Testing and Development of the CMS Silicon Tracker Front End Readout Electronics

James David George Leaver

Blackett Laboratory
Imperial College London

A thesis submitted to the University of London for the degree of Doctor of Philosophy and the Diploma of Imperial College London

November 2005

Abstract

The Compact Muon Solenoid (CMS) is a general purpose detector that will operate at the CERN Large Hadron Collider (LHC), a particle accelerator designed for the study of new physics at the TeV energy scale. A key component of CMS is the Silicon Tracker, which has ~9.3 million detector channels and is expected to generate over 70% of the total CMS data volume. The Tracker readout system must process data at a rate that is orders of magnitude higher than in any previous particle physics experiment.

On-detector readout is performed by the APV25 front end chip. To ensure a Tracker of the highest quality and efficiency, each APV25 must be rigorously verified; a wafer probing test station has been developed for post production quality assurance. The APV25 contains internal pipelines which buffer event data pending readout. An APV Emulator has been designed to prevent APV25 buffer overflow due to random fluctuations in the Level 1 trigger rate.

The first stage of the off-detector readout is performed by the Front End Driver (FED). This interfaces the Tracker to the CMS Data Acquisition system and reduces the enormous volume of data output from the detector to a manageable level through the process of Zero Suppression, which involves identifying strips that contain hit information and discarding the remainder. The FED is an important, highly complex device and it is vital for every board to function correctly. An Acceptance Test has been developed to check the functionality of each FED before it leaves the assembly plant. A FED Tester card has also been produced to enable testing of the FED under realistic CMS conditions. The data integrity of the S-Link connection used to read out the FED has been assessed and an extensive suite of software has been written to verify the performance of the FED hardware and firmware.

Finally, an alternative algorithm for Common Mode (CM) subtraction has been implemented in the FED firmware, with support for high occupancy events and non-uniform CM. In addition, a novel algorithm for baseline correction in APV frames has been developed.

Acknowledgements

It is difficult in so small a space to adequately express my gratitude towards all of the people who have contributed to my PhD, and have helped and supported me throughout my studies. I would firstly like to thank my supervisor, Prof. Geoff Hall, for all of his guidance, advice and encouragement over the past three years, and for proof reading this thesis. Greg Iles also deserves special mention; he has been a constant source of wisdom and inspiration, and my work has greatly benefited from his sage counsel.

I would like to thank everyone in the High Energy Physics department at Imperial College for creating a marvellous working environment. I wish to thank Costas for his guidance with the S-Link and FED firmware studies, Osman for his invaluable help in implementing the new CM subtraction algorithm within the FED, Jonathan for his support with the Fed9USoftware and Rob for his assistance with the FED Tester frame generating software (and, of course, for all of his work in developing the APV25 wafer probe station so that I could analyse the initial results). I would also like to express my gratitude towards the ladies of the Electronics Workshop for all their PCB work, namely Maria, Vera and especially Sarah for laying out the S-Link Transition card. In addition to the above, thanks in general go to Barry, Emlyn, Etam, Mark, Matt, and John; all have contributed towards my studies in some way and have made my time at Imperial both memorable and enjoyable. I would also like to thank Peter Dornan for letting me work with the group.

Outside of Imperial, a number of people from the Rutherford Appleton Laboratory deserve special mention for all of their work on the FED and for their feedback on the FED Acceptance Test and performance test software. I wish to thank Matthew Pearson, Ivan Church, Ian Tomalin (for his comments on FED data rates and CM subtraction) and particularly John Coughlan and Saeed Taghavi, for all of their help with the FED and for their hospitality on the many occasions that I have visited RAL.

Above all, I would like to thank my parents, Valerie and George, for their unwavering and selfless support throughout my PhD.

Contents

~ Chapter 1 ~

Introduction

1.1 The Large Hadron Collider

The Large Hadron Collider (LHC) [1] is a particle accelerator that is scheduled to commence operation at the European Laboratory for Particle Physics Research (CERN) in 2007. As CERN adopts a cost-effective strategy of building on previous investments, it will make use of the 27 km circular ring tunnel previously occupied by the Large Electron-Positron Collider (LEP). During operation, the LHC will collide bunches of protons at 40 MHz with a centre-of-mass energy of 14 TeV and a design luminosity of 10^{34} $cm^{-2}s^{-1}$; it is also sufficiently versatile to perform heavy-ion collisions with total energies of up to 1250 TeV. Providing access to previously unattainable energy regimes in particle physics, at interaction rates that are orders of magnitude higher than in any previous experiment, the LHC will enable the observation and measurement of a wide range of new and rare physics processes.

1.1.1 Motivation for the Large Hadron Collider

The LHC has been designed to address the most fundamental questions of modern particle physics. One of the principal aims of the accelerator is to investigate the origin of mass, related to the spontaneous electroweak symmetry breaking sector of the Standard Model (SM). It is postulated that the four forces of nature (gravitational, electromagnetic, weak and strong) are simply manifestations of one underlying force. Electroweak theory has already unified the electromagnetic (EM) and weak forces for equivalent kinetic temperatures in excess of 100 GeV. However, at the core of the SM, the mediators of the weak force (the W and Z bosons) have mass whereas the

mediator of the EM force (the photon) does not. This leads to a breaking of the symmetry between the EM and weak forces at energies below ~80 GeV. A mathematical explanation is incorporated into the SM via the Higgs mechanism, which predicts the existence of a new scalar particle: the Higgs boson. The mass of the SM Higgs is not accurately predicted, but a theoretical upper limit of ~1 TeV can be deduced while direct searches at LEP exclude masses less than 114.1 GeV (95% confidence level) [2]. With a proton-proton collision energy of 14 TeV, the LHC will be able to generate particles with masses over this entire range and should therefore be able to provide definitive evidence for the existence (or non-existence) of the Higgs boson [3, 4, 5, 6, 7, 8].

Another goal of the LHC is to examine Supersymmetry (SUSY). The mathematical description of the Higgs boson in the SM has an undesired feature; when calculated using perturbation theory, the mass of the Higgs is found to diverge quadratically. To counter this problem a new theory (SUSY) was proposed, in which every SM particle has a supersymmetric partner (a sparticle). A sparticle has the same quantum numbers as its partner, but a spin that differs by one half. Including SUSY particles in the SM has the effect of cancelling out the quadratic divergence of the Higgs, provided that they have masses less than a few TeV. At this energy scale, sparticles should be fully accessible to the LHC [9].

Other fundamental issues in particle physics include the origin of matter-antimatter asymmetry in the universe. It is believed that equal amounts of matter and antimatter were generated at the creation of the universe, but thus far no significant quantities of antimatter have been observed in nature. Theories describing the cooling of the early universe must introduce a form of symmetry breaking to explain the preference for matter production. One candidate is the violation of symmetry in

interactions related by the charge-parity (CP) transformation, first observed as a small difference between the decay rates of the neutral K-meson and its antiparticle. This indicates either the existence of an additional unknown matter-antimatter asymmetric force or a distinction by the weak force (which mediates meson decay) between matter and antimatter. Mesons with higher mass are expected to exhibit greater asymmetry and currently the most promising mechanism for investigating CP-violation is B-meson decays. A very large number of B-mesons will be generated by collisions at the LHC, providing an excellent opportunity for B-physics research [10].

The LHC should also enable studies of a new state of matter, the quark-gluon plasma (QGP) [11], which is believed to have existed before the early universe cooled. A QGP is predicted to form at very high temperatures or in heavy ion collisions, when nuclei interact with such energy that their constituent quarks and gluons become deconfined and mingle freely, analogous to electrons and protons in conventional plasma. If a QGP is produced following the collision of heavy ions at the LHC then it will rapidly hadronise, creating a decay signature that should be possible to detect.

1.2 The Compact Muon Solenoid

The Compact Muon Solenoid (CMS) [12] is one of two general purpose detectors that will be used at the LHC. It is designed to identify cleanly the diverse signatures of the new physics made accessible by the increased energy levels of the LHC, but is optimised for the discovery of the Higgs boson. As protons are composite entities, consisting of quarks and gluons, proton-proton collisions are complex interactions that generate a large number of particles. In order to isolate individual processes, the CMS detector must be able to locate and measure the momenta of muons, electrons and photons with high precision, over a wide energy range and at high luminosity.

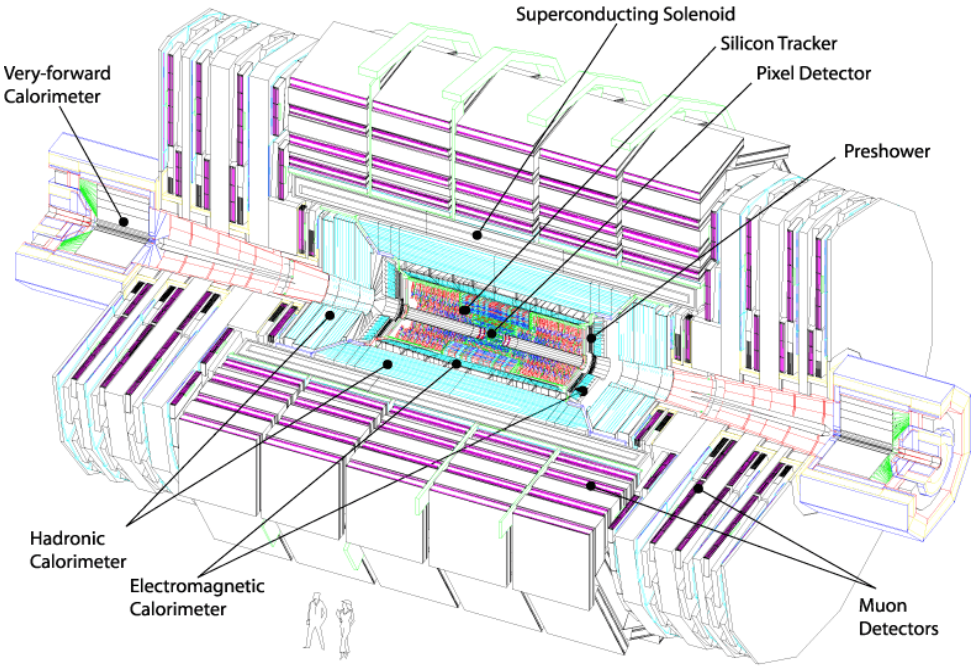


Figure 1.1: A cut-away schematic of the Compact Muon Solenoid.

A schematic of the CMS is shown in Figure 1.1. It is composed of a series of sub-detectors, arranged in concentric cylindrical layers around an axial beam pipe; each sub-detector also has 'endcaps', to ensure good coverage in the direction of the beams. Particles enter the CMS through both ends of the beam pipe, timed such that collisions occur at its centre. The resultant interaction products then pass radially through the concentric layers, as shown in Figure 1.2.

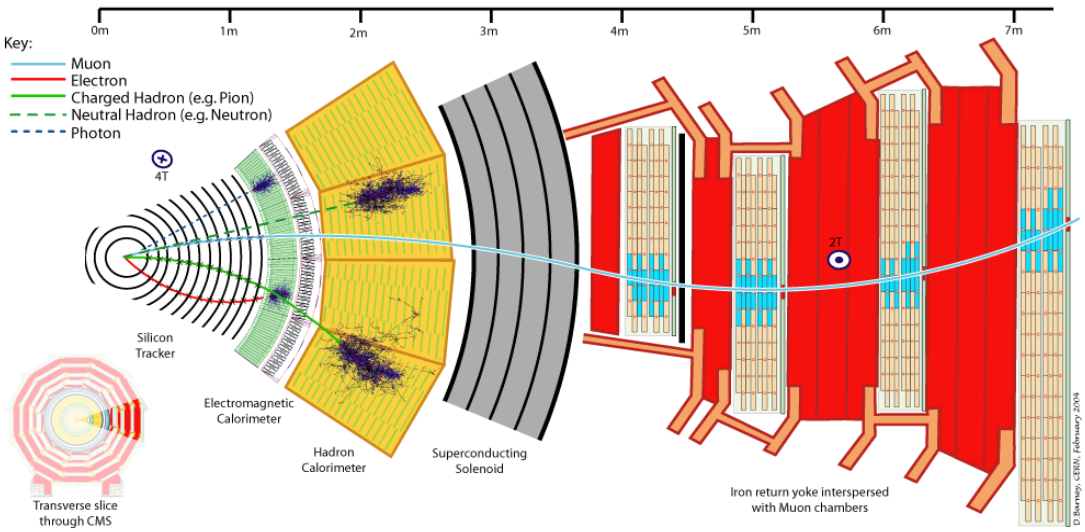


Figure 1.2: A schematic showing a transverse slice through the sub-detectors of the Compact Muon Solenoid, highlighting the paths taken by a number of particle types.

Nearest the interaction point are the Pixel Detector and Silicon Tracker [13, 14], which trace the paths taken by all ionising (charged) particles. The next layer is a lead tungstate crystal Electromagnetic Calorimeter (ECAL) [15], designed to absorb and measure the energy of electrons and photons. Other types of particle pass through the ECAL into a brass-scintillator sandwich Hadronic Calorimeter (HCAL) [16], where hadrons are absorbed and their energies are measured. Only muons and neutrinos can escape the HCAL, continuing to pass through a series of four Muon Chambers [17]. These detect the charged Muons (enabling them to be distinguished from other charged particles registered by the Pixel Detector and Silicon Tracker), but neutrinos do not interact with any part of the CMS and must be inferred from missing energy calculations.

A prominent feature of the CMS detector is a large superconducting solenoid [18], positioned between the HCAL and the Muon Chambers. This generates a 4 T magnetic field parallel to the beam pipe, extending to a radius of 3 m over a length of 6.25 m either side of the particle interaction point; iron return yokes interspersed with the Muon Chambers allow a weaker field to permeate the region beyond the solenoid. Charged particles moving through a magnetic field experience a force perpendicular to both the direction of motion and the field, causing their trajectories to bend. Particles with opposite charges bend in opposite directions and the radius of curvature increases with particle velocity and decreases with field strength. The Pixel Detector and Silicon Tracker record the curved trajectories, enabling the polarity and momentum of each charged particle to be determined. As the tracking detectors have a finite granularity, paths with smaller radii of curvature can usually be measured with greater accuracy. Consequently, the strong 4 T field is essential for good momentum resolution. The magnet is designed with a diameter that is large enough to

accommodate the calorimeters in addition to the tracking hardware, ensuring that the material of the solenoid has no impact on the ECAL and HCAL energy measurements.

1.3 The Silicon Tracker

All of the physics searches that will be conducted with the CMS rely heavily on the Silicon Tracker. In addition to providing the information required for measurements of particle momentum and charge, it enables the reconstruction of the physical layout of collision events and is thus essential for the identification of individual particles and the characterisation of interaction processes. The unprecedented energy levels and interaction rate at the LHC produce an extremely challenging experimental environment, requiring tracking detectors and readout hardware with exceptional performance.

1.3.1 Experimental Challenges of the Silicon Tracker

When operated at the design luminosity of 10^{34} $cm^{-2}s^{-1}$, each collision of proton bunches will cause an average of ~20 minimum bias events (a minimum bias event is defined as a non-diffractive inelastic interaction). This will lead to the generation of ~1000 charged particle tracks in the volume of the Tracker every 25 ns; a detector with very fine granularity is required for the precise identification and separation of each individual track. Particle fluxes of between 10^{13} and 10^{14} equivalent 1 MeV neutrons/cm^2/year are expected at the inner regions of the CMS, creating a harsh radiation environment in which the Tracker must survive, without significant performance degradation, for the 10-year lifetime of the experiment. The amount of material used in the Tracker should also be minimised in order to reduce the impact of particle absorption, energy loss and scattering on the accuracy of energy measurements made with the calorimeters, and on the reliability of the tracking itself.

Reading out the enormous volume of data generated by the Tracker is a highly challenging task. The readout electronics must be able to cope with a large number of detector channels and, with a bunch crossing rate of 40 MHz, must have a fast response time. Although inelastic proton-proton collisions will occur at a rate of 10^9 Hz, physical limitations on mass storage media mean that data from only 100 events each second can be recorded for future analysis; this should not reduce the physics performance of the detector, as 'interesting' events are very rare (it is expected that Higgs production will occur once every ~10^{13} collisions).

Data reduction is carried out by a multi-level trigger system [19, 20, 21], which discards events that do not fulfil a series of increasingly stringent selection criteria. A Level 1 (L1) trigger is designed to cut rapidly the acceptance of interaction data from the input rate of 10^9 Hz to ~100 kHz. It examines coarsely segmented data from a number of CMS sub-detectors, looking for trigger objects such as the presence of muons, photons, electrons and jets, and total energy/momentum levels above pre-defined limits. An L1 trigger for an event is generated and distributed to all of the sub-detector readout systems within 3.2 μs of the corresponding proton bunch crossing; this latency has important implications for the readout electronics, as all event data must be buffered whilst the L1 trigger decision is made. High Level Triggers (HLTs), operating on commercial processor farms, run more sophisticated event reconstruction and filtering algorithms on data read out from the CMS at 100 kHz, selecting ~100 'interesting' events for storage each second.

However, even with a data-reducing 100 kHz L1 trigger, the Tracker readout system must still be able to cope with output data rates of up to ~1.4 TB/s. This is orders of magnitude higher than in any previous particle physics experiment and

demands state of the art electronics; the development and testing of this hardware forms the subject matter for the following chapters of this work.

1.3.2 Implementation of the Silicon Tracker

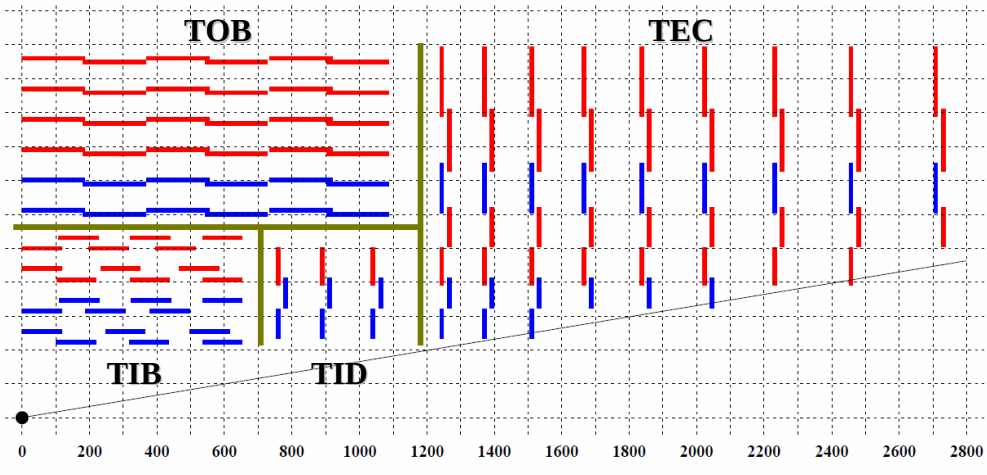


Figure 1.3: A diagram showing the layout of sensor modules in the CMS Silicon Tracker. The view is of one quarter of a transverse cross-section through the volume of the Tracker, with the particle interaction point marked with a black circle in the bottom left-hand corner. Single-sided modules are shown in red, whilst double-sided modules are highlighted in blue.

Figure 1.3 shows the layout of the Silicon Tracker. The barrel region consists of cylindrical layers of detectors that run parallel to the beam pipe. It is divided into the Tracker Inner Barrel (TIB), comprised of four layers, and the Tracker Outer Barrel (TOB), which contains six. Three inner disks (TID) enclose the TIB, while nine 'endcap' disks (TEC) are positioned at each end of the TOB. A relatively small number of detector layers are used, to minimise the Tracker material budget, but they provide very precise, high resolution measurements.

Each layer is made up of a number of sensor modules, each containing 512 or 768 silicon microstrips. The microstrips are simple p-n diode structures, readily produced at low cost and in large quantities using standard manufacturing techniques. A radiation-hard design has been chosen and they will be operated at a low temperature (-10 °C) to minimise the effects of radiation damage. In the TIB, TID and

the first four layers of the TEC, the silicon microstrips have a thickness of 320 μm, a length of ~12 cm and a pitch of ~100 μm. For the remaining layers, the lengths are extended to ~16 cm and the pitch is increased by a factor of two; this reduces the number of detector channels that must be read out but has little impact on physics performance, as particle track density decreases with distance from the interaction point. A larger surface area results in higher capacitance and noise levels, which are compensated for by increasing the thickness of these microstrips to 500 μm (which increases charge collection).

Most of the sensor modules are single-sided, but a number of double-sided modules, formed by placing two single-sided modules back-to-back at a relative angle of 100 mrad, provide high resolution two-dimensional hit detection in the inner regions of the TIB and TOB, and at various positions in the endcaps. The fine granularity of the detector results in a total of ~9.3 million channels, and enables measurements with a spatial resolution of 23-60 μm.

1.3.3 Silicon Tracker Readout System Design Choices

Analog readout has been chosen for the Silicon Strip Tracker. This can lead to greater complexity and power consumption than in systems that perform analog to digital conversion at the input of the on-detector readout stage and remain digital thereafter; for example, analog pipelines are required for the buffering of events instead of simple digital shift registers. It also places greater demands on the medium used to transmit the Tracker data off-detector, as analog signals are far more susceptible to the effects of attenuation and noise pickup than their digital counterparts. However, significant physics performance benefits are achieved.

For digital readout, an on-detector hit discriminator level is required (set above the noise but below the signal). If the signal to noise ratio is small then it may be

necessary to use a low discrimination threshold in order to maintain a high efficiency, which can generate a high proportion of ‘fake’ hits. With analog readout, hit identification can be performed more reliably through the use of sophisticated off-detector electronics, which are not constrained by the size, power and speed limitations of on-detector components. Similarly, off-detector Common Mode subtraction is possible; this would be difficult to achieve with any flexibility in a cost-effective on-detector integrated circuit. In the case of charge sharing between neighbouring strips, the availability of analog data also enables an estimate of the true hit location to be interpolated, increasing the positional accuracy of track reconstruction. Finally, any degradation in pulse height or signal to noise ratio can be directly observed off-detector, facilitating the diagnosis of radiation damage or any other problems.

It has also been decided that radiation-hard optical links will be used to transfer data from the Tracker to the off-detector readout systems. These were chosen for their considerable advantages over more traditional electrical connections, which compensate for the added technical difficulty of requiring electrical-optical signal conversions. Optical links have the highest transmission rates, low power consumption and excellent transmission quality. The latter is essential for preserving the integrity of the analog Tracker output. Physically, optical fibres have the benefits of low density, small size and high flexibility, and are thus handled more easily than large volumes of heavy, rigid electrical cabling. As the optical links are implemented using technology that is very similar to modern telecommunications standards, the Tracker has been able to take advantage of the drive by the communications industry to produce higher performance, lower cost components.

1.3.4 The Silicon Tracker Readout System

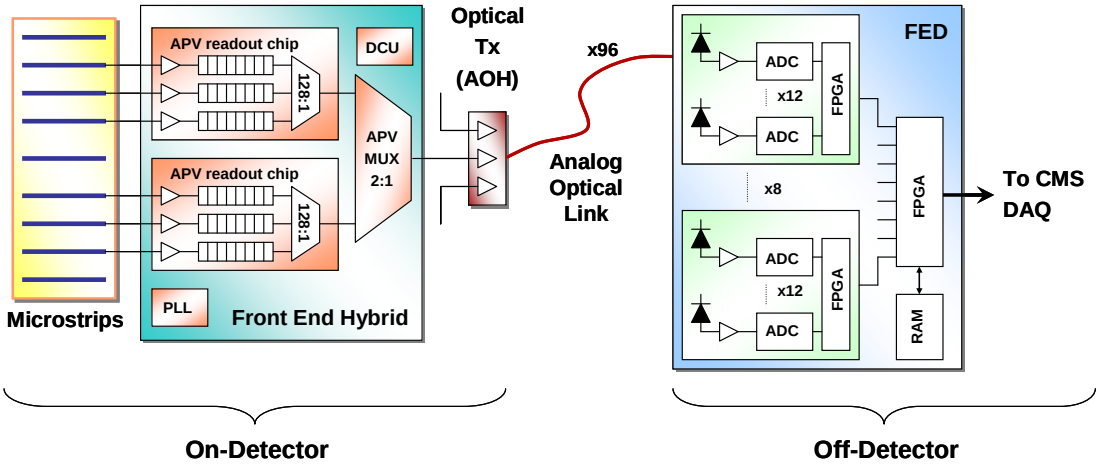


Figure 1.4: A schematic of the Silicon Tracker readout system

A schematic of the Silicon Tracker readout system is shown in Figure 1.4. The Tracker itself is divided into a large number of Front End Modules, each containing a sensor (512 or 768 microstrips) and a Front End Hybrid (FEH). Data are acquired from microstrips via the APV25 [22, 23, 24], a custom analog front end readout chip which samples, amplifies, shapes and buffers the signals from 128 detector channels (the APV25 is discussed in Chapter 2). A FEH houses four or six APV25s, grouped into pairs; data from each pair are interleaved by an APV multiplexer (APVMUX) [25], the electrical output of which is converted to optical by a semiconductor laser driver in an Analog Optohybrid (AOH) [26]. Signals from all APV25s are transmitted off-detector, along ~100 m optical fibre links [27], to Front End Driver (FED) [28, 29] modules in the CMS counting room (a description of the FED can be found in Chapter 4); each FED accepts the output from 96 multiplexed APV25 pairs.

The FED samples and digitises the optical signals, synchronising and reordering the result before applying algorithms to remove pedestals and Common Mode offsets. It then performs ‘hit finding’ or ‘Zero Suppression’ (see Chapter 4) in order to reduce the enormous volume of raw data output from the Tracker by a factor of ~60, to a level that is compatible with the CMS Data Acquisition System (DAQ).

Zero Suppressed collision event data are transmitted to the CMS DAQ via a fast S-Link connection [30] (described in Chapter 7), where more sophisticated filtering and track recognition algorithms are applied, culminating in the full reconstruction and archival storage of ‘interesting’ events.

As the interface between the Silicon Tracker and the CMS DAQ, the FED is an essential component of the Tracker readout system whose performance is fundamental to the success of the CMS experiment. The bulk of the work presented here (in Chapters 4 to 9) is concerned with the testing and development of this complex, high performance device.

~ Chapter 2 ~

The APV25 Readout Chip

The APV25 (Analog Pipeline [Voltage Mode] in 0.25 μm silicon CMOS technology) provides the initial on-detector readout of the CMS Tracker silicon microstrips. Raw data from each group of 128 detector channels are sampled in parallel by one APV25 chip, which amplifies and shapes the signal pulses before placing their (peak) values in an analog pipeline. Upon receipt of a Level 1 trigger from the CMS Trigger Control System (TCS), relevant hit information from the pipeline is further processed and multiplexed out in a standard frame format. Frames from pairs of APV25s are then externally multiplexed and transmitted optically to Front End Drivers (see Chapter 4) in the CMS counting room, which provide an interface to the Tracker Data Acquisition System.

2.1 Implementation of the APV25

The layout of the APV25 is shown in Figure 2.1. It was the first major integrated circuit for high energy physics to be implemented using a deep sub-micron (0.25 μm) Complementary Metal-Oxide-Semiconductor (CMOS) process, chosen for its inherent performance benefits. Although the APV25 contains analog circuitry in addition to digital logic, this implementation helps to reduce the overall power requirements of the chip. Each logic gate in a CMOS device consists of a pair of n-type and p-type transistors of which only one is turned on at any instant; power is only dissipated when the gate switches logic values.

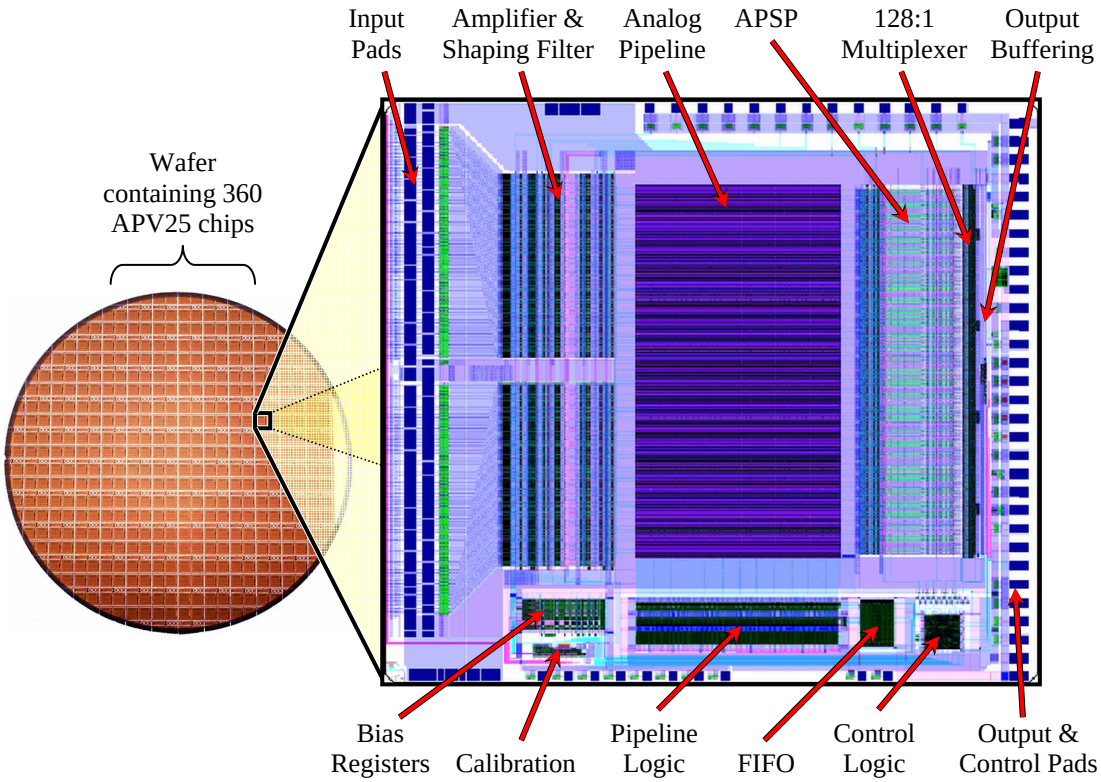


Figure 2.1: The layout of an APV25, with identification of the main functional blocks. The chips are fabricated on 8 inch (~200 mm) diameter wafers, an example of which is shown on the left side of the picture.

The small feature size enables the APV circuitry to fit within a die area of only 7.1 x 8.1 mm^2 and also leads to a high intrinsic radiation tolerance [31, 32]. Radiation damage in CMOS electronics is caused primarily through the generation of electron-hole pairs in the gate oxides of silicon transistors when they are crossed by ionising particles (a gate oxide is the layer which separates a gate electrode from the transistor substrate). While the electrons are rapidly swept out by internal electric fields, the holes have far lower mobility in the oxide and a proportion of them will become trapped at the oxide/silicon interface, modifying the performance of the transistor. 0.25 µm CMOS gate oxides are thin enough that the captured positive charge may be neutralized through electron tunneling. This helps to ensure that the APV25s will survive the exceptional levels of radiation that will be present in the Tracker, expected to reach a total dose of up to 10 Mrad during the lifetime of the experiment.

Finally, as 0.25 μm CMOS is a standard commercial technology, APV25 chips can be produced in industrial-scale quantities at relatively low cost using established production techniques; an important factor considering that over 100,000 APV25s (including spares) are required for the CMS Tracker.

2.2 APV25 Functionality

Signals entering the 128 APV channels initially pass through a low noise preamplifier followed by a CR-RC shaping amplifier, which convert strip hits into ~50 ns wide voltage pulses with magnitudes of ~100 mV per MIP of deposited charge (a MIP, or Minimum Ionising Particle, is equivalent to ~25,000 electrons for a 300 μm thick silicon detector). Sampled at the 40 MHz LHC bunch crossing frequency, the output from each shaping amplifier is read into a 192-column analog memory pipeline.

This stores the data from each particle interaction while a Level 1 trigger decision for the corresponding event is made. If the APV receives a Level 1 trigger, the relevant pipeline columns are marked and are prevented from being overwritten until they have been read out. 160 pipeline locations allow for a trigger latency of up to 4 μs (data can be stored for a maximum of 4 μs), whilst 32 locations are reserved to buffer events that occur while awaiting readout after a trigger; a FIFO maintains a list of the currently marked events, to ensure that they are output in the correct order.

Each of the 128 pipeline channels is read out through an Analogue Pulse Shape Processor (APSP), which has two principal modes of operation. In *peak* mode, one pipeline location is read per trigger, with the output of the APSP corresponding to the peak of the analog pulse generated by the shaping amplifier. This method has the best signal to noise ratio but is only appropriate for use when the Tracker occupancy is very low. If the hit rate is high enough then pile up can occur, as signals may be generated in the silicon strips at short enough time intervals that they overlap. In this

case, the APSP is run in *deconvolution* mode, where the weighted sum of three consecutive pipeline samples is calculated for every trigger (deconvolution is a technique widely used in signal processing, perhaps most commonly for the removal of blurring in optical images, and may be studied in References [33, 34, 35, 36]). This enables a good approximation of the 'real' signal to be extracted from the distorted input; the result is a pulse that peaks at 25 ns and falls rapidly to the baseline, narrow enough to permit single-bunch-crossing timing resolution.

Signals from all of the APSPs are sampled by buffers and subsequently fed into a 128:1 multiplexer, which generates a current output with a magnitude of 100 μA per MIP. Analog values leave the multiplexer in a different sequence from the physical silicon strip order, but the mapping is fixed and the data are trivially reorganised inside Front End Driver modules upon receipt (see Chapters 4 and 9). A differential op-amp amplifies the multiplexer output to produce a differential current signal of magnitude 2 mA per MIP. Header information is added to produce an APV frame, which is read out from the chip at 20 MHz; frames from pairs of APVs are then externally multiplexed and transmitted optically at 40 MHz to the CMS counting room. An example of a single APV frame is shown in Figure 2.2.

In the absence of Level 1 triggers, the APV outputs a single pulse or 'tick mark' every 70 LHC clock cycles, which is used to verify and maintain synchronisation with the downstream readout hardware. The event frames consist of a 12bit digital header followed by the 128 analog strip values. A header is composed of 3 start bits (used to distinguish events from tick marks), an 8bit pipeline address, corresponding to the analog pipeline column from which the strip data are being read, and finally a single APV error bit. It is possible to verify synchronisation between

APVs by comparing the pipeline address in each header with the value predicted by the APV Emulator; this is described in Chapter 3.

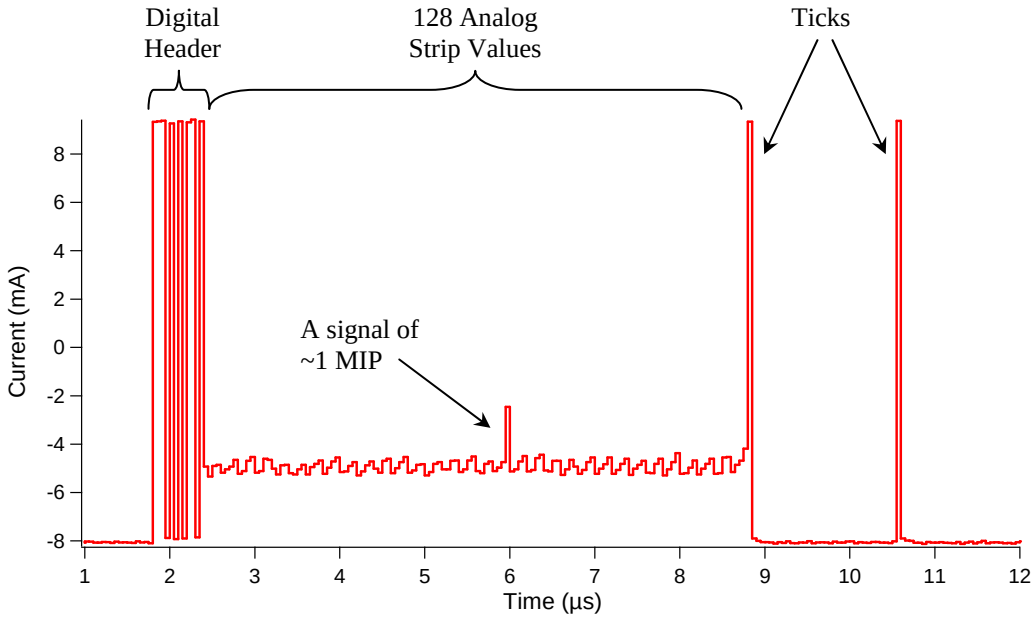


Figure 2.2: An example APV25 frame, containing a single ~1 MIP hit signal, followed by two 'tick marks'.

Control of the APV is via a serial interface, based upon the Phillips I^2C specification [37]. This provides access to internal registers which are used to set the operating modes of the chip and the internal voltage and current levels; error status information is also stored internally and may be read out. A calibration circuit enables charge to be injected before the preamplifier of each APV channel, allowing performance to be checked without applying external input signals. Pulses generated by the amplification stages of the chip upon receipt of a calibration signal can be reconstructed (to a resolution of 3.125 ns) by combining samples obtained whilst stepping though a series of delays between injecting charge and providing a Level 1 trigger. The shape of these pulses and the gain of the system can then be verified.

2.3 Production Testing of the APV25

APVs are used to read out every silicon strip in the CMS Tracker. The performance of the detector is therefore fundamentally dependent upon the quality and reliability of

the manufactured APV25 chips; faulty devices represent dead channels, which lead to a reduction in Tracker efficiency and the potential loss of physics data. It is therefore vital for every APV to be rigorously verified, in order to ensure that only fully operational chips with excellent analog performance are installed in the Tracker Front End Readout Modules.

2.3.1 The APV25 Wafer Probe Test

The APV25 functionality tests are performed using a semi-automated Micro-Manipulator probe station. As shown in Figure 2.1, the chips are manufactured on 8 inch (~200 mm) silicon wafers, each containing 360 devices. For maximum efficiency, screening is carried out before the wafers are diced; once a wafer has been correctly aligned by hand (requiring only ~10 minutes), the probe station can automatically scan every APV location, enabling all 360 chips to be tested in ~7 hours.

Figure 2.3 shows the complete experimental apparatus used for wafer probe testing. The probe station is controlled through an RS232 connection by a PC running an application written in the graphical programming language LabVIEW [38]. APV control and readout is also performed by the LabVIEW-equipped PC through a VME-based data acquisition system, accessed via a National Instruments VXI MXI-2 VME bus adaptor; a SEQSI (‘sequencer for use in silicon readout investigation’) drives the APV under test with clock and simulated Level 1 trigger signals, while an 8bit Struck Flash ADC digitises the output frames and a VME-to-I^2C converter enables the APV control registers to be set. Contact with the pads surrounding an APV is achieved using a custom probe card, which implements buffering, input/output termination and decoupling as near to the probe pins as possible.

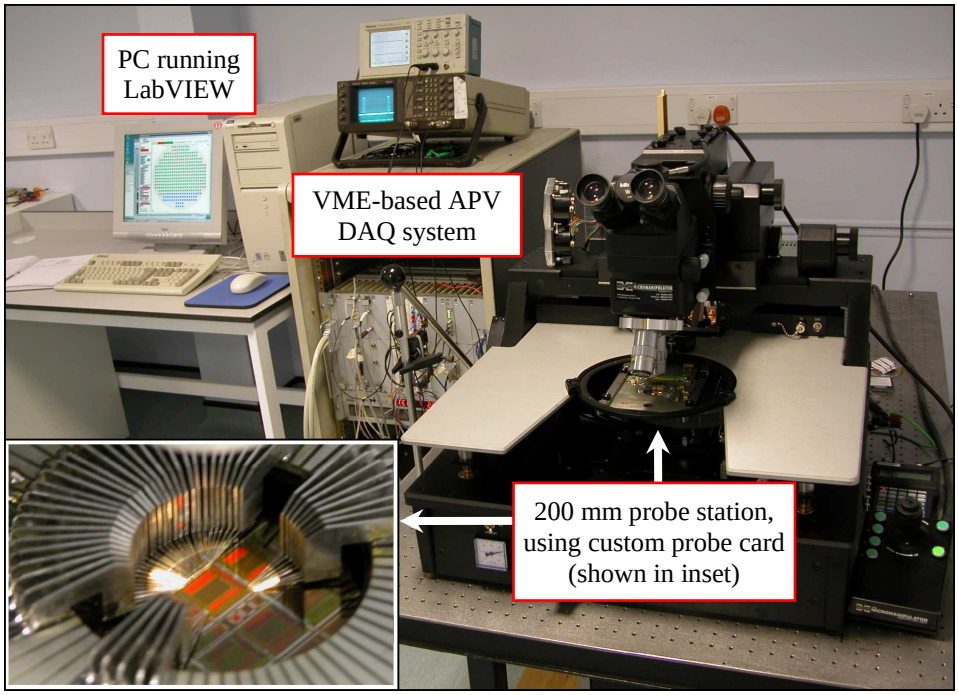


Figure 2.3: The hardware setup used for APV25 wafer probe testing.

The LabVIEW software automatically carries out a series of routines that comprehensively test the functionality and performance of each APV25 [39]. All results are written to file and a summary of the wafer status is generated in the form of a 'wafer map'. This shows the physical locations of all 'good' APVs which should be selected from the wafer after it has been diced.

Measures are initially taken to identify chips that could potentially suffer from premature failure. Following the manufacturers' recommendations, several functionality tests are conducted while the APV is biased at 50% higher than the nominal ±1.25V rating; the chip is then subjected to twice the normal bias for a period of 1s. Voltage stressing [40] is designed primarily to accelerate the breakdown of weak transistor gate oxides, exposing manufacturing faults that may not immediately become apparent under typical operating conditions.

Basic digital functionality is then assessed (at the nominal voltage level) by issuing read/write commands to all of the internal registers of the chip and checking for any transaction or data errors. Correct behaviour with random triggers is verified

(one frame should be output for every trigger) and one digital header is examined for formatting errors.

Analog performance tests involve configuring the APV with the typical settings required for normal operation and comparing a number of measured chip parameters with predefined acceptable limits. Checks on the power supply currents and both the channel and pipeline pedestal and noise values are performed. Additionally, the internal calibration feature is used to reconstruct the CR-RC pulses generated by the initial amplifying stages of the APV in peak and deconvolution modes; the uniformity of the pulse shapes is verified along with the uniformity and magnitude of the signal gains.

2.3.2 Wafer Yield

It is important for each wafer to provide as many correctly functioning APV25 chips as possible. At the financial level, a faulty chip represents wasted manufacturing effort, leading to increased production costs. More importantly, a consistently low yield (fraction of ‘good’ APVs) may indicate flaws in the circuit design or insufficient levels of quality control at the production foundry.

Wafers are manufactured in lots, typically consisting of 25 individual wafers that are simultaneously processed during a single production run. An initial engineering batch (lot 0) demonstrated a high average yield of 81% and two further lots were ordered. These, however, were found to have very low average yields of 27% and 10%. The chip manufacturer analysed samples of the problematic wafers and discovered a number of apparent flaws in the silicide ($TiSi_2$) layers that provide contact between the substrate of the APV transistors and the source and drain implants. Two replacement lots were provided once the silicide processing issues had been resolved but these also exhibited low average yields of 33% and 47%, indicating

that the silicide defects may not have been the reason for poor wafer quality. Thus began a major investigation into the cause of the phenomenon, by the manufacturers' failure analysis teams and groups at CERN, Imperial College (IC) and the Rutherford Appleton Laboratory (RAL).

During the initial studies at IC, software was written using the data analysis package Igor Pro to extract the wafer probe test results from the database generated by the LabVIEW control program, and format them such that measured APV parameters could be associated with physical regions of the wafers. A series of detailed 'maps' were produced for all of the wafers that were available at the time (lots 1 to 5) and a number of examples from lot 4 are shown in Figure 2.4; note that each probe test result is identified by a code word, the meanings of which are explained in Table 2.1.

Result Identifier	Unit of Measurement	Acceptable Range	Description
I_VDD	mA	70 – 100	The current measured in the nominal +1.25V power supply rail.
I_VSS	mA	120 – 170	The current measured in the nominal -1.25V power supply rail.
Peds_D	ADC Counts	40 - 90	The average channel pedestal measured in deconvolution mode.
CNoise	ADC Counts	< 2	The average channel noise measured in deconvolution mode.
PassFailX	N/A	N/A	The wafer probe tests can be divided into 5 general categories. Each APV is given a value (PassFailX) that identifies which category of test is failed in the event of an error: 0 – APV passes all tests 1 – Voltage stressing failure 2 – Digital functionality failure 3 – Current level failure 4 – Channel performance failure 5 – Pipeline performance failure The error status values are arranged from 1 to 5 in decreasing order of severity; only the most significant failure is noted.

Table 2.1: The code words used to identify the wafer probe test results in Figure 2.4.

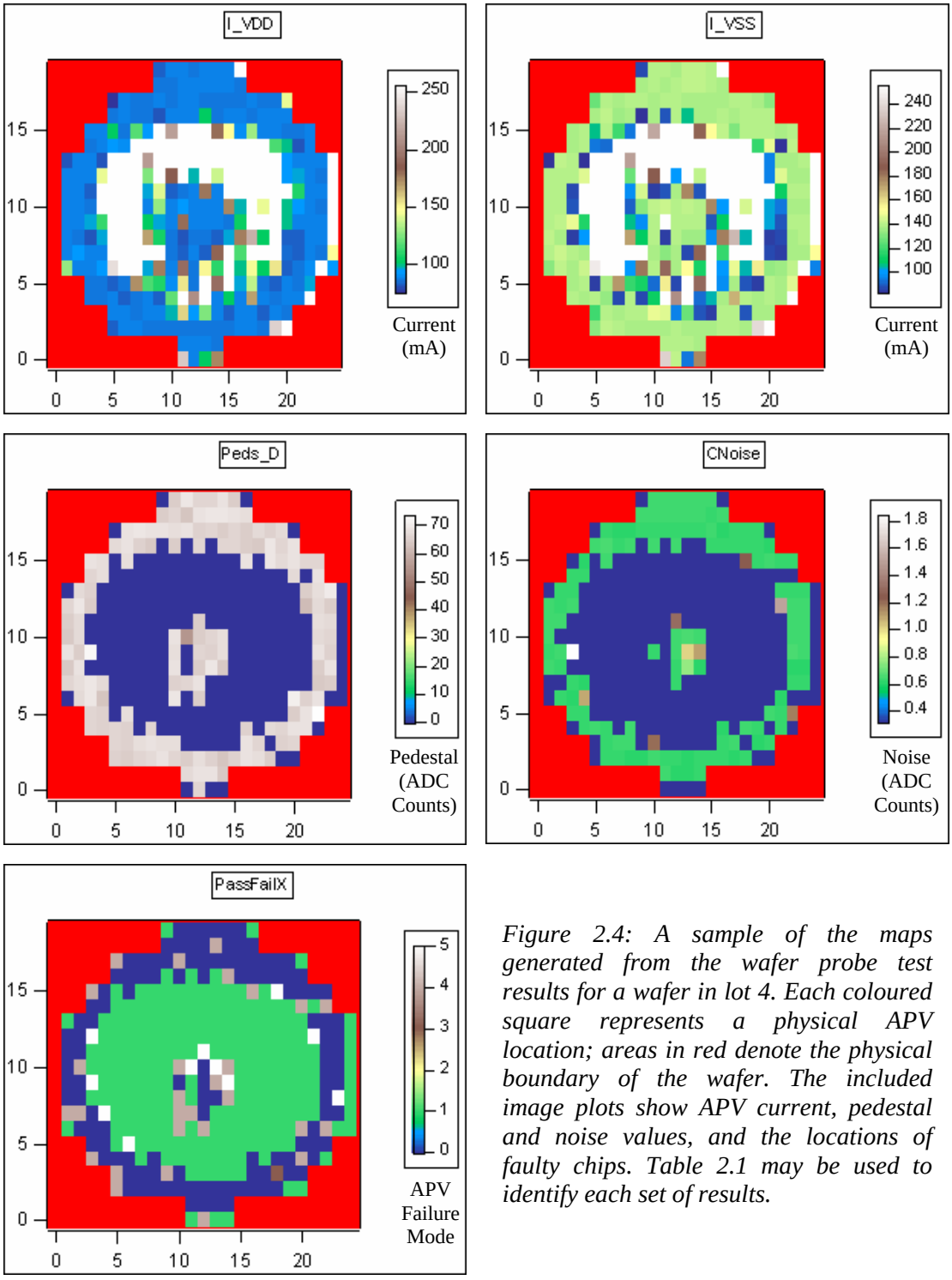


Figure 2.4: A sample of the maps generated from the wafer probe test results for a wafer in lot 4. Each coloured square represents a physical APV location; areas in red denote the physical boundary of the wafer. The included image plots show APV current, pedestal and noise values, and the locations of faulty chips. Table 2.1 may be used to identify each set of results.

Most of the wafers with low yields exhibit properties that are similar to those demonstrated in Figure 2.4. Correctly functioning APV25s typically occupy a thin ring near the outside edge of a wafer and a small circular region at the centre. Faulty chips, located in a concentric ring between these two features, are often grossly defective; they typically fail the initial voltage stressing test and have little or no

digital functionality. Such well defined failure patterns indicate a problem with the manufacturing process as opposed to an inherent flaw in the APV design (in which case a more uniform distribution of failures would be expected). A catalogue of these results was produced and supplied to the manufacturer, as an aid to their investigations.

