCR) to all sub-detectors at the same time. This is sent at 11kHz, once per orbit of the LHC, and also helps recover from problems caused by single event upsets during this period. At an ATLAS level, the counter is reset so that the BCID corresponding to the first crossing in a bunch train is one. In the SCT (and in some other sub-detectors), the BCR is sent to the detector modules over the same command link as the trigger. This takes several cycles to transmit and means that a trigger can't be sent at the same time which is required if a trigger is to read out the bunch at BCID=1. In order for the BCR not to collide with the trigger, triggers are vetoed in the CTP during this period. In order that a trigger can be sent to read out the first bunch, the BCR must be moved to a point in the orbit cycle during which there will be no collisions. As there are differing constraints between sub-detectors, this must be done at the SCT level.

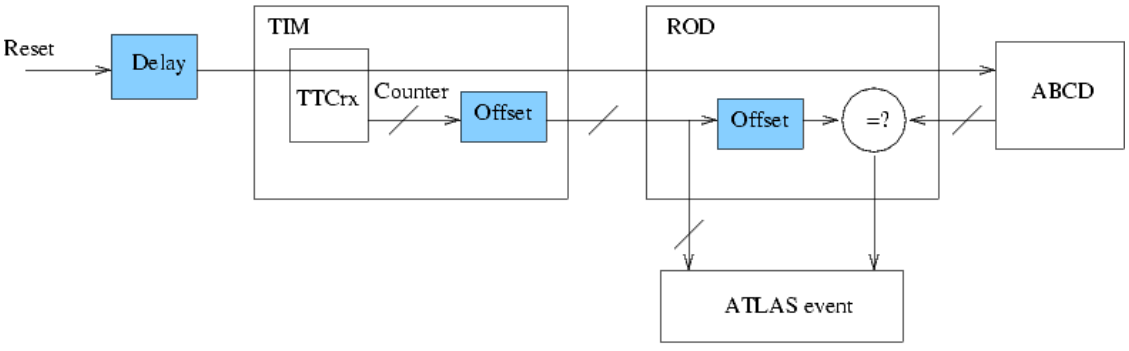

Figure 4: BCID handling, the shaded blocks are additions to allow the BCR to be moved

In order for the BCIDs to line up several delays and offsets are added. First, the BCR is delayed before it reaches the TIM so that it will arrive at the modules in the gap. The counter

which is put in the event for ATLAS is generated in the TIM, so in order for this to match the ATLAS counter it is offset to compensate for the delay. This BCID is sent to the ROD to compare against the front-end chip counters, which have also been reset by the delayed BCR. Finally, the ROD must also apply an offset to the BCID from the TIM before comparing it with the BCID from the front-end.

4.1.5 Diagnostics

Recent developments in the DAQ have been aimed at increasing the ability of the user to know about the system. Fundamentally, a shifter watching the system wants to know that things are proceeding as expected, and that good data is being accumulated. When the system is working, it receives triggers and sends out events without errors. This can be observed by watching the event counters increasing and also online monitoring histograms.

The online monitoring system monitors decoded events. So far this is done at the crate level, within the sub-detector, where some proportion of the events read out by Level 1 can be histogrammed; also in the ATLAS monitoring farm where events are processed at approximately 1Hz. It is also possible to carry out monitoring at the ROD level, where a higher event rate would be available, though so far this has not been a priority.

One important recent addition is the monitoring of the BUSY signal at the TIM and at the ROD level, which has greatly improved understanding of the failure modes. There are many other things that could be monitored, and a balance needs to be struck between the amount of data that is read out and how useful it is to diagnosing fault conditions.

4.1.6 Milestone Runs

During the past year, as ATLAS detector integration has proceeded, there have been a series of week long milestone runs. These are periods during which portions of the detector have been read out together with the aim of taking cosmics data. During the latter of these, the SCT and the DAQ have been exercised with the rest of ATLAS. In M5 only a small number of modules in a test box were exercised, this exercised the trigger system and synchronisation issues. In M6 almost the whole of the SCT barrel took cosmics data with ATLAS. Timing in with the cosmic trigger was done mainly by dead reckoning. For M7 the SCT took part using the simulator to produce data for a portion of the detector for one day. This enabled a validation of the software changes since the previous run. In M8 a high trigger rate was tested, with successful running up to 70kHz using simulated data. The new BC reset scheme is now used, instead of a manual trigger veto.

4.1.7 Current status

Since early September, the whole SCT has been turned on and running continuously. The majority of the time has been spent collecting data from cosmics. Increased scrutiny of the data revealed a timeout problem which has since been fixed.

5.Conclusion

The SCT DAQ has been exercised with almost the full complement of SCT modules[1]. There have been problems discovered as an increasing number of modules and the full ATLAS trigger system have been integrated with the SCT DAQ. This has flagged issues in different areas, including with BC counters and monitoring of the system status. One key area is ROD-level monitoring and feedback so we don't flag BUSY unnecessarily.

The simulation of ABCD chips in modules within the ROD has been a useful feature to test the dataflow in the absence of a powered detector. This would have been a very useful feature in earlier stages of development as we can't assume the support infrastructure is going to work all the time.

For the future, it seems that it would be useful to have a non-return to zero encoding on the receive links as well as the transmit links. Having a "dead-time" for triggers during resets has also caused problems.

The system has been shown to be flexible enough to allow solutions to problems as they arise. It will be useful to gain more experience running with the full quotient of modules. Other problems and areas for improvement are likely to emerge as we gain more experience, but are expected to be soluble.

References

[1] G. Aad et al.*The ATLAS Experiment at the CERN Large Hadron Collider,* JINST **3**:S08003, 2008.

[2] *ATLAS Inner Detector Technical Design Report.* CERN/LHCC/97-16 and 17, 1997.

[3] *The barrel modules of the ATLAS semiconductor tracker,* Nucl.Instrum.Meth.A568:642-671,2006.

[4] *The ATLAS semiconductor tracker end-cap module,* Nucl.Instrum.Meth.A575:353-389,2007.

[5] *Design and performance of the ABCD3TA ASIC for readout of silicon strip detectors in the ATLAS semiconductor tracker,* Nucl.Instrum.Meth.A552:292-328,2005.

[6] *The optical links of the ATLAS SemiConductor Tracker,* JINST **2:** P09003, 2007.

[7] *Resonant bond wire vibrations in the ATLAS semiconductor tracker,* Nucl.Instrum.Meth.A538:442-457,2005.

[8] *The data acquisition and calibration system for the ATLAS Semiconductor Tracker,* JINST **3**:P01003, 2008.

[1] Two cooling loops (33 + 13 modules) are turned off for this run