The APV25 resembles other HEP designs in that it combines analog and digital functionality; unlike most commercial devices (which are mainly digital), the layout has a relatively non-uniform distribution of tracks and capacitor components, leading to an inhomogeneous filling of metal in the final die. It was suggested by the manufacturer that this could potentially cause the metal and insulating layers of the chip to be formed with incorrect thicknesses. An extensive series of tests were therefore performed by the manufacturers' failure analysis team on samples from the low yield wafers, which included removing layers from and taking cross sections through faulty chips and checking for 'hot spots' (often indicative of faulty circuitry) using thermally responsive liquid crystal coatings.

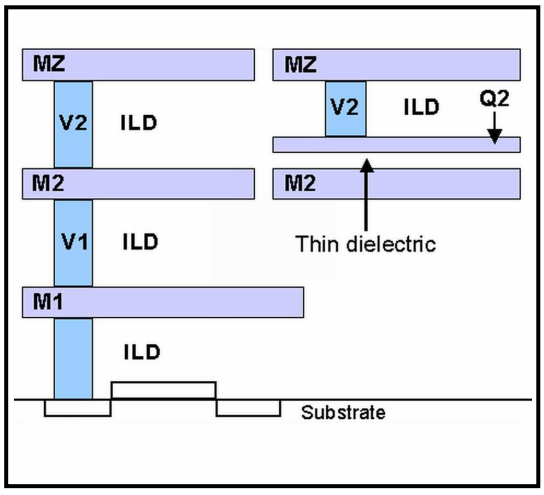


Figure 2.5: A schematic showing the 3 internal metal layers (M1, M2 and MZ) used for laying tracks in the APV25 (see text for explanation).

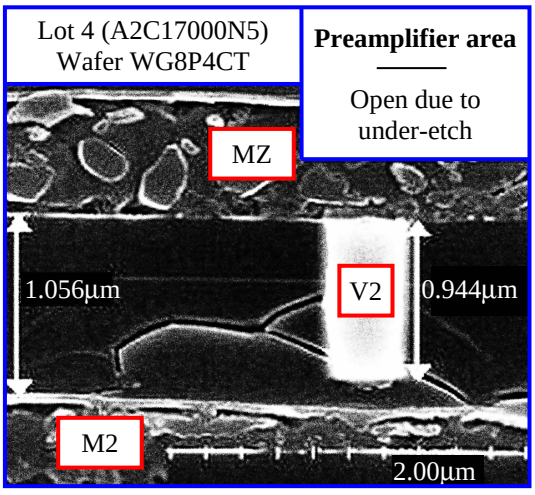


Figure 2.6: A cross section through the region of an APV25 preamplifier, showing an example of an open circuit due to the under-etching of a via.

A schematic of the material layers in the APV is shown in Figure 2.5. Three metal layers are used for tracks (M1, M2 and MZ), which are separated by an inter-level dielectric (ILD) and joined where necessary by vias (V1 and V2). An additional metal layer (Q2) between M2 and MZ, separated from M2 by a thin dielectric, enables the formation of capacitors. The investigations by the manufacturer revealed two kinds of fabrication error that can damage this structure. In lot 6, a number of instances of short circuits between M2 tracks were found, where Q2 material (which should have been removed) prevented etching through the full thickness of the M2 layer. However, this is now considered to be a rare isolated problem that is unlikely to recur. More commonly, in lots 4, 5 and 7, open circuits were detected between the M2 and MZ layers; vias had been etched to an insufficient depth, as shown in Figure 2.6.

This under-etching was caused by a thicker than expected ILD layer. During the manufacture of a wafer, microscopic topographic features left by each processing stage are ground away through Chemical Mechanical Polishing, to enable subsequent processes to begin from a flat surface. The atypical distribution of metal in the APV25 means that APV wafers are harder to polish than those containing more standard commercial chips. Although the APV was thus outside the original design specification of the production foundry, special dispensation had been given. Nevertheless, the allocated polishing time was inadequate, leading to increased layer thicknesses.

With the fundamental cause of low-yield wafer production identified, the manufacturer was able to solve the problem by altering the processing technique to allow for a longer polishing time, ensuring thinner ILD layers. This procedure has been used for all wafers from lot 9 onwards; to date, 29 lots have been tested, achieving excellent results with an average yield of 88% from lots 9 to 29. In total,

131,745 correctly functioning APV25 chips have been selected, meeting the full requirement of the CMS Tracker.

2.3.3 APV25 Wafer Probe Test Results

The wafer probe test is designed to check that every APV25 has the high level of performance necessary for the correct operation of the Tracker. As each chip is assessed with the same control register settings, any fluctuations in performance characteristics are due to natural (or induced) variations in the manufacturing process. Cuts are applied to the measured APV parameters in order to guarantee the uniformity of all the dies that will be used at CMS. Results from a subset of the tests, demonstrating this uniformity, are presented below.

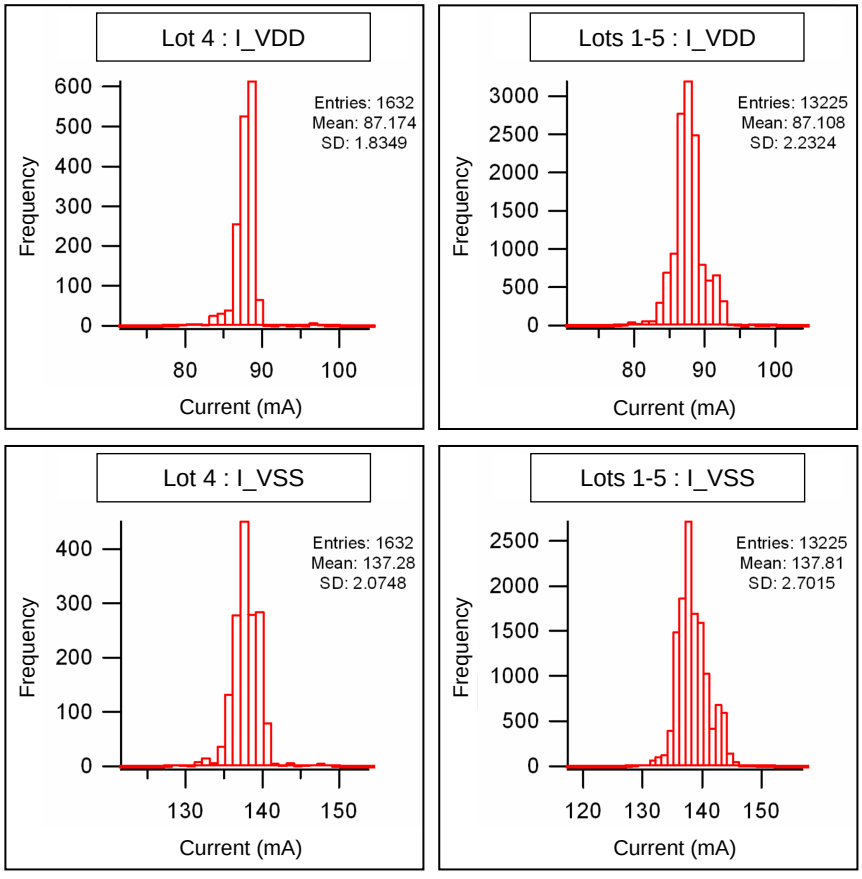


Figure 2.7: Histograms of the measured APV25 I_{DD} and I_{SS} power supply currents (excluding chips that fail the wafer probe test). Plots on the left show results from lot 4, whilst those on the right contain the combined data from lots 1 to 5. The test identifiers (I_VSS and I_VDD) are the same as those used in the wafer maps of Figure 2.4; see Table 2.1 for definitions.

Figure 2.7 shows the currents (I_{VDD} and I_{VSS}) measured in the positive and negative power supply rails of APV25s that have passed the wafer probe test. Data are given for all of the chips in lot 4, and for the combined totals of lots 1 to 5 (note that these were the only lots available at the time of the initial investigation; subsequent wafers have exhibited almost identical or improved behaviour). APVs on wafers from lot 4 demonstrate good current uniformity, with a spread (standard deviation) in I_{VDD} and I_{VSS} of ~2% or less of the mean values. Different lots have systematic variations in performance characteristics, but these are small; the currents for all chips from lots 1 to 5 have a spread (standard deviation) of less than 3% of the means. In all cases, the currents are well within the acceptable limits defined in Table 2.1.

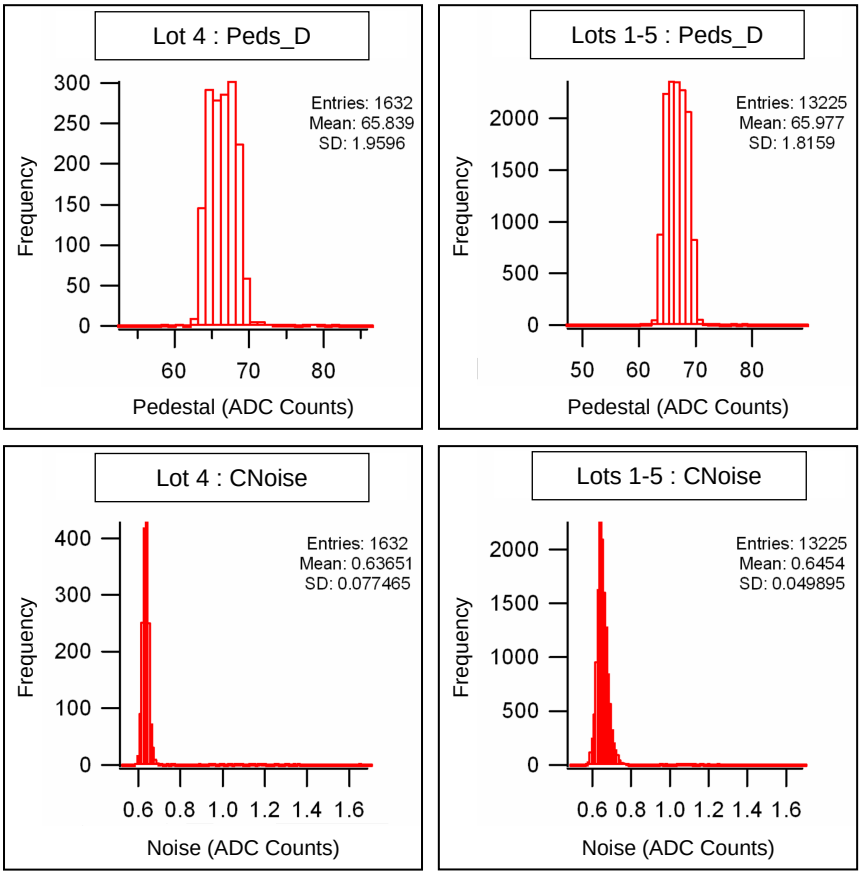


Figure 2.8: Histograms of the measured APV25 channel pedestal and noise values (excluding chips that fail the wafer probe test). Plots on the left show results from lot 4, whilst those on the right contain the combined data from lots 1 to 5. The test identifiers (Peds_D and CNoise) are the same as those used in the wafer maps of Figure 2.4; see Table 2.1 for definitions.

Histograms of the measured APV channel pedestals are shown in Figure 2.8. Each pedestal must have a value between 40 and 90 ADC counts; the mean of ~66 ADC counts for both lot 4 and lots 1 to 5 combined is very close to the centre of this range. Again, a spread (standard deviation) of less than 3% of the mean is observed for all of the chips in the five lots. Figure 2.8 also contains channel noise values. APVs from lot 4 exhibit a mean noise of 0.64 ADC counts, which is equivalent to ~540 electrons. This is near the expected minimum of ~430 electrons, obtained from more detailed studies of individual chips in a shielded environment [41].

Other measurements of pulse shapes, signal gains, and pipeline pedestals and noise demonstrate similar levels of high performance and uniformity between APV25 chips from different lots. A full account of the most recent results may be found in References [42, 43].

~ Chapter 3 ~

The APV Emulator

3.1 CMS Tracker Readout Limitations and the APVE Solution

The operation of the CMS Tracker is limited by the maximum rate at which data can be read out from an APV25 chip, corresponding to one event every 7 μs. The maximum average frequency of L1 triggers will be 100 kHz (an average read out period of 10 μs), but since L1 triggers occur with a Poisson distribution, random fluctuations in their rate will lead to trigger separations of less than 7 μs. This is the reason for buffering events that follow each trigger within an APV (see Chapter 2), to ensure that data corresponding to the next trigger are stored even if they cannot yet be read out.

However, on the occasions that random increases in the L1 trigger rate persist for more than a few triggers, it is possible that the APV buffer length might be insufficient for the storage of all required event data. If the pipeline of an APV were to fill up, all further data would be lost and it would be necessary to reset the chip before normal operation could be resumed. It is therefore clear that the status of the APV pipeline buffers must be closely monitored during the running of the CMS Tracker. L1 triggers can then be vetoed whenever buffers approach their limit, to avoid data loss and the disruption of resetting the readout system.

The geometry of CMS means that the status of APV chips in the Tracker itself cannot be monitored on-detector with a fast enough response. In the ~330 ns (~13 bunch crossings) it requires for a signal to travel along the 100 m optical link from detector to counting room it is possible that further triggers could be sent to the APVs.

Consequently, an APV Emulator board (APVE) [44] has been developed; the APVE is able to precisely simulate the buffer of an APV in the Tracker and determine its status within 50 ns, and will be installed inside the CMS counting room sufficiently close to the Trigger Control System that the transmission time for status signals will be negligible.

In addition to preventing buffer overflow, the APVE also outputs the address of the pipeline memory location from which data are read when an L1 trigger is received. The Silicon Tracker has been designed to take advantage of synchronous APV operation, as this ensures that all APV chips simultaneously output frames corresponding to the same bunch crossing events and thus simplifies the readout system. By comparing the APVE pipeline address (the "golden" pipeline address) with those obtained from real APVs, synchronisation of the Tracker can easily be verified.

3.2 The Tracker Data Acquisition Control System

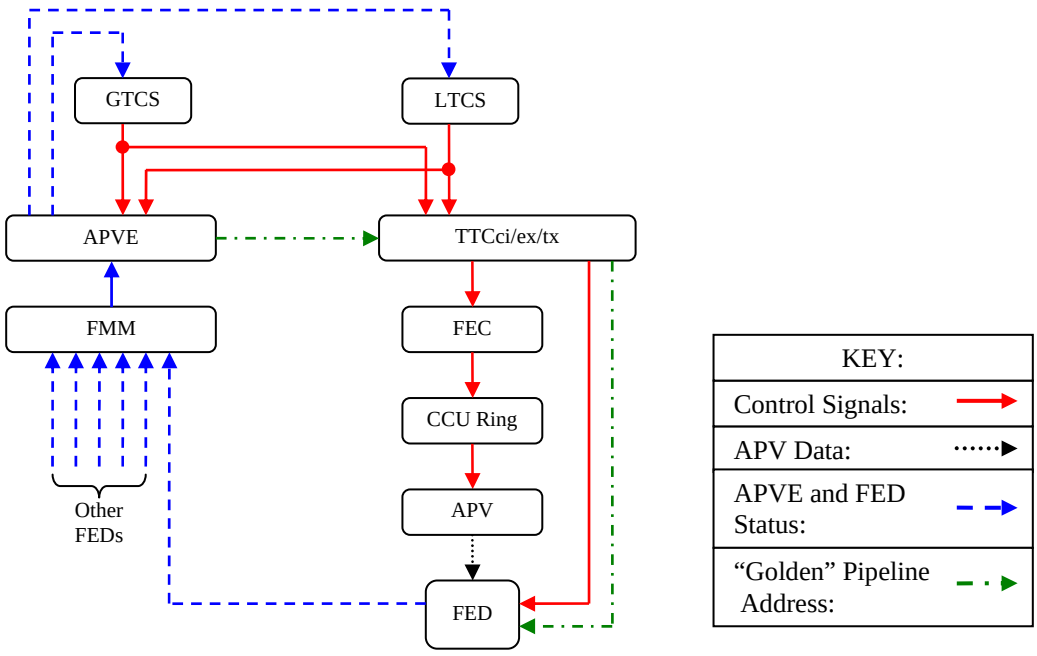


Figure 3.1: The CMS Tracker control system

A schematic showing the position of the APVE in the CMS Tracker control loop is given in Figure 3.1. The Tracker itself is divided into four major sections (called partitions), which have independent trigger distribution and readout systems. This permits greater flexibility when calibrating, commissioning or debugging the detector, although partitions are synchronously grouped during most normal physics data acquisition runs. Figure 3.1 shows the control loop for a single partition; four APVEs will be used in the final system.

At the heart of the control sequence are the Trigger Control Systems. The Global TCS (GTCS) is used during normal operation of the Tracker; it sends the LHC clock, L1 triggers and other relevant control signals to the APVE and to the Trigger, Timing and Control (TTC) system. An alternative Local TCS (LTCS) has the same functionality as the GTCS and allows operation of the Tracker to be maintained should the main control system be inhibited, for example during maintenance.

The TTC [45] and its associated sub-components distribute the LHC clock and L1 triggers, via a data channel ‘A’, and control signals, through a data channel ‘B’, to the rest of the detector. Signals initiated by the TTC are formatted onto the appropriate channel using a TTC*ci* (a CMS-specific version of the TTC*vi* [46]); channels A and B are then encoded with a TTC*ex* for transmission by a TTC*tx* [47] via optical links [48]. “Golden” pipeline addresses from the APVE are also routed through data channel B to the Front End Driver (FED) modules (described in Chapter 4).

Each APV receives L1 triggers and control information from a Front End Controller (FEC) module through the Communication Control Unit (CCU) ring [49]. Upon receipt of an L1 trigger, multiplexed data (including pipeline location information) from pairs of APVs are sent via an optical fibre to a FED. In addition to

processing the APV data, the FED checks the synchronisation of the APV pipeline locations with the “golden” APVE address.

Like the APV, the FED contains internal buffers that are used to store event data pending readout. The APVE can generate five different status signals depending (primarily) upon the occupancy level of the APV buffers. Similarly, the FED outputs status signals according to the occupancy of its internal memories. In order of increasing significance, the five possible status values are: Ready, Warning (when buffers are approaching overflow), Busy (when the buffers are full), Out Of Sync (when the “golden” pipeline address mismatches the actual APV25 memory location) and Error. Status messages from all FEDs are combined in a Fast Merging Module (FMM) [50]. When errors are received from multiple devices, the most significant is propagated. The APVE compares the FMM status with that determined from its own APV simulation and sends the overall most significant status signal to the currently active TCS. If a Warning state is detected, the L1 trigger rate is reduced; if the APVE transmits a Busy signal, all further triggers are vetoed until the APV (or FED) buffers have regained a sufficient number of free positions.

Of great importance when designing the CMS Tracker control loop is the length of time required for the APVE to receive an L1 trigger, determine the APV status and, if necessary, output a veto request to the L1 trigger inhibit gate of the TCS and for that veto to be applied. This time interval (the ‘magnitude’ of the control loop) should be as short as possible; there is no benefit in detecting that the APV buffers are full if it is possible for another trigger to be sent in the period before a veto is activated.

L1 triggers can occur with a minimum separation of three bunch crossings. Consequently, if maximum Tracker efficiency were to be achieved by requiring the

APVE to assert Busy only when all APV buffer locations are full, it would be necessary for the magnitude of the control loop to be less than three LHC clock cycles. In practice, the physical limitations of the control system may result in control loop times greater than three bunch crossing periods; in this case, Busy should be asserted before the pipeline buffer is full. The APVE permits the number of free buffer locations reserved for L1 triggers that occur during the control loop interval to be set programmatically by the user.

3.3 Implementation of the APV Emulator

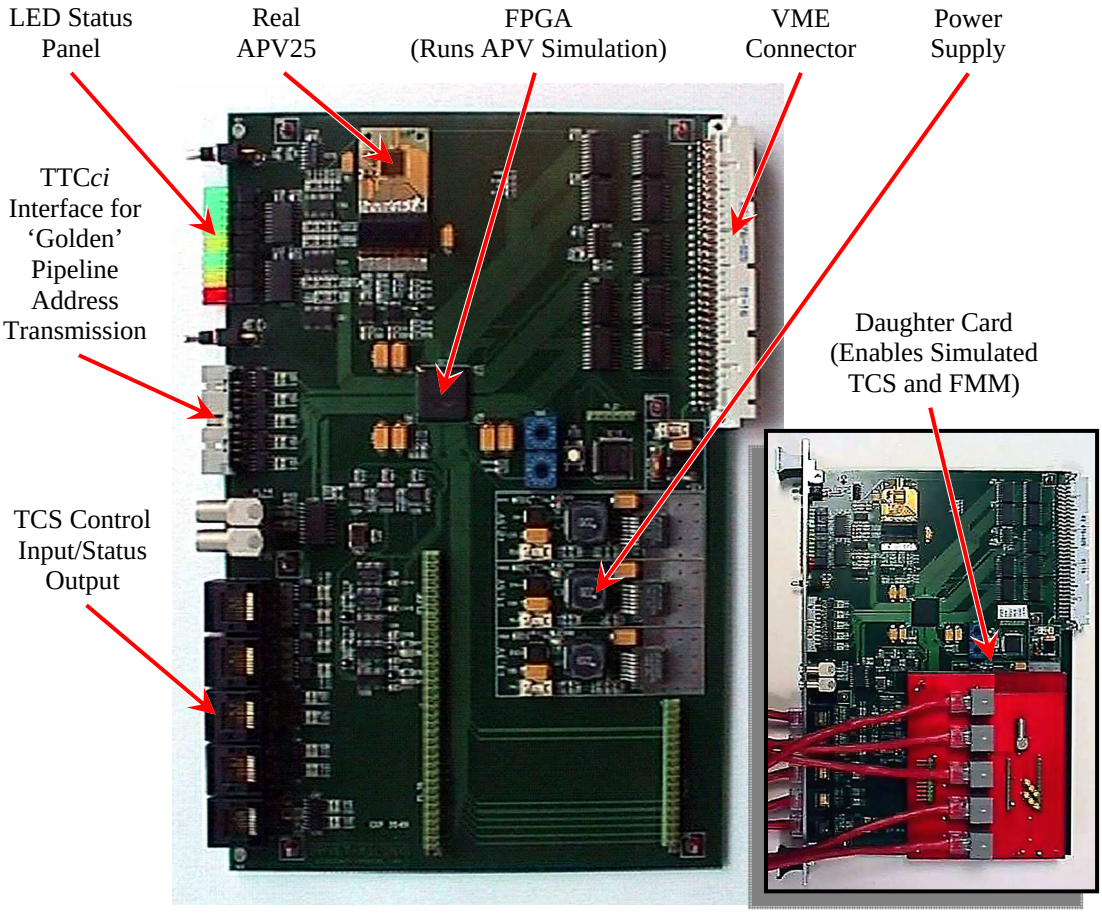


Figure 3.2: The APV Emulator VME board (shown with and without the daughter card used during simulated TCS and FMM operation).

A picture of an APVE is shown in Figure 3.2. It is implemented as a standard 6U (160 x 233 mm) VME [51] card with A24/D16 addressing. All of the board logic, with the exception of a backup clock generator, is contained within one Xilinx Virtex II

XC2V1000 Field Programmable Gate Array (FPGA) [52]. This enhances the flexibility of the device, ensuring that its functionality can be modified to accommodate potential changes to the Tracker control system. Of the FPGA logic resources, the current APVE firmware design utilises 30% of the differential flip flops (Dffs) and 67% of the look up tables (LUTs).

The pipeline buffer of an APV has two functions. It accommodates the L1 trigger latency by temporarily storing all bunch crossing samples (for a maximum of 4 μs) until a trigger arrives and additionally buffers triggered event data until they can be read out. For a given trigger latency, the period of time between each collision event and the corresponding L1 trigger is constant. Consequently, the number of pipeline locations required to buffer an event in anticipation of a trigger is known (it is the latency divided by the bunch crossing period), and can be subtracted automatically from the total pipeline length of 192 by the APVE FPGA. The remainder of the pipeline is used to store collision data that are awaiting readout, and a buffer counter inside the FPGA keeps track of its occupancy. Note that even in the case of zero latency, a maximum of only 32 triggered events can be stored; this is due to the limited size of the APV pipeline FIFO, which lists the pipeline locations of all such events to ensure that they are output in the correct order (see Chapter 2).

Two options for monitoring the APV status are provided, using either a real APV25 chip or an FPGA-based simulation of the APV pipeline logic. In both cases, the buffer counter is incremented whenever the APVE receives an L1 trigger. If the real APV is selected, the counter is simply decremented each time the chip reads out an event. Using actual APV hardware ensures that the APVE buffer status will exactly match that of the APVs in the Tracker, but in order to identify events the APVE FPGA has to check the real APV data stream for the 3 consecutive start bits of frame

digital headers (see Chapter 2); this adds a delay between a pipeline location becoming free and the detection of the corresponding output event. The APV simulation provides greater buffer efficiency, as the internal status of the APV pipeline logic is known; this ensures that the buffer counter can be decremented at the instant an internal pipeline FIFO location is cleared, without having to wait for the readout of a frame. Warning and Busy signals are generated by the APVE when the buffer counter reaches user-definable occupancy limits.

In addition to monitoring the APV buffer status, the APVE FPGA simulates the TCS and FMM described in Section 3.2. This is necessary for evaluation purposes, as the final hardware implementation of the TCS is not yet complete. Control signals from the real TCS and FMM enter the board via Ethernet connectors on the front panel; these can be replaced with simulated versions routed from the back of the board through a daughter card, shown in Figure 3.2. It is therefore possible to recreate the behaviour of the APVE-TCS control loop with an isolated APVE, enabling the functionality of the board to be verified (and the magnitude of the control loop to be measured) without additional hardware.

3.3.1 APV Glob Top Encapsulation

The real APV25 chip used by the APV Emulator is wire bonded to a small daughter PCB, shown in Figure 3.3, which fans out traces from the required APV power, control and data pads to a 17 pin connector. Most of the ~73,000 APVs at CMS will be mounted on detector modules, where they will be maintained in a thermally and atmospherically controlled environment at locations that are inaccessible during normal operation of the experiment. Conversely, the APVEs will be placed in the relatively uncontrolled environment of the CMS counting room, where it will be possible to remove and handle the boards if required. As a result, the exposed APV

die on each APVE is vulnerable to damage from static discharge and moisture in the air.

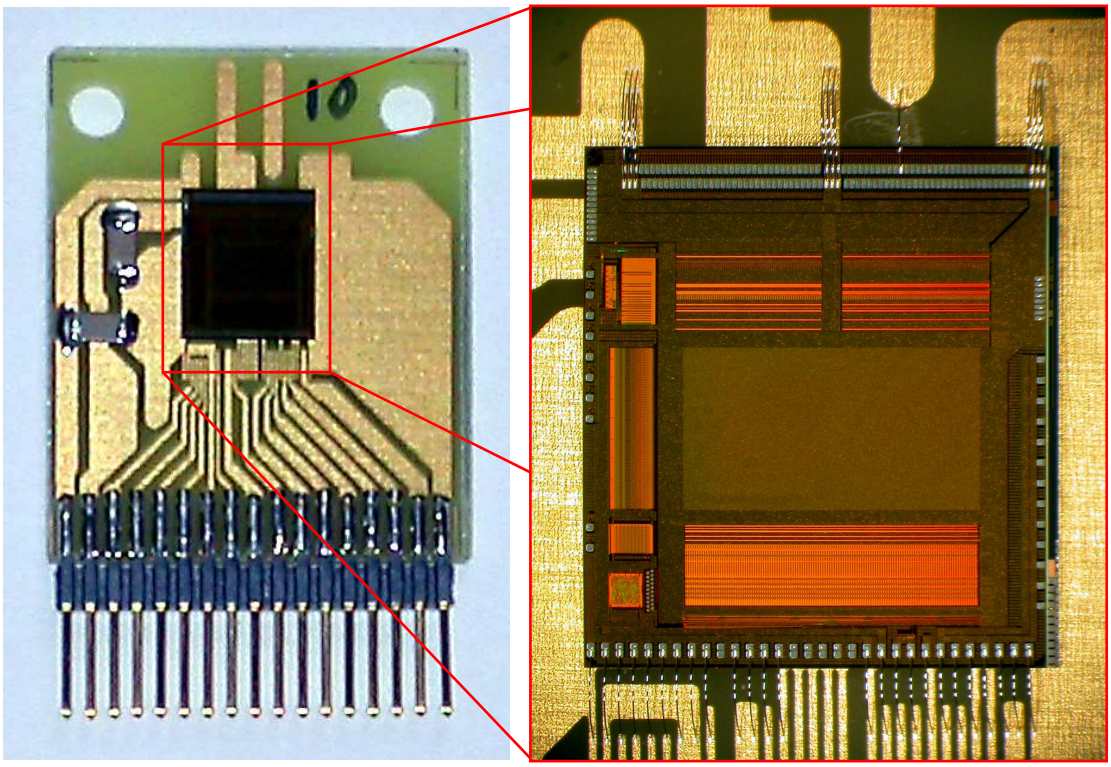


Figure 3.3: An APV25 daughter board. The enlarged view of the APV25 shows the fine pitch wire bonds (the front end of the chip is at the top of the picture, and the back end at the bottom). Note that only one of the 128 input data pads (at the front end of the chip) has been bonded; no silicon microstrips are read out by the device.

Additionally, the wire bonds connected to the APV are very fine and fragile; the back end of the chip has a pad size of 95 x 95 μm with a pitch of 225 μm, while the front end pads are 136 x 58 μm with a pitch of 44 μm. Even the slightest accidental touch whilst handing the board is sufficient to break them (although this has occurred on only a small number of occasions). It is therefore necessary to shield the APV and its bonds from human contact and moisture by encapsulating the chip with a protective seal.

Encapsulation of the die of an IC is common practice in commercial electronic component manufacture. Individual chips are often protected by a hard plastic shell, applied via an overmolding process. Alternatively, two forms of liquid encapsulation

are available; glob top and dam-and-fill. Glob top simply involves depositing a volume of liquid epoxy resin on top of a component and allowing gravity to induce the flow of resin around the die and any wire bonds. Once it has settled, the epoxy is cured to a hard finish by heating. Dam-and-fill is similar to glob top, but the spread of epoxy is better controlled by initially surrounding the component with a wall (or dam) of high viscosity material. The dam enables a lower viscosity resin to be used than in the case of glob top, which flows more easily around the die and is therefore less likely to produce vacancies. Pockets of air around a chip or wire bonds are undesirable as they lead to non-uniformities in the rate of thermal expansion throughout the volume of the epoxy coating. This may result in mechanical stress to the component, potentially leading to damage.

Plastic overmolding and dam-and-fill generally provide more reliable encapsulation than glob top. However, they also involve relatively complex procedures that make use of expensive commercial equipment. As only a very small quantity of APVs require protection, one for each of the four APVEs at CMS along with a number of spares, it is not financially viable to commission an IC manufacturer to carry out the encapsulation process. To enable sealing of the die by ‘in house’ technicians at minimal cost, the simplest option of glob top was chosen.

In order to confirm the viability of this method of encapsulation, an APV was wire bonded to a test PCB and coated with epoxy resin (EPO-TEK T7139 [53]). The resin is cured through baking at 125 °C for one hour; APVs are not typically heated above 100 °C, but the curing time is short enough that no damage is caused to the chip. As the glob top is mixed and applied by hand it is important to check the uniformity of the result, ensuring that there are no air vacancies near the die or wire bonds. This is achieved by using fine grade sandpaper to ‘cut’ a series of cross-

sections through the encapsulated device, each of which are examined under a microscope. Examples of these cross-sections for the original test are shown in Figure 3.4.

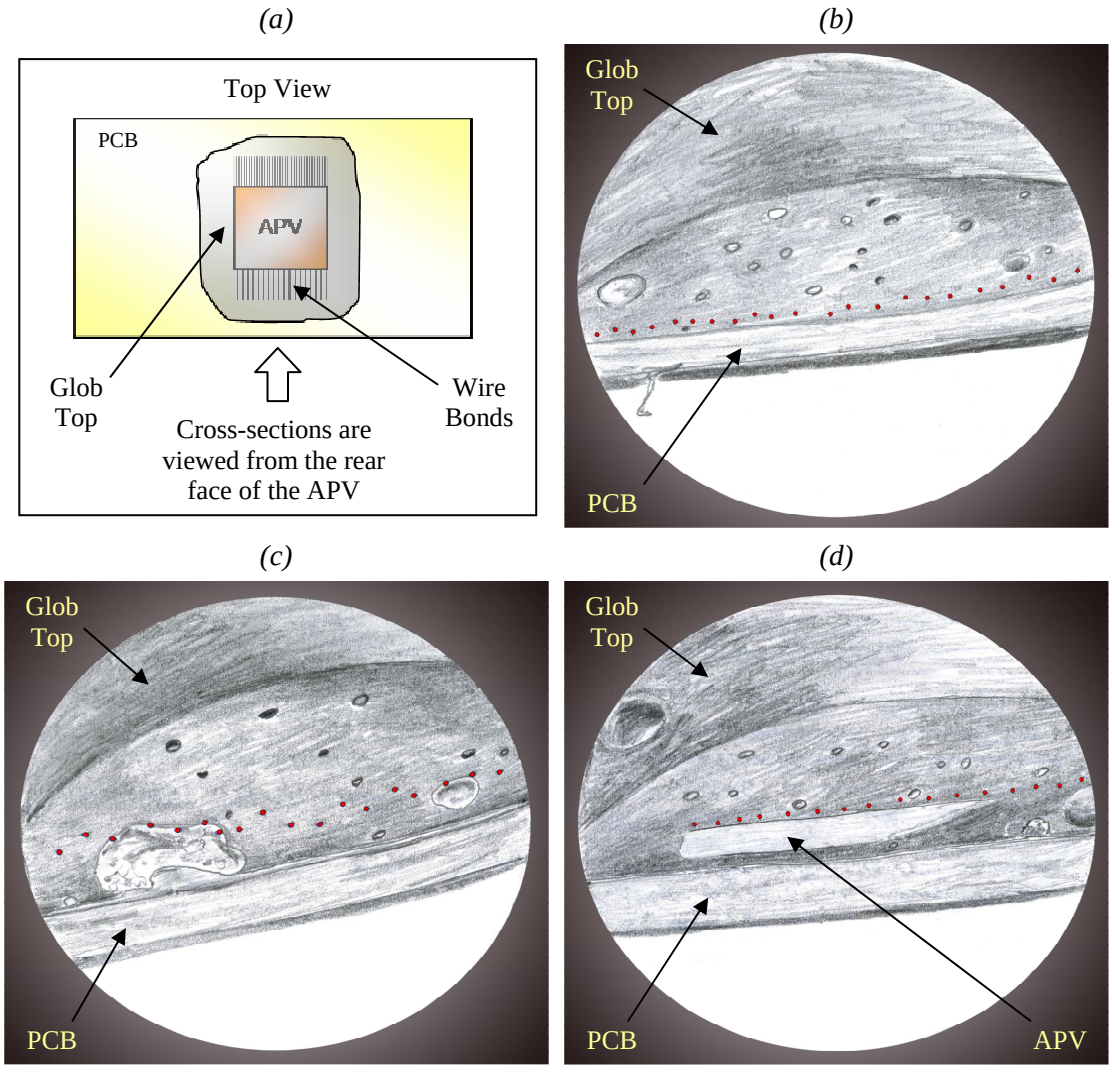


Figure 3.4: Cross-sections through a glob top encapsulated APV. Pictures (b) to (d) show progression from the outermost ends of the wire bonds to the rear face of the chip itself. Wire bonds are highlighted in red.

When the glob top is prepared and applied without special precautions, air entrapment is common throughout the volume of the resin. This is of greatest concern in Figure 3.4(c), where five wire bonds are shown to pass through the edge of a particularly large vacancy. The presence of air bubbles in regions of epoxy that are distant from the APV die indicates that the problem is not due to inadequate flow of

material around the device. Instead, air is introduced primarily during the initial resin-hardener mixing stage; it is therefore necessary to degas the glob top before applying it to the APV.

Degassing is performed by exposing the pre-mixed epoxy to a vacuum. In the absence of a standard evacuation chamber, a Boss 9000T table top vacuum sealer is used, which is intended primarily for the purpose of vacuum packing electronic devices in plastic bags for storage and transit. Sealing the resin directly within a bag would make it difficult to extract. Instead, it is first poured into a Petri dish which is then placed inside a clear Perspex box that has been designed to withstand atmospheric pressure. The box is vacuum packed in a bag at ~10 mb and the resin is allowed to degas for ~10 minutes, until all bubbles have cleared from its surface. Once removed from the box, the glob top is carefully applied to the APV and oven cured as normal.

A second test structure was prepared following this procedure, and again examined for vacancies. Cross-sections through the component were similar to those in Figure 3.4, but with a complete absence of trapped air in the vicinity of the die. At the outermost surface of the glob top a single cavity was observed, caused by air introduced from the action of spreading the material over the chip. It would be possible to remove these defects by degassing the resin again in between application and curing. However, the violent bubbling of the epoxy when it is exposed to a vacuum could potentially damage the wire bonds; as surface imperfections will have no effect on the APV, it was decided that single stage degassing would be used for all future chips.

The rate of thermal expansion of the encapsulated component is not only affected by glob top air vacancies. At room temperature, the Coefficient of Thermal

Expansion (CTE) of the epoxy resin is 28 x 10^{-6} $°C^{-1}$, whereas the CTE of the silicon used in the APV die is 2.6 x 10^{-6} $°C^{-1}$ and that of the PCB material is in the range 40-90 $°C^{-1}$. If the epoxy does not form a complete and secure bond with the PCB and APV, these differences in CTEs can lead to chip failure; temperature changes will cause each material of the component to expand by a different amount, potentially subjecting the die and wire bonds to damaging levels of stress. To ensure that the APV will survive thermal fluctuations, a temperature cycling test was conducted.

A final version APV Emulator APV25 daughter board was prepared and glob top applied, along with two PCB test structures each comprised of 128 wire bonds (also glob top encapsulated). The connectivity of the wires on each board was verified, and the functionality of the APV checked by providing it with an L1 trigger signal from the APVE and reading out a number of frames. An environmental chamber (DEL Series 3000) was then used to vary the temperature of the boards between ~0 °C and ~70 °C, the standard commercial operating range for integrated circuits. 100 cycles from ~0 °C to ~70 °C and back were repeated, taking approximately 5 days.

After thermally stressing the components, each APV and test structure wire bond was checked and no damaged connections were found. The functionality of the APV was again verified by driving it with L1 triggers and reading out events; no performance changes were observed. Thus, glob top encapsulation of the APV die provides the necessary level of chip protection without adversely affecting the thermal vulnerability of the final encapsulated component.

3.4 APVE Software

Operationally, the APVE is a fairly simple device. Once configured the logic may be left to run almost without intervention, as the APV status is automatically determined

from the input L1 trigger signals and transmitted directly to the TCS. During normal operation at CMS it should only be necessary to poll the board at regular intervals in order to check for APV errors and synchronisation loss, recording the internally generated status report if any problems occur.

Despite the simplicity of the final user interface, actually configuring the APVE and controlling the optional simulated TCS and FMM systems require sequenced read and write access to over 70 command registers within the APVE FPGA. Individual register access is provided by an ApveObject class written in C++, which makes use of the Hardware Access Library (HAL) [54] for VME read and write operations.

The HAL is a general purpose library developed for the CMS online software environment to facilitate the development of hardware control routines. It essentially wraps the low-level drivers of the hardware interface used to access a board with a set of generic read and write functions. This enables the interface to be changed very easily without significant modifications to the higher-level user code; three different VME bus adaptors have been used during the development of the APVE and it is possible to switch between them trivially without editing the ApveObject code. Additionally, the HAL allows the register address locations of a device to be stored in an address table file that is external to the source code, permitting register offsets to be changed in the firmware of the board without having to recompile the software.

An ApveApplication class provides the user interface to the APVE, wrapping the ApveObject routines with a small number of essential top-level control functions. It is divided into three sections, corresponding to the three main logic blocks of the APVE FPGA; the simulated TCS, the simulated FMM and the APV Emulator itself. Each system is configured by its own simple initialisation function, while additional

routines enable the detection of board errors and the starting and stopping of the simulated TCS. Furthermore, the ApveApplication class inherits from the ApveObject so that lower-level access functions remain available for debugging purposes. Board configuration parameters are stored in and managed by an ApveDescription object, which reads settings from (and writes them back to) an XML (EXtensible Markup Language) [55] ApveDescription file; XML is the format used for the storage of all device configurations at CMS (a description of the XML format is given in Chapter 5). The ApveApplication constructor requires an ApveDescription object, which is then accessed by each of the three system initialisation routines.

The APVE software provides complete control of an APVE, with full logging of all operations (via log4cplus [56]) and graceful exception handing in the event of fatal errors. Before it can be integrated with the Tracker online software environment it must be wrapped with an additional class layer in order to make it fully compatible with XDAQ [57]. This is a run-time environment for general data acquisition applications, used by all of the hardware readout systems at CMS. The XDAQ executive manages all networking and data transport throughout any networked cluster of computers and custom hardware so that applications can be written without regard for the network they are running on. A preliminary APVE XDAQ framework exists but additional work on the network transfer of logged messages is currently ongoing. However, this should prove to be a fairly simple extension of the code.

3.5 Impact of the APV25 Buffer Size on Tracker Deadtime

In order to prevent APV buffer overflow, the APVE must request a veto on a certain number of L1 triggers. This will increase the deadtime of the Tracker, as data corresponding to those triggers will be lost. To assess the impact of the APVE and the magnitude of the APVE-TCS control loop (described in Section 3.2) on Tracker

efficiency, the deadtime was measured as a function of the maximum APV buffer occupancy threshold; that is, the number of APV pipeline locations that can be filled before the APVE asserts 'Busy'.

The simulated TCS was used to supply the APVE with Poisson-distributed L1 triggers at the maximum expected LHC rate of 100 kHz and the performance of the board was monitored with both the real and FPGA simulated APV implementations. For an APV in peak mode (see Chapter 2), 32 of the 192 pipeline buffer locations are available for storing event data whilst awaiting readout following a trigger. In deconvolution mode, three data samples are required for each trigger and so there is sufficient pipeline space for only 10 events. Deconvolution mode therefore presents the greatest challenge to the APV buffers, and was used during the measurement in order to obtain worst case values.

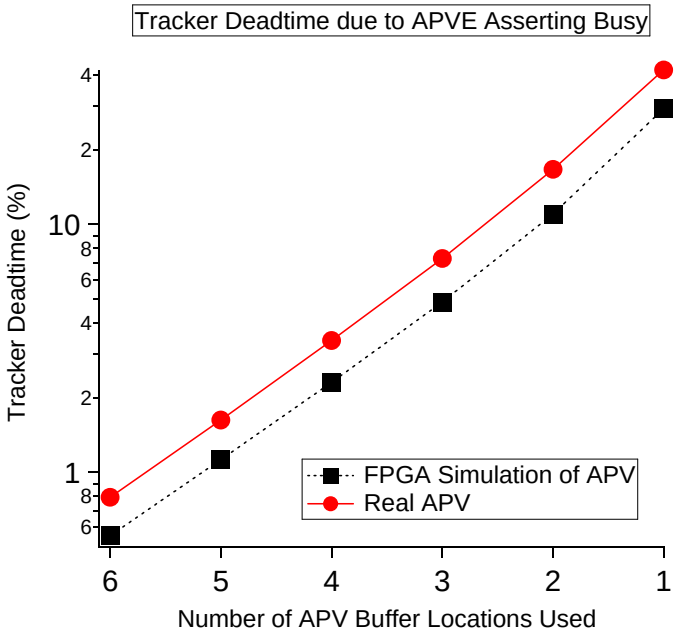


Figure 3.5: A plot of the Tracker deadtime as a function of the effective APV buffer occupancy level at which 'Busy' is asserted. Results are shown for measurements made with both the real and simulated implementations of the APV.

A plot of the Tracker deadtime versus the (effective) number of APV buffer locations in use when the APVE asserts Busy is shown in Figure 3.5. This deadtime

represents the fraction of L1 triggers discarded while the APVE is in a Busy state. It is a rule of CMS that triggers occurring with a separation of less than three bunch crossing periods (75 ns) are automatically vetoed, and these are excluded from the results.

It can be seen from Figure 3.5 that the Tracker deadtime decreases (approximately logarithmically) as the maximum allowed APV buffer threshold increases; clearly, a larger available pipeline has a lower probability of overflowing. As expected, the Tracker operates more efficiently when the buffer status is determined from the APV simulation, due to the fact that the response of the system is not limited by the readout delay associated with the real APV. Using the simulated TCS, it is only possible to operate the APVE when it is configured to assert Busy once six or less of the available pipeline locations are full. At least four locations are required to buffer events that occur between the receipt of an L1 trigger by the APVE and the application of a resultant veto signal. This indicates that the magnitude of the control loop implemented on the APVE board is of the order of 12 LHC clock cycles (300 ns). Consequently, the minimum achievable Tracker deadtime is 0.55%.

The magnitude of the simulated Tracker control loop on the APVE is larger than that expected in the final CMS system. This is a direct result of running the APV Emulator logic and the TCS model within a single FPGA. At CMS, the TCS is driven by the main 40 MHz system clock; this clock is transmitted to the APVE via the TCS output, and used to drive the APV status signal that is passed back to the TCS. As a result, the TCS and APVE are automatically synchronised. In the simulated control loop there is no external hardware available to provide a main system clock, and so there is no fixed starting point for a TCS-to-APVE-to-TCS clock loop. The APVE board was therefore designed such that the simulated TCS and APVE logic blocks

each run with a local 40 MHz clock source; whenever a transmission is made from one system to another, these clocks must be synchronised with a sync unit. This process requires approximately 5 clock cycles to complete, and must be performed at both the input and output stages of the APVE logic. Without the sync units, the control loop would have a magnitude of approximately 2 clock cycles and it would be possible for virtually all of the APV pipeline locations to be used for buffering events pending readout.

In the final Tracker control system it will still be necessary to synchronise the data lines between the TCS and APVE, but this can be achieved simply by tuning the lengths of the physical cable connections. The APVE itself requires one clock cycle to determine the APV status after receiving an L1 trigger, and the transmission of signals between the TCS and APVE is also expected to take of the order of one clock cycle. If the final TCS is able to impose a veto on triggers within three clock cycles then it should be possible to allow 9 APV buffer locations to be filled before the APVE asserts Busy.

In order to predict the Tracker deadtime that will be caused by the limited APV buffer size at CMS, data were taken from an existing C code model of the APV pipeline logic (based upon the work in Reference [58]) in which the magnitude of the control loop is ignored; this enables all of the buffer locations to be used. The results from the model, compared with the measured APVE deadtime when using the APV simulation, are shown in Figure 3.6. It can be seen that the model agrees well with experimental data, indicating that it may be used to extrapolate the performance of the Tracker.

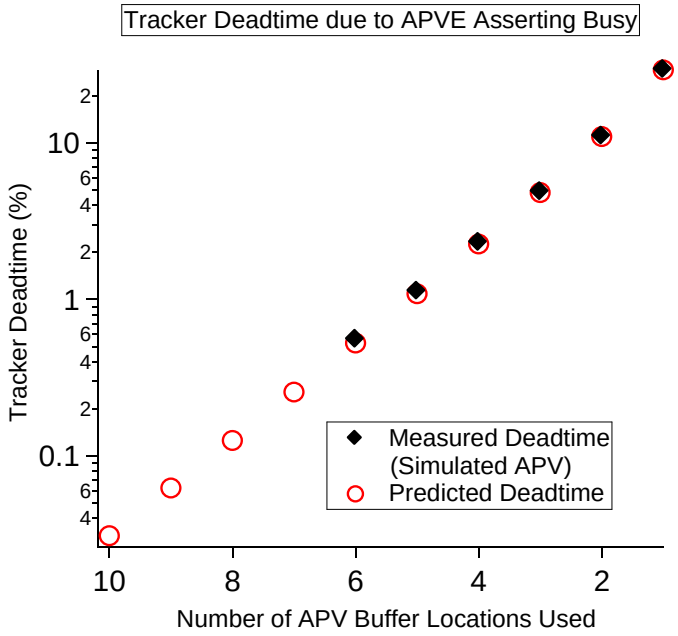


Figure 3.6: A plot of the Tracker deadtime as a function of the effective APV buffer occupancy level at which 'Busy' is asserted. Results measured from an APVE running the FPGA-based APV simulation are compared with predicted values from a C code model of the APV pipeline.

If the magnitude of the final control loop is between 3 and 5 LHC clock cycles then the maximum buffer occupancy threshold may be set to 9 pipeline locations, as stated above. In this case, the model predicts that, with APVs running in deconvolution mode, 100 kHz L1 triggers and the APVE utilising the pipeline logic simulation, the deadtime introduced by the APVE asserting Busy will be of the order of 0.06%. Within the Tracker, the vetoing of triggers that have a separation of less than three bunch crossing periods leads to a corresponding deadtime of ~0.5%. Consequently, the limited APV buffer size should not have a significant effect on the normal operational deadtime of the CMS Tracker.

~ Chapter 4 ~

The Front End Driver

The CMS Silicon Tracker has over 9 million detector channels, which are read out by approximately 73,000 APV25s at average L1 trigger rates of up to 100 kHz. At expected track occupancies, the Tracker will generate over 70% of the total data output by the CMS detector [28]; the combined throughput from the multiplexed pairs of APVs amounts to a data volume of approximately 1.4 TB/s. This is far higher than the rate at which the CMS DAQ can process hit information. If data from all of the detector channels were required by the DAQ at every event, it would limit the L1 trigger rate to approximately 2 kHz [59].

The Front End Driver (FED) reduces the volume of data from the Tracker to a level that is compatible with the DAQ by identifying which of the Tracker strips contain hit information and discarding data from those which do not, a process referred to as ‘Zero Suppression’. Approximately 440 FEDs are required to process data from all of the APVs; each one will accept an input data rate of ~3 GB/s which is reduced through Zero Suppression to ~50 MB/s per percent of Tracker occupancy, expected to vary typically between ~0.5% and ~3% [60]. As the first stage in the off-detector readout chain, the FED additionally carries out the conversion of analog optical signals from the APVs to the digital electronic format required by the DAQ.

4.1 Implementation of the Front End Driver

The FED is implemented as a 9U x 400 mm VME64X [51] card, a picture of which is shown in Figure 4.1. It receives inputs from 96 optical fibres, each corresponding to a multiplexed APV pair; the large number of input channels leads to a very densely

populated PCB, but reduces the total number of boards required and hence minimises production costs. Eight identical ‘Front End (FE) Units’ occupy the front most region of the PCB, with each one accepting signals from 12 of the optical fibres, grouped into a single ribbon cable. The FE Units perform the majority of the signal processing on the FED, digitising the input data and running the Zero Suppression algorithms in FE FPGAs [61].

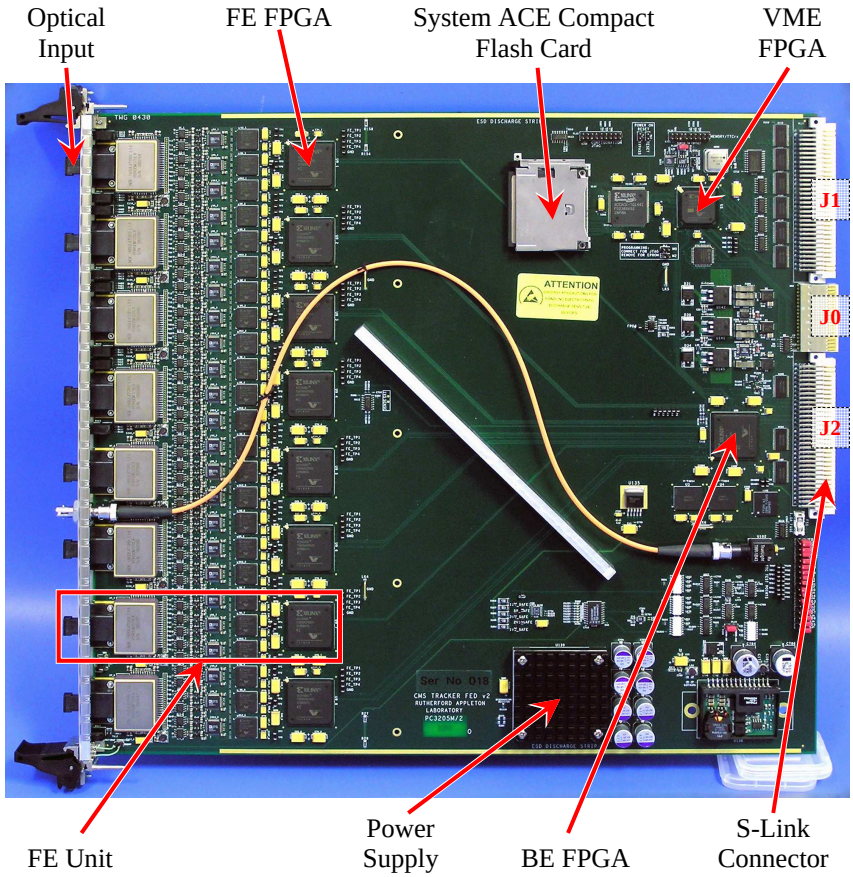


Figure 4.1: The primary side of a FED PCB.

A Back End (BE) FPGA [62] collects and formats the output of the FE Units into event packets, buffering them in memory until they are requested by the DAQ. Events are read out via a high speed electrical S-Link interface, which is discussed in detail in Chapter 7. The component density of the FED is such that it would be impractical to mount the S-Link Transmitter on the PCB itself; instead, it is connected to the rear of the board by means of a Transition Card. To permit testing of the FED in

the absence of S-Link hardware, it is also possible to read events at a reduced rate through the FED VME connection.

Programmable logic devices are used for the majority of the FED functionality to enable modifications to be made to the data processing algorithms once CMS is fully operational, if required. All but one of the FPGAs on the board are configured using a Xilinx System ACE Compact Flash (CF) product [63]; firmware for the FE, BE and Delay FPGAs [64] is stored on a standard removable CF card. It is even feasible to load a CF card with multiple configuration files, perhaps for alternate Zero Suppression algorithms, and select between them via software. A VME FPGA [65], which provides the interface to the VME bus and enables external communication with the FED, has independent configuration settings stored in an onboard EPROM. This ensures that VME access to the board is always possible, even in the event that an incorrect CF file is loaded.

4.1.1 The FED Front End Units

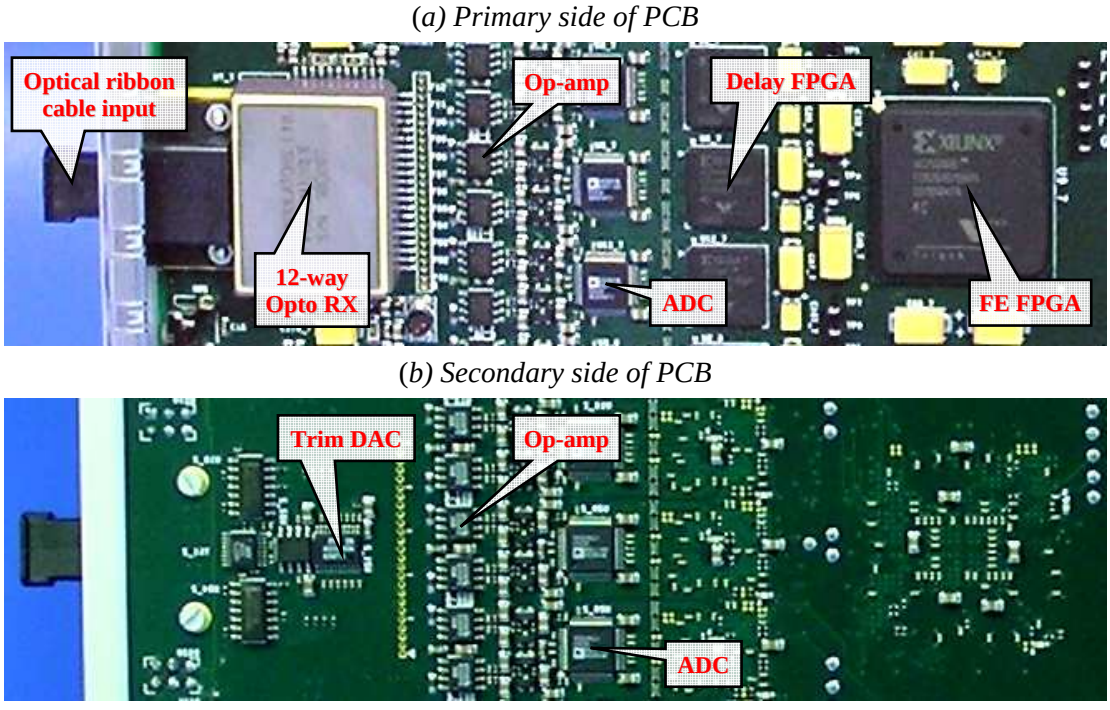


Figure 4.2: A close up of one Front End Unit of a FED PCB

A detailed picture of a single FE unit is shown in Figure 4.2. The input fibre ribbon cable is connected to an optical receiver (Opto RX) package [66, 67], which contains 12 p-i-n photodiodes each connected to an amplifier. This converts the optical signals to single-ended analog current outputs. A differential line driving op-amp [68], in conjunction with a load resistor, converts each of the Opto RX outputs into the differential voltage signal required by the FED ADCs [69]. The op-amp reference voltage for each channel is set by a Trim DAC [70], enabling a programmable offset to be introduced to the data in order to optimise its level relative to the ADC range (calibration of these offsets is discussed in Chapter 8).

In addition to digitising the analog voltage signals, the 10bit Front End ADCs enable the data for each channel to be individually delayed to compensate for variations in arrival time caused by differences between optical fibre lengths (calibration of these delay settings is also described in Chapter 8). The ADCs have independent clock signals driven by Delay FPGAs, which may be skewed relative to the main 40 MHz LHC system clock in intervals of 1/32 clock cycles (~781 ps). Digitised signals from the ADCs are then sampled by the Delay FPGAs, and programmable coarse delays of 0 to 16 clock cycles are provided by internal variable length pipelines. Finally, the data from all 12 channels are read by a Front End FPGA, which identifies APV frames and extracts the strip information.

Each FE FPGA typically runs in Zero Suppressed mode, which involves the subtraction of pedestals and Common Mode offsets from the APV frame data followed by hit identification and the removal of channels containing background signals (the operation of the FE FPGA is discussed in detail in Chapter 9). A number of alternative operating modes, used primarily for system tests, provide different levels of event processing; these are summarised in Table 4.1. For each event, a

packet containing strip data from all 12 optical channels (24 APVs) and APV status information is formed and stored in an internal 4 kb buffer, awaiting a readout request from the BE FPGA.

FED Operating Mode	Processing Steps	Comments
Scope	1. Read *n* raw data samples following L1 trigger 2. Output *n* raw data samples	FED functions as a simple oscilloscope. Number of samples ('scope length') is programmatically controlled. Enables readout of event data in the absence of optical inputs.
Virgin Raw	1. Find APV frame following L1 trigger 2. Extract raw strip values 3. Output raw strip values	FED outputs raw APV frame payload data.
Processed Raw	1. Find APV frame following L1 trigger 2. Extract raw strip values 3. Subtract APV pedestals 4. Reorder strip values to match physical locations in detector 5. Output processed strip values	Processes APV frame payload normally, but omits data reduction. All strip values are output. Can potentially be used for high occupancy heavy-ion collision events.
Zero Suppressed	1. Find APV frame following L1 trigger 2. Extract raw strip values 3. Subtract APV pedestals 4. Reorder strip values to match physical locations in detector 5. Subtract Common Mode Offset 6. Reduce strip value data width from 10bit to 8bit 7. Perform cluster (hit) finding, discarding empty strips 8. Output Zero Suppressed strip values	Normal operating mode for proton-proton collisions. Reduces input data rate from ~3 GB/s to ~50 MB/s per percent of Tracker occupancy.

Table 4.1: The operating modes of the Front End Driver.

4.1.2 The FED Back End FPGA

For every L1 trigger, the variable length data packets from each of the 8 FE Units are collected by the BE FPGA, via fast point to point links, and built into a FED event. Information from the TCS signals driving the FED is also gathered, enabling the event to be tagged with trigger and bunch crossing numbers for later synchronisation checks in the DAQ. Finally, a Cyclic Redundancy Check (CRC) algorithm is applied to the event data to generate a CRC code [71], which is appended to the event trailer word and can be used to test for transmission errors.

Each event is stored in an external 2 MB memory buffer, to help absorb fluctuations in data rates, prior to being read out by the DAQ. The BE FPGA additionally provides fast feedback status signals, which are required by the TCS to alert the system of board errors and to request a reduction in the L1 trigger rate in the

event that any of the data buffers in the Front or Back End of the FED risk overflowing.

4.2 Expected Performance at CMS Data Rates

Each fully populated FED reads a constant volume of data from 24,576 detector channels at every L1 trigger. However, the size of the Zero Suppressed events output by the FED is dependent upon Tracker occupancy, which can vary considerably between particle interactions. The rate at which the FED can be read out is limited by the S-Link connection and the processing speed of the DAQ. Although the average Tracker occupancy is expected to be low, approximately 1.7% [60], random increases for protracted lengths of time may lead to an output data rate from the FED that exceeds the capability of the readout hardware.

The buffers in the Front and Back Ends of the FED are designed to absorb these data rate fluctuations, but for high occupancies it is possible for the buffer memory to become full. In this case, it is necessary for the FED to request a reduction in the L1 trigger rate from the TCS; in practice, these requests are made before the memories are completely filled (the FED enters a ‘full’ state at a BE buffer occupancy of 87%), as there is a latency in the response of the TCS to fast feedback status signals. This is the same principle that applies to the APV pipeline (see Chapter 3), though the APV is more vulnerable to buffer overflow as its internal memories are orders of magnitude smaller than those in the FED.

It is clearly undesirable to veto L1 triggers as data from the corresponding events are lost, and so it is important to verify that the FED buffers are sufficient for the Tracker occupancies expected at CMS. Before the design of the board was finalised, the performance of the FED buffers was simulated by E. Corrin [71]; a summary of the results is presented in the following sections.

4.2.1 The Front End Buffer

Each FE FPGA contains a 4 kB buffer, which is sufficient to hold approximately 500 Zero Suppressed data packets from the 12 input channels. Point to point links between the Front and Back End FPGAs support a data rate of 80 MB/s. The buffer simulation assumed a L1 trigger frequency of 140 kHz, instead of the average CMS value of 100 kHz; as the LHC bunch crossing rate is 40 MHz and a multiplexed pair of APV frames is 280 LHC clock periods in length, this represents (to a good approximation) back to back frames and hence serves as a worst case (maximum input data rate) example.

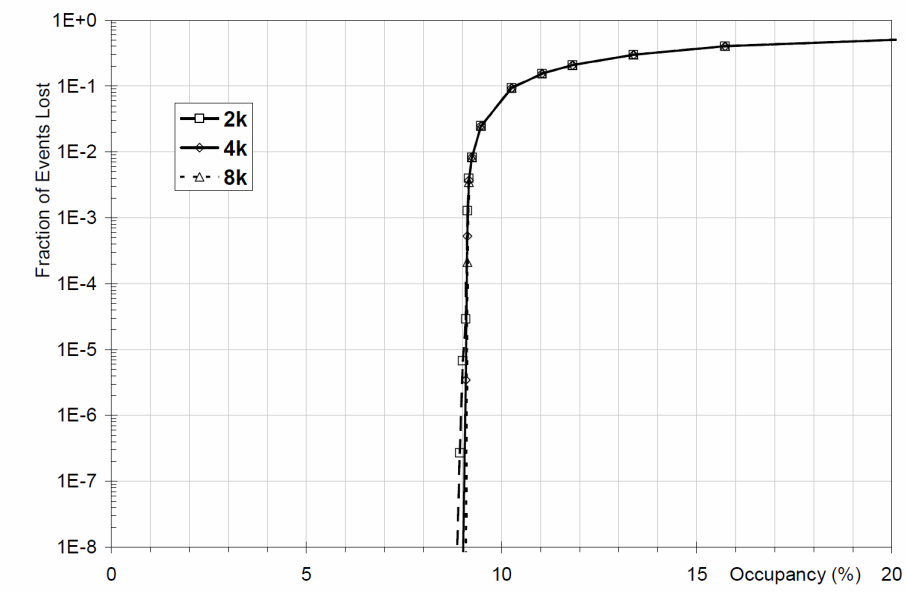


Figure 4.3: Fraction of events lost due to FE buffer overflow in Zero Suppressed mode Vs Tracker occupancy. Plots are shown for simulated FE buffer sizes of 2 kB, 4 kB and 8 kB. (Figure taken from Reference [71])

A plot of the predicted fraction of L1 triggers vetoed by the FED due to FE buffer overflow under these conditions is shown in Figure 4.3. The large buffer size means that the FE readout rate dominates (doubling or halving the 4 kB memory has little effect), but even so the FED is able to cope with a Tracker occupancy of 9% before losing events; this greatly exceeds the expected worst case average occupancy of 3%. According to the simulation, the probability of the FE overflowing is

effectively zero provided that the BE FPGA can always accept data at 80 MB/s. The performance of the BE buffer should therefore determine the behaviour of the FED as a whole.

4.2.2 The Back End Buffer

The BE buffer in the final FED has a size of 2 MB, corresponding to approximately 1,300 Zero Suppressed events. This had not been finalised at the time the simulation was written and a 1 MB buffer size was used instead, again with 140 kHz L1 triggers. Plots of the predicted fraction of events lost due to BE buffer overflow are shown in Figure 4.4; it can be seen that the occupancy at which the BE buffer reaches capacity increases with BE readout rate, which is limited by the S-Link data connection and the processing speed of the DAQ.

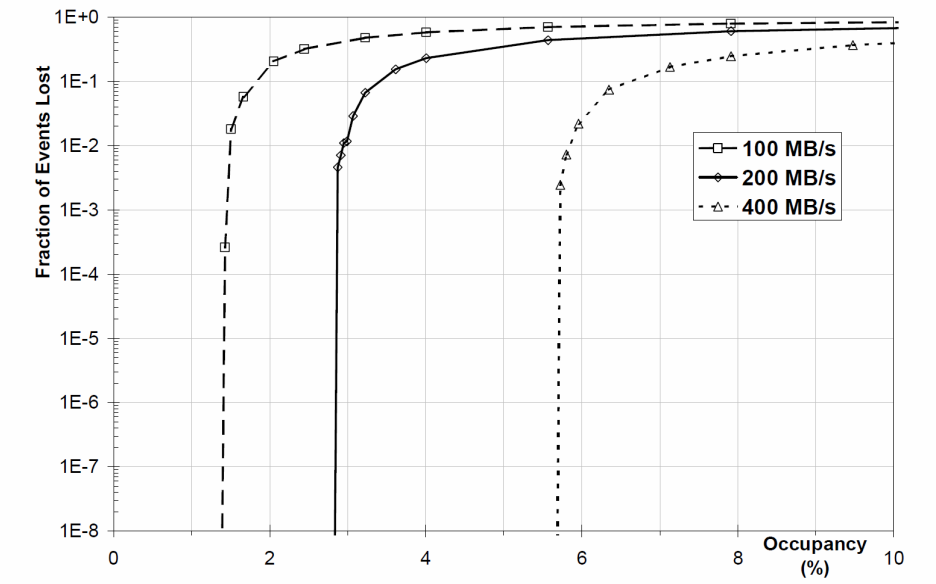


Figure 4.4: Fraction of events lost due to BE buffer overflow in Zero Suppressed mode Vs Tracker occupancy. Plots are shown for maximum BE FPGA readout rates of 100 MB/s, 200 MB/s and 400 MB/s. The x-axis intercept of each plot represents the occupancy at which the BE buffer reaches capacity. (Figure taken from Reference [71])

In practice, pairs of FEDs will be read out through each S-Link connection with the DAQ supporting a peak combined data rate of 400 MB/s [19]. Consequently, each FED has a nominal peak output rate of 200 MB/s, but additional bandwidth for individual boards is gained by pairing FEDs that read out low and high occupancy

regions of the Tracker (high occupancy FEDs may ‘borrow’ unused bandwidth from their partner). At the nominal 200 MB/s limit, the FED can cope with an occupancy of ~2.8% before events are lost. However, when the trigger rate is reduced to 100 kHz the simulation predicts that the readout performance is sufficient for occupancies of up to 4%, exceeding the worst case requirements. Doubling the buffer size to 2 MB has little effect, as readout rate dominates up to the limit of the S-Link connection speed; the extra memory in the final FED was added principally to simplify the data connections between the BE FPGA and the external memory chips.

Although the simulation provided an essential confidence check during the initial design stage of the FED, it is important to verify that the performance predictions are accurate when actual hardware becomes available. A measurement of the fraction of events lost as a function of Tracker occupancy under realistic CMS conditions was therefore made, once the full FED system test apparatus had been completed; full details are given in Chapter 8.

~ Chapter 5 ~

FED Industry Testing

5.1 The Importance of Industrial Acceptance Testing

The FED is one of the core components of the CMS Tracker readout system. Every bit of data output by the Tracker will have been processed by a FED; if the Tracker is to produce physically useful results, it is vital that every FED should function correctly. It is essential for the functionality, performance and firmware design of the FED to be rigorously tested (and this is the subject matter for the following chapters), but it is just as important to verify at a more basic level that each instance of the FED hardware is fully operational.

The FED is designed to accommodate as many channels of the Tracker as possible, in order to minimise the number of boards that will need to be manufactured, and as such is very densely populated. It is a large (9U, where 1U = 43.60 mm, is the largest standard VME size), double sided board which contains ~6,000 components joined by ~25,000 tracks. These components include approximately 40 Ball Grid Array (BGA) [72] devices and over 30 FPGAs, the largest of which have 676 pins on a 1 mm pitch. Most of the surface mount passives have a small 0402 footprint, which means that they occupy a surface area on the PCB of only 1.0 mm x 0.5 mm (with pads at each of the shortest ends). The PCB itself has 16 layers, with 6 for power and ground, and all tracks are impedance controlled.

Manufacturing a board of this complexity is a technically difficult task; consequently, there is great potential for errors in assembly. Hardware faults will be discovered during any laboratory-based performance tests, but in order to minimise

any possible delay to the FED production schedule it is crucial to identify manufacturing flaws as early as possible, preferably before the finished FED boards leave the assembly plant. Not only does this prevent the financially wasteful and time consuming process of returning faulty boards to the manufacturer, it also enables the assembly company to immediately adjust the manufacturing process should any fundamental assembly problems become apparent.

5.2 Industrial Acceptance Test Requirements

Developing an Acceptance Test that is to be run at an assembly plant presents a challenge beyond the task of verifying that a FED functions correctly. The test will be run by assembly plant operatives who have limited knowledge of the operation of the FED, and who cannot be expected to learn the low level software that is routinely used by experienced researchers in the laboratory. Consequently, the test station provided to the assembly plant must be as user friendly and automated as possible, with clear error reporting that enables even the relatively inexperienced user to quickly identify and locate any hardware faults on a board. It is also necessary to provide an interface for entering the results of any manual tests that cannot be automated using software, to ensure that they are recorded in a convenient structured format. Finally, the test results for all FEDs must be automatically organised into a database, so that the status and potential repair history of any board can be easily accessed once the FEDs have left the assembly plant and are in general use.

5.3 Testing During the Assembly Process

The company chosen for the large scale manufacture of the FED carries out extensive quality assurance checks during the assembly process, prior to the final Acceptance Test. These are intended to identify the standard types of failure modes associated with PCB production and particularly with the mounting of BGA components;

attaching a BGA to a PCB can be problematic, due to the high density of solder points on the rear of the device that are inaccessible once it has been placed. Poor BGA assembly techniques [73], PCB quality and diagnosis tools led to a number of faulty boards during the initial FED prototype runs, prior to the establishment of thorough quality assurance tests.

Before assembly commences, the tracks on each bare board are tested for connectivity by a Flying Probe system (Takaya 9400 [74]) and their impedance is checked. After the surface mount components have been attached, each board is examined using an Automated Optical Inspection (AOI) machine. This essentially compares a picture of the board with a ‘perfect’ master copy, and is able to detect almost all component misplacements (even incorrect part numbers) or soldering errors, provided that they are in visually accessible areas.

The undersides of the BGAs are not visually accessible and checking the quality of their assembly is a lengthy process that is carried out on only a subset of the FEDs. A 3D X-Ray inspection is initially used to check the placement of each BGA component and then the quality of the soldering reflow is assessed by placing endoscopic inspection tools, such as Ersascope [75], underneath the IC packages in order to visually inspect each solder ball. Finally, a sample of the assembled boards is checked with another Flying Probe system, which is able to verify the connections of all accessible devices and check the component values of the accessible passives.

5.4 The Acceptance Test

The checks carried out during assembly help to identify fundamental manufacturing flaws but they do not verify the actual functionality of the non-passive components on the board, they are not completely infallible, and the length of time required for the BGA and final Flying Probe tests is such that it is impractical to apply them to every

FED. The Acceptance Test is designed to address these issues, providing a full assessment of every assembled board in a user friendly manner.

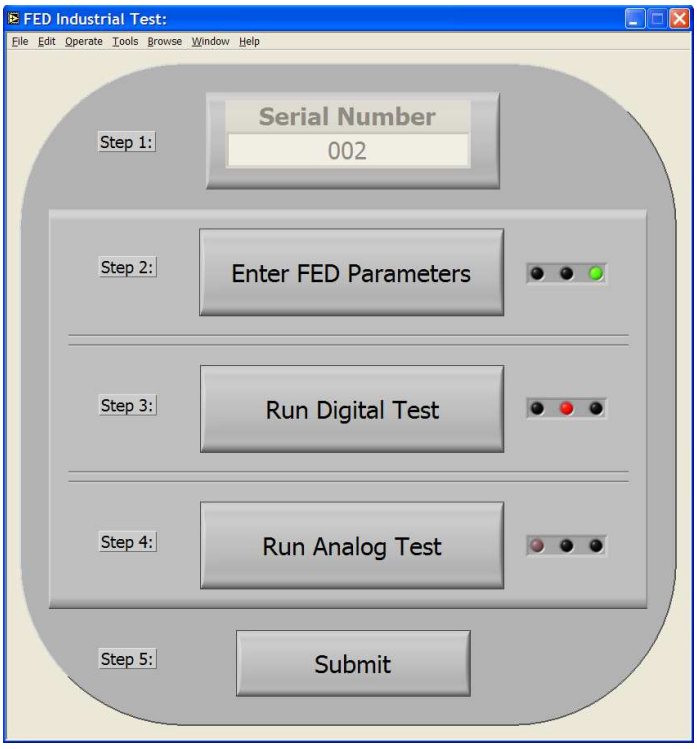


Figure 5.1: The top level interface to the FED Acceptance Test.

The top level Acceptance Test window is shown in Figure 5.1; the graphical interface is produced with National Instruments LabVIEW. It is necessary for the software to run under Linux as this is the only platform supported by the existing low level FED hardware access libraries. However, the LabVIEW development environment is available for all major operating systems and programs written with LabVIEW are essentially platform independent. This enabled the majority of the Acceptance Test to be developed under Microsoft Windows, on the most conveniently available computer.

A user is initially required to enter the serial number of the FED under test, which provides the handle by which all subsequent results are referenced. They are then prompted for a user name, as it is likely that a number of assembly plant staff will carry out the Acceptance Test and it is necessary to keep track of each one should

any queries arise. For this purpose, date and time stamps for each operation are also automatically saved.

Access is then granted to the three subsections of the Acceptance procedure. The ‘Enter FED Parameters’ button opens a panel which provides a structured entry form for the results of tests that cannot be automated and must be carried out manually. Pressing the ‘Run Digital Test’ button launches an automated test of the digital functionality of the board, and the ‘Run Analog Test’ button commences an automated high level assessment of the FEDs’ analog readout path. For clarity, the current status of each subsection is shown via a coloured light next to the corresponding button (brown → un-attempted, red → test failed, green → test passed) and each section, with the exception of the first, is disabled until the previous one has achieved a pass status.

At any point the user may press the ‘Submit’ button in order to finish operations on the current FED and add the results from each subsection to the results database. The user may re-run the Acceptance Test multiple times on a single FED, after fixing a fault for example, by re-entering the relevant serial number at the start; all additional user input and test results are automatically indexed and added to the FED database entry.

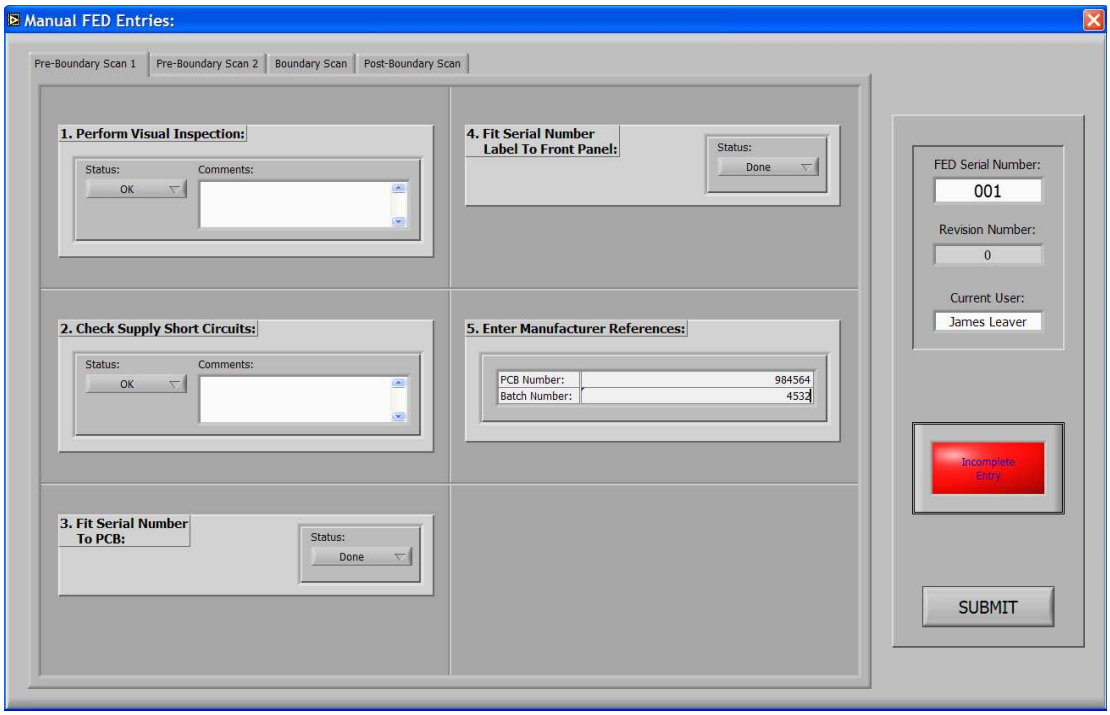
5.5 Manual Acceptance Tests

The section of the Acceptance Test that is to be carried out manually consists primarily of simple ‘sanity checks’, to ensure that the FED is sufficiently operational to take part in the automated Digital and Analog test sequences, combined with a Boundary Scan test (see Section 5.5.2) to verify digital connectivity. It is also enables basic information about a FED, such as component serial numbers, to be recorded for

future traceability. The essential steps of the manual Acceptance procedure are described in the following sections.

5.5.1 Pre Boundary Scan

(a)



(b)

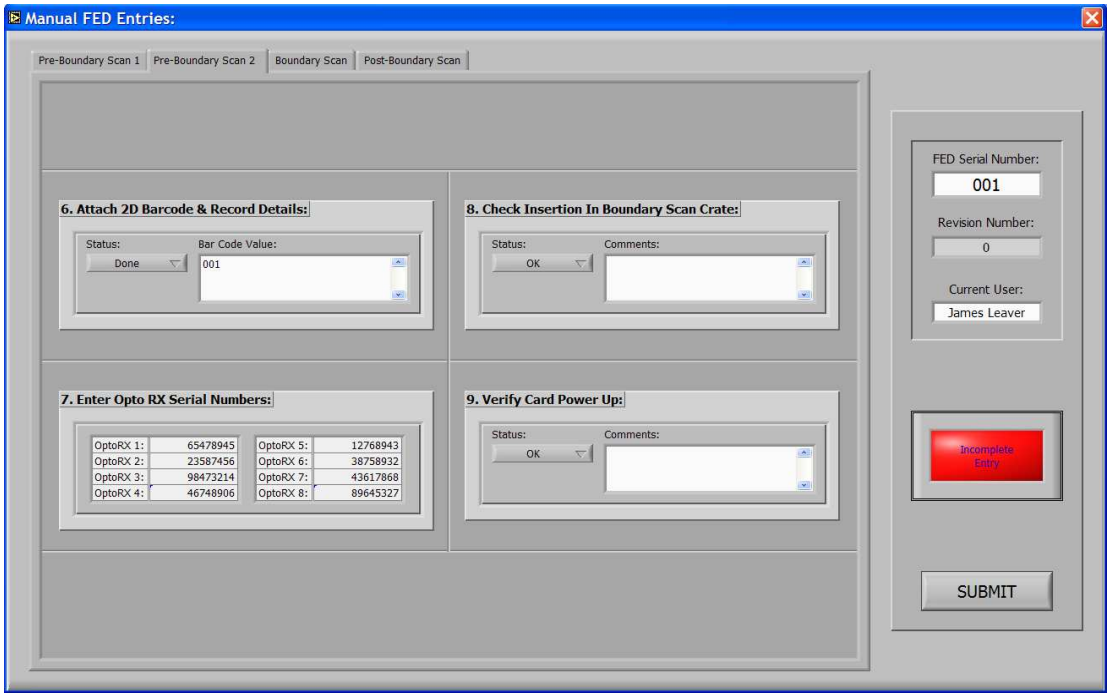


Figure 5.2: Pre-Boundary Scan manual Acceptance Test entries.

Screenshots of the main panel used for manually entering FED parameters and test results are shown in Figure 5.2. A tabbed interface permits access to a number of subsections; Figures 5.2(a) and (b) demonstrate the steps that must be completed before running the Boundary Scan test.

The assembly plant operative must first perform a visual inspection of the board for any obvious errors that may have been missed by the AOI, and input a fault description if necessary. They must then check for short circuits on the board power supply using a Digital Multimeter, and again describe any faults. To identify the FED under investigation, the user must apply a serial number label to both the PCB and the front panel and record the manufacturers' batch and PCB reference numbers. A standard CMS 2D barcode is then attached to the front panel, the barcode value is logged, and the serial numbers of the 8 Optical Receivers on the board are entered. Finally, the operative must confirm that the board can be inserted into the VME crate that will be used for the Boundary Scan test, checking for any warping of the PCB or misplacement of the front panel or VME connectors, and then verify that the FED powers up correctly by reading a number of on-board status LEDs.

5.5.2 Boundary Scan Testing

JTAG Boundary Scan [76] is used to verify the digital connectivity of most of the active components on the FED. Boundary Scan is a technology which enables the signals at the boundary pins of a JTAG compatible device to be controlled and observed via software. This is achieved through a series of boundary cells which are built into the device between the internal logic of the chip and every input or output pin. During normal operation, the boundary cells remain inactive and have no effect; when the device is switched into test mode, the cells capture all input signals and preset the levels on all output pins.

A JTAG compatible IC contains a Test Access Port (TAP) controller, which enables boundary cell readout via a Test Data Out pin (TDO) and the loading of output levels via a Test Data In (TDI) pin. If multiple devices are connected together in a circuit, the TDI pin of each may be connected to the TDO pin of the next to form a single Boundary Scan Chain. The devices may then be tested as a single unit. Short circuits between pins or bad solder joints can be found by sending known test vectors from every output pin of each component and reading them back wherever a connection is made to the input pin of another. If the signals do not match then a problem is identified, as demonstrated in Figure 5.3.

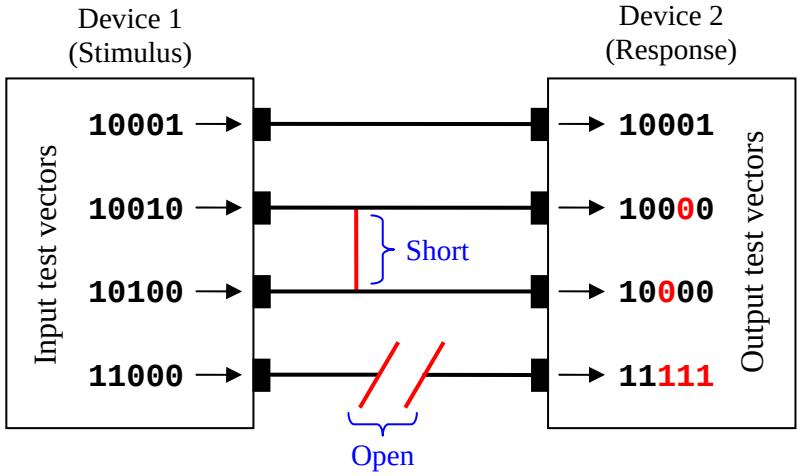


Figure 5.3: A demonstration of Boundary Scan connection testing via the transmission of known test vectors (Adapted from Reference [77])

The hardware setup required for the FED Boundary Scan test is relatively simple. A PC is connected to the main Test Access Point on the FED (which is supplied with power from a VME crate) via a JTAG USB interface box. JTAG Technologies Vip Manager is then used to access the Boundary Scan compatible devices. Initially, an infrastructure test is run, which checks that each device is properly installed by reading out the instruction register of its internal TAP controller. The last two bits of this register should always be “01” if the device is correctly powered and operational. Secondly, a board level interconnect test is performed; the

Vip Manager software automatically generates the vectors required to check for shorts and open circuits between Boundary Scan devices and analyses the values read back, providing a full report of faulty pins and net connections.

Results from both procedures are entered into the Acceptance Test software via the panel shown in Figure 5.4. Report files output from the Boundary Scan software are also specified; indexed copies are made and added to the Acceptance Test results database.

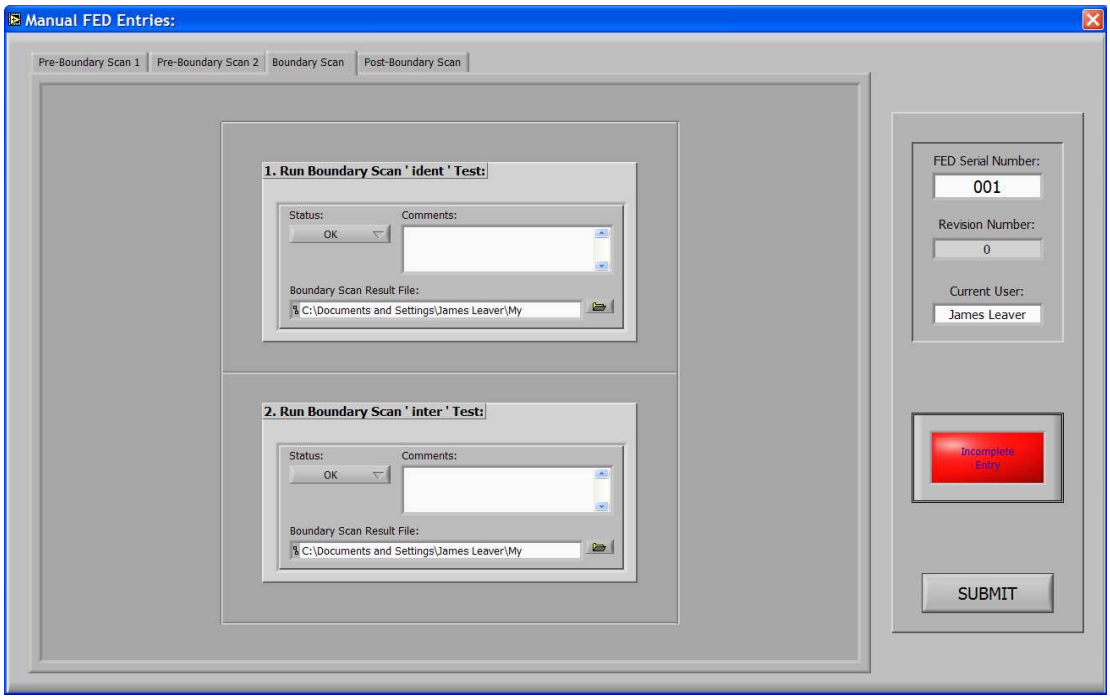


Figure 5.4: Boundary Scan manual Acceptance Test entries.

5.5.3 Post Boundary Scan

Once the Boundary Scan has completed successfully, it is necessary to prepare the FED for the automated Digital and Analog tests; the remaining steps are shown in Figure 5.5.

The plant operative must plug a Compact Flash card (programmed with an appropriate ACE file) into the FED, insert the board into a standard VME crate and then verify that it again powers up correctly by reading the FED status LEDs. In order for the BE and FE FPGAs to load successfully via System ACE (and to enable access

to the FED through VME) it is first necessary to program the VME FPGA, as this routes all of the clock signals to the rest of the board. This is done manually through the VME Boundary Scan interface; the VME FPGA is not part of the System ACE chain and instead loads its firmware from the VME EPROM at boot up. The user must verify that the VME EPROM programming step completes successfully, entering a description of any errors that occur.

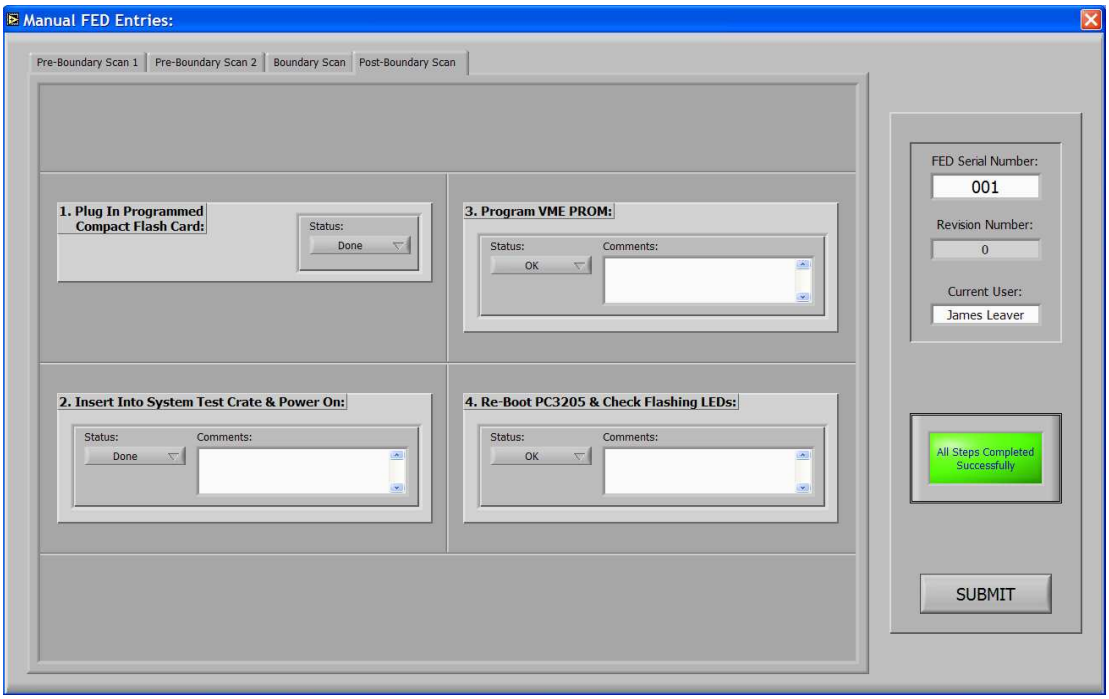


Figure 5.5: Post-Boundary Scan manual Acceptance Test entries

After rebooting the FED, causing the VME FPGA to read the EPROM, the operative must check the flashing LED start up sequence of the board while the BE and FE FPGAs are loaded from the Compact Flash card. Basic faults in the System ACE chain will be indicated by error LEDs, and must be recorded. If all of the manual procedures have been completed successfully, and all results have been entered, the Acceptance Test software will indicate to the user that it is possible to begin the automated VME-based test procedures.

5.6 Automated VME Crate Testing

An automated full system test of the FED is performed once the board has been placed in a standard VME crate. A series of C++ routines drive and acquire data from the FED, with Fed9USoftware [78] libraries providing hardware access. The C++ routines are wrapped with a LabVIEW-based graphical interface, similar to that of the manual Acceptance Test, designed to make the software easy to use. VME crate testing consists of two phases, which check both the digital and analog functionality of the board.

5.6.1 Software Architecture

To facilitate the modification and addition of routines, the automated test software has been developed using a simple modular approach. Each aspect of the FED functionality is checked with a single standalone C++ executable, which performs any required register access and dumps the data that are read (or an error statement if access fails) to the command line; output to a file is also possible, for instances when a large volume of data is produced. LabVIEW has an inbuilt function which can launch an executable and capture any command line values. Utilising this feature, LabVIEW programs (commonly referred to as ‘Virtual Instruments’ or VIs) have been written to encapsulate each test executable and analyse its output, generating a set of raw unformatted results and report messages.

These VIs represent the building blocks of an automated test. A complete Acceptance Test procedure is formed by executing the blocks in sequence inside a higher level VI. This collects the individual sets of results and formats them for output to screen; for clarity, each set is typically presented in a separate tabbed pane, with user controls added as required. Additional LabVIEW-only sub-VIs enable the collated results to be written to a standard report file.

Two high level VI ‘executors’ have been created, one for the digital functionality checks and one for the analog. Although much of the code is application specific, the architecture is highly flexible and could easily be used in the development of Acceptance Test software for future hardware modules.

5.6.2 Automated Digital Tests

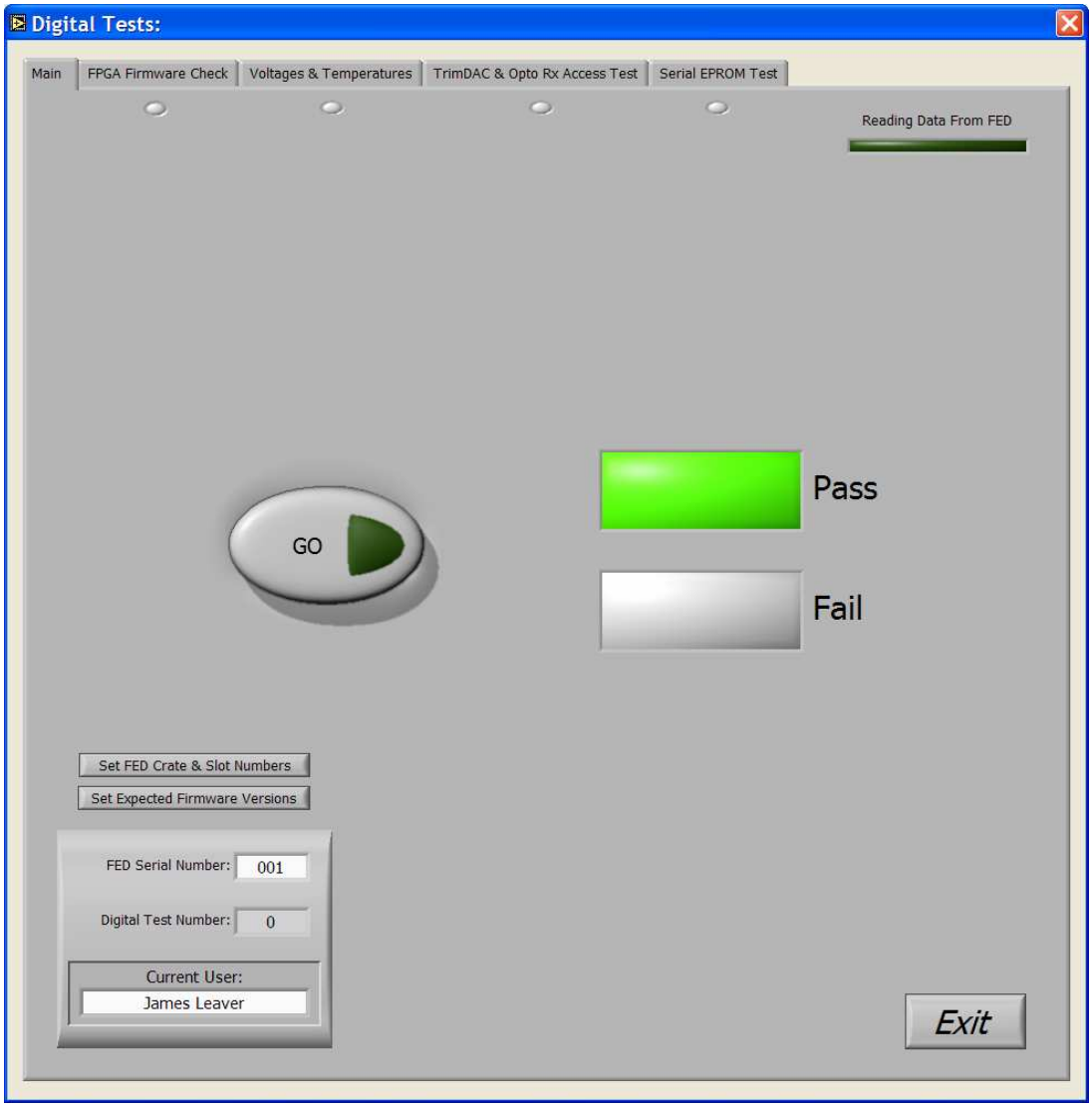


Figure 5.6: The initial Automated Digital Test panel, used to launch the test and display the overall pass/fail result. Tabs provide access to the results from each subtest.

The automated Digital subsection of the Acceptance Test verifies the basic electrical connectivity and functionality of the FED components that are not Boundary Scan compatible, and additionally checks the operation of the FPGA devices. Figure 5.6 shows the initial panel used to start the Digital Test sequence; minimal user

interaction is required, with the plant operative merely having to press the ‘GO’ button and wait for a pass or fail result. If the FED fails then the results and error report from each subtest of the sequence may be viewed in the various tabbed panes of the test window.

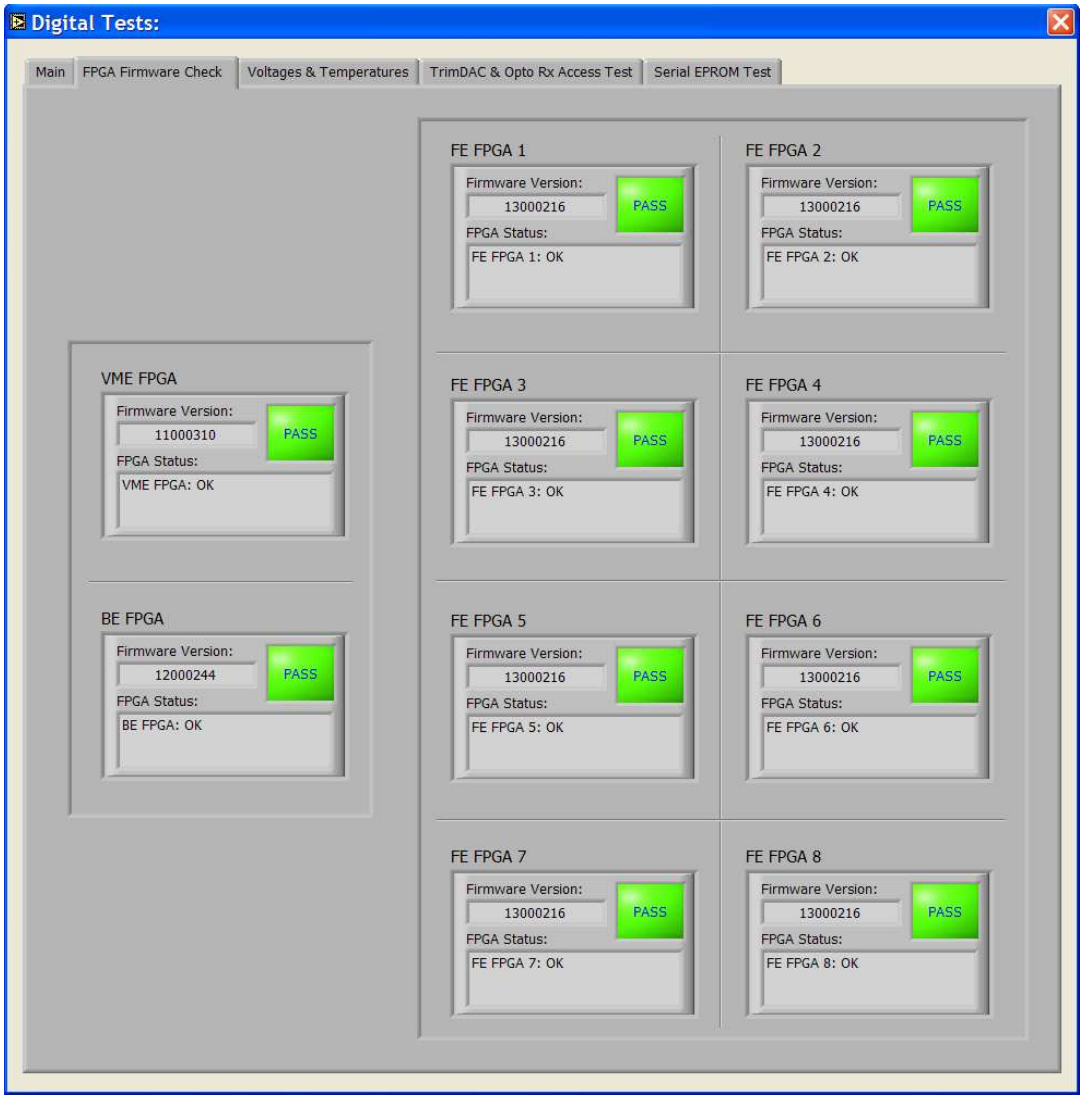


Figure 5.7: Example results from an ‘FPGA Firmware Versions Check’. This is the second panel of the Automated Digital Test window.

The first step of the Digital Test procedure involves reading the firmware versions of the Front End, Back End and VME FPGAs, and checking them against the expected values. This simple test provides a clear indication of the state of the FPGA devices. If the firmware version register value is read incorrectly then either part of the FPGA core is faulty, requiring the chip to be replaced, or the firmware has not

been loaded correctly, signifying a problem between the FPGA and the System ACE controller. If the FPGA cannot be accessed at all, but it passed Boundary Scan, then either the core of the device is again faulty or the FPGA has lost its clock signal. An example of the FPGA firmware version test results is given in Figure 5.7.

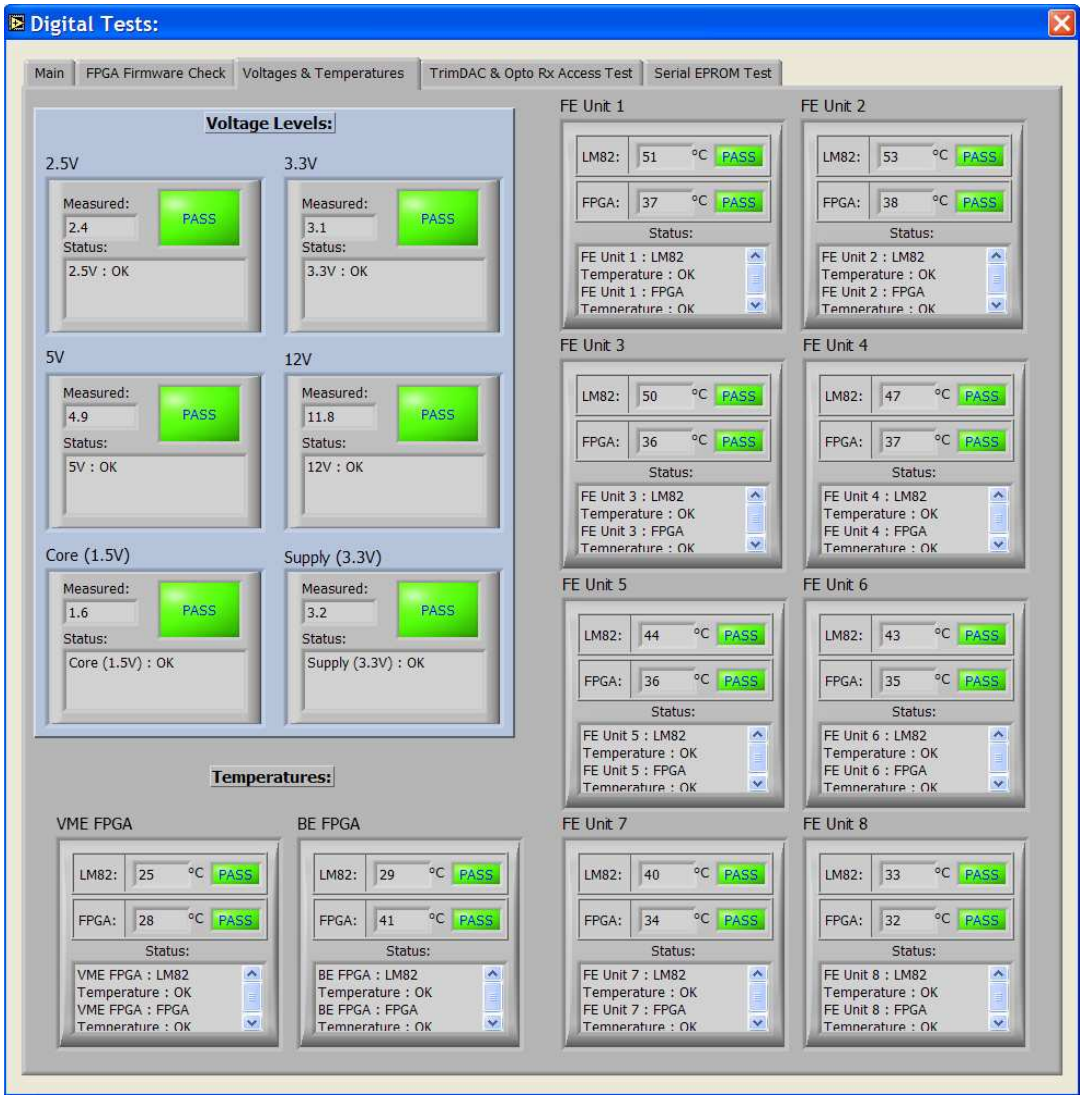


Figure 5.8: Example results from a 'Voltages & Temperatures' check. This is the third panel of the Automated Digital Test window.

The second test attempts to read the temperature of the FE, BE and VME FPGAs and the voltage levels of the 6 power supply rails. This verifies that the thermocouple and LM82 [79] temperature sensor associated with each FPGA, and the MAX708 [80] used to monitor the FED power supply circuitry, are functioning correctly. Threshold limits are additionally applied to the measured temperatures in

order to highlight any potential thermal problems with the FPGAs (defective chips may run at increased temperatures), and to the observed voltage levels so that faults on each of the power supply rails may be identified. An example of the obtained results is presented in Figure 5.8.

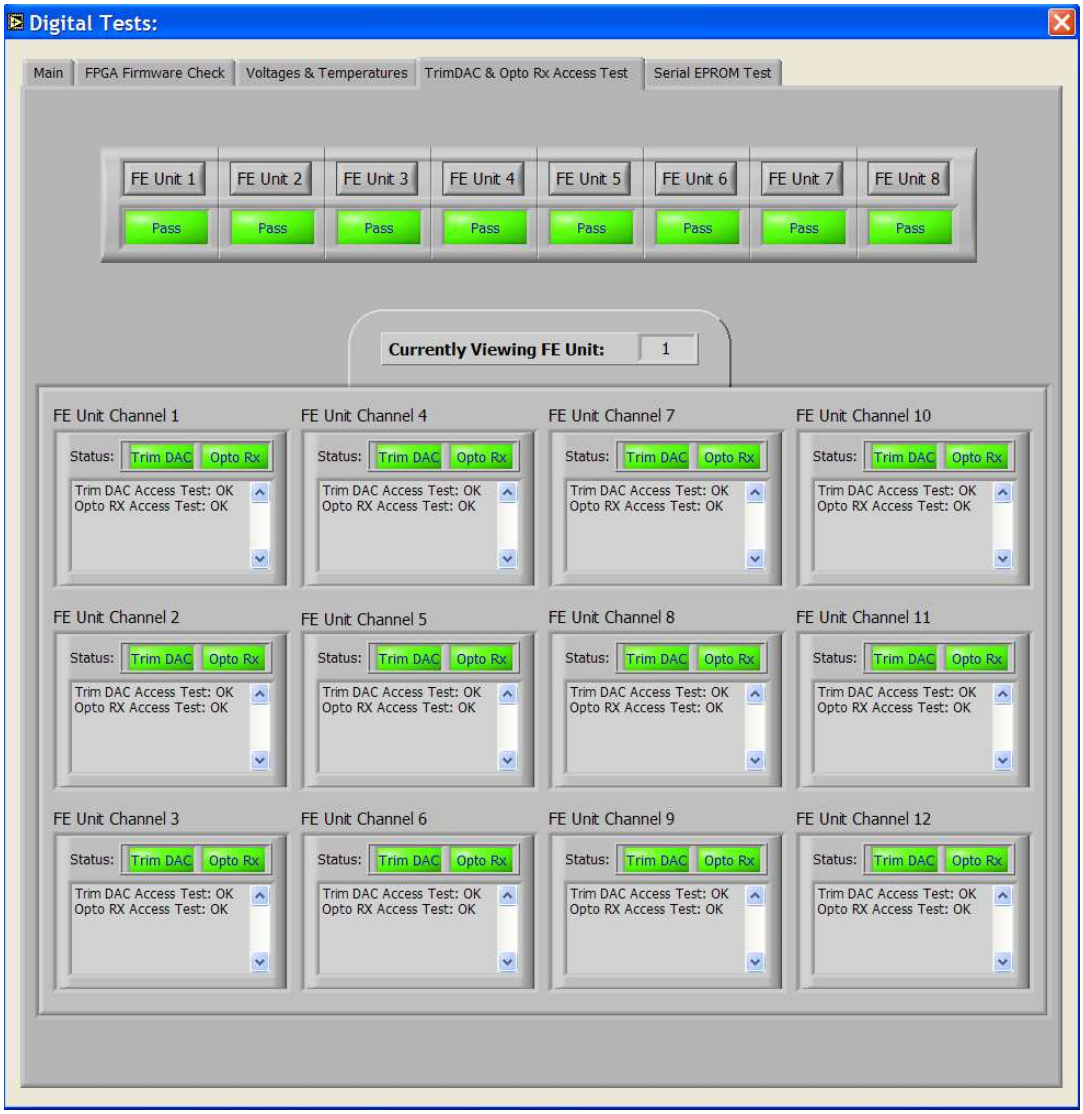


Figure 5.9: Example results from a 'Trim DAC & Opto RX Access Test'. This is the fourth panel of the Automated Digital Test window. Results for a particular FE Unit may be displayed by pressing the appropriate button at the top of the screen.

The third phase of the sequence is a simple hardware access test of all the Opto RX and Trim DAC components on the board. This verifies that the devices are powered and have at least basic functionality, and that the tracks providing access to them are properly connected. Standard Opto RX settings are written for every FED

channel, read back and then the values are compared. If read or write access fails then either the Opto RX is malfunctioning, the power pins are not soldered correctly or there is a fault along the control data path to the chip. If the device can be accessed but the wrong value is read back then either the Opto RX is again faulty or there is a problem with the read or write data lines. The Trim DACs' settings cannot be read back, so only a write access check is performed. Example test results can be seen in Figure 5.9.

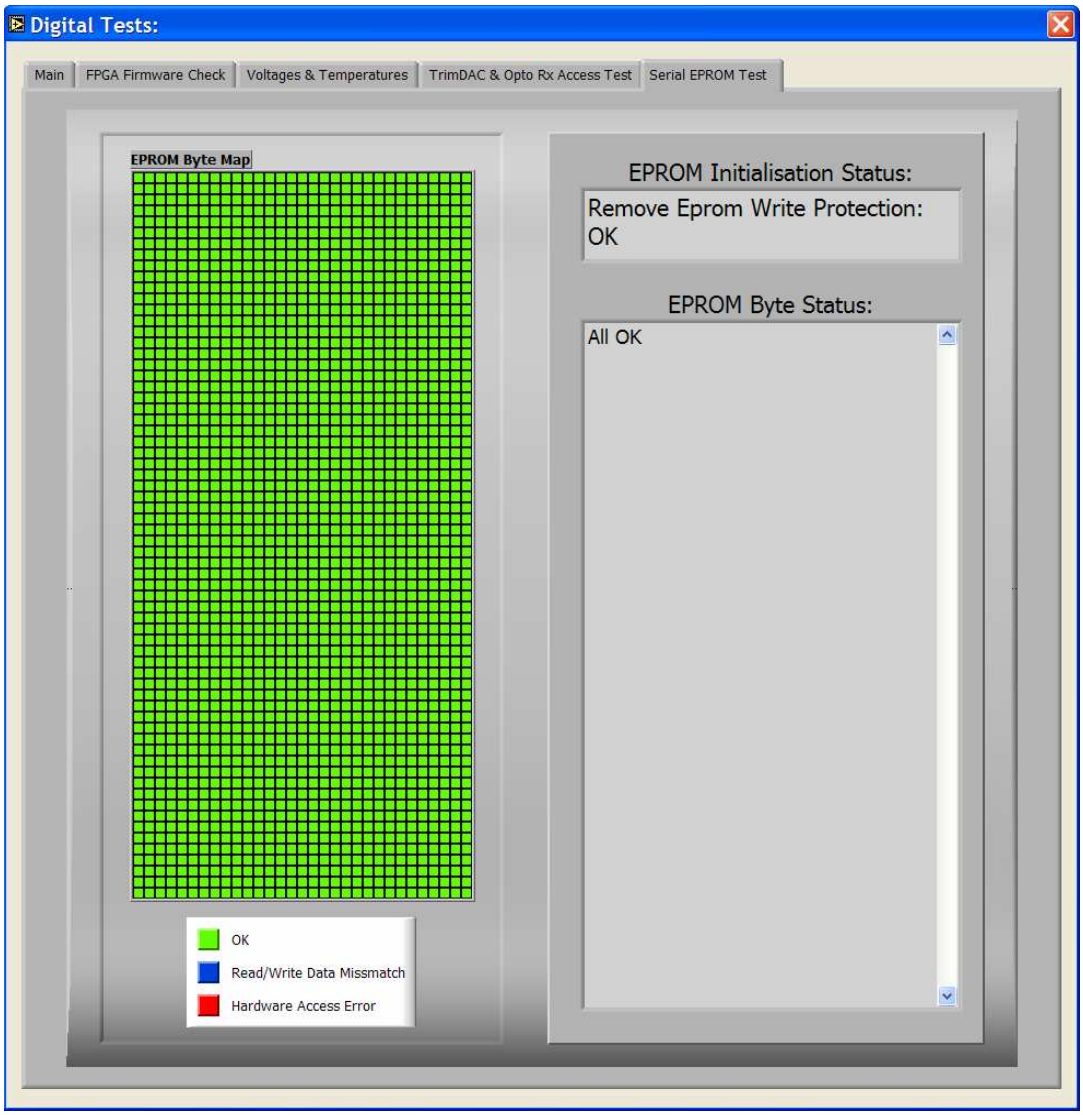


Figure 5.10: Example results from a 'Serial EPROM Test'. This is the fifth panel of the Automated Digital Test window. Each square of the EPROM map to the left of the screen represents one byte of the EPROM internal memory.

When the full Acceptance Test is complete, a small subset of the results and identifiers are written to a serial EPROM on the FED, so that they may be accessed even if the full results database is unavailable. It is therefore important to test that the EPROM is fully operational. The final section of the Digital test sequence verifies that the EPROM can be accessed and then fills the EPROM memory with an alternating '10' binary pattern and checks it against what is read back. The pattern is then inverted and the read write cycle repeated. In this way, faults in the data lines connected to the EPROM and any dysfunctional areas of the EPROM memory are identified. An example of the test results, with a visual 'map' of the status of each byte of the EPROM memory, is given in Figure 5.10.

5.6.3 Automated Analog Tests

The automated Analog Test verifies the complete analog readout chain for each FED channel, with the exception of the photodiodes in the Opto RX component at the input stage of every FE Unit. These are important elements of the FED design, which perform the optical to electrical conversion of all APV25 signals received from CMS Tracker (the importance of the optical readout links is discussed in Chapter 1). It would be desirable to check Opto RX functionality during the Acceptance Test but it is impractical to supply FEDs with optical inputs while they are stationed at the assembly plant.

Although a device that enables the FED to be driven with controlled optical signals has been developed (see Chapter 6), requiring relatively inexperienced plant operatives to handle the optical fibres that carry these signals greatly increases the potential for human error. Optical fibres are easily broken, and their ends must be thoroughly cleaned every time that they are connected to a FED. Any damage to the fibres or improper cleaning can attenuate the light received by the FED, leading to test

failures even when the Opto RX components are functioning correctly. The increased test system complexity and manpower requirements (due to the cleaning, connection and disconnection of optical fibres) would also raise the fee charged by the assembly company. As the optical transmissions are produced by semiconductor laser diodes, additional difficulties could arise at the plant due to laser safety issues.

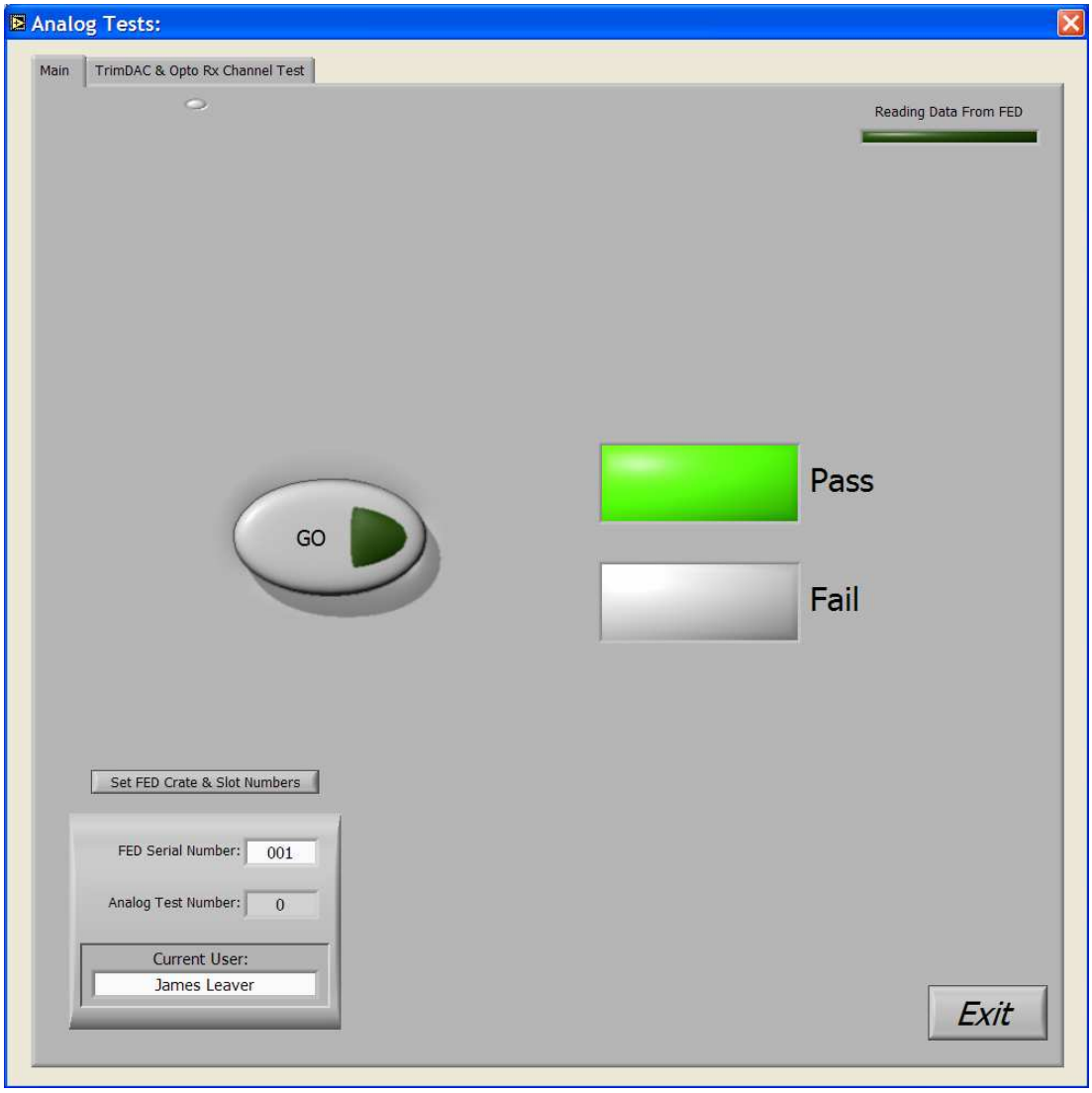


Figure 5.11: The initial Automated Analog Test panel, used to launch the test and display the overall pass/fail result. A second tab provides access to more detailed results.

Consequently, the Acceptance Test must attempt to validate as much of the FED as possible without the use of external signal generators. In the absence of an optical input, outputs from the FE Opto RX devices should be constant. It is possible, however, to simulate a varying optical signal by changing the device parameters. The

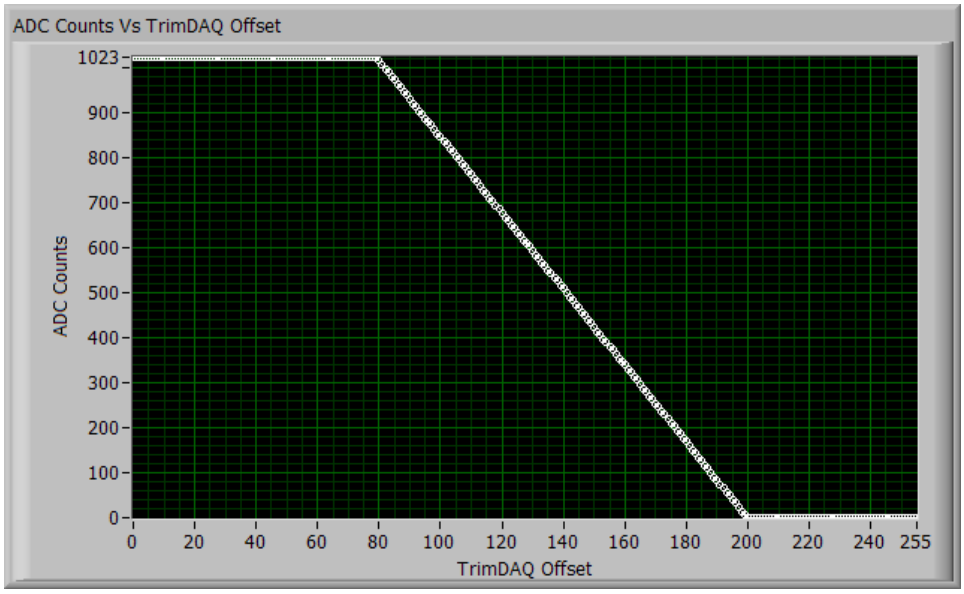
Opto RX has an adjustable input and output offset, which can be used to shift the output voltage level. Each Opto RX signal then passes through an op-amp, which enables a further offset to be introduced to the data (controlled via a Trim DAC) before it is sampled by a FE ADC. To the remainder of the analog readout chain, a signal that is artificially shifted by an Opto RX and Trim DAC is essentially the same as one that is produced by a changing optical input. This technique does not permit 'fast' signal variations (the devices must be loaded with new offset settings before each data sample is taken) and it cannot exercise the photodiodes in the Opto RX chips, but it enables the readout chain to be tested for most significant hardware faults.

The initial panel used to start the Analog tests is shown in Figure 5.11. It is the same as the Digital test interface, as the user again presses the 'GO' button and waits for a pass or fail verdict, with results shown in detail in a separate tabbed pane. There are three phases to the test, which consist of scanning through all of the Trim DAC offsets, the Opto RX input offsets and finally the Opto RX output offsets whilst recording the average channel values read out from the FED.

Example plots showing the ADC values measured by the FED as functions of the Trim DAC and Opto RX offsets are given in Figure 5.12. In each case, the signal level should vary linearly with the offset value (with possible saturation at the offset limits, if the range of the ADC is exceeded). The Acceptance Test software analyses the linear region of these plots and checks the length, gradient, monotonicity and noise levels of the data. This enables much to be deduced about the analog readout path; principally, lifted pins or shorts can be identified from signals that are stuck high or low, or that have discontinuities within the linear region. Faulty Opto RX, op-amp, Trim DAC and ADC components (or incorrect termination or voltage reference

resistors) can be identified by gradients or noise levels that are outside acceptable limits.

(a)



(b)

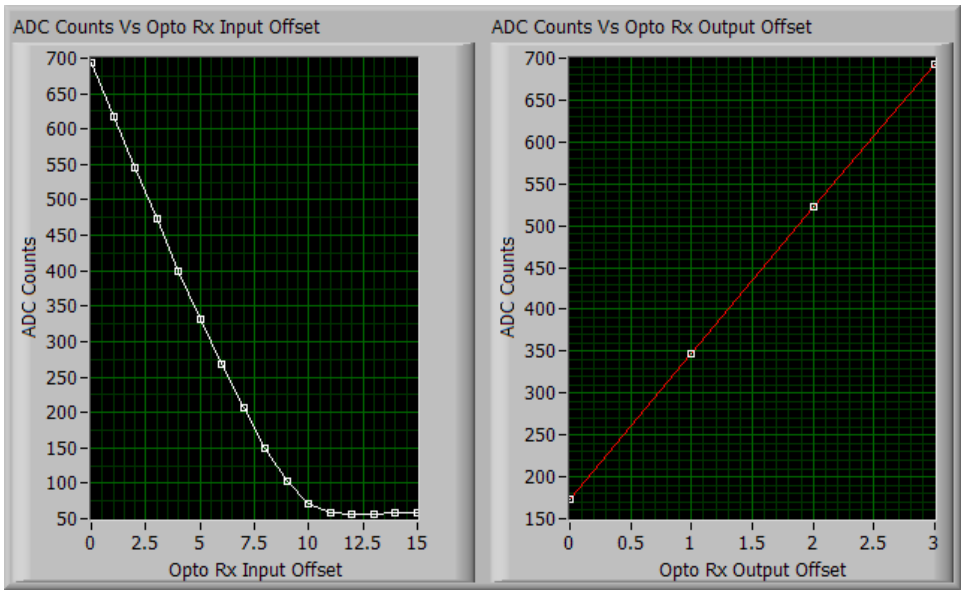


Figure 5.12: Plots showing the average ADC value measured by the FED as a function of (a) Trim DAC offset and (b) Opto RX input and output offsets.

It is not always possible to locate faults with great precision (for example, the results from an Opto RX scan may show errors if the output from a working Opto RX is read through a faulty op-amp), but the Acceptance Test software can always isolate particular types of fault on specific FED channels. This accuracy is quite sufficient; as

the analog readout path for a single channel represents a very small area of the board, it is relatively straightforward to manually check every connection along the route if the error diagnosis does not specify an individual component.

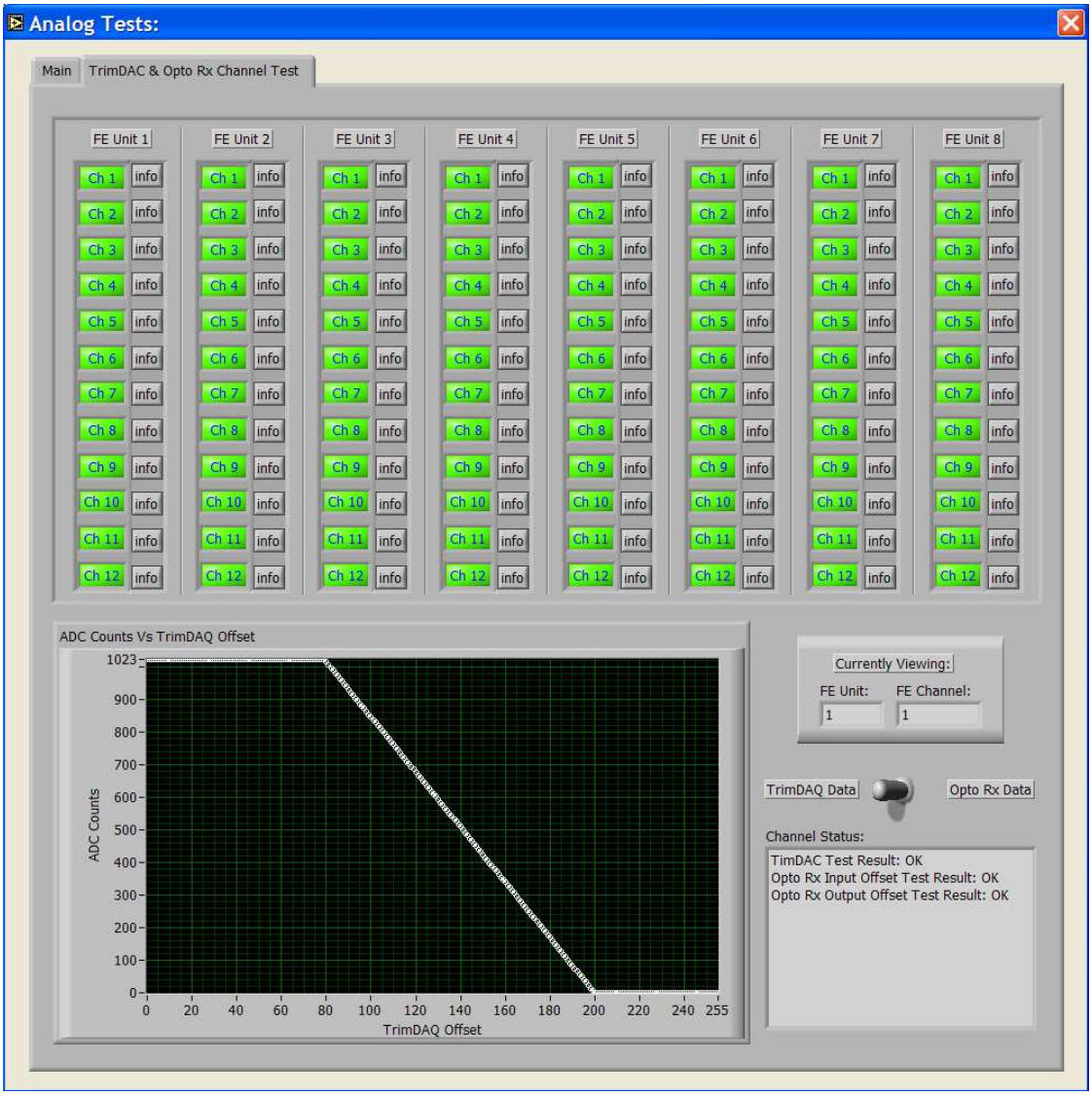


Figure 5.13: Example Automated Analog Test results. The graph in the lower left hand corner of the screen can be switched between Trim DAC and Opto RX offset results using the toggle switch to its right.

The output format of the test results is shown in Figure 5.13. A set of coloured lights, one for each channel of every FE Unit, clearly indicate which channels have passed (green) and failed (red). Pressing the 'info' button next to a failed channel causes the relevant offset plots and fault diagnosis to be displayed. If errors are found, the plant operative must remove the FED from the VME crate and repair the affected

channels. The Analog test is then used again to verify the repairs, and the process repeated as necessary until all channels function correctly.

5.7 Acceptance Test Results Storage

All of the results from the Acceptance Test are stored in XML [55] format. XML stands for EXtensible Markup Language. A markup language includes codes that are used to describe the structure and nature of information. When text written in a markup language is read by a computer, these codes enable values to be automatically associated with particular data constructs or object properties; the markup language therefore provides a standardised method for organising data. Extensible refers to the fact that XML permits new type codes to be defined by the user. This enables any type of information to be written in the standardised format of the markup language.

The layout of an XML document is quite simple. It consists of data values wrapped with descriptive identifier tags, typically grouped into particular data classes. Figure 5.14(a) shows the general format of an XML 'element'; an example is given by the output from one of the FPGA firmware version tests from the Digital section of the Acceptance procedure, shown in Figure 5.14(b).

(a)

```
<Class ID="instance">
    <property1>data value</property1>
    <property2>data value</property2>
    <property3>data value</property3>
</Class>
```

(b)

```
<Firmware_Data FPGA="VME FPGA">
    <Test_Result>Pass</Test_Result>
    <Firmware_Version_Found>11000310</Firmware_Version_Found>
    <Firmware_Test_Status>VME FPGA: OK</Firmware_Test_Status>
</Firmware_Data>
```

Figure 5.14: XML elements. (a) shows the general format of an XML element, while (b) gives an example taken from a set of Automated Digital Test results.

In Figure 5.14(b), the class of the data is the result of the firmware version test for a single FPGA, identified by the tag code 'Firmware_Data'. The identifier for the

class is the FPGA type, with this instance corresponding to the “VME FPGA”. Within the bounds of the class tags, a series of sub-elements are defined for the test results (class properties), where each result is tagged by its own descriptive code word. XML elements can be grouped into higher level classes and nested to any degree, but the tag codes mean that it is trivial to find and access a particular element regardless of the length or complexity of the data structure. As the tag codes can be recognised by a computer, it is also a simple task to programmatically extract and sort elements, a feature that is used by the Acceptance Test software to manage the sets of results from each subsection and build a complete results file for each FED.

Ease of data organisation is not the only benefit of XML. The Acceptance Test results will need to be accessible throughout the lifetimes of the FED boards. Using an industry standard format such as XML, where data are stored in plain text files, prevents any problems that may arise in the foreseeable future through having to access or convert data which are written in potentially obsolete proprietary formats. XML was originally chosen for the Acceptance Test purely on its own unique merits, but it is in fact the default format for data storage at CMS; consequently, all of the Acceptance Test results are automatically compatible with the CMS databases. The only real disadvantage of XML is that overheads are caused by having to store all of the ‘tag’ words in addition to the actual data content. However, this is not an issue for the Acceptance Test as the report files generated for each FED have a size of only a few hundred kB. Once all of the FEDs have been tested, it should be possible to fit the entire results database on a single writeable CD.

The XML technique of separating form (class structures and data types) from content also makes it a simple task to convert XML data into other formats. It is possible to write a ‘style sheet’ in XSL (the Extensible Stylesheet Language) [55]

which associates particular formatting styles with any of the XML element tags. Typically, the style sheet contains rules for converting the XML data structure into an HTML equivalent; HTML stands for Hypertext Markup Language and is similar to XML, but its markup code words are concerned primarily with the visual display properties of data elements (it is designed for the formatting of information into graphical internet Web pages). This technique is used by the Acceptance Test software to produce a copy of the XML results in a graphical, tabulated, indexed human readable HTML form, enabling the Acceptance Test results for all FEDs to be easily accessed through a standard Web browser. Examples of the HTML output are given in Figures 5.15 to 5.18.

Figure 5.15: The main Acceptance Test results index, in tabulated HTML form. Each FED is represented by one row of the table. The first column indicates the serial number of the FED, while the remaining columns provide links to results from each of the three major subsections of the Acceptance procedure: Manual Tests, Automated Digital Tests and Automated Analog Tests. Links are colour coded to enable rapid identification of FEDs that require attention (blue = pass, pink = fail/incomplete, beige = not attempted).

Manual FED Entries / Test Results

Revision 2

Operator Name:

Ivan

Time Stamp:

2005-07-22, 11:28:18

Pre Boundary Scan:

Action	Status	Comments
Visual Inspection	OK	
Power Supply Short Circuit Check	OK	
Fit Serial No. To PCB	Done	
Fit Front Panel Serial No. Label	Done	
Enter Manufacturer Reference Numbers	Done	
Attach 2D Barcode	Done	
Enter Opto RX Serial Numbers	Unattempted	
Check Insertion In Boundary Scan Crate	OK	
Check Card Power Up	OK	

Manufacturers Reference No.	Value
PCB_No	005306
Batch_No	005306

2D Barcode Value
30201300000044

Opto RX	Serial Number
1	
2	
3	
4	
5	
6	
7	
8	

Boundary Scan:

Action	Status	Comments	Boundary Scan File
Boundary Scan 'Ident' Test	OK		/home/fed/FedIndustryTest/LabVIEW_Executable/TestReports/044/BS_Files/BS_Ident_File_044_Revision2.txt
Boundary Scan 'Inter' Test	OK		/home/fed/FedIndustryTest/LabVIEW_Executable/TestReports/044/BS_Files/BS_Inter_File_044_Revision2.txt

Post Boundary Scan:

Action	Status	Comments
Plug In Compact Flash Card	Done	
Power On In System Test Crate	Done	
Program VME PROM	OK	
Reboot PC3205, Check Flashing LEDs	OK	

Revision 1

Operator Name:

Ivan

Figure 5.16: Example HTML results from the Manual Test subsection of the FED Acceptance Test. Links are provided to local copies of the Boundary Scan result files that are specified during the testing procedure. As with the index in Figure 5.15, any failures are highlighted in pink and incomplete entries are highlighted in beige. A new set of results is recorded each time that the test is run; these are presented consecutively, in reverse chronological order (the same is true for the Automated Digital and Automated Analog Tests).

FED Digital Test Results

Test 1

Operator Name:

Ivan

Time Stamp:

2005-07-22, 11:35:28

Firmware Version Test Results:

FPGA Name	Test Result	Firmware Version Found	Comment
VME FPGA	Fail	2100031C	Accessed FPGA, but incorrect data read
BE FPGA	Pass	2200036D	OK
FE FPGA 1	Pass	23000319	OK
FE FPGA 2	Pass	23000319	OK
FE FPGA 3	Pass	23000319	OK
FE FPGA 4	Pass	23000319	OK
FE FPGA 5	Pass	23000319	OK
FE FPGA 6	Pass	23000319	OK
FE FPGA 7	Pass	23000319	OK
FE FPGA 8	Pass	23000319	OK

Voltage Test Results:

Voltage Level	Test Result	Measured Voltage	Comment
2.5V	Pass	2.486980V	OK
3.3V	Pass	3.162500V	OK
5V	Pass	4.869790V	OK
12V	Pass	11.750000V	OK
Core (1.5V)	Pass	1.476560V	OK
Supply (3.3V)	Pass	3.145310V	OK

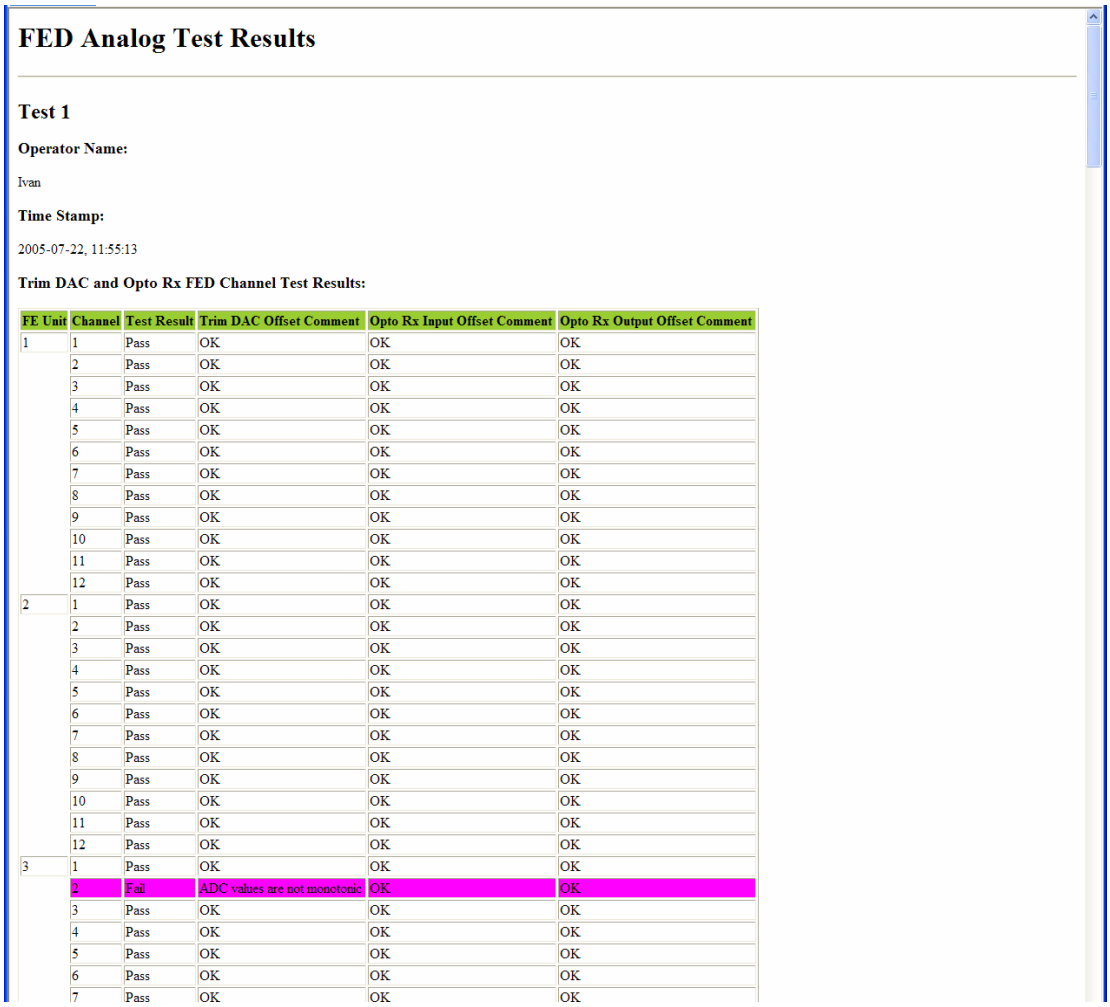
Temperature Test Results:

FE Unit	Chip	Test Result	Measured Temperature	Comment
FE Unit 1	LM82	Pass	33degC	OK
	FPGA	Pass	32degC	OK
FE Unit 2	LM82	Pass	37degC	OK
	FPGA	Pass	33degC	OK
FE Unit 3	LM82	Pass	40degC	OK
	FPGA	Pass	35degC	OK
FE Unit 4	LM82	Pass	42degC	OK
	FPGA	Pass	36degC	OK
FE Unit 5	LM82	Pass	44degC	OK
	FPGA	Pass	37degC	OK
FE Unit 6	LM82	Pass	46degC	OK
	FPGA	Pass	38degC	OK
FE Unit 7	LM82	Pass	48degC	OK
	FPGA	Pass	38degC	OK
FE Unit 8	LM82	Pass	46degC	OK
	FPGA	Pass	37degC	OK
BE FPGA	LM82	Pass	30degC	OK
	FPGA	Pass	41degC	OK
VME FPGA	LM82	Pass	26degC	OK
	FPGA	Pass	25degC	OK

Trim DAC and Opto Rx HW Access Test Results:

FE Unit	Channel	Test Result	Trim DAC Comment	Opto Rx Comment
1	1	Pass	OK	OK
	2	Pass	OK	OK
	3	Pass	OK	OK
	4	Pass	OK	OK
	5	Pass	OK	OK

Figure 5.17: Example HTML results from the Automated Digital Test subsection of the FED Acceptance Test. Here the VME FPGA has failed the firmware version test. The fact that the FPGA was accessed correctly, combined with the format of the read value, indicates that the device is functional but the wrong firmware has been loaded.



FED Analog Test Results

Test 1

Operator Name:

Ivan

Time Stamp:

2005-07-22, 11:55:13

Trim DAC and Opto Rx FED Channel Test Results:

FE Unit	Channel	Test Result	Trim DAC Offset Comment	Opto Rx Input Offset Comment	Opto Rx Output Offset Comment
1	1	Pass	OK	OK	OK
	2	Pass	OK	OK	OK
	3	Pass	OK	OK	OK
	4	Pass	OK	OK	OK
	5	Pass	OK	OK	OK
	6	Pass	OK	OK	OK
	7	Pass	OK	OK	OK
	8	Pass	OK	OK	OK
	9	Pass	OK	OK	OK
	10	Pass	OK	OK	OK
	11	Pass	OK	OK	OK
	12	Pass	OK	OK	OK
2	1	Pass	OK	OK	OK
	2	Pass	OK	OK	OK
	3	Pass	OK	OK	OK
	4	Pass	OK	OK	OK
	5	Pass	OK	OK	OK
	6	Pass	OK	OK	OK
	7	Pass	OK	OK	OK
	8	Pass	OK	OK	OK
	9	Pass	OK	OK	OK
	10	Pass	OK	OK	OK
	11	Pass	OK	OK	OK
	12	Pass	OK	OK	OK
3	1	Pass	OK	OK	OK
	2	Fail	ADC values are not monotonic	OK	OK
	3	Pass	OK	OK	OK
	4	Pass	OK	OK	OK
	5	Pass	OK	OK	OK
	6	Pass	OK	OK	OK
	7	Pass	OK	OK	OK

Figure 5.18: Example HTML results from the Automated Analog Test subsection of the FED Acceptance Test. Here, channel 2 of FE Unit 3 has failed due to a lack of monotonicity in the plot obtained during the Trim DAC offset scan (refer to Figure 5.12(a)).

5.8 Acceptance Test Usage at the Assembly Plant

A picture of the Acceptance Test station is shown in Figure 5.19. It was installed at the assembly plant in August 2005 and, at the time of writing, has been in constant use for over two months. The assembly plant Test Section staff were initially provided with a training session, in which a RAL engineer (I. Church) worked through the test procedure in detail with a small number of FEDs. Since then, the plant operatives have required assistance in repairing 7 of the boards (from the first batch of 50) that were identified as faulty.

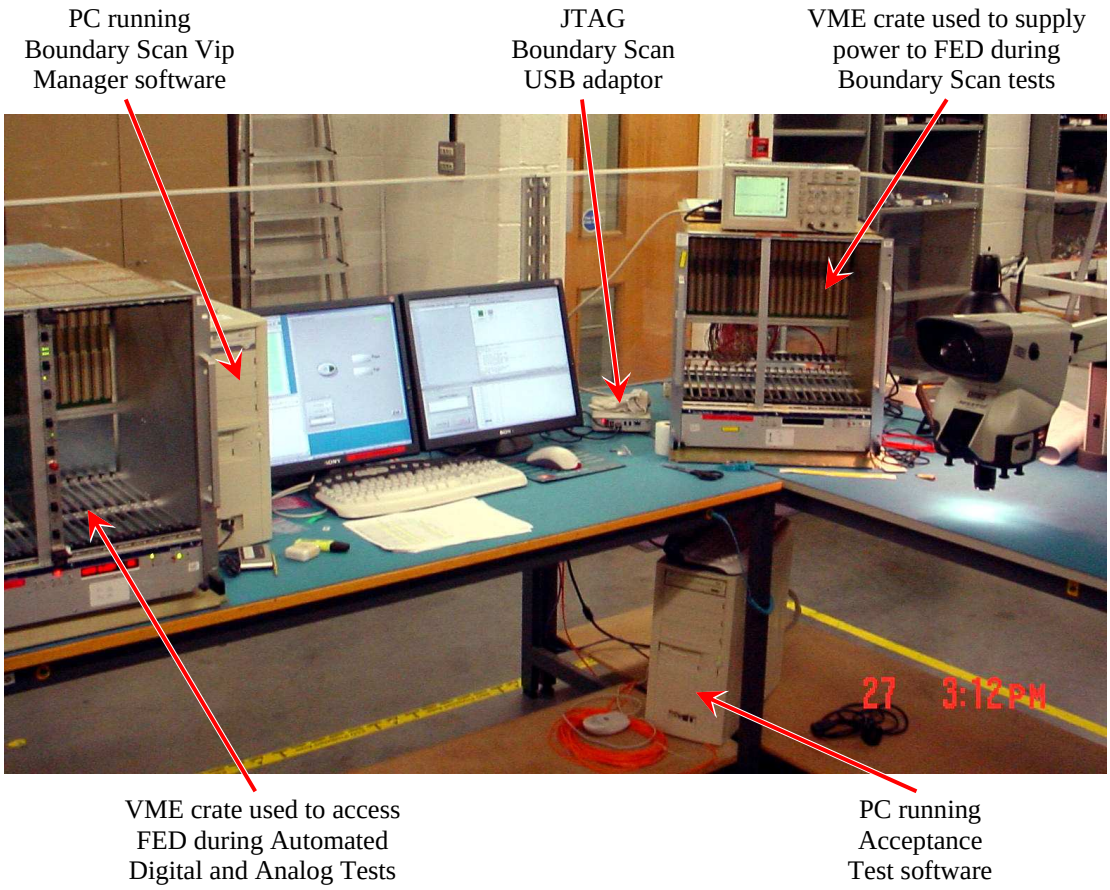


Figure 5.19: A picture of the Acceptance Test station installed at the FED assembly plant.

In terms of the operation of the Acceptance Test, no support (other than the initial training) has been necessary. Over 100 boards have been tested without encountering a single software problem. The Acceptance Test has been carefully designed to ensure bug-free operation; barring a hardware fault in the computer running the test, it is essentially impossible for the software to crash or behave in an unexpected manner. Feedback from the plant operatives indicates that they are very impressed with the software package and find it easy to use. They also appreciate the simple pass/fail approach to the automated testing, with more detailed diagnostics available on demand.

5.9 Acceptance Test Limitations

While the Acceptance Test performed at the assembly plant can verify almost all of the FED functionality and diagnose most manufacturing errors, it has a few

limitations. To reduce the complexity of the test station (one of the most important aspects of testing at an assembly plant), the amount of auxiliary hardware has been minimised. The lack of optical signals with which to drive the FED Opto RX components has already been discussed in Section 5.6.3, but there are two other main weaknesses.

During the automated tests, all data are read out from the FED via VME, as additional S-Link readout hardware (see Chapter 7) is unavailable. Consequently, the quality of the S-Link connection cannot be assessed. When the FED is operated in the CMS Tracker, it receives trigger and control signals from the TTC system through an optical link (see Chapters 3 and 8). As with the FE Opto RXs, an optical signal cannot be applied to the TTC receiver on the FED while it is situated at the assembly plant.

To identify any of the small number of faults that can be missed, each FED received from the manufacturer undergoes rigorous post production testing in a simulated CMS environment; this is discussed in Chapter 8. Of the FEDs that have passed the Acceptance Test, the entire first batch of 50 has already been delivered to RAL for further testing. A few had very minor imperfections, such as a surface mount LED attached the wrong way, or a dry joint which caused intermittent failure. Four boards had genuine faults; one was warped (this should have been reported by the plant operative), one had non-standard TTC optical receiver behaviour, one had a defective S-Link connection and one experienced intermittent power up problems.

So far, the flaws detected by the post production test procedures are of types that the Acceptance Test is not expected to identify. Any of the numerous readout channel faults that were present in early versions of the board are now being discovered and repaired at the assembly plant. The Acceptance Test has therefore

proved to be a great success, and it will continue to be used in confidence throughout the entire FED production cycle.

~ Chapter 6 ~

The FED Tester

It is essential to evaluate the performance of the FED before it is used for real data taking at CMS in 2007. If the Tracker is to be relied upon, the ability of the FED to correctly process Tracker data must be rigorously verified. In particular, it is important to check the functionality of the FED hit finding algorithms using optical data that realistically simulates the Tracker output under realistic CMS conditions.

6.1 Testing the FED Under Realistic CMS Conditions

During normal operation at CMS, each FED will receive optical multiplexed event frame data from 96 APV pairs at the 100 kHz L1 trigger rate. It will respond to trigger and control signals from the Tracker Trigger Control System, and provide the TCS with a throttle signal which reflects the status of its internal data buffers. Once hit finding has reduced the input data volume by a factor of ~60, events will be read out from the FED via a high speed S-Link connection. Consequently, with the exception of the readout data path (described in Chapter 7) and the means to analyse the received data (discussed in Chapter 8), there are two requirements for testing the FED under realistic CMS conditions. The FED must be supplied with optical data that closely matches the output of real APVs during a CMS experiment, and it must be driven with clock, trigger and control signals that are identical to those output by the final TCS system.

A FED Tester (FT) [81] board has been designed to enable the production of these signals in the laboratory. Through the use of CMS Analog Optohybrids (AOHs) [82] it is able to precisely simulate the shape and timing of optical signals from the

Tracker, and can be loaded with any APV frame data or test patterns. The FT is therefore capable of exercising the FED under a wide range of operating conditions, and provides an effective mechanism for testing all data processing aspects of the FED hardware and firmware.

6.2 Implementation of the FED Tester

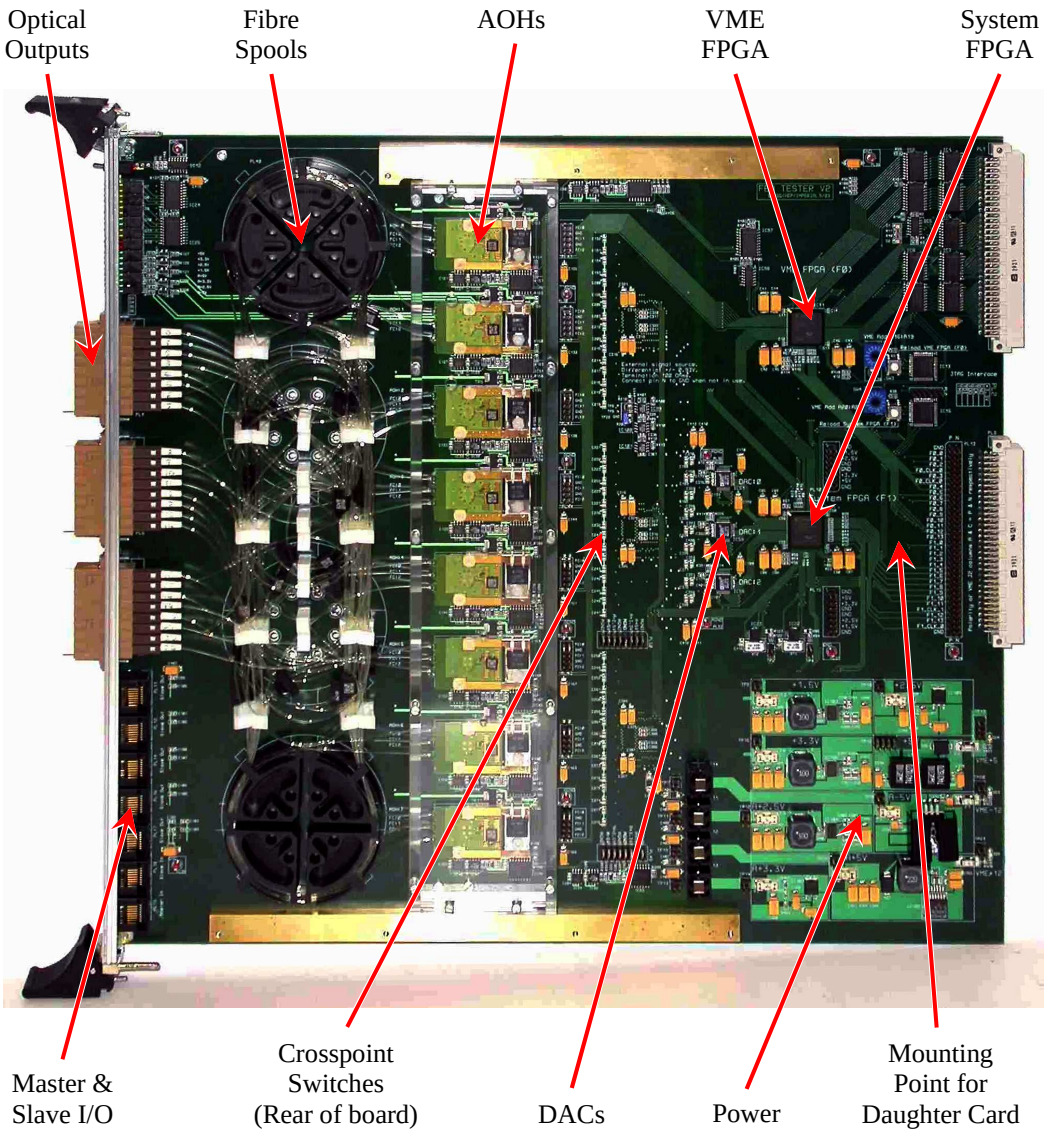


Figure 6.1: The FED Tester.

A picture of an FT is shown in Figure 6.1. It is implemented as a 9U x 400 mm VME card, acting as an A24/D16 VME slave. The FT PCB is 2 mm thick and comprises of 8 layers; 2 ground, 4 power and 2 signal. When fully populated, each FT occupies the

width of 2 VME slots. The board logic is contained primarily within two Xilinx Virtex II XC2V1000 FPGAs [52] and the overall architecture of the FT is outlined in Figure 6.2. A System FPGA is used to generate virtual APV25 data and simulate the TCS system, while a VME FPGA handles VME access and auxiliary control routines.

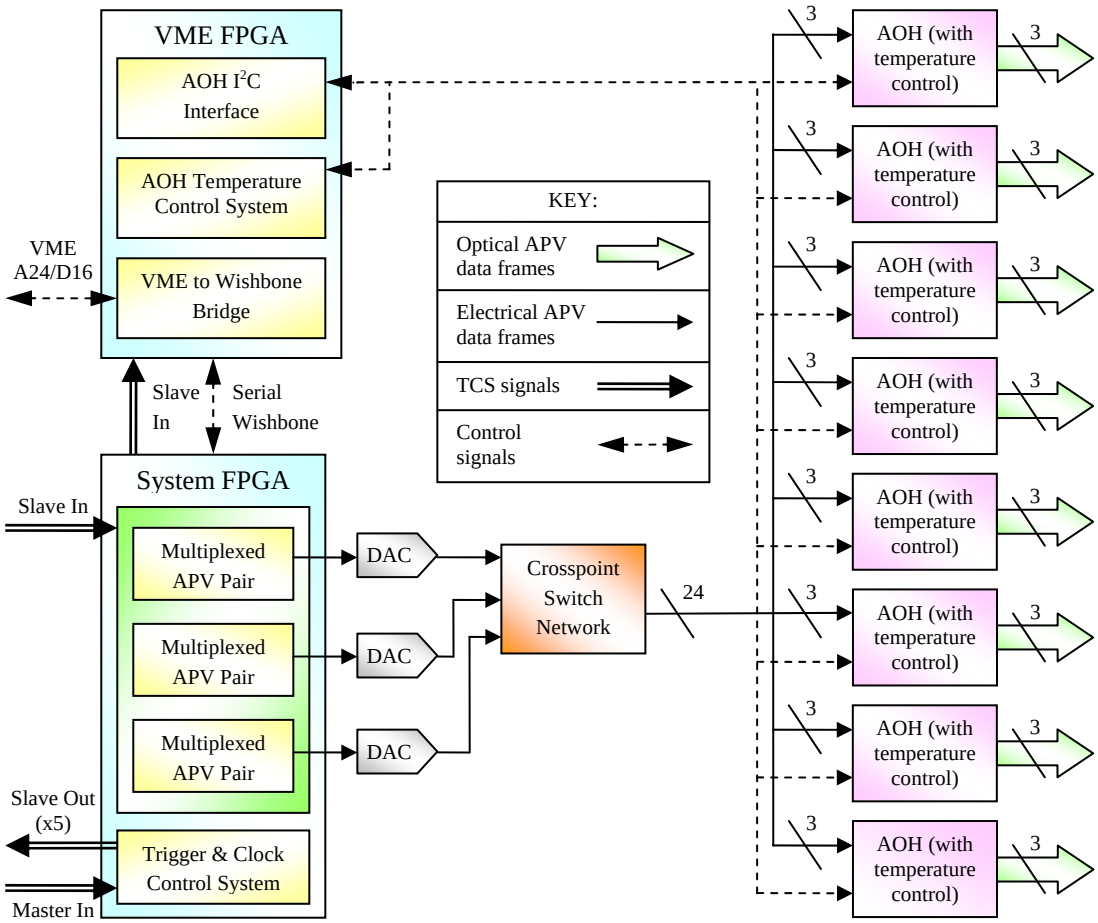


Figure 6.2: The architecture of the FED Tester.

6.2.1 The FT System FPGA

The System FPGA emulates the output from 3 multiplexed pairs of APVs. Frame data for all 6 APVs are stored in a common frame pattern memory block, 12bit wide by 232 frames deep (~350 kb in size). This frame memory is relatively small (it is limited by the size of the FPGA, which is in turn limited by cost and board complexity considerations), but it is possible to upload new frame information while the FT is in operation.

Each virtual APV has associated with it a 1 kb deep pointer memory, with every pointer value referencing one of the frames in the common memory block. Whenever an L1 trigger occurs, each virtual APV outputs the frame corresponding to its current pointer; the pointer memory indexes for all APVs are then incremented. 1024 L1 triggers can therefore be received before the sequence of output frames is repeated. In the absence of L1 triggers the APVs transmit ‘tick marks’ (for synchronisation purposes), the form of which are also programmable in memory.

Once the outputs from adjacent pairs of APVs have been multiplexed, a delay can be added to the data in order to simulate differences in fibre lengths between the Tracker and the FEDs. This delay can be finely controlled, from 0 to 32 LHC clock cycles (800 ns) in steps of ~100 ps. The digital data from pairs of APVs are converted to analog by three AD9753 DACs [83], the outputs of which are used to drive three AD8108 [84] programmable analogue crosspoint switches; an optional external input source is also routed to the crosspoint switches, to permit testing of the FED with arbitrary user-generated signals. Acting as a 4-to-24 fan-out, the switches deliver the analog multiplexed APV data (and the external source, if connected) to 8 CMS Analog Optohybrids, which convert the electrical signals into the standard Tracker optical format expected by the FED.

The TCS implemented in the System FPGA is a complete simulation of the CMS Tracker Local TCS, generating user-definable L1 trigger and control signals in the correct CMS format and responding to the FED throttle. It is essentially identical to the virtual TCS in the APVE (described in Chapter 3); TCS signals are again routed out from the rear of the board via a daughter card. A FED Tester may also be run with an external TCS source (bypassing the simulation entirely), enabling the system to be driven with real TCS hardware when it becomes available.

6.2.2 The FT VME FPGA

The VME FPGA acts as a VME to Wishbone Bridge, enabling control of the board through a standard VME interface. Wishbone [85] is a standard format for making data path interconnects between logic cores within an FPGA, and is used to enhance design reusability; for example, if a future version of the board were required with a USB or Ethernet interface, it would only be necessary to modify the Wishbone Bridge and the rest of the firmware would remain intact. The VME FPGA also provides I^2C [37] access (via VME) to the Analog Optohybrids, and runs the AOH temperature control system (described in Section 6.3).

6.2.3 The FT Ensemble

It would be desirable to have a single testing device which is able to drive all 96 channels of a FED. However, as each AOH has a maximum of three optical outputs, this would require the mounting of 32 AOHs on a single board. An AOH is relatively small (23 mm x 30 mm), but a large area is needed to coil the 1 m optical fibre pigtails that carry the optical signals from each one. This would have resulted in a PCB that exceeds the dimensions of any VME standard, leading to complications in accessing and controlling the board (normal VME access would not be possible). Consequently, it was decided to divide the AOHs across a number of VME cards. Each FT outputs 24 optical data channels and so 4 boards are required to fully populate the 96 channels of a FED. A group of FTs are collectively referred to as a FED Tester Ensemble (FTE).

To enable simultaneous operation, the FT has been designed to facilitate synchronisation between multiple hardware instances. The emulated TCS in each FT is a separate entity from the frame generation system, with no internal data connections. For a single FT, TCS clock, trigger and control signals are routed from

an output on the FT daughter card to a ‘Master In’ connection on the front panel. From here, they fan out to 5 ‘Slave Out’ connectors, one of which is routed back to a ‘Slave In’ port from which the signals are distributed to the rest of the board.

If an Ensemble of FTs are used together, one of them is defined to be the ‘Master’. The Master is configured in the same way as the isolated FT above, with the local TCS output connected to the Master In, and one of the Slave Outs connected to the Slave In. For each of the other FTs, the internal TCS simulation is ignored; instead, the Slave In ports are also connected to Slave Outs on the Master. In this way, 4 FTs can be set up to use the clock and control signals from a single TCS source, ensuring synchronisation between the output optical data. The fifth Slave Out port on the FT Master may be used to add an additional FT to the Ensemble, or it can be connected to the Master In of a second FT Master to enable TCS clock and trigger signals to be ‘daisy chained’ through multiple Ensembles.

The 24 channels of a single FT may be connected to any of the 3 distinct APV pair data streams. With a 4 FT Ensemble it is possible to drive a FED FE Unit with 3 outputs from each FT, providing every Front End channel with data from a different APV pair for maximum test flexibility. For typical configurations, the limitation is that all of the FE Units will receive the same 12 channels of data. However, as they process data independently, this does not have any significant detrimental effect on FED performance verification tests (and of course, channels may be arbitrarily repeated within a Front End if it is necessary to operate with imbalanced data rates).

6.3 Analog Optohybrid Temperature Sensitivity and Control

Each FT makes use of 8 Tracker Outer Barrel AOHs. The 3 channels of optical data output from an AOH are generated by semiconductor laser diodes (driven by a Linear Laser Driver IC [86] with adjustable gain and bias current), which transmit infrared

light at a wavelength of 1310 nm. In very simple terms, a semiconductor laser diode generates light when an applied current excites electrons across the band gap of the diode and photons are emitted when the electrons return to their ground state. Thermal energy may also excite electrons, and this affects the lasing properties of the semiconductor material. As a result, the output of an AOH is temperature dependent.

The TOB AOHs will be used at CMS itself, to transmit real APV data from the Tracker. AOH temperature sensitivity is not expected to have a significant impact in the final Tracker system, as the CMS detector will be operated in a temperature controlled environment at -10 °C. Temperature and optical intensity fluctuations will be minimal, and should be dealt with by the Common Mode subtraction algorithm of the FED (described in Chapter 9).

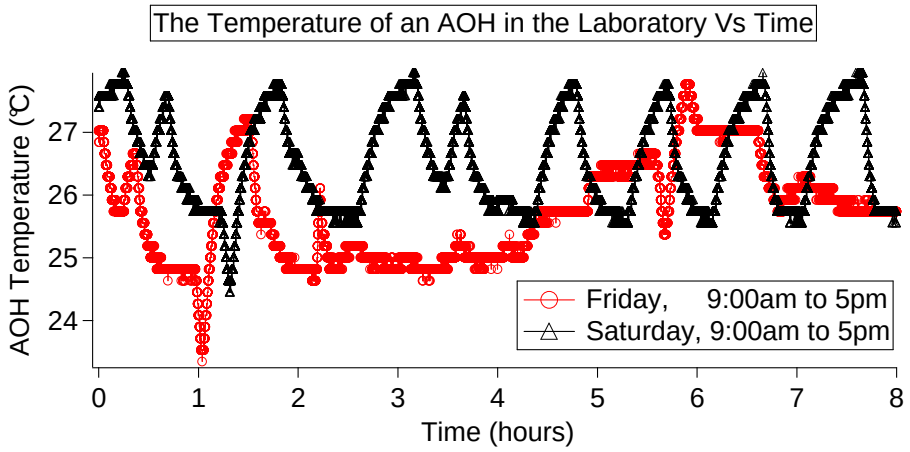


Figure 6.3: Measured temperature variation of a powered AOH over two days in the laboratory.

However, the FTs are intended for use in the laboratory, where they will operate at room temperature with a standard air conditioning unit providing the only environmental control. The temperature of a powered AOH in the laboratory has been measured to fluctuate by ~5 °C peak to peak (see Figure 6.3), and the temperature sensitivity of the laser drivers is such that considerable variation in output light intensity is observed. As an Ensemble of FTs occupies a large VME crate, it is

impractical to place the test system in an environmental chamber; instead, it became necessary to develop an AOH temperature control system that can be mounted on the FT boards themselves.

6.3.1 An On-Board AOH Temperature Control System

A prototype of the temperature control system developed for the FT is shown in Figure 6.4. It was later used to perform a preliminary characterisation of the temperature sensitivity of a TOB AOH.

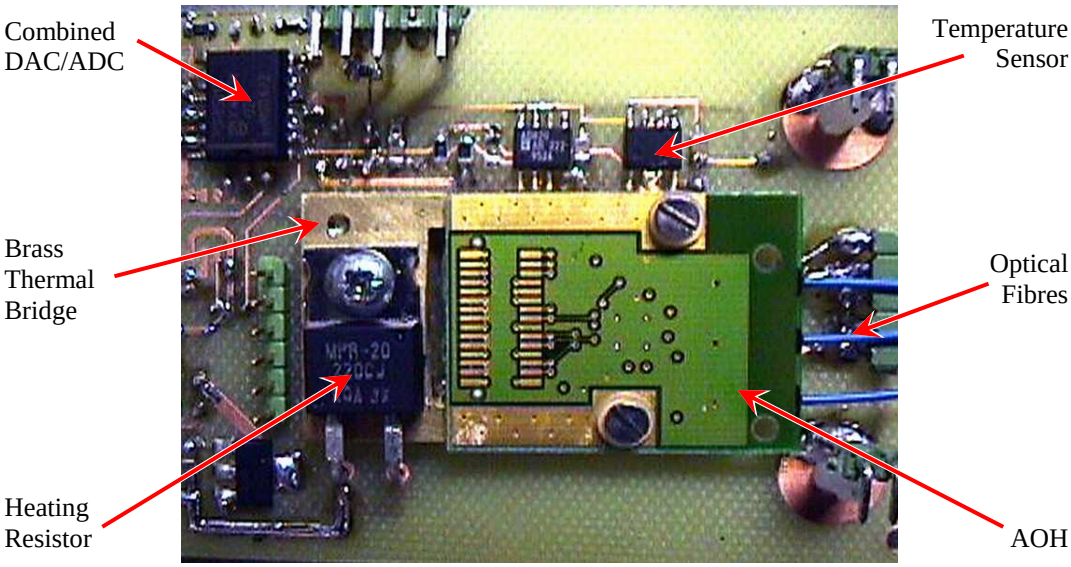


Figure 6.4: The prototype AOH temperature control system.

The system operates by applying a controlled heating level to the AOH in order to maintain it at a set point a few degrees above room temperature. The AOH itself is mounted on a PCB on top of a brass support structure, which is in thermal contact with a temperature sensor and a heating resistor. Temperature values are read from the sensor using an 8bit combined DAC/ADC in ADC mode; in DAC mode, the converter is used to set the heating resistor voltage. Communication with the DAC/ADC is achieved via an I^2C interface, with a second I^2C interface allowing the gain and bias current of each AOH channel to be set.

The optimum heating power is calculated with a standard Proportional-Integral-Differential (PID) feedback algorithm [87], the formula for which is shown in Equation 6.1:

$$W = P\left[(T_S - T_0) + D\frac{d}{dt}(T_S - T_0) + I\int(T_S - T_0)dt\right] \qquad \textit{Equation 6.1}$$

where W is the heating power required to maintain a set point temperature of T_S, T_0 is the current temperature of the system, P is the Proportional Gain, D is the Damping Constant and I is the Integral Gain. This algorithm is implemented using National Instruments LabVIEW, where the differential and integral terms are calculated by numerical methods for simplicity. Values for the P, I and D constants are determined experimentally through manual tuning of the system. An Optobahn Helix HRX9001 optical receiver hybrid is used to measure the intensity of the AOH output.

6.3.2 Calibration of the Laser Driver and Optical Receiver

In order to relate the change in AOH output with temperature to the level shifts caused by physical signals, it is useful to consider the magnitudes of the AOH input and output in terms of Minimum Ionising Particles (MIPs). This is achieved by calibrating the system with a known MIP-equivalent input.

During calibration, the AOH was held at 29.42 ± 0.18 ^{o}C (quoted error: maximum deviation) using the temperature control system. AOH gain and bias registers for all channels were set to 0 and 22 respectively. The gain register is 2 bits wide; gain settings of 0, 1, 2 and 3 correspond to laser driver output current ranges of ±2, ±3, ±4 and ±5 mA for an input differential voltage range of ±400 mV. Bias current, which is an offset added to the laser driver output so as to raise it above the laser diode threshold current (the point at which lasing begins), can be varied between 0 and 55 mA in steps of 0.45 mA (the bias current register is 7 bits wide). A bias

current register setting of 22 is at approximately the centre of the laser diode linear response region.

A signal generator was used to apply a controlled voltage level to the input of channel 0 of the AOH; 1 MIP was defined to be a 100 mV differential voltage signal across the AOH input termination resistors, since the nominal input range of ±400 mV corresponds to an equivalent nominal optical output range of ~8 MIPs. Plotting the output from the optical receiver against the laser driver input in MIPs yields a straight line, the gradient of which is 66.5 ± 0.5 mV/MIP (quoted error: standard deviation). This provides a scaling constant for converting between optical output in mV and optical output in MIPs.

6.3.3 Measuring the Temperature Dependence of the AOH Output Intensity

To characterise the temperature sensitivity of the AOH, the optical receiver was used to measure the output from channel 0 whilst the temperature control system held the laser drivers at nominal set point temperatures between 26 and 36 ^{o}C. At each temperature level, the laser driver bias current was incremented across its full range; at each bias current register value, the laser was allowed to reach thermal equilibrium with its surroundings and the output from the optical receiver was recorded. Plots of the AOH laser diode output as a function of bias current for a number of the set point temperatures are shown in Figure 6.5(a). The actual temperature of the AOH when each measurement was made is shown in Figure 6.5(b).

For most of the range of Figure 6.5(b), the temperature of the AOH is maintained by the control system to within ±0.18 °C. The actual measured temperature tends to be lower than the set point value; this is due to the parameters used in the PID control algorithm and is of little consequence, as temperature stability (and not actual temperature) is the important factor in the final system.

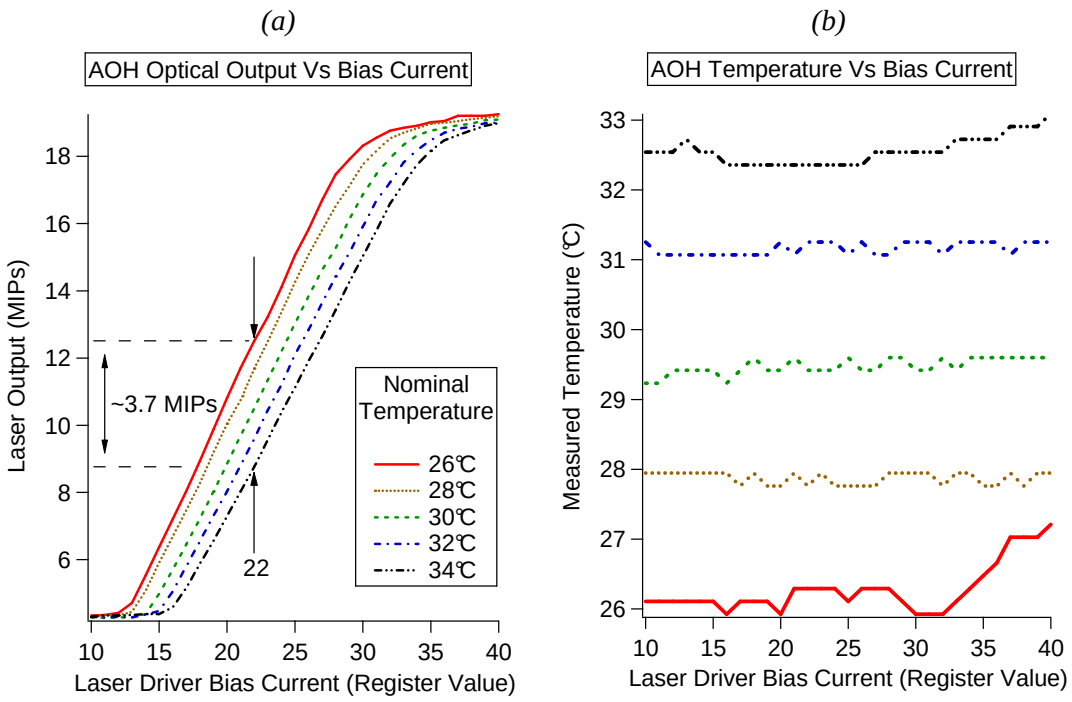


Figure 6.5: Plots of (a) the optical output, in MIPs, of the 'channel 0' AOH laser diode and (b) the temperature of the AOH, both as a function of the laser driver bias current register setting. The legend in (a) also applies to (b).

The plots in Figure 6.5(b) demonstrate that the temperature control system fails in certain limiting conditions. Firstly, the temperature of the AOH increases with bias current and for 'high' register settings of greater than 32 the operating temperature (in the absence of a heating resistor) of a laser diode exceeds the lowest set point temperature of 26 °C. Secondly, the heating resistor used in the prototype dissipates insufficient power to raise the AOH temperature to the highest set point value, and so the PID algorithm is compromised above ~32 °C. These issues are dealt with in the final FT temperature control system by ensuring that a set point temperature significantly greater than the operating temperature of the AOH is always used, and the efficiency (and maximum temperature) of the heating resistor is increased by means of thermal insulation (described in Section 6.3.4).

At each temperature level, the transfer curves in Figure 6.5(a) show the output intensity of the AOH increasing linearly with laser bias current after some threshold

current is reached. The rate of change of optical output with temperature is determined from the separation between the curves at a bias current register setting of 22; this represents the approximate centre of the linear regions of the transfer curves (the bias current at which the laser output was calibrated in Section 6.3.2) and is in a region where the temperature control system maintained stability at all set points. The measurements were repeated a total of four times and plots of optical output as a function of temperature for each experimental run are shown in Figure 6.6.

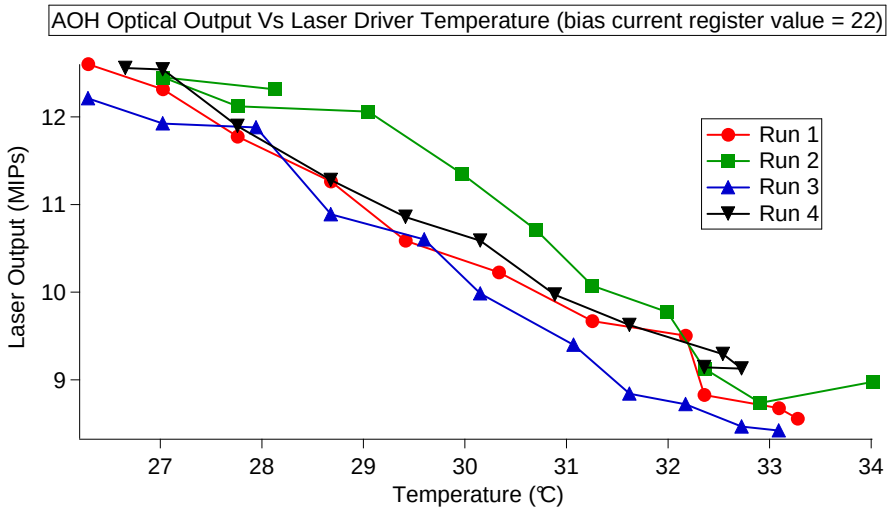


Figure 6.6: Plots of the output, in MIPs, of the 'channel 0' AOH laser diode as a function of measured AOH temperature, for a laser driver bias current register setting of 22. Results are shown for 4 independent experimental runs.

The relationship between optical output and temperature is approximately linear over the range of Figure 6.6. From the mean gradient of straight lines fitted to each of the data sets, the rate of change of the laser diode optical output with temperature is found to be -0.60 ± 0.02 MIP/^{o}C (quoted error: standard deviation).

To put this in context, the nominal input range of the FED Front End is 13 MIPs and the typical digital low to digital high region of an APV frame is equivalent to 8 MIPs. If a FED is calibrated such that frame data are centred at the midpoint of the input range, then it can at best tolerate variations of approximately ±2.5 MIPs in the AOH optical signals. With measured laboratory temperature fluctuations of ~5 °C,

the potential shift in laser diode output due to thermal effects is of the order of ~3 MIPs. Consequently, without temperature stabilisation, there is a risk of APV frame headers drifting beyond the thresholds of the Front End frame finding logic or strip data falling outside the FED input ADC range. With the temperature control system in place, signal fluctuations should be contained within limits of the order of ±0.1 MIPs, which will be dealt with effectively by Common Mode subtraction in the FED.

6.3.4 Implementation of the Temperature Control System on the FT

Having determined that the prototype temperature control system provides sufficient AOH output stability, it was transferred to the FT itself. The implementation is shown in Figure 6.7; two AOH mounting stations are pictured, one occupied and one empty. The control mechanism and hardware are the same as that described in Section 6.3.1, with the exception that the optimum heating power for each AOH is calculated by the FT VME FPGA. As a result, the temperature control systems for each FT are completely self contained and no user intervention is required once the desired AOH temperature has been set.

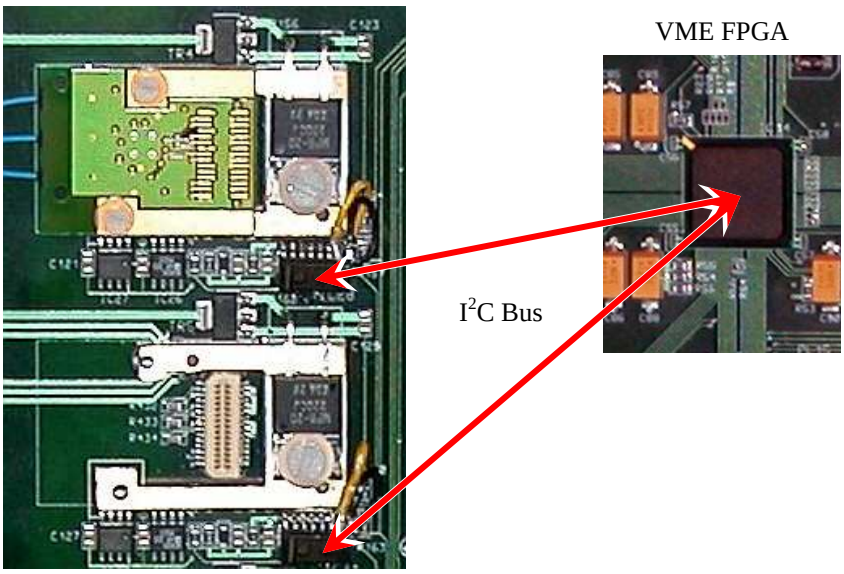


Figure 6.7: The AOH temperature control system implemented on the FED Tester itself. The FT VME FPGA runs the PID algorithm, reading the AOH temperature and setting the required heating resistor voltage level via an I^2C bus. Two of the 8 AOH mounting points on the FT are shown; the lower one is unoccupied.

An additional consideration when adapting the prototype design for the FED Tester is that the FT boards are operated in a VME crate which is fan cooled. Coupled with the relatively large surface area of the board (compared with the prototype PCB), this means that the thermal energy from the AOH heating resistors is rapidly dispersed into the air. Consequently, it becomes difficult to achieve a significant rise in the AOH temperature, given the maximum power dissipation limit of 1.6 W for each heating resistor. To alleviate this problem, steps are taken to thermally insulate each of the AOHs.

A Perspex box is mounted over the AOHs on the front side of the board, and a Perspex sheet is fixed to the rear of the PCB just behind the AOH mounting points, in order to minimise convective air cooling. The FT power planes around the temperature control systems are removed, to reduce conductive heat loss, and the brass thermal bridges are insulated from the surface of the PCB by additional thin sheets of Perspex (the brass only makes contact with the PCB in the direct vicinity of the temperature sensor). With this protection in place it is possible to vary the temperature of an AOH over a range of more than 20 °C above its normal operating point. It should therefore be possible to absorb any normal fluctuations in laboratory temperature.

6.4 FED Tester Software

The software used to control the FTs is written in a similar style to the drivers for the APVE (described in Chapter 3), but with significantly greater complexity. To enable the greatest possible flexibility in testing a FED, the APV frame generation, conditioning and simulated TCS operation are heavily customisable. A substantial amount of software is required to provide access to all of the FT configuration options and organise them in such a way as to make the FT easy to use. Further complications

are added due to the fact that an Ensemble of distinct FT boards must operate transparently as a single object when accessed programmatically. A simple overview of the FT software class structure is shown in Figure 6.8, and discussed in the following sections.

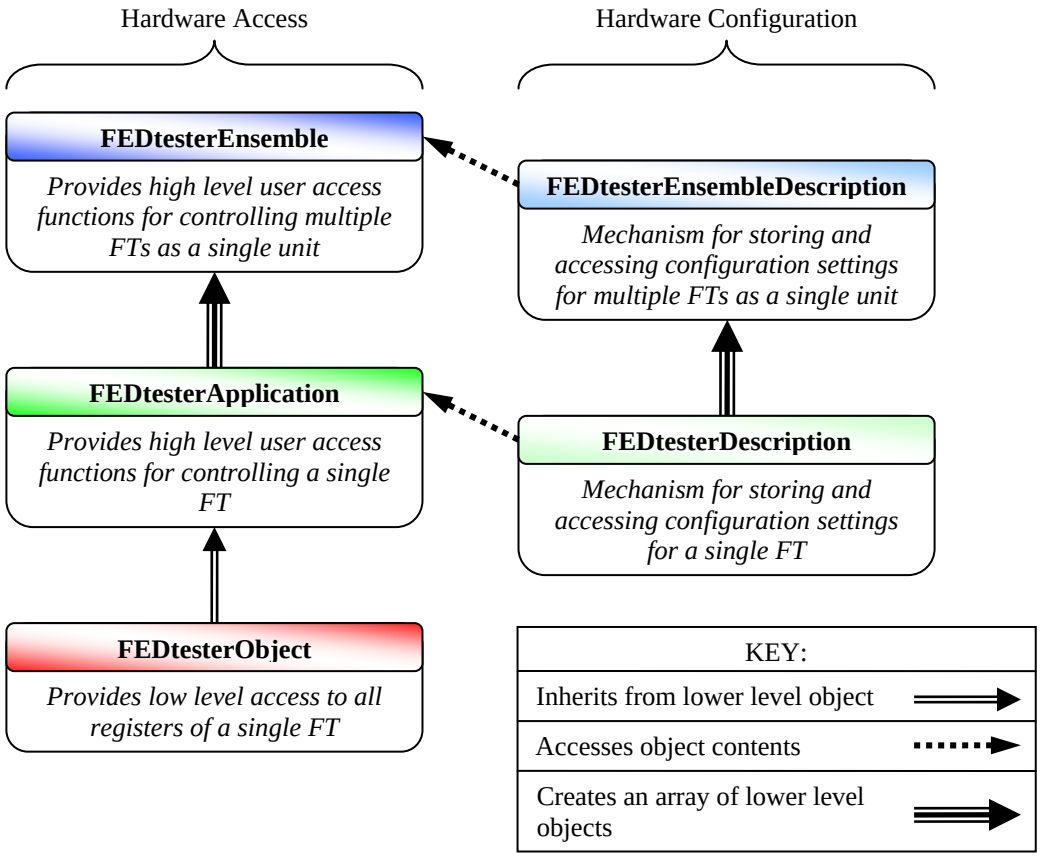


Figure 6.8: The class structure of the FED Tester software.

6.4.1 The FEDtesterObject Class

A FEDtesterObject class provides the lowest level access to the 186 individual control and data registers of a single FT. As with the APVE software, it makes use of the Hardware Access Library [54] for basic VME reads and writes. The FEDtesterObject functions are divided into six categories. APV frame methods enable data values to be written to the FT frame and frame pointer memories. Multiplexer functions allow the delay of the APV multiplexers to be set, for simulating fibre length differences. Crosspoint switch functions enable the APV frame data to be allocated to particular

FT channels. A large section of the code deals with the simulated TCS configuration, permitting the L1 trigger and control signals to be adjusted. Further methods allow the various PID parameters and set point temperatures of the AOH temperature control systems to be set, and finally, I^2C control functions provide low level access to the AOHs for setting gains and bias currents.

6.4.2 The FEDtesterApplication Class

Many of the FEDtesterObject functions are at too low a level for efficient control of the FT in user applications. Procedures such as writing a complete set of APV frames to the FT memory from file or communicating with AOHs via I^2C require many sequential register access operations, often with precise timing. The FEDtesterApplication class provides a user-friendly interface to the FT software by wrapping FEDtesterObject methods to simplify the execution of these common tasks. It also provides functions for initialising all of the FT configuration settings from a FEDtesterDescription (described in Section 6.4.4). As the FEDtesterApplication inherits from the FEDtesterObject class, and so has direct access to all of the low level functions in addition to the high level wrappers, a FEDtesterApplication object is all that is required for complete control of a single FT.

6.4.3 The FEDtesterEnsemble Class

An FT Ensemble consists of an arbitrary number of FT boards (typically 4); this is represented in software as an array of FEDtesterApplication objects. The FEDtesterEnsemble class contains the FEDtesterApplication array and maps the connections between all of the output channels of each FT to the input channels of the FED it is driving. All of the FEDtesterApplication functions that deal with specific channel settings are wrapped to include this mapping scheme. Consequently, a FEDtesterEnsemble object enables distributed access across all of the boards of an

FTE as though they were a single entity; it is only necessary to reference settings by a FED channel number and the relevant FT board and channel are accessed automatically.

A second consideration when dealing with an FTE is that the simulated TCS in only one of the boards should be used. The FEDtesterEnsemble class additionally wraps the FEDtesterApplication TCS control functions to ensure that only the FT Master is accessed (TCS systems on the other boards are disabled). With the ability to initialise the complete FTE from a FEDtesterEnsembleDescription (described in Section 6.4.5), a FEDtesterEnsemble object provides complete, well ordered control of an FTE, greatly simplifying the development of FED performance tests.

6.4.4 The FEDtesterDescription Class

In order to correctly initialise a single FT, it is necessary to configure it with 135 individual settings (the 32,690 frame data and 6,144 frame pointer values required to fill the FT memories are read from external text files, so only file paths are needed for initialisation). These settings are stored and organised through the use of a FEDtesterDescription class.

Configuration parameters can be loaded into a FEDtesterDescription object from a plain text, human readable FEDtesterDescription file or set programmatically. The Description object can then be passed cleanly to a FEDtesterApplication, where initialisation values are easily extracted through the use of the FEDtesterDescription member functions. It is also possible to write the current contents of a Description object back to a FEDtesterDescription file. This is an important feature for a Description class, as it means that a Description file can easily be modified with new values after calibration or user input, facilitating the organisation and maintenance of test setup parameters.

6.4.5 The FEDtesterEnsembleDescription Class

Initialising a FED Tester Ensemble comprised of 4 FTs requires 686 unique configuration parameters, 192 of which describe the mapping between FT outputs and FED channel inputs. With the increased complexity of the system compared to a single board, it is of even greater importance to store and organise the configuration settings effectively.

Resembling the FEDtesterEnsemble class, a FEDtesterEnsembleDescription consists of an array of FEDtesterDescription objects combined with additional channel mapping information. In order to simplify the management of an FTE, the configuration of a FEDtesterEnsembleDescription object is split into two sections; fixed structural settings and varying test parameters.

At construction time, the Description object is provided with the number of FTs in the Ensemble and a path to a FEDtesterMap file, which defines the physical board layout. An FTE should consist of a fixed number of FTs for most single-FED test procedures. Adding and particularly removing FT boards can drastically change the way in which the FED is driven, and should not be attempted without fully understanding the consequences. Requiring the number of boards in the FTE to be set in the object constructor conceptually isolates this parameter, and, in practice, encourages the value to be fixed as a constant at the top level of most test procedures (discouraging arbitrary changes).

```
1 0#-------- The number of the Bus Adaptor to be used.
2 10000#---- Base Address of FT 0.
3 20000#---- Base Address of FT 1.
4 30000#---- Base Address of FT 2.
5 40000#---- Base Address of FT 3.
6 MASTER#--- Enable state of FT 0.
7 ON#------- Enable state of FT 1.
8 ON#------- Enable state of FT 2.
9 ON#------- Enable state of FT 3.
```

Figure 6.9: The contents of an example FEDtesterMap file for an FTE containing 4 FTs.

An example of a FEDtesterMap file for an Ensemble containing four FTs is given in Figure 6.9. This contains the index of the VME crate in which the FTs are situated, (i.e. the index of the VME Bus Adaptor) and also the VME base address of each board. It additionally defines the enable states of the FTs; one board in the Ensemble must be a 'MASTER', distributing the TCS clock and trigger, while the others should be declared to be 'ON'. It is possible to disable a particular FT by setting its enable state to 'OFF'. Should a board develop a hardware fault, this enables it to be temporarily removed from the system for repair without having to modify the software structure of the Ensemble. For FTEs that contain more or less boards than in the example of Figure 6.9, 'base address' and 'enable state' lines are simply added or removed. The FEDtesterMap cannot be modified in software by a FEDtesterEnsembleDescription object as it describes the fundamental physical arrangement of the Ensemble, which can only be changed manually.

The remainder of the FTE configuration settings consist of parameters that are likely to vary between experimental runs. They are loaded into an existing Description object from a FEDtesterEnsembleDescription file, the format of which is demonstrated by Figure 6.10.

Figure 6.10(a) shows 13 of the 96 lines that define the properties of the optical connections between an FTE and a FED. Each line gives the FT board and channel number associated with a FED channel, followed by the enable state of the corresponding laser driver, the index of the simulated multiplexed APV pair chosen for output and the bias current and gain of the AOH. A FED channel may be disabled very simply by setting the matching FT board number to 'NC' (not connected).

(a)

```
(FT Board Number) <TAB> (FT Channel Number) <TAB> (Enable State) <T
0       0       ON      MUX0    31      1       # FED channel 0
0       1       ON      MUX1    34      1       # FED channel 1
0       2       ON      MUX2    32      1       # FED channel 2
1       0       ON      MUX0    33      2       # FED channel 3
1       1       ON      MUX1    31      1       # FED channel 4
1       2       ON      MUX2    31      1       # FED channel 5
2       0       ON      MUX0    31      1       # FED channel 6
2       1       ON      MUX1    30      1       # FED channel 7
2       2       ON      MUX2    29      1       # FED channel 8
3       0       ON      MUX0    30      1       # FED channel 9
3       1       ON      MUX1    32      1       # FED channel 10
3       2       ON      MUX2    32      1       # FED channel 11
0       3       ON      MUX0    32      2       # FED channel 12
```

(b)

```
||| General settings - Master TCS Settings and General MUX Settings:
OFF#------------ TCS L1 auto-reset status.
ON#------------- TCS status input enable state.
ON#------------- TCS control output enable state.
RANDOM#--------- L1A trigger type.
LOCAL_SRC#------ TCS source.
1#-------------- Minimum period between TTC B channel commands.
1000000000#----- TCS timeout period.
3#-------------- TCS Trigger Rule, Minimum spacing between L1As.
400#------------ Time period between L1As when L1As are generated repetitively.
40000#---------- Time period between L1As when L1As are generated repetitively and the TCS has switched to low L1A rate.
3564#----------- Time period between the TCS sending BC0s.
164#------------ L1A rate when multiplied by 40,000,000/65,536, when generated randomly.
-34#------------ Minimum possible MUX phase shift fine value.
50#------------- Maximum possible MUX phase shift fine value.
```

(c)

```
||| Specific FT Settings:
FTeConfigFiles/ApvFrameExample.txt#----- FT 0: APV frame data file.
32620#---------------------------------- FT 0: The number of frames in the APV frame data file, times the length of one frame (140).
FTeConfigFiles/ApvFramePtrExample.txt#-- FT 0: APV frame pointer data file.
1024#----------------------------------- FT 0: The number of frame pointer entries for each of the 6 virtual APVs.
DEG_240#-------------------------------- FT 0: MUX0 coarse phase.
0#-------------------------------------- FT 0: MUX0 fine phase shift.
0#-------------------------------------- FT 0: MUX0 coarse frame delay.
0#-------------------------------------- FT 0: MUX0 fine frame delay.
DEG_240#-------------------------------- FT 0: MUX1 coarse phase.
0#-------------------------------------- FT 0: MUX1 fine phase shift.
0#-------------------------------------- FT 0: MUX1 coarse frame delay.
0#-------------------------------------- FT 0: MUX1 fine frame delay.
DEG_240#-------------------------------- FT 0: MUX2 coarse phase.
0#-------------------------------------- FT 0: MUX2 fine phase shift.
0#-------------------------------------- FT 0: MUX2 coarse frame delay.
0#-------------------------------------- FT 0: MUX2 fine frame delay.
194#------------------------------------ FT 0: TEMP0 set point temperature.
138#------------------------------------ FT 0: TEMP0 proportional constant.
276#------------------------------------ FT 0: TEMP0 integral constant.
3045#----------------------------------- FT 0: TEMP0 differential constant.
194#------------------------------------ FT 0: TEMP1 set point temperature.
138#------------------------------------ FT 0: TEMP1 proportional constant.
276#------------------------------------ FT 0: TEMP1 integral constant.
3045#----------------------------------- FT 0: TEMP1 differential constant.
FTeConfigFiles/ApvFrameExample.txt#----- FT 1: APV frame data file.
```

Figure 6.10: Samples from an example FEDtesterEnsembleDescription file.

After the individual channel parameters the general FTE settings are given. As shown in Figure 6.10(b), these are concerned primarily with the simulated TCS of the FT Master, configuring the L1 trigger type, trigger rate and the handling of TCS status signals. The global maximum and minimum of the APV multiplexer phase shift range are also provided; it is necessary to experimentally establish these limits to ensure that fibre delays introduced by the multiplexers have a linear response.

Finally, Figure 6.10(c) shows an example of the distinct settings for each FT in the Ensemble. These consist of directory paths to the APV frame and frame pointer files that are to be used to fill the board memories, the multiplexer fibre delays and the AOH temperature control parameters. Lines 114 to 137 are simply repeated for every FT in the system.

The FEDtesterEnsembleDescription file can be modified in software, and provides a convenient mechanism for organising and storing the myriad parameters of each FED test procedure. Once configured with a FEDtesterEnsembleDescription, it is generally possible to run an FTE simply by enabling the TCS of the FT Master and loading additional APV frame files as required. Consequently, the FED Tester software package provides an easy to use interface to the relatively complex hardware configuration of the typical FTE.

~ Chapter 7 ~

S-Link Testing

7.1 The S-Link Connection

Approximately 440 data links are needed to transfer the output from the CMS Tracker FEDs to the Tracker DAQ system. Other sub-detectors at CMS have different front end readout modules which must also be connected to their respective DAQs. All of the four major experiments at the LHC (CMS, ATLAS, ALICE and LHCb) have sub-detector systems that include some form of front end electronics which must be connected to subsequent readout hardware; in total, several thousand individual data links will be required.

In order to facilitate the development of the detector readout systems it was decided that the format of these data links should be standardised throughout the LHC [88]. However, many of the detector subsystems have different physical readout requirements, in terms of the link length, bandwidth, latency, radiation tolerance, power consumption and cost. It is therefore difficult to standardise the actual hardware implementation of the connections; additionally, any components chosen during the early planning stages of the experiments would have been obsolete once large scale construction of the detectors had commenced. Instead, a specification has been adopted which defines the signals and protocol that should be used for communication between a front end module and the next stage of its readout system, without referring to the hardware of the physical data path. This concept is referred to as the S-Link (Simple Link) interface [89].

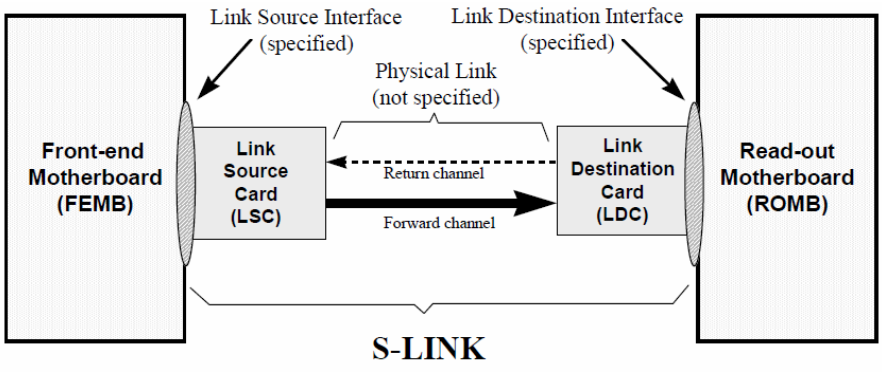


Figure 7.1: The S-Link concept. (Figure taken from Reference [88])

Figure 7.1 demonstrates the S-Link principle. A Link Source Card (LSC) accepts signals from the front end module (referred to as a Front-end Motherboard, or FEMB), encoding and transmitting them via an arbitrary physical path to a Link Destination Card (LDC). This decodes the received data and outputs the formatted signals to the next layer of readout hardware (referred to as a Read-out Motherboard, or ROMB). Only the FEMB-LSC and LDC-ROMB junctions are defined by the S-Link standard, using a simple synchronous FIFO-like interface. The link itself, consisting of the LSC and LDC daughter cards and their physical connection, is effectively a single component with known inputs and outputs, which may be implemented using any hardware configuration. As the ‘ends’ of every S-Link are identical it is also possible to change the transmission technology, from electrical to optical for example, without modifying the FEMB or ROMB (although the daughter cards may potentially be integrated into the motherboards to minimise space requirements, if necessary).

The S-Link standard has evolved to incorporate, with minimal complexity, all of the essential features that are required for a readout data connection. A normal S-Link may take one of two forms, simplex or duplex. Both have a maximum clock rate of 40 MHz, and include internal error detection and an inbuilt self-test function which verifies a fixed pattern that can be sent automatically from the LSC to the LDC. Data

words transmitted across the link may have a variable width of 8bit, 16bit or 32bit, and each word is accompanied by a control bit that enables the identification of headers, trailers and any other command signals. The duplex S-Link has an additional return path from the LDC to the LSC, which includes a dedicated flow control line (for requesting the FEMB to cease transmission if the ROMB cannot cope with the data rate) and a small number of optional user-definable feedback channels.

An S-Link connection supports a maximum data rate of ~150 MB/s. This is adequate for many applications, but insufficient for the read out of the Tracker FED; as demonstrated in Chapter 4, a total bandwidth of 400 MB/s is required to safely accommodate typical Tracker occupancies. To allow for the increased capacity of modern DAQ systems, the S-Link specification was extended to create the S-Link64 [90]. Incorporating all the features of the duplex S-Link, the S-Link64 standard adds 32 additional data lines (for a data width of 64bit) and permits clock rates of up to 100 MHz, enabling a maximum throughput of 800 MB/s.

7.2 The Common Data Format

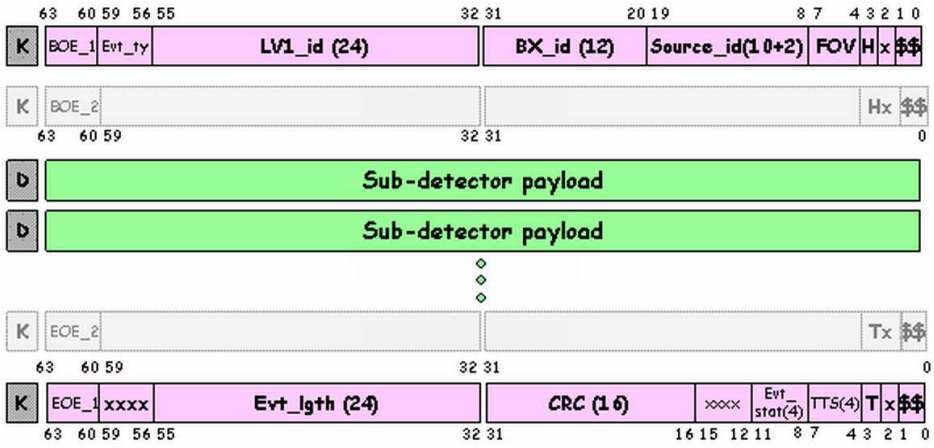


Figure 7.2: The Common Data Format for CMS events. (Figure taken from Reference [91])

In addition to the standardised S-Link signal transmission mechanism, the CMS DAQ Readout Unit Working Group (RUWG) [91] has established a Common Data Format

(CDF) for the bunch crossing event data that are read out from each CMS detector subsystem upon receipt of a Level 1 trigger. Figure 7.2 shows the structure of a CMS event. The first 64bit S-Link word is a DAQ header which identifies the L1 trigger (LV1_id), LHC bunch crossing number (BX_id) and readout hardware module (Source_id) associated with the data. After this follows the sub-detector payload; a 64bit trailer word at the end of the data packet provides information about the event length (Evt_lgth) and status, including a CRC code [71] for error checking. It is possible for additional header and trailer words to be used, but they are currently omitted from the Tracker FED output. The individual fields of the CDF are described in greater detail in Table 7.1.

Data Field	Width (Bits)	Description
K	1	S-Link control bit, indicating a control word
D	1	S-Link control bit, indicating a data word
BOE_*n*	4	Indicates the beginning of the event packet
Evt_ty	4	Event type identifier (e.g. test, calibration or physics data)
LV1_id	24	Level 1 trigger number associated with the event
BX_id	12	LHC bunch crossing number associated with the event
Source_id	12	Identification number of the front end module that generated the event data
FOV	4	Version number of the event data format
H	1	Indicates final header word (for when more than one header word is used)
x	-	Reserved bit
$$	2	Ignored (used for internal S-Link error checking)
Sub-detector payload	64 x *N*	Event data combined with detector-specific formatting information
EOE_*n*	4	Indicates the end of the event packet
Evt_lgth	24	The total length of the event packet in terms of 64bit S-Link words
CRC	16	Cyclic Redundancy Code
Evt_stat	4	Event status summary (as yet undefined)
TTS	4	Current values of the TTS status bits
T	1	Indicates final trailer word (for when more than one trailer word is used)

Table 7.1: The Common Data Format fields

7.3 The FED Kit

The FED Kit [92] is a prototype of the S-Link64 hardware that will be used with the Tracker FEDs at CMS. It also includes a ROMB which enables data from the FED to

be read out directly via a computer, an essential feature for testing the FED and S-Link prior to the implementation of the final CMS DAQ system. A diagram of the FED Kit hardware configuration is shown in Figure 7.3.

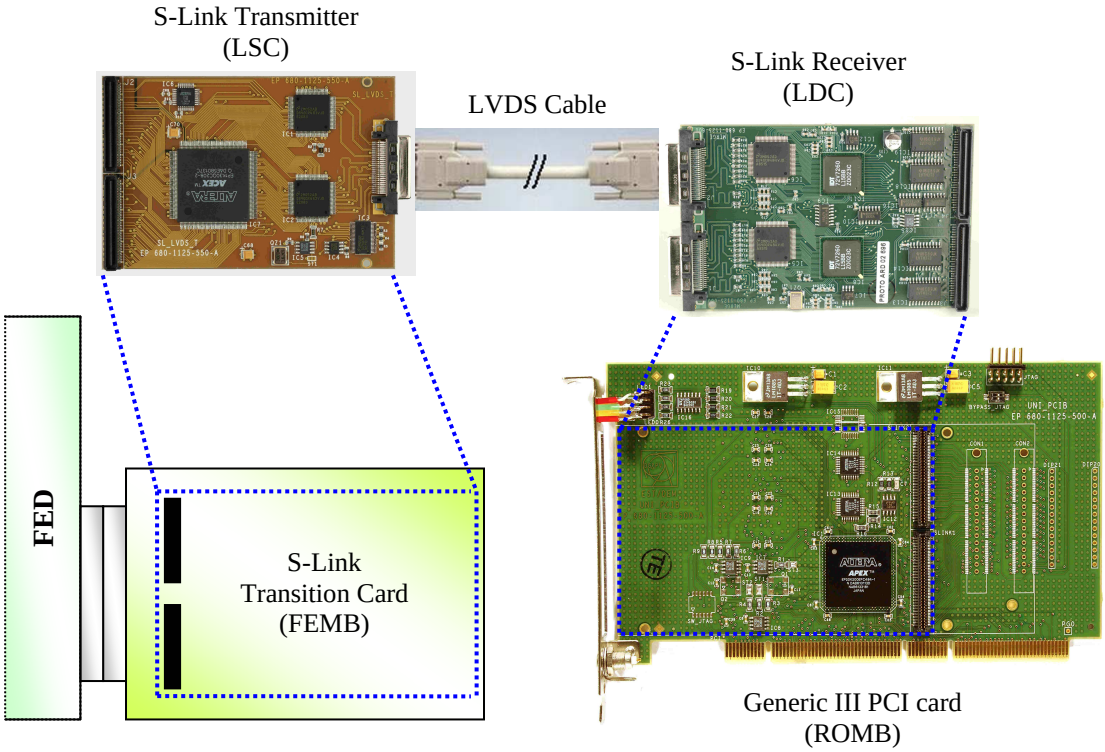


Figure 7.3: The FED Kit implementation of the CMS Tracker FED S-Link64 connection. The ROMB is used inside a PC and accessed via a 64bit 66 MHz PCI bus. Blue dotted lines denote daughter card mounting locations.

S-Link signals are output from the FED via its VME64X J2 connector to an S-Link Transition Card (the FEMB). Mounted on the Transition Card is the first component of the FED Kit; an S-Link Transmitter (the LSC). This accepts 64 data and 10 control lines (returning 10 feedback lines) as single-ended Low Voltage CMOS (LVCMOS) signals at a maximum clock frequency of 100 MHz, according to the S-Link64 specification. In practice, the FED generates a lower frequency 80 MHz S-Link clock as it is most convenient for all clocks in the readout system to operate at an integer multiple of the LHC bunch crossing rate of 40 MHz. Chosen for its relatively low cost and technical design simplicity, the physical medium used to transmit S-Link

data is a Low Voltage Differential Signalling (LVDS) cable. An electrical connection of this type has a short maximum transmission length, of ~15 m, but this is adequate for CMS since FEDs will always be situated near to their corresponding ROMBs.

The S-Link Transmitter multiplexes the input data and control lines and converts the result into a differential format that is sent to an S-Link Receiver (the LDC) via 14 twisted wire pairs contained within the LVDS cable. Four additional twisted pairs provide a data path for flow control and other feedback signals. One S-Link Receiver can accept the output from one or two Transmitter cards; in either case the received signals are converted back into the standard S-Link format and are then presented to the ROMB, upon which the S-Link Receiver is mounted. The ROMB is a Generic III (GIII) PCI card [93], a general purpose DAQ board that is based upon an ALTERA APEX20K FPGA. It may be programmed with a number of different applications and is used by the FED Kit to simulate the CMS Tracker Front End Readout Links.

Employing a 64bit 66 MHz PCI [94] data bus, the GIII can access the memory of a host PC at a maximum data rate of ~500 MB/s (close to the theoretical hardware limit of 528 MB/s). A contiguous section of this memory is divided into pre-allocated blocks of a user-definable size. Events are extracted from the input S-Link data stream via the identification of the CDF header and trailer words. The event contents are passed directly into the allocated memory blocks, from which they can be read out using software applications running on the host PC. Blocks are filled and emptied in a first-in first-out order, and a number of software/hardware FIFOs keep track of the stored event memory locations. If all of the memory blocks are in use, the GIII card asserts a ‘link full’ status via the S-Link flow control path which prevents the FED from outputting further events. An alternative mode of operation simply dumps every

word received through the S-Link into the host PC memory, enabling the system to be debugged should transmission errors corrupt the header or trailer words such that events cannot be correctly identified by the hardware. The FED Kit includes a device driver for the GIII card and a low-level software library for reading events, either with a (slow) user-friendly interface or via direct access to the memory blocks for greatest efficiency [95]. A maximum readout rate of 480 MB/s is supported; although this limit is less than the bandwidth of the PCI bus (it is in fact the wire speed of the LVDS cable at the normal driving clock rate), the S-Link performance still exceeds the 400 MB/s throughput requirement of the CMS DAQ.

7.4 S-Link Data Integrity

Checks have been carried out by the developers of the FED Kit to verify the quality of the LVDS S-Link connection [92]. These involved the transmission of test patterns through LVDS cables of several different lengths at a number of data rates. Over a period of one month, ~2 PB of data were received via a 2 m cable and no errors were observed. The maximum cable length for error-free transmission was found to be 17 m, tested at a data rate of 528 MB/s for 8 hours (note that for these simple hardware checks it is possible to drive the S-Link Transmitter at a higher clock frequency than that typically used when reading out the FED, enabling greater throughput of data).

Verification of the data integrity of the physical S-Link connection itself is vital. However, it is also important to check that signal quality is maintained when the FED Kit is used to read out real FED boards (unavailable at the time of the original tests) under (approximately) realistic CMS conditions.

7.4.1 Hardware Configuration for S-Link Testing

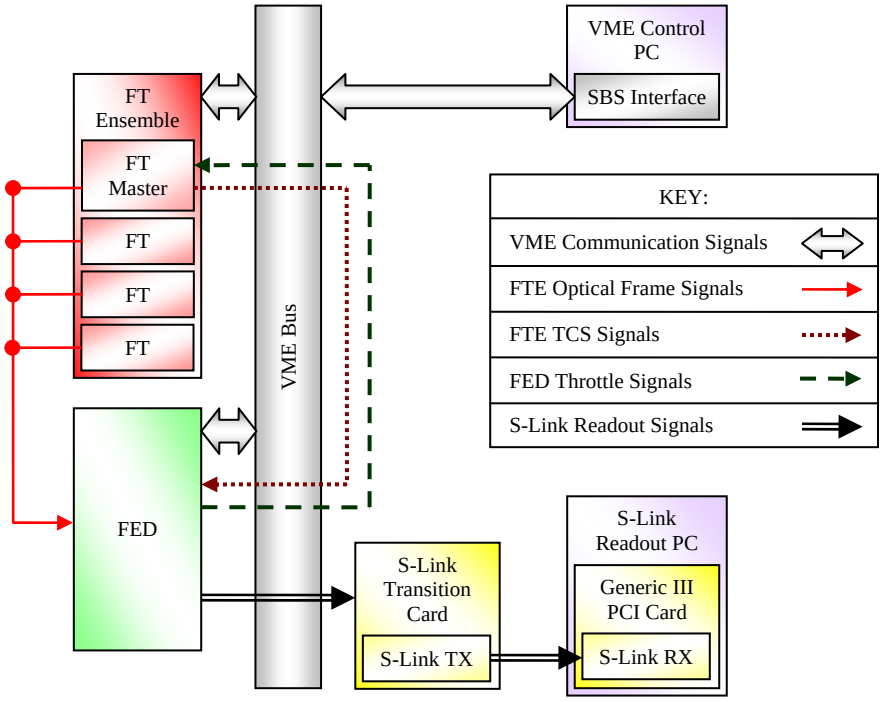


Figure 7.4: A block diagram of the hardware used for S-Link data integrity tests.

A block diagram of the hardware configuration required for testing the S-Link connection with a real FED is shown in Figure 7.4. The FED is operated with a FED Tester Ensemble (see Chapter 6); clock and Level 1 trigger signals are provided by the simulated CMS Global Trigger Control System running in the FT Master, through direct wire connections on the VME backplane. For completeness, the FTE drives the FED with 96 channels of optical data. This enabled the FE readout path to be verified during the early debugging stages of the experiment, but no input is necessary for the generation of test patterns.

Events are read out via the FED Kit, with a Transition Card developed for the ECAL sub-detector used to connect the S-Link Transmitter to the FED. If the S-Link Receiver asserts ‘link full’ then the FED is prevented from transmitting further events and they remain stored within the memory buffer of the BE FPGA (see Chapter 4). Throttle signals, requesting a reduction of the input Level 1 trigger rate (or a veto on

triggers) when the FED internal buffer occupancies reach predefined limits, are fed back to the TCS in the FT Master (again via direct backplane wire connections).

The FED Kit GIII card requires a host computer with a 3.3V 64bit PCI bus. It is mounted inside a high performance dual processor PC equipped with 6 PCI-X slots [94]. VME control is achieved by means of an SBS 620 VME Bus Adaptor [96]. Unfortunately, the only SBS interface available at the time had a 32bit 5V-only PCI card, which is incompatible with the PCI-X standard. Consequently, two computers are used; one for S-Link readout and the other for VME control of the system. This has the advantage that all of the resources of the S-Link PC are available for the processing of the received data, but care is required to ensure that the two machines operate synchronously.

In order to verify the S-Link connection it is necessary for the FED to output known test patterns that can be checked for transmission errors upon receipt. It is possible to fix the size of FED events by operating the board in Scope mode with a constant scope length, but the event content depends upon the optical signals present at the FED input. Even if constant optical levels are provided by the FTE, random noise fluctuations will lead to unknown differences between events. Consequently, the FED firmware was modified to include a ‘test pattern mode’.

When this is enabled, the ‘sub-detector payload’ (see the CDF in Section 7.2) of each event is overwritten with fixed data. Two patterns are available; a counter, in which each byte of the S-Link data is incremented by one between successive words, and a repeating sequence of ‘As and 5s’. For the latter, 64bit S-Link words are alternately set to a value of 0xAAAAAAAAAAAAAAAA and then 0x5555555555555555. This means that adjacent bits in each word have the opposite

parity and all bits swap parity between successive words, representing the worst-case switching example.

7.4.2 Testing S-Link Data Integrity

Software was written for the VME control PC to configure the FED and FTE, and drive the FED with a user-defined number of Level 1 triggers. During normal operation at CMS, the FED will run in Zero Suppressed mode with an average expected trigger rate of 100 kHz. For the generation of test patterns it is necessary to operate the FED in Scope mode, but a scope length of 6 samples is used in order to produce events that are approximately equal in size to those output in Zero Suppressed mode at typical Tracker occupancies. The FTE provides the FED with repetitive 100 kHz triggers, but the effective trigger rate due to the FED throttling action while verifying events is ~17 kHz.

A second package was developed for the S-Link PC, which wraps the FED Kit software to enable automatic configuration and readout of the GIII card and provides event analysis functions (this later evolved into the FK_Object class seen in Chapter 8). For a true S-Link data integrity check, a bit by bit comparison of the received event payload with the expected test pattern is performed; any errors are flagged in an output report file. Synchronisation between the control and readout computers is achieved manually. The VME PC resets and configures the FED and FTE, and then asks for the number of L1 triggers, *N*, that should be sent. Whilst it is waiting, the S-Link PC configures the GIII card and then polls the receiver for events until the next *N* have been acquired. Providing the VME PC with the number of triggers then starts the experimental run.

Using this software, 1TB of ‘As and 5s’ test pattern data were successfully transmitted through a 1.5 m S-Link cable, over a period of 10.8 hours, without

receiving any bit errors. It is possible to estimate the probability that no S-Link words will be transmitted incorrectly by assuming that transmission failures occur with a random Poisson distribution, as shown in Equation 7.1:

$$P(n) = \frac{e^{-\lambda}\lambda^{n}}{n!} \qquad \textit{Equation 7.1}$$

where P is probability of finding n transaction errors when the mean number is λ. The probability that no errors are obtained is given by $P(0, \lambda) = e^{-\lambda}$. Consequently, the probability that λ is less than or equal to a value λ_0 is given by Equation 7.2:

$$P(\lambda \le \lambda_0) = \int_0^{\lambda_0} e^{-\lambda} d\lambda = 1 - e^{-\lambda_0} \qquad \textit{Equation 7.2}$$

For a Confidence Limit (CL) of 95%, Equation 7.3 is true:

$$1 - e^{-\lambda_0} = 0.95 \Rightarrow \lambda_0 = 3.00 \qquad \textit{Equation 7.3}$$

A sample of S-Link data transactions should therefore include a maximum average of ~3 failures. Thus, if n S-Link words are transmitted without errors, the probability (at a Confidence Limit of 95%) that one word will contain an error is less than $3/n$. 1 TB of data corresponds to ~1.25 x 10^{11} 64bit words and so it can be calculated that the probability of receiving an incorrect S-Link word is less than ~2.4 x 10^{-11} @ 95% CL.

It is useful to place this in context by estimating the number of words that would have to be received (without errors) to guarantee no more than one error per week of normal CMS operation. Given the maximum average FED readout rate of 200 MB/s, a maximum of ~1.6 x 10^{13} S-Link words should be transmitted in 7 days. If one of these words were incorrect, the probability of obtaining an error would be ~6.3 x 10^{-14}. From the $3/n$ rule derived above this corresponds to the bit by bit verification of ~4.8 x 10^{13} words, assuming that no errors are found. The processing speed of the S-Link readout PC limits data verification to a maximum rate of ~3.8 x 10^{6} words/s. Consequently, guaranteeing to a 95% Confidence Limit that no more than one

transmission error will occur each week would require an S-Link verification test lasting at least 146 days (~4.9 months). A test of this duration has not yet been run, as the FED and FTE hardware have been required for other applications (many of which are described in Chapter 8). However, a reduction in data verification time could easily be achieved by operating multiple FEDs in parallel (perhaps using the final CMS DAQ when available).

In addition to checking the integrity of S-Link data transmissions, the S-Link verification software provided the first real test of the FED hardware and firmware at high data rates with CMS-like driving signals. It highlighted a number of internal buffer overflow, throttle response and status reporting errors that were present in early versions of the firmware. As an essential debugging tool, the S-Link data integrity test prepared the FED for the advanced validation procedures described in Chapter 8.

7.5 The S-Link Transition Card

During the initial S-Link data integrity checks, the interface between the FED and the S-Link Transmitter was an ECAL Transition Card, as no alternative hardware was available. The physical pin mapping of the VME64X J2 connector on the ECAL card does not match that of the final CMS Tracker FED design; this was accommodated by temporarily modifying the order of the S-Link data lines in the FED firmware. FED throttle signals are also output via the J2 connector, but the ECAL Transition Card provides no readout path for these data and prevents external access to the relevant backplane pins. Copies of the throttle signals were therefore sent to the FED J0 connector and transmitted to the simulated TCS of the FTE through direct wire links.

While these modifications to the operation of the FED are acceptable for tests of the S-Link hardware, they are not compatible with the final Tracker readout system. It therefore became necessary to produce a new S-Link Transition Card, with

the correct J2 pin mapping and the ability to convert the raw throttle signals from the FED into the standard CMS format required by the real TCS.

7.5.1 A New Transition Card for the CMS Tracker FED

(a) *(b)*

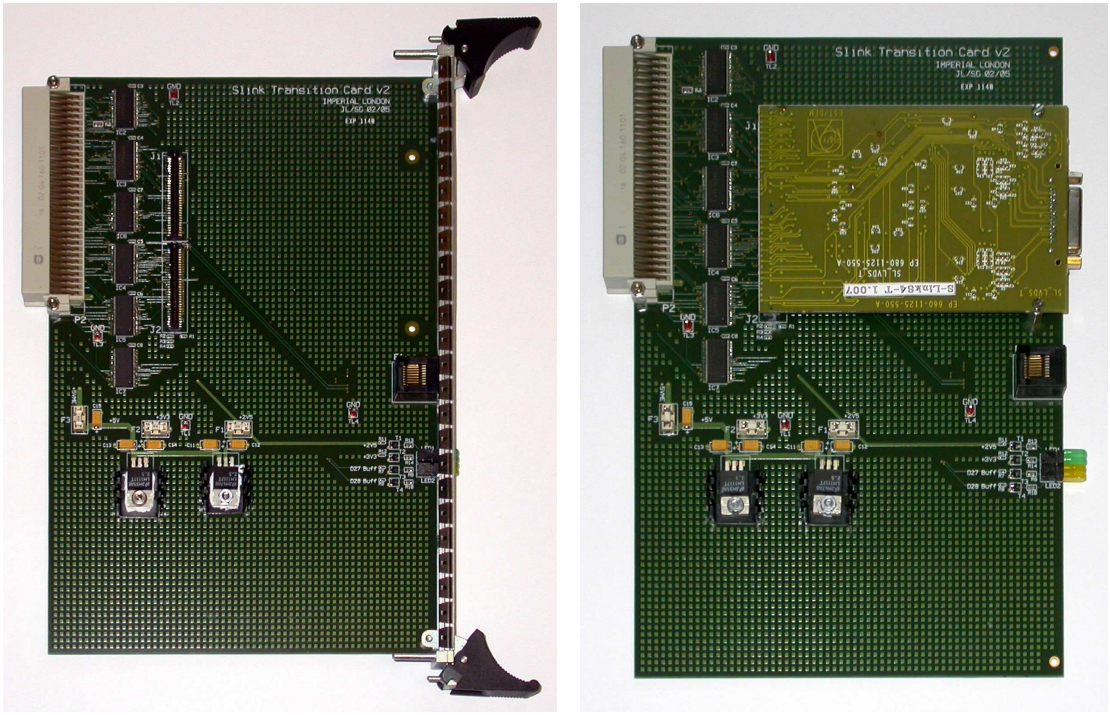


Figure 7.5: (a) is a picture of a CMS Tracker FED S-Link Transition Card. (b) shows the Transition Card with the front panel removed and a mounted FED Kit S-Link Transmitter.

The CMS S-Link Transition Card was developed using Mentor Graphics DesignView and Expedition PCB [97]. It is implemented as a 6-layer (2 power and ground, 4 signalling) standard 6U (160 x 233 mm) VME transition card, pictures of which are shown in Figure 7.5. Functionally, the Transition Card is a simple device which routes S-Link data and control lines between the VME backplane and the S-Link Transmitter, provides access to the FED throttle signals and includes a basic power supply. The fundamental elements of the board are highlighted on the PCB layout shown in Figure 7.6, and discussed in the following sections.

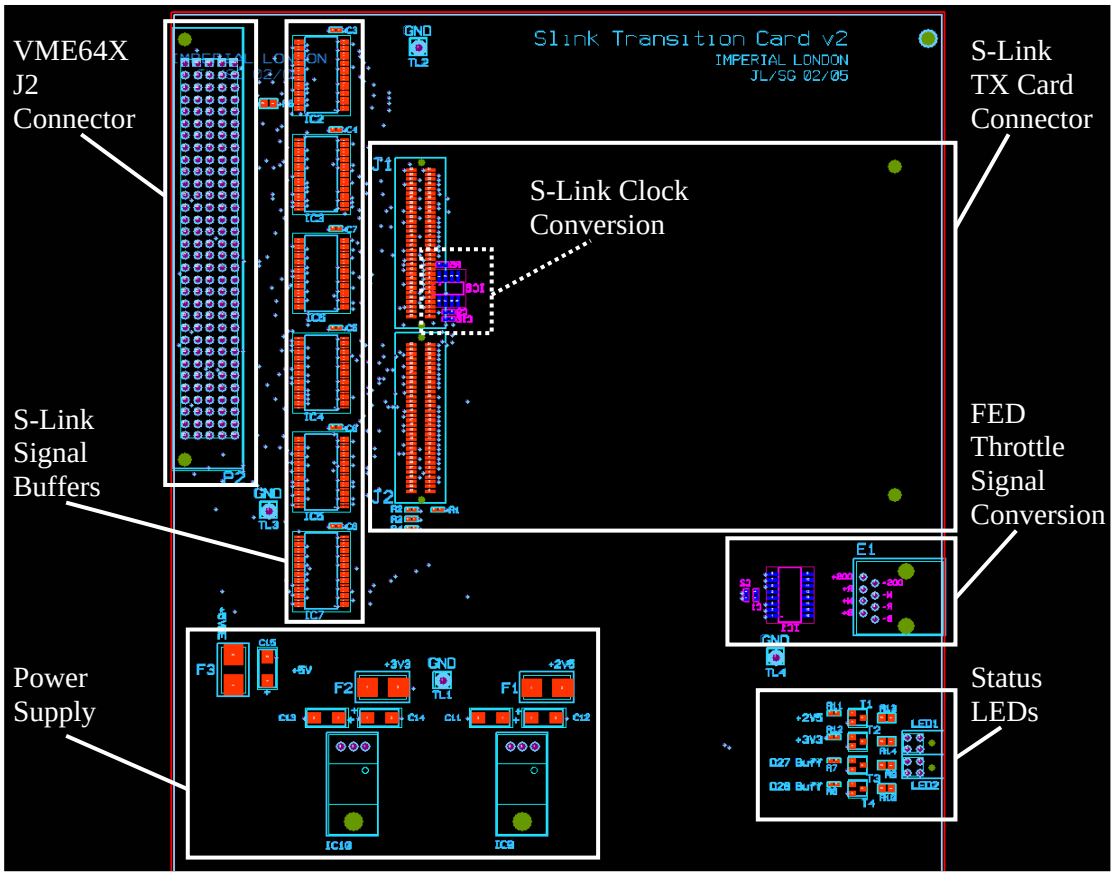


Figure 7.6: The CMS FED S-Link Transition Card PCB layout. Traces have been removed, for clarity. Important regions are highlighted with white outlines.

7.5.2 S-Link Data and Control Signal Path

All of the S-Link data and control lines (with the exception of the clock, discussed in Section 7.5.3) leave the FED and enter the S-Link Transmitter as single-ended LVCMOS signals. The ECAL Transition Card has direct PCB traces between the VME64X J2 and S-Link Transmitter connectors. Although the data integrity test described in Section 7.4.2 did not indicate any performance issues with this configuration, as no transmission errors were ever observed, it was decided that the new Transition Card should attempt to improve signal quality and reduce the load on the FED BE FPGA, whose S-Link output pins are connected to the FED VME connector without intermediate protection. 74LVTH16245 [98] low voltage 16bit transceivers, configured for one-directional operation, are therefore used to buffer the S-Link signals between the FED and the S-Link Transmitter.

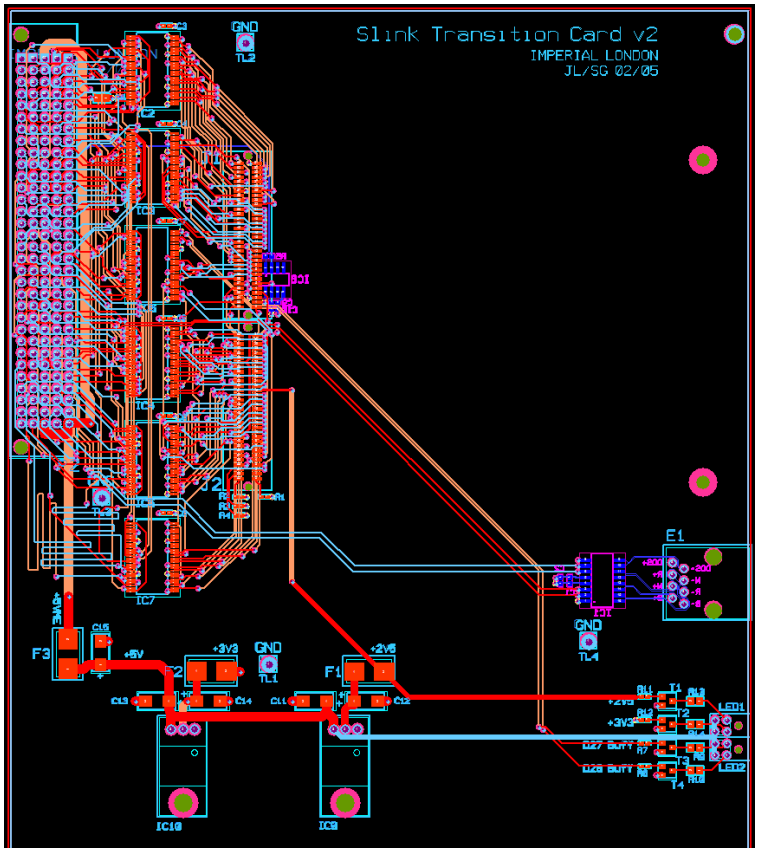


Figure 7.7: The CMS FED S-Link Transition Card PCB layout, shown with signal tracks.

A second picture of the new Transition Card PCB layout, showing signal tracks, is presented in Figure 7.7. The large number of S-Link data and control lines (84 in total), combined with the requirement that all tracks should be as short as possible to reduce noise pickup, results in a high track density in the region of the buffers. An additional constraint is introduced by the relatively high frequency of the S-Link transmission; the variation in propagation delay between signals that would ideally arrive at the same instant should be less than 10% of the driving clock period, or less than ~1 ns for the 80 MHz S-Link clock. Using three of the PCB layers (1, 3 and 4) for single-ended LVCMOS signalling enabled the data and control tracks to be routed successfully, all with propagation delays of less than 0.5 ns. The LVCMOS traces are also impedance matched to the FED and the S-Link Transmitter boards (60Ω), to reduce signal degradation due to transmission line reflections.

7.5.3 The S-Link Clock

The most important signal provided to the S-Link Transmitter by the FED is the clock, as this drives the entire system and determines the location of all the data sampling points. Particular care is therefore taken to minimise noise pickup on the S-Link clock. Whereas all other S-Link data and control lines are single-ended, due to the limited number of pins on the VME64X J2 connector, the clock is output from the FED as a differential LVDS signal. Layer 6 of the board is dedicated to the transmission of LVDS signals, shielded from the single-ended LVCMOS tracks by a ground plane on layer 5. Conversion from the differential to a single-ended clock, required by the S-Link Transmitter, is carried out by a DS90LV018A [99] 3V LVDS single CMOS differential line receiver. The DS90LV018A is positioned as close to the clock pad on the S-Link Transmitter connector as possible, such that the single-ended clock trace on the Transition Card has a length of less than 1mm.

It should be noted that the DS90LV018A has a maximum propagation delay of 2.5 ns, while the transceivers used to buffer the other S-Link data and control signals have a maximum delay of 3.5 ns, leading to a small temporal offset between the clock and data/control lines. However, this is easily dealt with by tuning the delay of the S-Link clock (via changes to the FED firmware) as it leaves the BE FPGA.

7.5.4 FED Throttle Signals

A 4bit fast feedback throttle signal indicates the current status of the FED. Each bit corresponds to a particular operating condition (Ready, Overflow Warning, Busy and Out of Synch) but they are used in conjunction to encode a total of seven possible FED states, summarised in Table 7.2.

4bit Value [Ready] [Busy] [Out of Synch] [Overflow Warning]	Status	Description
0000	Disconnected	Hardware failure or damaged/missing FMM connection
1111	Disconnected	Hardware failure or damaged/missing FMM connection
0001	Warning Overflow	Internal memory buffers almost full (overflow imminent)
0010	Out of Synch	Loss of synchronisation with the TTC
0100	Busy	Internal memory buffers full (cannot accept Level 1 triggers)
1000	Ready	Ready to accept L1 triggers
1100	Error	Any other state that prevents normal operation of the FED
All other values are reserved	Illegal	Must not be generated by the FED

Table 7.2: The encoded values of the FED throttle status

The S-Link Transition Card receives the FED throttle through the VME64X J2 connector as four single-ended LVCMOS signals. These are converted to LVDS by a DS90LV047A [100] 3V LVDS quad CMOS differential line driver and output via an 8/8 way RJ-45 Ethernet connector. Status signals can then be transmitted to the simulated TCS in the FTE, or an FMM in the final system [50], using a standard low-cost UTP5 patch cable.

7.5.5 Transition Card Status LEDs

Four status LEDs are visible on the front panel of the S-Link Transition Card. Two of these are connected to the on-board generated voltage lines (described in Section 7.5.6) and are used to indicate that the board is correctly powered. The remaining LEDs are connected to two undefined pins on the VME64X J2 connector. Currently, they do not show any status information but the FED BE FPGA can easily be configured to output any two signals via the spare pins; for example, it may be useful in the future to display the S-Link clock on one LED (to indicate clock failures) and the FED Kit ‘link full’ status on the second.

7.5.6 Transition Card Power Supply

The VME64X J2 connector only provides access to the +5V backplane power supply. This meets the requirements of the status LEDs but the S-Link Transmitter operates at

+2.5V, and the LVDS and buffer components at +3.3V. Two LM1117 [101] fixed voltage low-dropout linear regulators (supporting a maximum current of 1.5A) generate independent +2.5V and +3.3V power lines from the input 5V source. Switched mode voltage regulators are more efficient than linear regulators but they introduce noise to the output voltage level and so were considered inappropriate for use on the Transition Card, where signal quality is paramount.

Linear regulators dissipate as heat a power equal to the dropped voltage multiplied by the supplied current; it is therefore important to verify that adequate cooling is provided. An S-Link Transmitter draws ~200 mA at +2.5V, corresponding to a dissipated power of ~0.5 W in the +2.5V LM1117 component. The maximum operating temperature of the LM1117 is 125 °C and as a result the maximum allowable value of the junction to ambient thermal resistance (θ_{JA}, which defines the temperature increase per Watt of dissipated power) is ~200 °C/W. As θ_{JA} of the LM1117 package chosen for the Transition Card is only 79 °C/W, unaided conductive (via the PCB) and convective (air) cooling is sufficient.

For the LVDS and buffer components, the total required current is ~700 mA, leading to a dissipated power of ~1.2 W and a maximum allowable θ_{JA} of ~84 °C/W. Consequently, an LS205 heat sink is used with the +3.3V LM1117 component, which reduces θ_{JA} to 30 °C/W and provides a generous safety margin. As the LS205 is inexpensive, one is also fixed to the +2.5V regulator; it should be unnecessary, but ensures that the device will never overheat.

7.5.7 Production and Testing of the New S-Link Transition Card

A number of prototype S-Link Transition Card PCBs were manufactured by Express Circuits [102], with component assembly performed by technicians at Imperial College. The design was initially tested by repeating the S-Link data integrity check

described in Section 7.4, replacing the ECAL Transition Card with one of the new boards and using the Ethernet connection to route FED throttle signals to the simulated TCS in the FTE. 1 TB of ‘As and 5s’ test pattern data were again transmitted; both the S-Link data and FED throttle signals were received without errors. The prototype boards have since been used extensively at IC and RAL for all of the FED performance tests described in Chapter 8 and no problems have ever been observed.

Approximately 440 S-Link Transition Cards (and spares) will be required at CMS, one to provide readout for each FED. Manufacture and assembly will be performed by Cemgraft [103], at cost of £89.93 per board. An initial production run of 70 Transition Cards has already been completed and they have passed routine acceptance testing. Verifying a board is a simple task; reading out one ‘As and 5s’ test pattern event from a FED will immediately identify any short circuits or unsoldered pins along the S-Link data and control paths. A second and final batch of 400 Transition Cards will be ordered once the existing boards have been integrated and fully tested in the final CMS DAQ.

~ Chapter 8 ~

FED Performance Testing

The Industrial Acceptance Test ensures that each FED is electronically operational and the S-Link data integrity checks demonstrate the ability of the FED to process event information at high data rates. With the basic functionality of the FED thus verified, it is possible to carry out rigorous performance testing under realistic CMS operating conditions utilising the full capabilities of the FED Tester. The hardware and software environment developed for complete FED system tests and an evaluation of the board performance are discussed in the following sections.

8.1 Hardware Configuration for Full System Tests

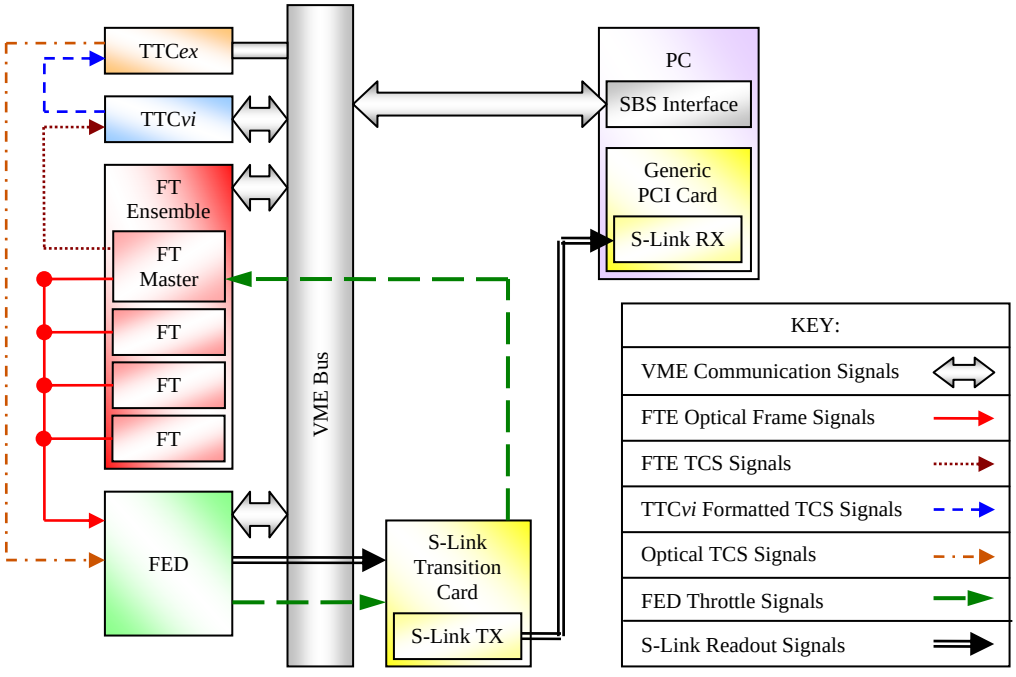


Figure 8.1: A block diagram of the hardware used for full FED system tests. The components and their operation are described in the text.

A block diagram of the hardware configuration required for full FED system tests is shown in Figure 8.1 (a photograph of the setup is given in Figure 8.2). The

experimental apparatus is similar to that used during the S-Link data integrity checks described in Chapter 7. As before, a FED Tester Ensemble provides the FED with controlled optical input signals; this FTE may contain any number of FED Testers but typically four are chosen, enabling all 96 channels of the FED to be populated. The Master FED Tester simulates the CMS Global Trigger Control System, generating L1 triggers and control signals. It also responds to the FED throttle.

In Chapter 7 the trigger and control signals were sent directly to the FED through a VME Backplane connection. At CMS, TCS commands are formatted by various TTC components and broadcast to each FED via an optical link (a description of the TCS system may be found in Chapter 3). Actual TTC hardware is therefore used in this instance to achieve a more accurate simulation of CMS operating conditions. The FT Master TCS signals are transmitted to a TTC*vi* (VME Bus Interface), where they are converted into A- and B-Channel data streams which are output to a TTC*ex* (Laser Encoder/TX[Transmitter]). This multiplexes the A- and B-Channel commands into a single optical signal that is finally transmitted to a TTC*rx* (Receiver) component [104] on the FED.

Data may be read out from the FED through VME, or via S-Link using the FED Kit hardware described in Chapter 7 (incorporating the new S-Link Transition Card). For S-Link data integrity checks, separate computers were used for the VME control and S-Link readout of the system. This maximised the data processing rate but was also a hardware limitation, caused by the differing voltage requirements of the FED Kit receiver and VME Bus Adapter cards. Full performance testing demands closer integration between control and readout, especially when attempting to compare arbitrary FTE input data with the processed output from the FED. Consequently, an SBS VME Bus Adapter PCI card was obtained which is compatible

with the PCI-X slots used by the FED Kit. All hardware access is now performed by a single computer.

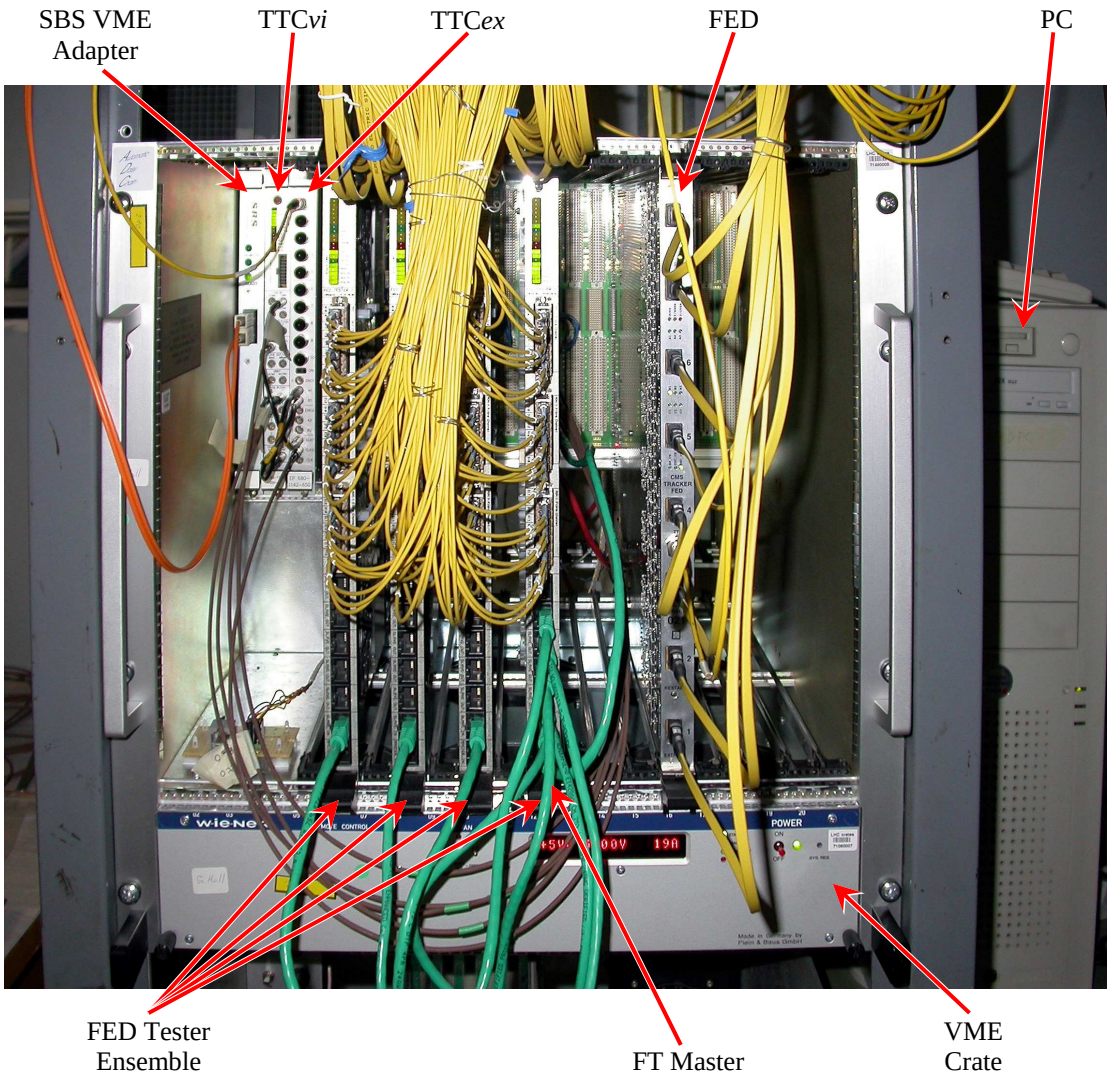


Figure 8.2: A photograph of the hardware used for full FED system tests (the S-Link Transition Card is connected to the rear of the VME crate, and cannot be seen).

8.2 A Unified Software Architecture for Full System Tests

The experimental apparatus required for full FED system tests is a complex arrangement of multiple hardware and software elements. Although most of the hardware had previously been employed for the S-Link data integrity tests described in Chapter 7, these made use of only a small subset of the available functionality; the primary aim in that case was simply to transmit a large quantity of data through an S-Link connection and check for erroneous bit values. Testing the performance of the

FED under arbitrary operating conditions is a significantly greater challenge, which requires complete and flexible control over every piece of hardware in the system.

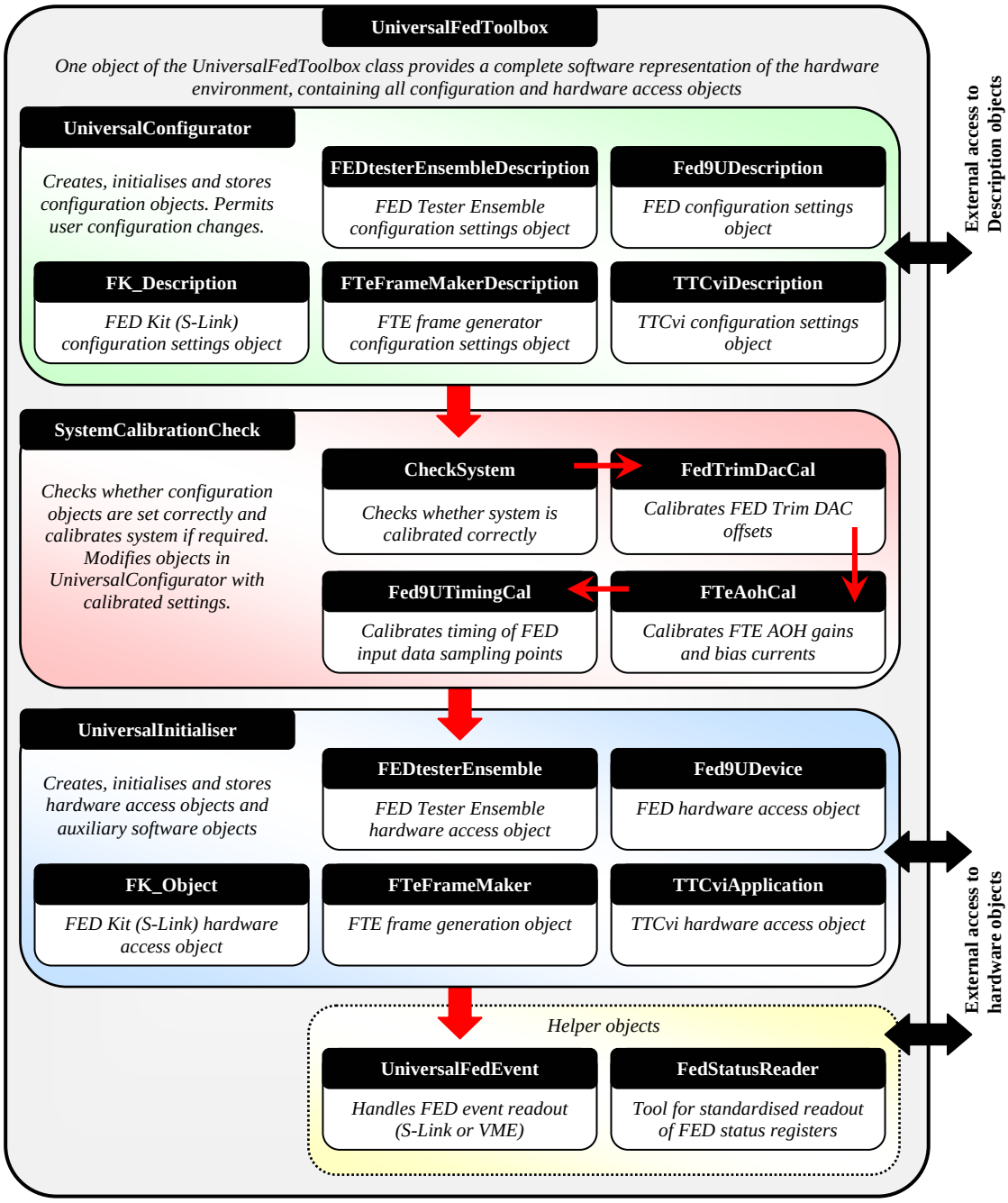


Figure 8.3: The structure of a UniversalFedToolbox class object, which forms the core of all FED performance test procedures. Each internal block corresponds to one object of a UniversalFedToolbox sub-class (class names are shown in white text). Red arrows indicate the sequence in which objects are instantiated.

It soon became apparent that merely configuring and accessing the various subsystems would grow into a cumbersome task if the software used to drive them were not well organised. A nested class structure was therefore developed in order to

manage all of the tools and hardware access objects required to control the test environment. Inbuilt routines enable the entire system to be initialised with a single function call and all software elements can be individually accessed through a simple layered interface. A diagram outlining the class architecture is shown in Figure 8.3. Wrapped at the highest level by a UniversalFedToolbox class, the software is divided into three main sections; system configuration, calibration and initialisation.

8.2.1 System Configuration

Configuring every item of hardware in the full FED system test setup requires a total of over 200,000 individual device parameters. To simplify the organisation and storage of this information, every configurable element in the test environment has an associated Description class (the FTE Description is described in Chapter 6). These enable settings for the FED, FTE, TTC*vi*, S-Link connection and artificial APV frame generator (discussed in Section 8.2.4) to be loaded from Description files and modified using simple function calls.

A UniversalConfigurator class wraps the assorted Description objects and provides an interactive user interface for changing the most commonly varied parameters; all changes are automatically written back to the input Description files, ensuring continuity between tests. The UniversalConfigurator object stores the current Descriptions in memory and provides direct access to them via pointers. They can be used by any class object within the UniversalFedToolbox to initialise the hardware they describe and may also be read externally, in the event that the system settings are required for decision making in higher-level test procedures.

8.2.2 System Calibration

After the Description files have been loaded, and any changes to the settings have been made by the user, it is necessary to check whether the system is correctly

calibrated. A number of factors influence the optical signals transmitted from the FTE to the FED. The initial light intensity depends upon the FT AOH bias currents and gains (see Chapter 6), but attenuation is caused by imperfect optical connections and the radius of curvature of each optical fibre. To guarantee measurements of the highest resolution, the intensity of each optical signal entering the FED must be tuned such that it occupies as much of the FE ADC range as possible (leaving a small buffer to absorb signal drift). It is also important to ensure that attenuation does not reduce the magnitude of the APV digital headers to a level below the user-defined digital high threshold, as this will cause a failure of the FED frame finding logic (described in Chapter 9).

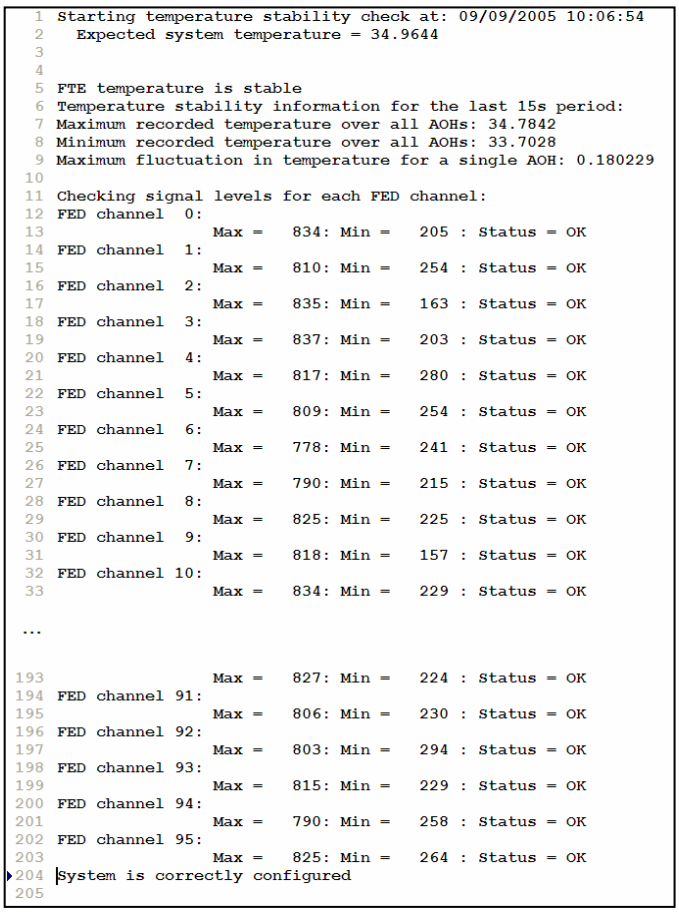
Subtle differences between the individual AOH components and the lengths of their optical fibres can additionally lead to a spread in the arrival times of multiplexed APV frames. If the timing variations for each channel are not compensated for by the FED Delay FPGAs then the sampling points of the FE FPGAs may occur at the rising or falling edges of data signals, causing levels to be read incorrectly.

System calibration is dealt with by a SystemCalibrationCheck class object. This wraps a number of sub-classes which automatically check the status of signals entering the FED with the current Description parameters and, if necessary, determine and apply any changes to the Descriptions that are required to accommodate laser intensity and frame timing variations. The calibration sequence is shown in the SystemCalibrationCheck block of Figure 8.3; its elements are described in the following sections.

8.2.2.1 Calibration Check

The calibration check is a simple procedure which the user is given the option to run immediately after the various object Descriptions have been configured. It initialises

the test hardware and waits for the AOHs on the FED Testers of the Ensemble to stabilise at the set point temperatures of the FT temperature control systems. This ensures that thermally induced signal fluctuations (discussed in Chapter 6) will be minimised during all subsequent operations.



```
Starting temperature stability check at: 09/09/2005 10:06:54
  Expected system temperature = 34.9644


FTE temperature is stable
Temperature stability information for the last 15s period:
Maximum recorded temperature over all AOHs: 34.7842
Minimum recorded temperature over all AOHs: 33.7028
Maximum fluctuation in temperature for a single AOH: 0.180229

Checking signal levels for each FED channel:
FED channel  0:
                Max =   834: Min =   205 : Status = OK
FED channel  1:
                Max =   810: Min =   254 : Status = OK
FED channel  2:
                Max =   835: Min =   163 : Status = OK
FED channel  3:
                Max =   837: Min =   203 : Status = OK
FED channel  4:
                Max =   817: Min =   280 : Status = OK
FED channel  5:
                Max =   809: Min =   254 : Status = OK
FED channel  6:
                Max =   778: Min =   241 : Status = OK
FED channel  7:
                Max =   790: Min =   215 : Status = OK
FED channel  8:
                Max =   825: Min =   225 : Status = OK
FED channel  9:
                Max =   818: Min =   157 : Status = OK
FED channel 10:
                Max =   834: Min =   229 : Status = OK

...


                Max =   827: Min =   224 : Status = OK
FED channel 91:
                Max =   806: Min =   230 : Status = OK
FED channel 92:
                Max =   803: Min =   294 : Status = OK
FED channel 93:
                Max =   815: Min =   229 : Status = OK
FED channel 94:
                Max =   790: Min =   258 : Status = OK
FED channel 95:
                Max =   825: Min =   264 : Status = OK
System is correctly configured
```

Figure 8.4: An example calibration check report. A number of FED channels are omitted here, for brevity.

Events are then read out from the FED in Scope mode, capturing standard APV tick marks from the FTE. For every event, all but one of the FTE channels are disabled; the enabled signal is applied to each FED channel in turn. Incorrectly calibrated channels are identified by comparing the tick mark digital high and low values (and their separation) with preset threshold limits. Applying data to one FED channel at a time additionally permits the detection of mapping errors between the FTE and the FED. If it is found that the system is improperly configured, details are

given in the generated status report and a request is made for a full calibration run. An example calibration check report is shown in Figure 8.4.

8.2.2.2 FED Trim DAC Calibration

The calibration sequence is divided into two sections, dealing with optical signal intensity and APV frame timing delays. It is necessary to adjust the signal intensity levels before attempting to compensate for the variation in frame arrival times; incorrect data sampling points cannot be accurately identified unless the raw data amplitudes occupy a well defined region of the FED input ADC range.

The signal level calibration is itself undertaken in two stages, the first of which occurs at the point where frames are received. Signals input to the FED may be shifted across the full range of the FE ADCs through the use of the on board Trim DACs. In the absence of an optical input, the baselines of all FED channels should occur at approximately the same level, near the lower limit of the FE ADC range. A function for calibrating the Trim DAC offsets is included in the Fed9USoftware [78] library. With the FTE outputs disabled, this operates by reading Scope mode events from the FED whilst modifying the Trim DAC value for each channel until the average received signal level is between 30 and 60 ADC counts.

At CMS it is a requirement for the FED to read a non-zero ADC value when no input is connected, which has the disadvantage of slightly reducing the ADC range available for data measurements. As no such requirement is necessary during testing, the Trim DAC offsets generated by the Fed9USoftware routine are shifted by a fixed value across all channels to ensure that the FED baseline is at a level below the ADC limit. Thus, the full range of the FED input ADCs is accessible through adjustments to the FTE output intensity, increasing the maximum possible resolution of test measurements and providing greater flexibility for the calibration of the FTE AOHs.

8.2.2.3 FTE AOH Calibration

Once the Trim DAC offsets have been configured correctly and the channel baselines are known, the FED can be used as a measuring device to enable the calibration of the output stage of the FTE. The AOH bias current is proportional to the offset of the output optical signal, while the gain determines the relative magnitude of the signal offset and range. Calibrating the FTE AOHs consists of driving the FED with 'DC' optical signals at the centre, minimum and maximum of the FTE output DAC range and adjusting the gain and bias current until the received levels coincide with the matching regions of the FED input ADC.

For each FED channel, the AOH gain is initially set to its minimum value and the bias current is calibrated. The FTE is loaded with APV data that is constant for every strip, at the mid point of the FT DAC, and the corresponding optical signal is transmitted to the FED. The use of fixed levels ensures that the calibration is valid regardless of whether the frames are timed in correctly. An event is read out from the FED in Scope mode and the average channel value calculated. This should be equal to the midpoint of the FED ADC range (512); if it is greater or smaller then the bias current is accordingly decremented or incremented by one register value and another event is read out and analysed. The process is repeated until the channel average is as close to 512 as possible (if a tolerance limit is exceeded for all bias currents, the channel is flagged and disabled).

Once an acceptable bias current is found, the gain setting is checked. Constant APV data at first the maximum and then the minimum of the FT DAC are loaded into the FTE and output to the FED, events are again read out in Scope mode and the channel averages are calculated. If the maximum and minimum APV values are too close to the limits of the FED ADC range (a small region must be reserved to prevent

the FED ADC from saturating should the AOH intensity fluctuate during a run) then the offending channel is identified and disabled. Calibration is considered complete when the difference between the maximum and minimum values is between ~50-80% of the FED ADC range; if it is less, the gain setting is incremented and the bias calibration and gain checks are repeated. If it is greater, or the limit of the AOH gain has been reached, the channel is again flagged and disabled.

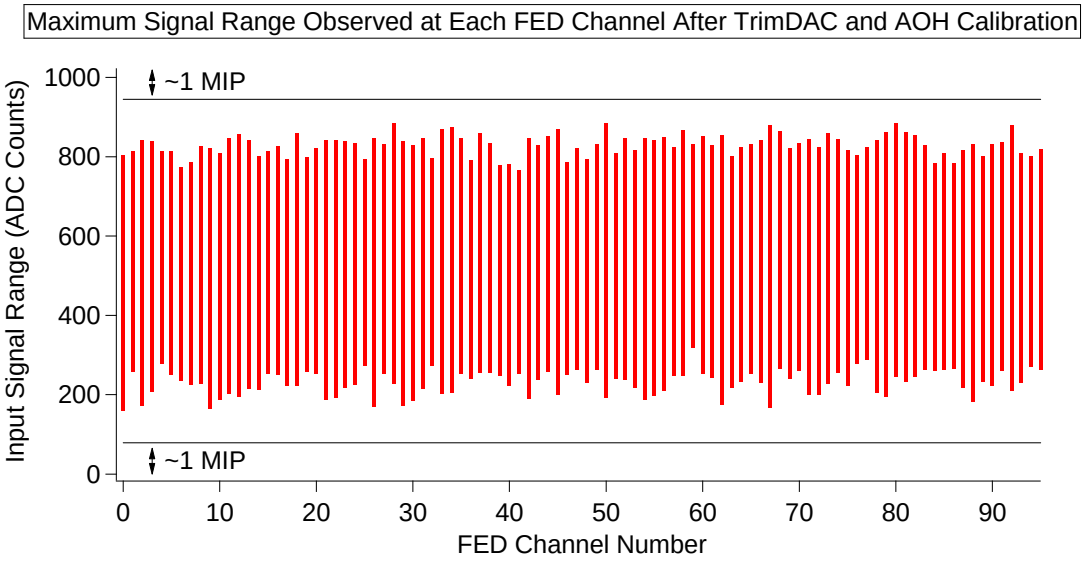


Figure 8.5: An example of the range of the optical signals received by the FED following Trim DAC and AOH calibration. Each vertical bar represents the region of the FED input ADC range occupied by the digital header of a typical APV frame.

The process is repeated for every connected FED channel. Finally, the Description initially used to configure the FTE is updated with the calibrated AOH bias currents and gains, and enable states in the event that any channels are found to be faulty. A plot showing the typical range of optical signals received by the FED after Trim DAC and AOH calibration is given in Figure 8.5. In this example the mean magnitude (upper limit minus lower limit) of the APV digital headers over all channels is 600 ± 48 ADC counts, or 7.6 ± 0.6 MIPs (quoted errors: standard deviation), with an ADC range equivalent to at least ±2 MIPs available for absorbing signal drifts that may occur during measurements.

8.2.2.4 FED Timing Calibration

With the intensity levels of the optical signals correctly calibrated, it is finally possible to adjust for the variation in their arrival time between channels. The FED Delay FPGAs allow signals entering the FED to be delayed by fractional and whole clock cycle intervals (see Chapter 4); these fine and coarse delay shifts are calibrated in two stages.

Throughout the timing run, the FTE drives the FED with standard APV tick marks. The FED is placed in Scope mode, with a scope length of sufficient samples to ensure that at least one tick mark will always be captured. Calibration requires the analysis of signal features that vary over a smaller time scale than the sample period of the FED (25 ns). Consequently, an event is read out from the board at each of the fine delay settings and the resultant frame data are interleaved to generate high (temporal) resolution tick marks for all channels.

By scanning through the interleaved event data and applying digital logic high and low boundary threshold limits, the software is able to find the first tick mark rising edge for each FED channel. Gradient, monotonicity and adjacent point spacing checks are carried out on the rising edge data fragments in order to identify potential errors; if the tick mark for a channel is valid then the correct fine delay setting is chosen to be the value that places a FED sample point 80% of the fine delay range after the start of the rising edge. This provides the optimum balance between data setup and hold times.

The Delay FPGA is able to skew data at nominal intervals of 1/32 LHC clock cycles. However, the initial timing calibration data revealed a bug in the FED hardware which causes an overlap between data offsets for several fine delay settings, as shown in Figure 8.6. Consequently, the FED fine delay was remapped in software

to a reduced number of 25 steps, which cover a one clock cycle range with greater uniformity.

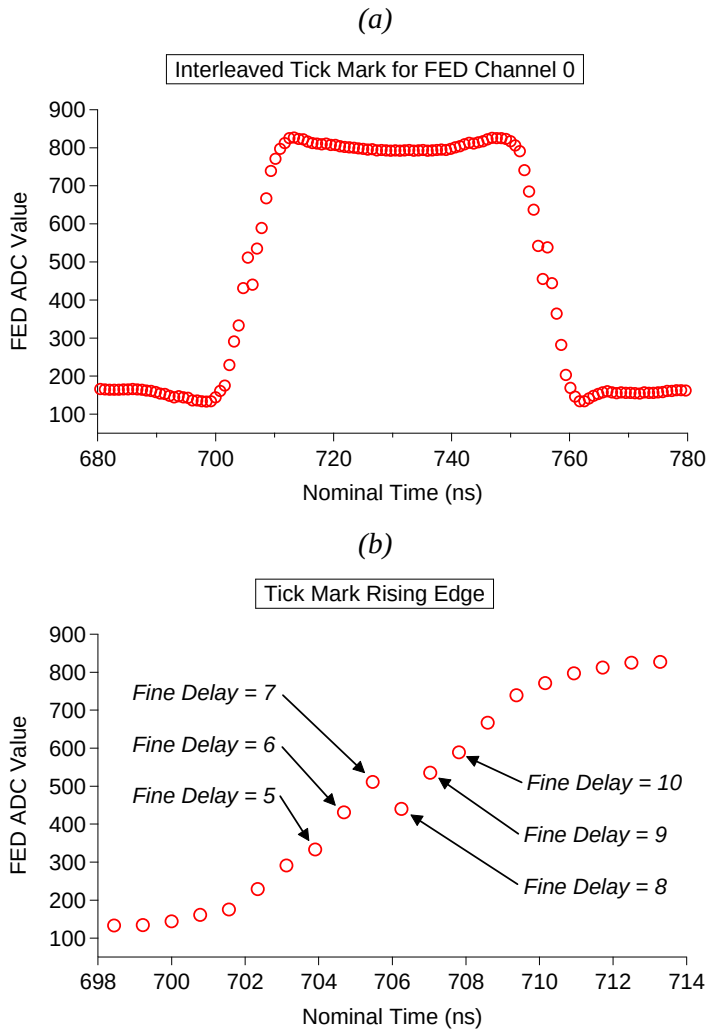


Figure 8.6: A demonstration of the FED fine delay timing bug. (a) shows an entire interleaved tick mark, while (b) shows a close up of the tick mark rising edge. It can be seen that the FED sample points at fine delay settings of 6 and 7 occur at almost the same locations as those at fine delay settings of 8 and 9.

Once the correct fine delay settings have been found and applied, the coarse delay is calibrated. A further event is read out from the FED and the first tick mark in each data channel is identified. The coarse delay value for a channel is then simply the offset, in whole clock cycles, between this tick and the corresponding mark in channel 0. If the offset is greater than the 16 clock cycle range supported by the Delay FPGAs then the offending channel is flagged with an error message.

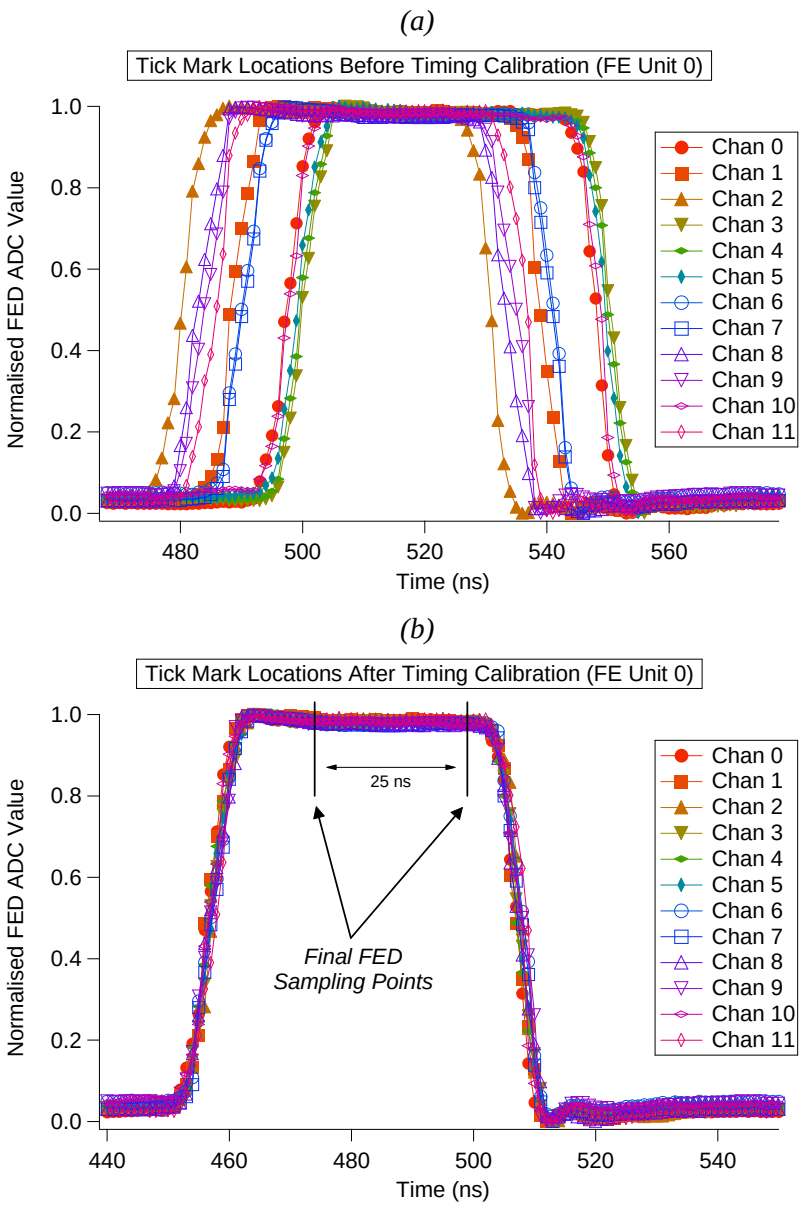


Figure 8.7: Interleaved tick marks obtained from all 12 channels of one FED FE Unit. The plots show the typical relative temporal offsets between tick marks (a) before and (b) after the FED timing calibration has been performed. (b) also shows the final locations of the 40 MHz FED sampling points after calibration is complete. They are positioned for optimal data setup and hold times.

At the end of the calibration procedure the FED Description is updated with the new delay settings and all received event data are written to a series of output files; these provide a useful mechanism for verifying the functionality of the Delay FPGAs. Examples of the typical (interleaved) tick marks recorded from one FE Unit of a FED before and after calibration are shown in Figure 8.7. Once calibration is complete, the signals input to the FED are shifted such that they all arrive at the FE

FPGA within an interval of ~2 ns. They are also given a common offset relative to the FED sample point to ensure ~20 ns of settling time before data are recorded after any signal transition.

8.2.3 System Initialisation

A UniversalInitialiser class wraps the individual hardware and software access objects required to communicate with the FTE, FED, TTC*vi*, S-Link receiver and artificial APV frame generator. The objects are constructed and initialised using the calibrated Descriptions stored in the UniversalConfigurator. All instantiated devices are held in memory, with direct access to their class member functions enabled via pointers. These are passed though the UniversalFedToolbox layer in order to provide a convenient tiered interface to the hardware, greatly simplifying the preparation and use of the full FED system test apparatus in any external test procedure.

8.2.4 Simulated APV Frame Generation

The FTE is able to produce optical signals that mimic the output of the CMS Tracker. However, it can only transmit the APV data that have been loaded into its internal memories; if the FED is to be tested under realistic CMS operating conditions, appropriate raw event data must be obtained. An FTeFrameMaker class was therefore written, to enable the random generation of realistic APV frames.

An FTeFrameMaker object produces FTE frame and frame pointer files (see Chapter 6) for a user-specified Tracker occupancy. It determines the number of unique frames that an FTE can store (a single FT holds 233) and randomly allocates a specific number of hits to each one, using a Poisson distribution, such that the average strip occupancy matches the requested value. The hits associated with a frame are given random magnitudes (between user-defined limits) and are placed at random strip locations; randomly-sized clusters of hits are formed according to an input

cluster width distribution profile, which may be derived from real or simulated Tracker data. Once pedestal offsets (with optional Gaussian noise) and digital headers have been added to the frames, they are randomly disordered and divided equally among the FTs in the Ensemble. A final step produces a randomised frame pointer list for each FT, ensuring that no multiplexed pairs of APV frames are repeated.

The FTeFrameMaker additionally extracts the FTE to FED channel mapping scheme from the current FEDtesterEnsembleDescription. Using this information it is able to read back the generated FTE frame and frame pointer files and reformat the APV strip data to match the structure of the expected FED event for any given L1 trigger. This enables the comparison of data input to the FED, via the FTE, with the processed output result, a feature that is essential for the verification of the FED hit finding algorithms (discussed in Section 8.3.3).

8.2.5 Helper Objects

Two helper objects are generated after the system is initialised, which may also be accessed via pointers through the UniversalFedToolbox. The first, a FedStatusReader, simply provides a user friendly interface for reading the FED status registers and allows a full status summary to be dumped to file should any irrecoverable errors occur. The second is a UniversalFedEvent object, an essential tool which transparently enables read out of the FED through either VME or S-Link. Choice of read out path is automatically determined from the input FED Description and the same access functions are used in either case. Methods are provided for efficiently converting data received via the S-Link into standard FED event objects and for formatting events read out via VME into the S-Link raw data format (useful for low level debugging). Full error checking and handling is supported through the use of low-level FED and S-Link software routines, with additional fast sanity checks

implemented for raw S-Link data (which can be executed at the maximum S-Link readout rate). Wrapping the Fed9USoftware methods for extracting strip information from FED events, a UniversalFedEvent is the only object required for performing read out of the FED in any full system test application.

8.3 Performance Tests

The UniversalFedToolbox provides the control and access framework for a comprehensive set of test procedures that verify the functionality of the FED hardware and firmware and characterise the performance of the board. Details of the testing methodology and results are given in the following Sections.

8.3.1 FED Channel Noise

In order to measure the typical noise introduced to an input data signal by the FED hardware, it is necessary to remove the noise component from the FTE by disabling all of the FED Tester outputs. As this causes the values sampled at each channel of the calibrated FED to fall below the minimum of the input ADC, the Trim DAC and Opto RX offsets are artificially set to constant levels, for all channels, such that the signal baseline occurs at approximately the centre of the ADC range. It should be noted that this is the only instance in which the calibrated Trim DAC offset values are ever overwritten during testing; an unbiased noise measurement requires all FED channels to be configured identically.

The FED is triggered to read out an event in Scope mode, with the maximum scope length of 1,020 samples used to increase the statistical accuracy of the results. Noise is calculated simply as the standard deviation of the recorded data. An example of the average (mean) measured signals and corresponding noise values for all 96 channels of a FED is shown in Figure 8.8.

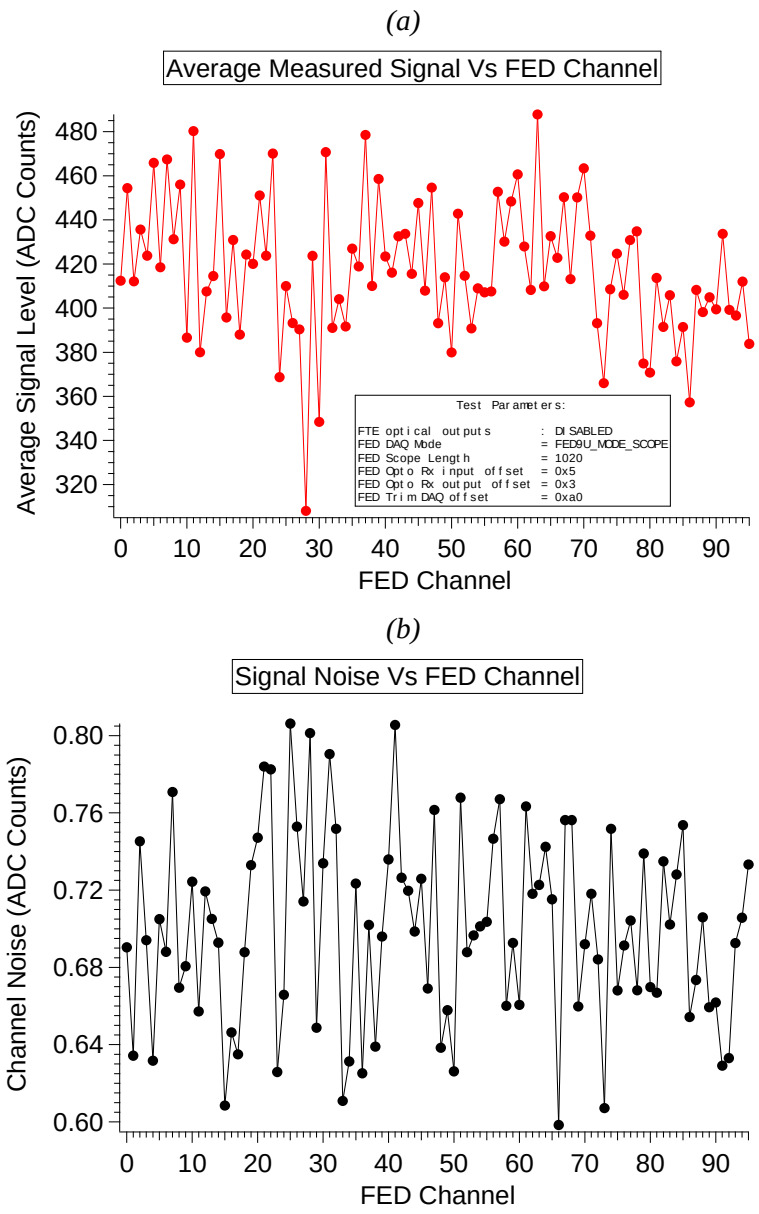


Figure 8.8: Plots showing (a) the average signal level recorded by the FED and (b) the calculated channel noise, both as a function of FED channel. The test parameters described in (a) also apply to the data set in (b).

From Figure 8.8(b), the mean channel noise is 0.70 ± 0.05 ADC counts (quoted error: standard deviation); noise for all channels is less than 0.81 ADC counts (similar values have been obtained from three other FEDs). For comparison, a MIP produces a signal with an ~80 ADC count magnitude. As a MIP is equivalent to ~25,000 electrons (for a 300 μm thick silicon detector), the FED therefore contributes a mean equivalent noise charge of ~220 electrons. This is close to the design value of

150 electrons [13] and is almost an order of magnitude less than the noise estimate for the complete CMS Tracker readout system. Consequently, the noise introduced by the FED is unlikely to have a significant effect on the input optical data.

8.3.2 FED Channel Crosstalk

Crosstalk between FED channels should be as small as possible; in addition to increasing signal noise, if a hit on one channel were to induce a response on another of large enough magnitude to be mistaken for a hit, the physics performance of the Tracker would be compromised. The inter-channel crosstalk among the analog readout paths of the FED Front End is measured by using the FTE to drive a number of the channels of one FE Unit with a multiplexed pair of empty ‘signal’ APV frames, whilst the remainder are provided with ‘noise’ frames that contain a user-definable hit object; the magnitude of any disturbance in a ‘signal’ frame at the location of the corresponding driving feature in a ‘noise’ frame may then be found.

As stated in Chapter 6, the FT is able to apply a fine delay to the output APV data in steps of ~100 ps. For each set of crosstalk frames, events are read out from the FED in Scope mode at ~250 FTE fine delay settings from 0 ns to 25 ns (one LHC clock period). The resultant event data are interleaved in order to generate high resolution images of the sampled frames, enabling crosstalk effects to be examined in greater detail.

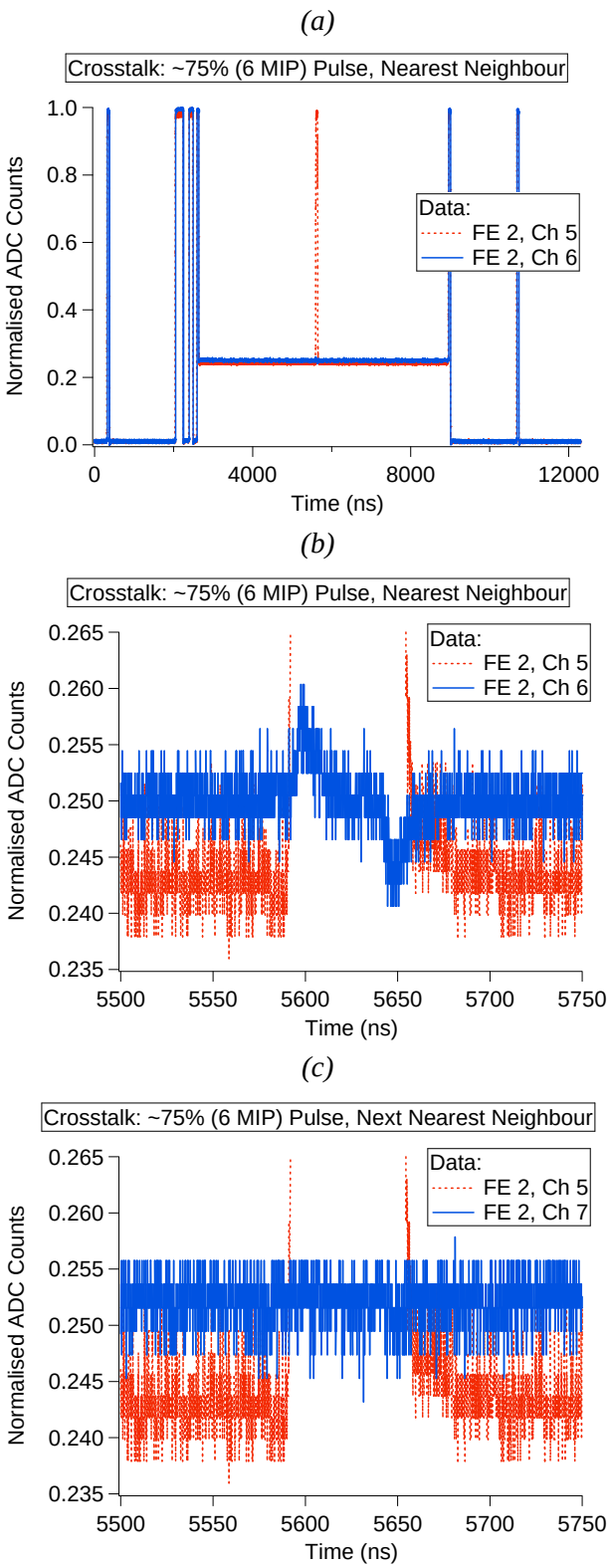


Figure 8.9: Inter-channel crosstalk caused by a ~6 MIP 'noise' feature on one channel of a FED FE Unit. (a) shows a full view of the signals received by the 'noisy' channel (FE 2, Ch5) and its nearest neighbour (FE 2, Ch 6). (b) is a close up of the signal received by the nearest neighbour channel at the location of the 'noise' feature. (c) is a close up of the signal received by the next nearest neighbour (FE 2, Ch 7) to the 'noise' channel, at the location of the 'noise' feature.

Initially, empty ‘signal’ frames were sent to all 12 channels of each FE Unit apart from channel 5; this was exposed to multiplexed pairs of ‘noise’ frames containing a central two-strip-wide hit feature with a magnitude equal to ~75% of the APV digital header range (~6 MIPs, the maximum expected range of physics data). Normalised interleaved event data from channels 5 and 6 of FE Unit 2 are shown in Figure 8.9(a), with the hit area expanded in Figure 8.9(b). In this case, the nearest neighbour crosstalk has a maximum excursion of ~0.79% of the digital header range, corresponding to ~0.06 MIPs or ~5 FED ADC counts. Figure 8.9(c) shows data from FE Unit 2 channel 7, the next nearest neighbour to the ‘noise’ frame, demonstrating that crosstalk effects are not significant beyond nearest neighbours.

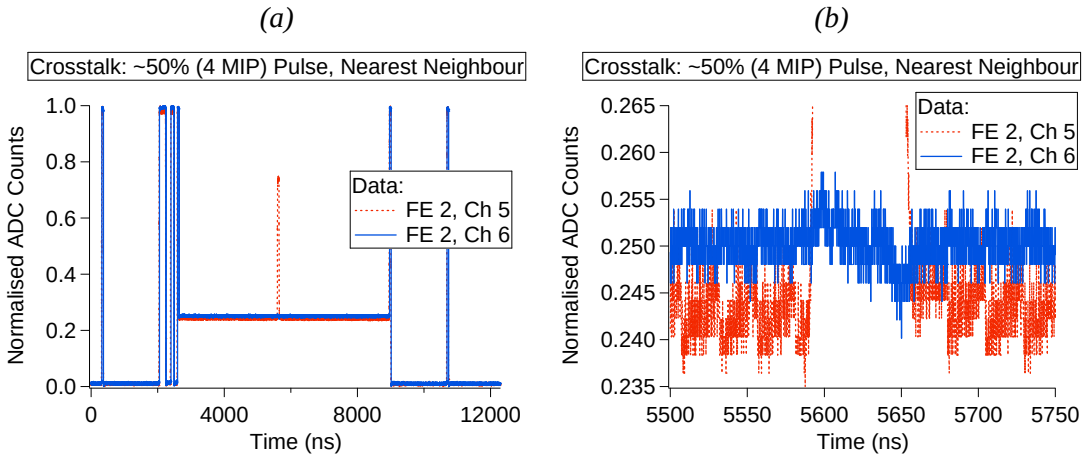


Figure 8.10: Inter-channel crosstalk caused by a ~4 MIP ‘noise’ feature on one channel of a FED FE Unit. (a) shows a full view of the signals received by the ‘noisy’ channel (FE 2, Ch5) and its nearest neighbour (FE 2, Ch 6). (b) is a close up of the signal received by the nearest neighbour channel at the location of the ‘noise’ feature.

The test was repeated with reduced ‘noise’ hit magnitudes of ~50% (~4 MIPs) and 25% (~2 MIPs) of the APV digital header range, and results obtained from FE Unit 2 are given in Figures 8.10 and 8.11. Measured over all FE Units, an isolated hit received by a single FED channel induces crosstalk on the nearest neighbouring channels with an amplitude equal to 1.4 ± 0.4 % (quoted error: standard deviation) of the hit magnitude (a second FED has also exhibited similar behaviour). Crosstalk

should therefore have little effect on the performance of the FED for typical physics data.

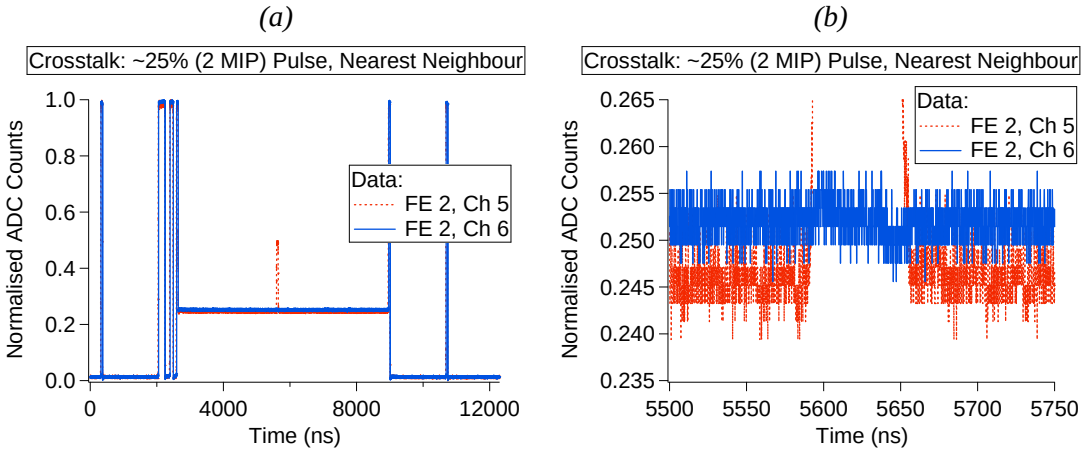


Figure 8.11: Inter-channel crosstalk caused by a ~2 MIP 'noise' feature on one channel of a FED FE Unit. (a) shows a full view of the signals received by the 'noisy' channel (FE 2, Ch5) and its nearest neighbour (FE 2, Ch 6). (b) is a close up of the signal received by the nearest neighbour channel at the location of the 'noise' feature.

A further measurement was made in order to examine the worst case example of crosstalk, when data arriving at one channel of a FE Unit are out of phase with the remaining channels; if the difference in arrival times is sufficiently large then the digital headers of eleven frames will overlap the analog data region of the twelfth. 'Noise' frames were sent to all of the FED channels with the exception of channel 5 of each FE Unit, which received the standard empty 'signal' frame. The 'noise' frames were delayed relative to the 'signal' by 10 LHC clock cycles and sampled events for channels 5 and 6 of FE Unit 0 are shown in Figure 8.12.

With effective hits of ~8 MIP magnitude on all neighbouring channels, the induced crosstalk on channel 5 has an amplitude of over 2% of the digital header range or ~13 ADC counts. This is approximately an order of magnitude greater than the FED channel noise and could potentially lead to the misidentification of hits in the 'signal' frame. It is therefore important to always achieve correct frame timing, by ensuring that the FED optical fibres are of similar length and by using the calibration

sequence described in Section 8.2.2.4. Optical links in the final CMS Tracker readout system will have lengths with a maximum deviation of no more than 5 m, leading to a variation between APV frame arrival times of ±12.5 ns [105]; this is well within the range that can be compensated for by the FED Delay FPGAs.

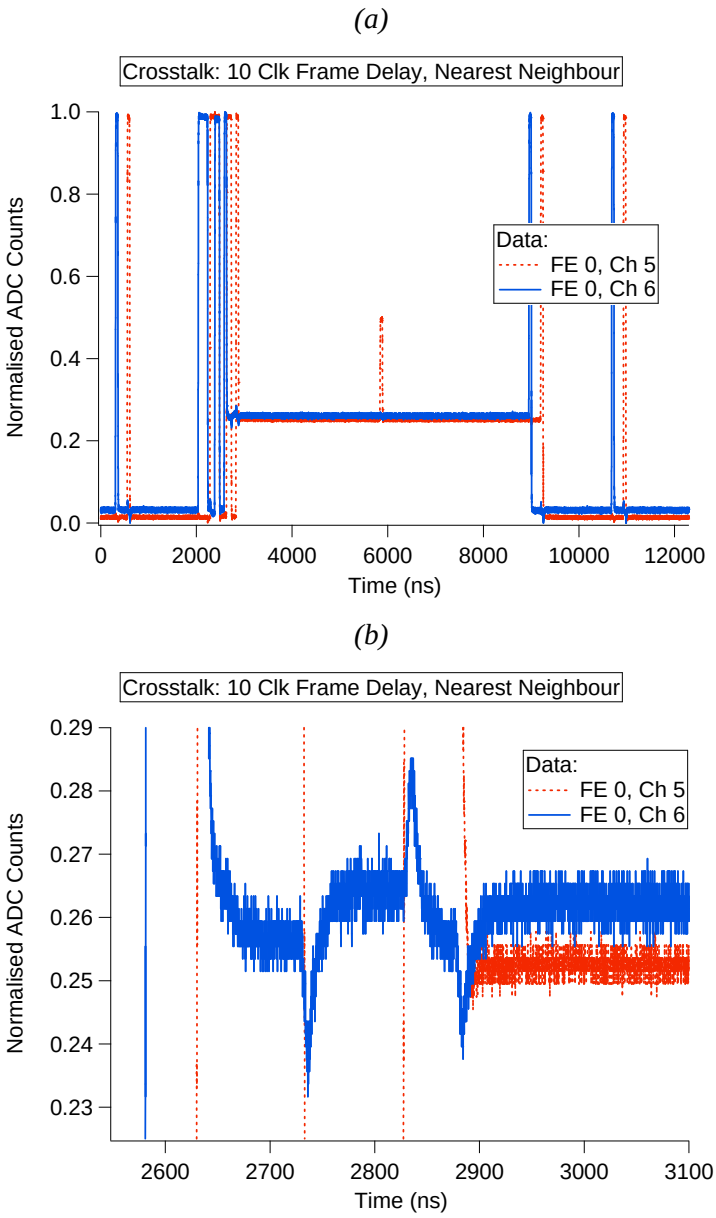


Figure 8.12: Inter-channel crosstalk due to the signals at one FED channel arriving out of phase with the others. A 'Signal' frame was transmitted to channel 6 of FED FE Unit 0, while 'noise' frames were sent to the remaining channels after a delay of 10 LHC clock cycles. (a) shows a full view of the 'signal' frame (FE 0, Ch 6) and the 'noise' frame received by the nearest neighbouring channel (FE 0, Ch 5). (b) is a close up of the 'signal' frame at the location where its analog payload region is overlapped by the digital headers of the 'noise' frames.

8.3.3 Evaluation of the FED Hit Finding Algorithms

It is vital that the cluster finding algorithms used in the FED FE FPGAs (described in Chapter 9) correctly identify all hits in the input Tracker data; if the algorithms were to fail, causing the FED to incorrectly add or remove hit information from the processed event stream, then all data from the Tracker would have to be discarded. Verifying the hit finding ability of the FED is only possible when the APV frames entering each channel are known; hence, it is only by using the FED Tester that such a check can be performed.

During a hit finding test the FED is operated under typical CMS conditions, in Zero Suppressed mode with a randomly distributed L1 trigger rate of 100 kHz and a maximum S-Link readout rate of 200 MB/s. The FTeFrameMaker is used to generate APV frames that contain hits at random strip locations with random heights that cover approximately the full range of the FTE output DACs (between a typical APV pedestal and a digital high).

Once these data have been loaded into the FTE, the 1,024 unique multiplexed APV frame pairs are output to the FED in a repeating cycle. Events are read out at full speed and discarded, until the Front and Back End buffer occupancies have reached normal operating levels. At this point, 1,024 successive events are captured from the FED and each one is compared with the corresponding input data from the FTeFrameMaker. With all of the unique frame combinations sampled, new frames are randomly generated and the process is repeated until the desired number of events have been acquired.

A scatter plot of the ADC values of each hit recorded by the FED against the matching DAC value output by the FTE for a single channel is shown in Figure 8.13;

100 cycles of 1,024 events were sampled during this run, corresponding to ~260,000 individual hits (and ~26,000,000 empty strips) per FED channel.

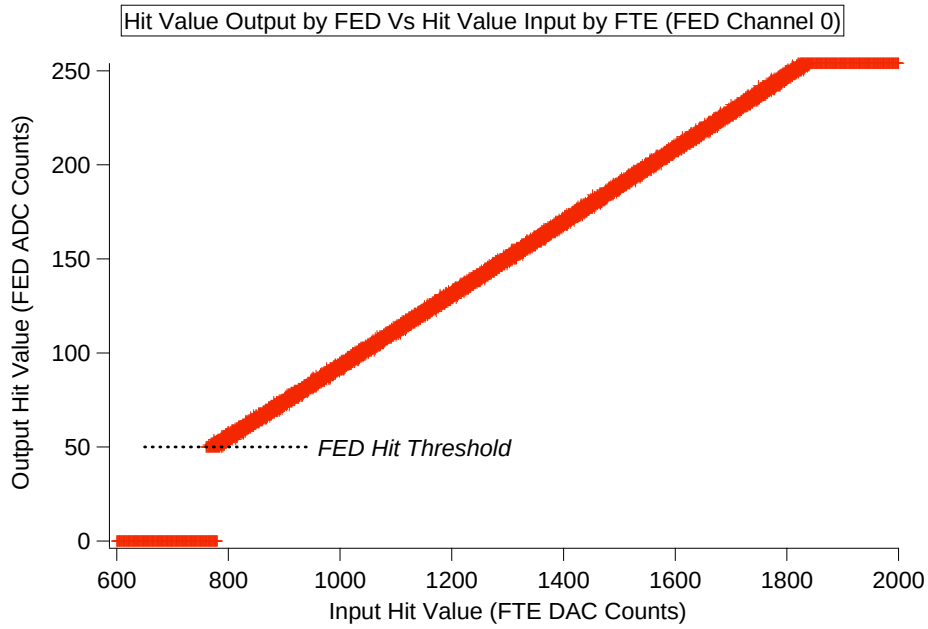


Figure 8.13: A scatter plot for FED channel 0 showing the ADC value of each hit signal identified by the FED against the corresponding DAC value output by the FTE (note that hit signals output by the FTE that are not identified by the FED are also included). The FED detects all input hits (with no false positives) apart from those with levels below the predefined 'hit threshold'.

The plot is divided into three distinct regions. Below a FED ADC count of 50 all hits are discarded, as this is the FED 'hit threshold' used during the test. Since data recorded by the FED in Zero Suppressed mode are truncated to 8 bits (see Chapter 9), the output of the FED becomes saturated for ADC values above 254. In between these points, the FED hit value increases linearly with the FTE hit value, demonstrating that all of the input hit signals are correctly identified (to within noise resolution); any discrepancies between the FED and FTE would be apparent as clear outliers from the linearly increasing line.

An intensity plot showing the sample distribution for a small section of this linearly increasing region is given in Figure 8.14. At each input FTE DAC value, the hit levels output by the FED have an approximately Gaussian distribution. The mean standard deviation over all output hit values is 1.25 ADC counts; this is greater than

the maximum of 0.81 ADC counts recorded in Section 8.4.1 due to the added noise contribution from the FTE.

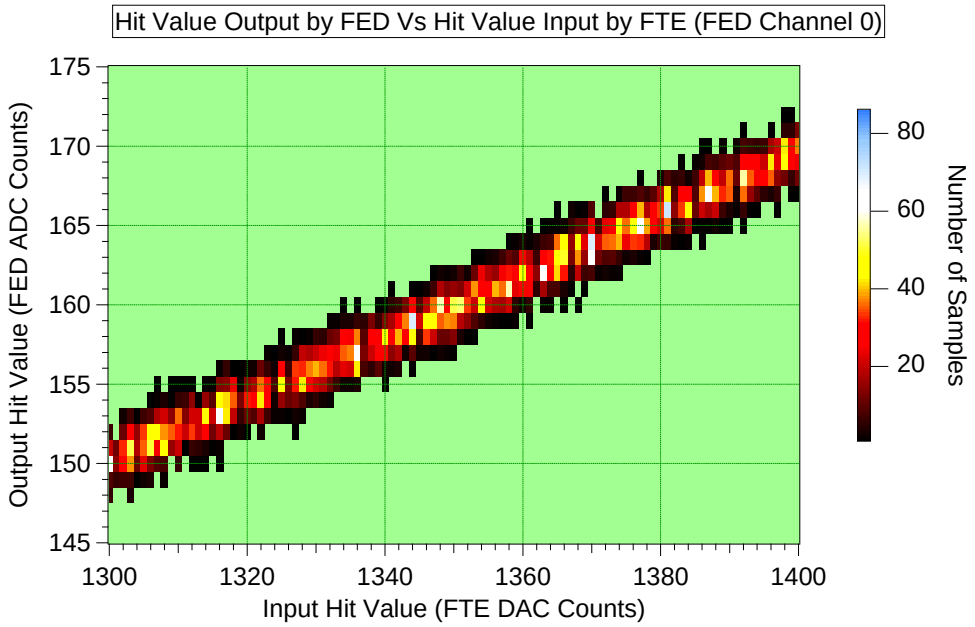


Figure 8.14: An intensity plot showing the ADC value of each hit signal identified by the FED against the corresponding DAC value output by the FTE, for a section of the linearly increasing region of the scatter plot in Figure 8.12.

8.3.4 Performance of the FED at High Data Rates

Another principal aim of the Laboratory tests is to characterise the performance of the FED at realistic CMS data rates. It is necessary to verify that the FED Front and Back End buffers are sufficient to prevent the vetoing of L1 triggers at expected Tracker occupancies, as predicted by the simulation discussed in Chapter 4.

For the purpose of this test, the FED is again operated under normal CMS conditions; events are read out in Zero Suppressed mode via S-Link, with a randomly distributed 100 kHz L1 trigger rate. In order to replicate CMS Tracker data, the FTeFrameMaker is configured to produce APV frames with a hit cluster width distribution that is derived from an ORCA simulation of high luminosity (10^{34} $cm^{-2}s^{-1}$) p-p events, described in Reference [106]. A histogram of the generated cluster width distribution is shown in Figure 8.15.

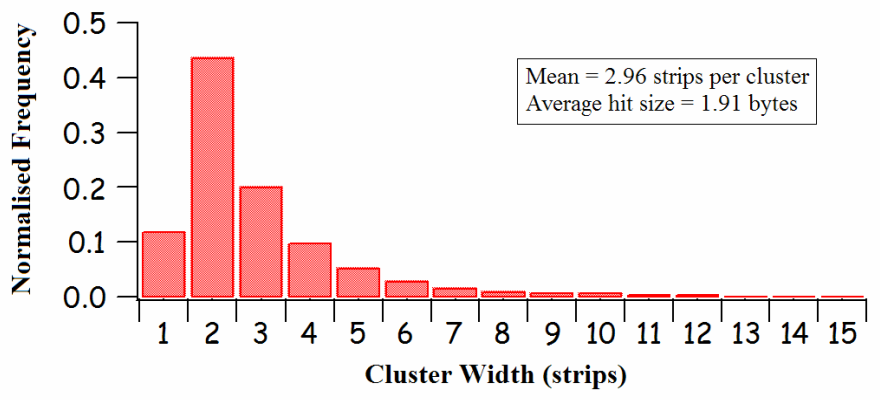


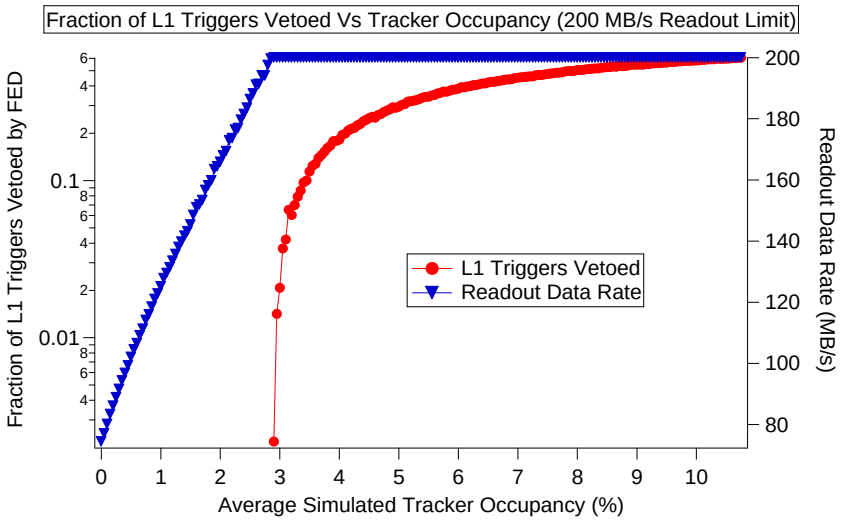
Figure 8.15: The cluster width distribution of simulated CMS Tracker APV frames generated by the FTeFrameMaker.

All 96 FED channels are driven with APV frames at a range of simulated average Tracker occupancies. At each occupancy step, appropriate frames are loaded into the FTE and the FED is triggered until typically 10^7 events have been read out. The data volume generated by the FED increases with Tracker occupancy; a software limit on the maximum S-Link readout rate is used to simulate the physical limitations of the CMS DAQ.

Plots of the fraction of L1 triggers vetoed by the FED as a function of Tracker occupancy are shown in Figure 8.16, for maximum S-Link readout rates of 200 MB/s and 400 MB/s. As predicted in Chapter 4, events are lost once the average data rate from the FED reaches the bandwidth of the S-Link connection.

At the nominal maximum 200 MB/s average sustained readout rate of the CMS DAQ, the FED is able to cope with an average occupancy of ~2.8 %. However, the DAQ supports temporary peak readout rates of up to 400 MB/s, provided that the average rate over timescales of a second remains at or below the nominal maximum. Figure 8.16(b) therefore demonstrates that the FED should be able to handle fluctuations in occupancy up to ~7%. This is well in excess of the current estimates of mean strip occupancies for proton-proton collisions at CMS, which are all under 3% [60].

(a)



(b)

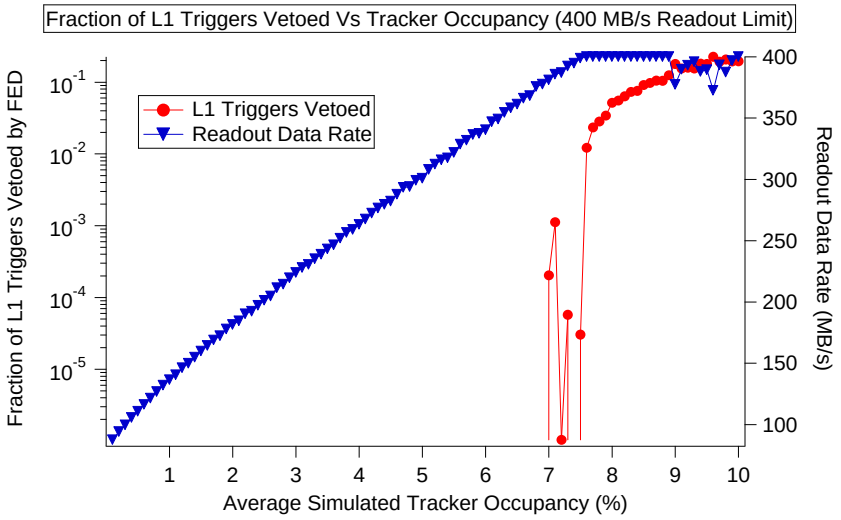


Figure 8.16: Plots of the fraction of L1 triggers vetoed by the FED against average simulated Tracker occupancy. The measured output data rate from the FED at each Tracker occupancy is also shown. Results are given for maximum S-Link readout rates of (a) 200 MB/s and (b) 400 MB/s.

It should be noted that the plots in Figure 8.16(b) are a little distorted at high occupancies, as the processing speed of the computer running the test can become an issue when reading out data at ~400 MB/s. This is an intensive, time-critical operation that requires virtually all of the system resources of the PC; any background processes have to 'steal' CPU time from the main data analysis thread in order to function. A number of software drivers also periodically access the PCI bus, momentarily reducing the bandwidth available to the S-Link receiver card. As a result, a very small

fraction of events are vetoed unnecessarily. The FED should therefore exhibit marginally better performance when it is read out by the final CMS DAQ system.

8.3.4.1 Future FED Performance at High Data Rates

The highest occupancy region of the Silicon Tracker will be layer 1 of the Tracker Inner Barrel, which is nearest the particle interaction point. During full-design-luminosity proton-proton collisions, it is predicted that the average strip occupancies of sensor modules in this area will be 2.8-2.9% [60]. According to an earlier Monte Carlo simulation of Tracker data rates (described in Reference [60]), the readout rate from a ‘full’ (180 out of 192 APVs occupied) TIB layer 1 FED at these occupancy levels is expected to be 109.5 MB/s. However, in Figure 8.16(b), the measured data rate at an occupancy of 2.9% is 217.8 MB/s. Much of this discrepancy is due to the fact that, as of writing, the final data format for events has not yet been implemented in the FED firmware.

When operating in Zero Suppressed mode, the FED packages strip hit values with excess status and structural information that is useful for error checking whilst developing the firmware but which will not be required at CMS. Currently, the formatting (non-data) sections of each event occupy 816 bytes, whereas the Monte Carlo simulation assumes 112 bytes; this more efficient format will be incorporated into a new ‘Zero Suppressed Lite’ mode of operation before data taking at CMS commences. Additionally, a variable number of padding bytes, typically an average of 32 per event, are required to byte align the clustered hit data leaving the FED; this also is not included in the Monte Carlo simulation and will be removed in the final version of the FED firmware.

Accounting for these two factors, and scaling for the 180/192 occupancy of the simulated FED, gives a ‘corrected’ measured data rate at a Tracker occupancy of

2.9% of 139 MB/s. The remaining discrepancy indicates that the cluster width distribution used in the simulation differs slightly from that generated by the FTeFrameMaker. However, even the potentially inflated data rate of 139 MB/s corresponds to only ~70% of the maximum sustained S-Link bandwidth for a single FED. Consequently, the Tracker readout hardware should be able to cope easily with occupancies in excess of the predicted maximum average levels at CMS.

8.3.4.2 Potential Performance of the FED at TOTEM data rates

TOTEM is an additional experiment planned to run at the LHC, which is dedicated to the measurement of total cross section, elastic scattering and diffractive processes. It is beyond the scope of this work to describe the TOTEM experiment (a full account may be found in References [107, 108]), but of relevance to CMS is that APV25 chips will provide the on-detector readout stage for part of the TOTEM GEM (Gas Electron Multiplier) Tracker. To further reduce costs and development time, the TOTEM group also intend to adopt the existing CMS FED.

One concern with the use of CMS electronics is that the expected average Tracker occupancy at TOTEM is relatively high, at approximately 16%. In Figure 8.16(b) it is shown that a FED is limited to an occupancy of ~7%, even with a 400 MB/s readout rate. This limit will increase with the future implementation of the Zero Suppressed Lite mode of operation, but constraints on the planning schedule for TOTEM required the FED to be demonstrated as a feasible choice for its read out system before such additions to the firmware could be made.

In order to support TOTEM occupancy levels it is therefore necessary to divide the Tracker output channels amongst a number of FEDs. If the 192 APVs normally associated with a single FED were instead read out by four FEDs, with half of the channels of each FED disabled and each channel reading a single APV instead

of a pair, then the effective occupancy would be reduced to 8%. However, it was initially unclear whether the FE firmware would be fully compatible with this configuration. The measurement of Section 8.3.4 was therefore repeated, but with the FTE driving only alternate FED channels. Plots of the fraction of L1 triggers vetoed as a function of Tracker occupancy for S-Link readout rate limits of 200 MB/s and 400 MB/s are shown in Figure 8.17.

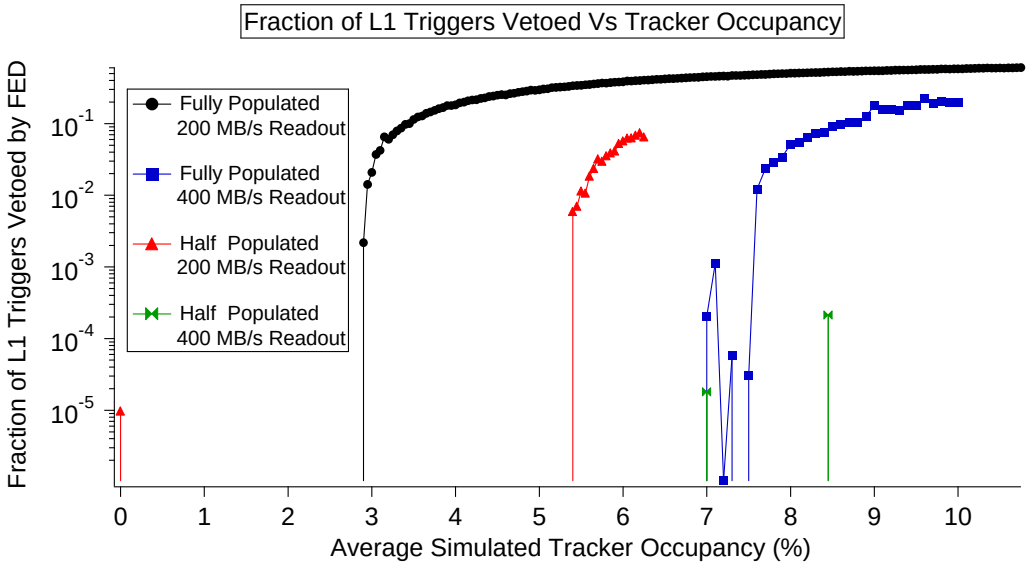


Figure 8.17: Plots of the fraction of L1 triggers vetoed by the FED against average simulated Tracker occupancy. Results are given for fully and half (alternate channels) populated FEDs, with S-Link readout rate limits of 200 MB/s and 400 MB/s.

With a sustained S-Link data rate of 200 MB/s, a half-populated FED is able to cope with an occupancy of 5.4% before events are lost. This is clearly insufficient for TOTEM readout, indicating that the TOTEM DAQ must support a higher readout rate than that of CMS. If TOTEM is able to maintain sustained readout at the peak CMS data rate of 400 MB/s, Figure 8.17 demonstrates that the FED would be able to operate with a Tracker occupancy of at least 10.8% (the limit of the FTeFrameMaker). It therefore seems that the FED is a suitable candidate for the TOTEM project; these results have been submitted to the group developing TOTEM and are currently being used in the planning of their readout system.

8.4 Post Production Acceptance Testing

The procedures developed to verify the fundamental design and performance of the FED may also be used for the routine post production acceptance testing of all boards received from the assembly plant, complementing the Industrial Acceptance Test described in Chapter 5. Each test is highly automated and generates comprehensive records of system settings and results, with a series of LabVIEW applications enabling the display of any required graphical output (the plots in Section 8.3 have been re-formatted using WaveMetrics Igor Pro [109] for clarity). Two routines in particular have been written for general purpose functionality tests, and provide a minimum requirement for post-production acceptance.

8.4.1 Optical Ramp Test

The primary (necessary) shortcoming of the Industrial Acceptance Test system is that it cannot supply the FED with an optical signal and so does not therefore verify the functionality of the photodiodes in the Opto RX components. This is readily checked in the Laboratory by a simple optical ramp test.

An FTE is used to drive the FED with APV frames whose pedestals consist of a ramp from digital low to digital high. One event is triggered and read out in each of the operating modes of the FED; Scope, Virgin Raw, Processed Raw and Zero Suppressed. As a result, the test provides an additional basic verification of the FED firmware in all these modes. A LabVIEW application takes the output events and calculates the gradient and noise of each ramp. Example results from an imperfect FED are shown in Figures 8.18 to 8.21.

The APV frames sampled by one FE Unit of the FED in Scope Mode are displayed in Figure 8.18. A scope length of 512 clock cycles, twice the length of a frame, is used to guarantee that the entire ramp is always captured.

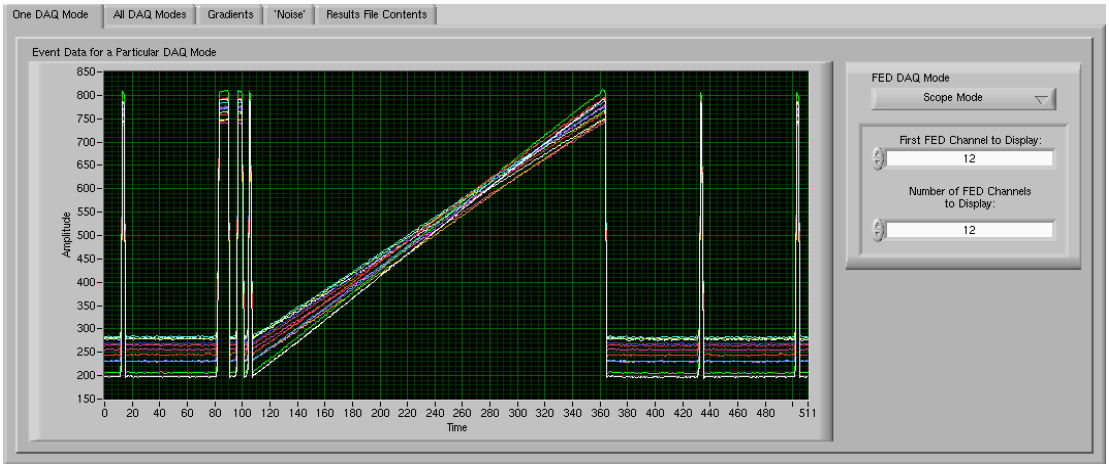


Figure 8.18: APV 'ramp' frames sampled from all 12 channels of one FED FE Unit, with the FED operating in Scope mode (scope length = 512). The picture is taken from the first panel of the LabVIEW-based optical ramp test results viewer.

Figure 8.19 shows the event data obtained from a working FED channel in all four operating modes. In each case the ramp signal is accurately reproduced, indicating that the Opto RX is functioning correctly; note that the Zero Suppressed ramp is truncated, due to Common Mode subtraction and the reduction of the ADC data width from 10 to 8 bits (see Chapter 9).

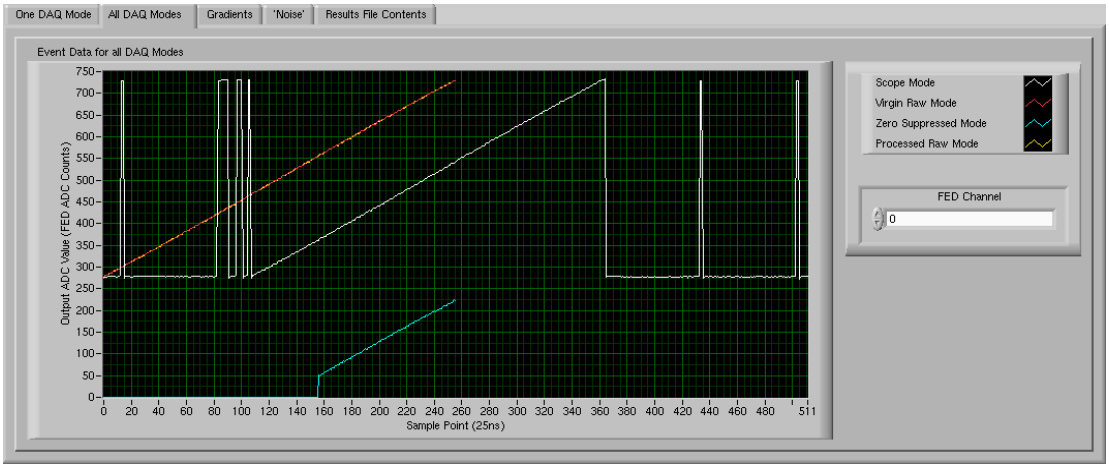


Figure 8.19: APV 'ramp' frames sampled from one FED channel in all four FED operating modes (white = Scope, red = Virgin Raw, yellow = Processed Raw, blue = Zero Suppressed). The picture is taken from the second panel of the LabVIEW-based optical ramp test results viewer.

The gradients of the ramps for all FED channels, displayed in Figure 8.20, are found by performing straight line fits to the linearly increasing regions of each frame. An effective noise value for each event is then determined by calculating the mean of

the residuals of the line fit, with results shown in Figure 8.21. Potential ‘bad’ channels are identified by outliers in the gradient plot and excessively high effective noise values (above 2-3 ADC counts).

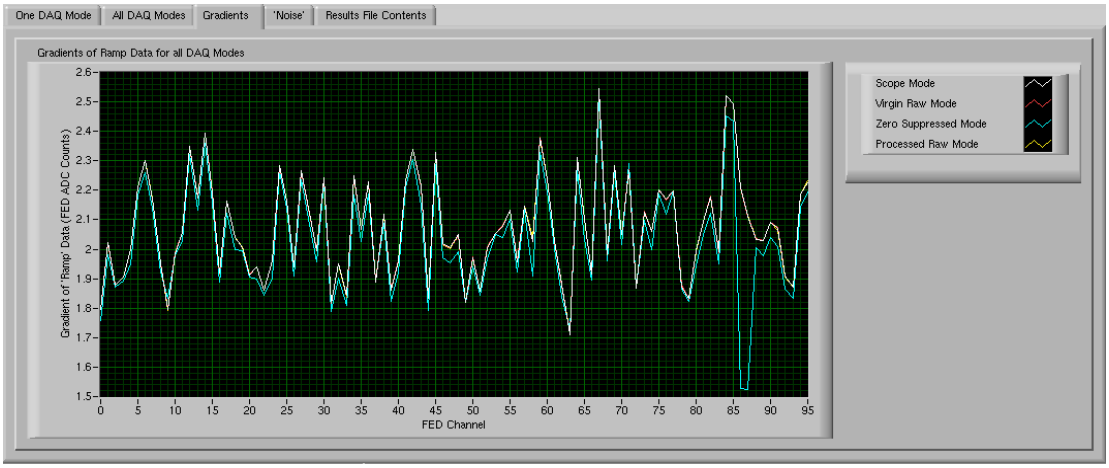


Figure 8.20: The gradients of the linearly increasing regions of the ramp frames received by all 96 FED channels in all four FED operating modes (white = Scope, red = Virgin Raw, yellow = Processed Raw, blue = Zero Suppressed). A problem on channel 86 is identified by a low gradient value. The picture is taken from the third panel of the LabVIEW-based optical ramp test results viewer.

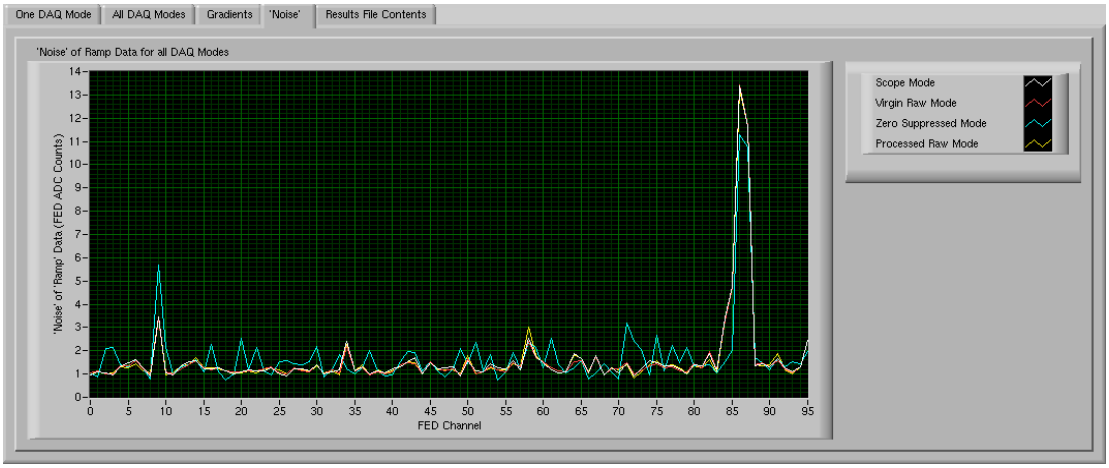
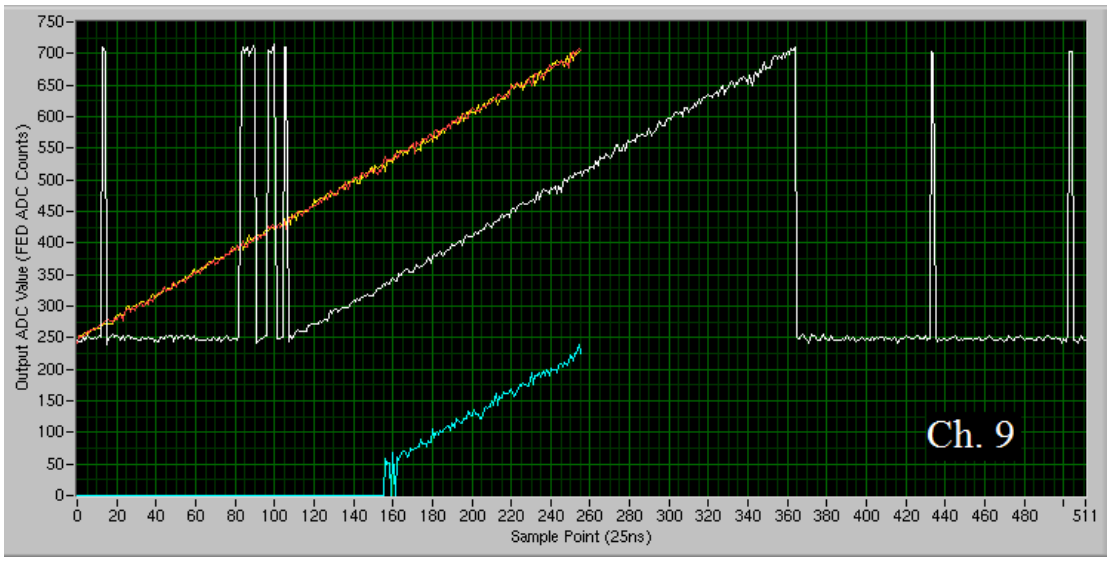


Figure 8.21: The effective noise of the linearly increasing regions of the ramp frames received by all 96 FED channels in all four FED operating modes (white = Scope, red = Virgin Raw, yellow = Processed Raw, blue = Zero Suppressed). Problems on channels 9 and 86 are identified by high noise values. The picture is taken from the fourth panel of the LabVIEW-based optical ramp test results viewer.

Ramps from two of the problematic channels highlighted by Figure 8.21 are given in Figure 8.22. In this example, the noise amplitude for channel 9 of the FED is approximately twice that of a normally functioning channel, equivalent to ~0.05

MIPs. Channel 86 has a large effective noise due to the fact that the corresponding Opto RX photodiode exhibits a non-linear response for high intensity optical signals. An experienced electrical engineer may examine these channels and attempt a repair if any faults are discovered or, in the worst case, replace the offending Opto RX modules.

(a)



(b)

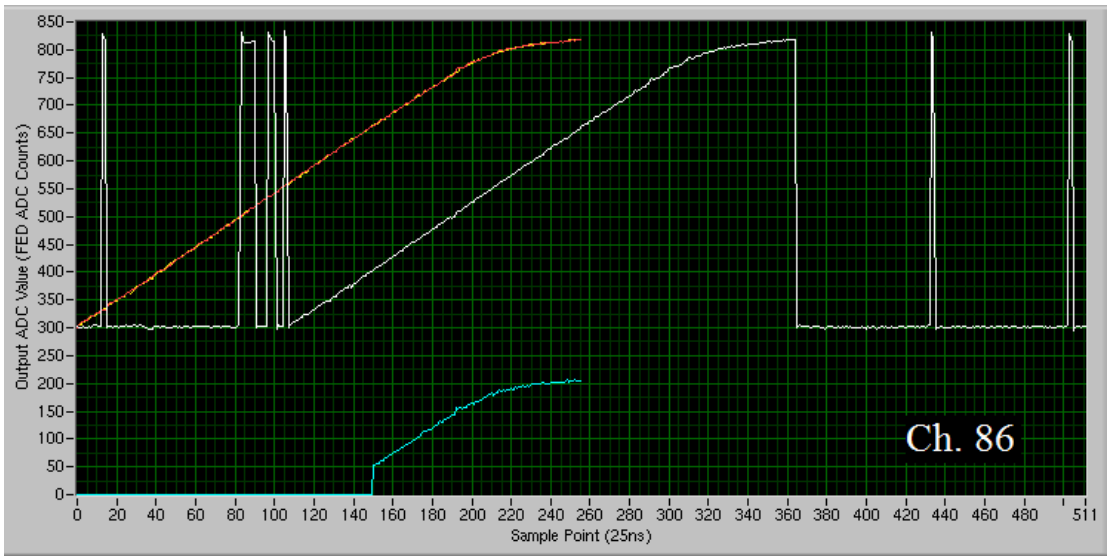


Figure 8.22: Example 'ramp' frames sampled from two problematic FED channels identified by the gradient and noise plots in Figures 8.20 and 8.21. Results are shown for all FED operating modes (white = Scope, red = Virgin Raw, yellow = Processed Raw, blue = Zero Suppressed). The frames in (a) exhibits an unusually high noise level while those in (b) demonstrate a non-linear response of the FED Opto RX to high intensity optical signals.

8.4.2 The Universal FED Readout Test

A ‘universal’ readout test provides comprehensive general verification of the FED functionality under arbitrary operating conditions. Supporting all possible system configuration parameters, it enables the FED to be driven with a specific number of frames at a user-definable Tracker occupancy and L1 trigger rate, with a controllable readout rate through S-Link or VME, two levels of output event structural integrity checks and optional real-time display of the FED status.

By default, the index and length of each event read out during a universal FED test are inspected, and an S-Link CRC error check is performed if S-Link readout is specified. These simple analysis steps require very little CPU time and enable the output from the FED to be processed at up to the maximum S-Link data rate. At the expense of a reduction in readout rate, a more rigorous examination of the events may be carried out; the raw data from the FED are used to construct a Fed9UEvent object (part of the Fed9Usoftware framework), the constructor of which extracts the formatting information of an event and throws an exception if any structural errors are found. A second CRC check is then conducted by the Fed9USoftware, for comparison with the S-Link hardware generated result.

Enhanced error detection typically reduces the readout rate to a level that is compatible with 1 kHz L1 triggers. Driving the FED under these conditions with 100 kHz triggers provides good verification of the FED throttling system. Counters keep track of the number of events that exhibit each type of error and the results are presented in a report at the end of the experimental run, together with dumps of the raw data from any problematic events. An example report file is shown in Figure 8.23.

```
********************
* General Settings *
********************


*** Current FED settings:

       FED - Read Route              = FED9U_ROUTE_SLINK64
       FED - DAQ Mode                = FED9U_MODE_ZERO_SUPPRESSED
       FED - Scope Length            = 6
       FED - Test Register           = NORMAL
       FED - Strip Pedestal          = 0
       FED - Strip High Threshold    = 50
       FED - Strip Low Threshold     = 50
       FED - Crate Number            = 0
       FED - Slot Number             = 16
       FED - Clock Source            = FED9U_CLOCK_TTC
       FED - Trigger Source          = FED9U_TRIG_TTC

*** Current FTE settings:

       FTE - Trigger State           = RANDOM
       FTE - L1A Period              = 400
       FTE - L1A Low Rate Period     = 40000
       FTE - L1A Random Rate         = 164
       FTE - Minimum Trigger Spacing = 3
       FTE - TCS Source              = LOCAL_SRC
       FTE - TCS Status Input Enable = ON

*** Current FTE FM settings:

       FTE FM - Cluster Definition   = MULTI_STRIP
       FTE FM - Cluster Widths File  = FTeFrameMakerFiles/FTe_CluWidthDistr_Simulation.txt
       FTE FM - Digital High Level   = 3600
       FTE FM - Digital Low Level    = 400
       FTE FM - Pedestal Level       = 1200
       FTE FM - Maximum Hit Level    = 2050
       FTE FM - Minimum Hit Level    = 1950
       FTE FM - Gaussian Noise Level = 0



######################################################################

Test 1:

  Initial settings:

    Number of triggers to send   = 100000
    Maximum readout rate (MB/s)  = 400
    Tracker occupancy percentage = 2
    Fed9UEvent error checking enabled
    System status checking enabled



Results For Test 1:

  Number of triggers sent                             = 100000
  Number of events received                           = 100000
  Time taken                                          = 30.54
  Data rate (MB/sec)                                  = 5.94505
  Fraction of samples for which throttle was enabled  = 0.98921

 Errors:

  Number of events with WC errors                     = 0
  Number of events with Event Number errors           = 0
  Number of events with Event Length errors           = 0
  Number of events with Slink CRC errors              = 0
  Number of events with Fed9U CRC errors              = 0
  Number of events with Fed9UEvent construction errors = 0



######################################################################

Test 2:

  Initial settings:

    Number of triggers to send   = 10000000
    Maximum readout rate (MB/s)  = 400
    Tracker occupancy percentage = 2
    Fed9UEvent error checking disabled
    System status checking disabled



Results For Test 2:

  Number of triggers sent                             = 10000000
  Number of events received                           = 10000000
  Time taken                                          = 100.21
  Data rate (MB/sec)                                  = 181.318

 Errors:

  Number of events with WC errors                     = 0
  Number of events with Event Number errors           = 0
  Number of events with Event Length errors           = 0
  Number of events with Slink CRC errors              = 0
```

Figure 8.23: An example report generated by the 'universal' FED readout test. The results from two tests are shown; in Test 1, 100,000 triggered FED events were analysed with full error checking and in Test 2, 10,000,000 triggered events were subjected to the less rigorous fast error checking. In both cases, no problems were found.

The universal readout test does not check the actual APV channel data content of the received events, but verifies all other aspects of the readout hardware functionality. It has been used extensively throughout the development of the FED firmware to identify and diagnose potential problems. These included a buffer indexing flaw which occasionally caused data from successive events to become merged when the FED applied a throttle signal and incorrect readout clock delays that lead to erroneous data sampling points and CRC errors in a small fraction of events. Since the firmware issues have been resolved, between 1×10^{11} and 1×10^{12} FED events have been analysed from over 30 different boards without receiving errors.

8.4.3 The Post Production Test Procedure

A FED that has passed the Industrial Acceptance Test must be subjected to the following Laboratory test procedure before it is deemed to be acceptable for use at CMS. The FED is placed in the full system test environment and the optical ramp test of Section 8.4.1 is performed, following a calibration run. Both the ramp test output and the calibration results, specifically the tick marks obtained form the timing calibration (see Section 8.2.2.4), are checked for anomalies.

Two passes of the universal readout test are then conducted, using 100 kHz randomly distributed L1 triggers, a simulated Tracker occupancy of 6% and full error checking; the first analyses 100,000 triggered event packets read out through VME (taking ~10 minutes) and the second verifies 1,000,000 triggered events read out via S-Link (requiring ~1 hour). It should be possible to complete the post production acceptance test for a single FED within ~90 minutes. Less than two months should therefore be required for the testing of all of the 450 FEDs that will be used at CMS.

8.5 Full Crate Tests

In addition to characterising the performance of individual FEDs it is important to verify that they function correctly when multiple boards are operated simultaneously within a single VME crate. At CMS, the rack layout has been organised such that each crate will hold up to 16 FEDs [110]; while the board has been designed to avoid any access conflicts, with power requirements that are within the LHC crate specifications, functionality cannot be completely guaranteed until a full crate of FEDs has been tested under realistic CMS conditions.

This procedure essentially involves the simulation of the entire Tracker readout system. Multiple FEDs must be driven by optical inputs from multiple FTEs, with independent event handling and synchronisation via a single TCS system. The complexity of the testing setup is thus increased by an order of magnitude.

It is possible to manage the system by splitting it into a number of almost independent sub-units, where each unit consists of a FED combined with its own individual set of driving and readout hardware. Ideally, each FED should be provided with optical inputs from a four-FED-Tester FTE; in practice, only 9 FT boards exist and so it is necessary to either split them into smaller Ensembles and partially populate FEDs, or run a number of FEDs in Scope mode with no optical signals. Similarly, all FEDs should be read out via S-Link, but the limited availability of FED Kits means that a mixture of S-Link and VME readout will be used for the initial Laboratory tests.

One sub-unit is equivalent to the standard single-FED full system test apparatus, with two principal exceptions; the number of hardware elements in each unit is variable and a single ‘Master Unit’ will provide the TCS source for all the others. A Master Unit must therefore consist of an FTE, a TTC*vi* and a TTC*ex*, and

may optionally contain a FED and a FED Kit. Other sub-units should contain at least a FED, with an additional FTE and/or a FED Kit depending upon hardware availability. The TTC*ex* distributes optical clock, trigger and control signals to the FED in each unit, whilst the FT Masters in each FTE are daisy chained together (as described in Chapter 6) in order to achieve synchronisation with the FTE in the Master Unit. A block diagram showing an example of the hardware configuration for a system containing five base units with a mixture of FED input and readout options is given in Figure 8.24.

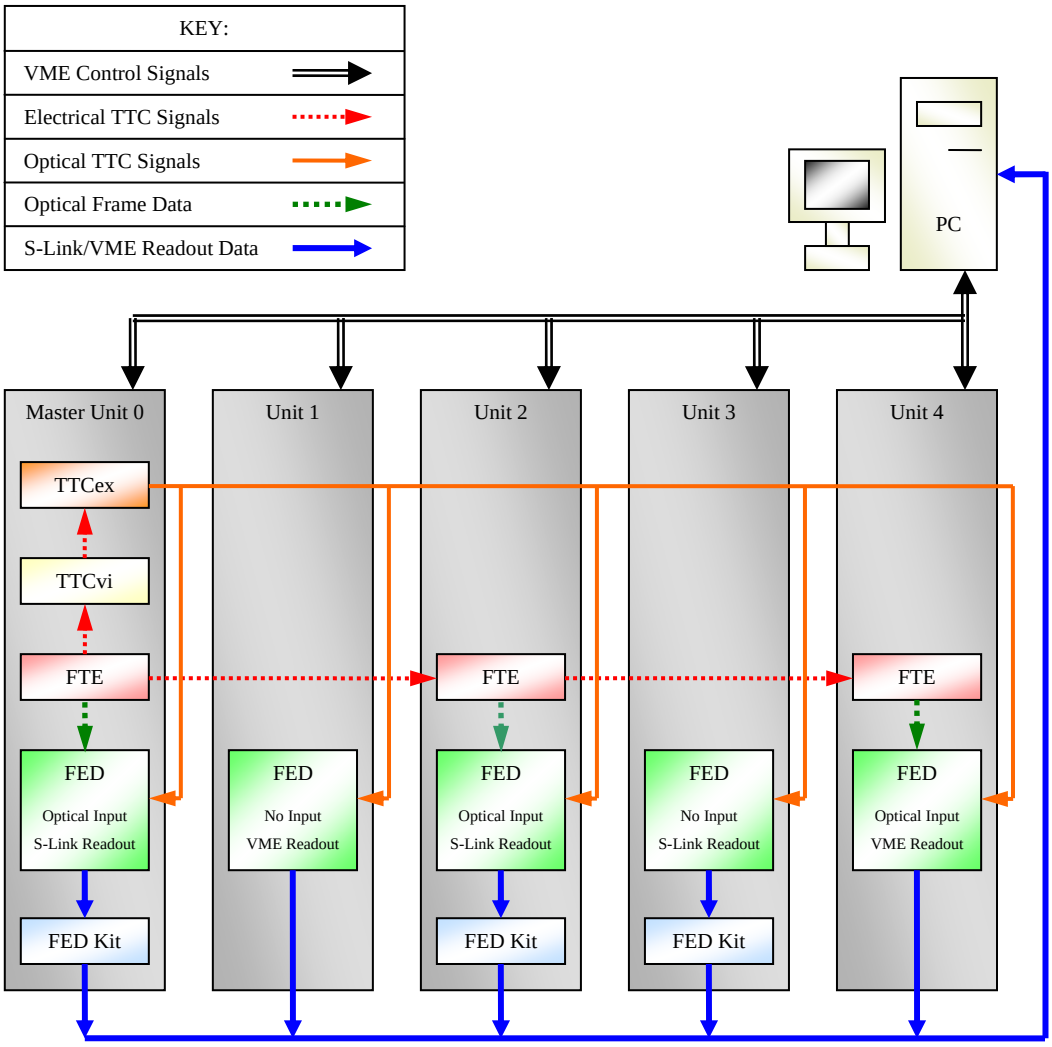


Figure 8.24: A block diagram of an example 'multiple FED test' setup. This system contains five FEDs; three are supplied with optical inputs from an FTE, while the others are operated in Scope mode with no input. A mixture of S-Link and VME readout is used.

The UniversalFedToolbox described in Section 8.2 is intended for the control and read out of a complete test system utilising a single FED. However, it was designed for maximum flexibility and reusability and as such is able to ‘intelligently’ configure itself depending upon the Description files that are passed to the internal UniversalConfigurator object. If a system does not contain a TTC*vi*, FTE, FED or FED Kit then it is sufficient to simply omit the corresponding Description files and the UniversalFedToolbox will automatically construct only the required hardware access and helper objects (for example, if there is no FTE or FED then it is not necessary to implement an FTeFrameMaker).

It is therefore possible to represent each unit of crate test hardware as a single UniversalFedToolbox object. Applying a similar technique to that used with the FED Tester software, the entire crate test system is then embodied by a top level UniversalFedToolboxEnsemble class, which contains an array of UniversalFedToolbox’s combined with mapping information about which unit is the Master (TCS source). As the final FMM hardware (see Chapter 3) used to merge the throttle signals from a number of FEDs is not yet available, it is also necessary to define a ‘Throttle Master’; this corresponds to the unit whose FED throttle signals are output to the Master TCS and, to prevent buffer overflow, is read out last at each L1 trigger.

The UniversalFedToolboxEnsemble is a powerful construct which supports any permutation of crate test hardware, with the same ease of configuration, calibration and readout (tuned to the items contained within each unit) that is found with the UniversalFedToolbox. It is limited only by the number of PCI slots that the host computer contains; the current PC has six slots, each of which may accept a FED

Kit receiver or a PCI VME Bus Adapter card (which in turn enables access to the 20 slots of a VME crate).

A ‘multiple FED test’ has been written using the UniversalFedToolboxEnsemble framework. This has exactly the same functionality as the universal readout test described in Section 8.4.2, but includes simple error checking between the events that are read out from each FED in order to verify board synchronisation. The event number and LHC bunch crossing numbers are compared, as too are the lengths of any events that are read out in Scope mode with the same scope length; if any mismatches are found, the raw data from each event are dumped to the automatically generated report file.

The potential of the multiple FED test software has not yet been fully exploited, as the test station situated at RAL (the only institute in the UK with sufficient boards to populate an entire crate) is in constant use for the post production acceptance testing of new FEDs. However, simple tests involving 2 FEDs driven in Zero Suppressed mode by 2 FTEs with mixed S-Link and VME readout, and 8 FEDs all read out in Scope mode via VME, have been conducted and no event errors or synchronisation problems have been observed.

For a crate containing 8 FEDs, events may be read out and checked for errors at a maximum rate of ~19 Hz. This is very low, caused primarily by the bandwidth limitations of the VME readout path, but is adequate for basic commissioning. It is perhaps interesting to note that the initial crate test using the final CMS DAQ to read out 10 FEDs in Scope mode through VME achieved a data rate of only ~7 events per second without any error checking (although this will improve immeasurably when the event filter farm is in place). An example report file generated by the multiple FED test is shown in Figure 8.25.

```
********************
* General Settings *
********************


--------------------------- System summary: ----------------------------

Number of UniversalFedToolboxEnsembles: 8
FTE TCS Master index                  : 0
FED Throttle Master index             : 1

  - UniversalFedToolbox   0 contains:

    FTE 0
    FED 0
    FTE Frame Maker 0
    Universal FED Event 0
    FED Status Reader 0

  - UniversalFedToolbox   1 contains:

    FED 1
    Universal FED Event 1
    FED Status Reader 1

  - UniversalFedToolbox   2 contains:

    FED 2
    Universal FED Event 2
    FED Status Reader 2

  - UniversalFedToolbox   3 contains:

    FED 3
    Universal FED Event 3
    FED Status Reader 3

  - UniversalFedToolbox   4 contains:

    FED 4
    Universal FED Event 4
    FED Status Reader 4

  - UniversalFedToolbox   5 contains:

    FED 5
    Universal FED Event 5
    FED Status Reader 5

  - UniversalFedToolbox   6 contains:

    FED 6
    Universal FED Event 6
    FED Status Reader 6

  - UniversalFedToolbox   7 contains:

    FED 7
    Universal FED Event 7
    FED Status Reader 7


---------- Configurator settings for UniversalFedToolbox   0: ----------


*** Current FED settings:

       FED - Read Route              = FED9U_ROUTE_VME
       FED - DAQ Mode                = FED9U_MODE_SCOPE
       FED - Scope Length            = 6
       FED - Test Register           = NORMAL
       FED - Strip Pedestal          = 0
       FED - Strip High Threshold    = 50
       FED - Strip Low Threshold     = 50
       FED - Crate Number            = 1
       FED - Slot Number             = 12
       FED - Clock Source            = FED9U_CLOCK_TTC
       FED - Trigger Source          = FED9U_TRIG_TTC

*** Current FTE settings:

       FTE - Trigger State           = REPETITIVE
       FTE - L1A Period              = 40000
       FTE - L1A Low Rate Period     = 40000
       FTE - L1A Random Rate         = 164
       FTE - Minimum Trigger Spacing = 40
       FTE - TCS Source              = LOCAL_SRC
       FTE - TCS Status Input Enable = ON

*** Current FTE FM settings:

       FTE FM - Cluster Definition   = MULTI_STRIP
       FTE FM - Cluster Widths File  = ConfigFiles/FTeFrameMakerFiles/FTe_CluWidthDistr_Simulation.txt
       FTE FM - Digital High Level   = 3600
       FTE FM - Digital Low Level    = 400
       FTE FM - Pedestal Level       = 1200
       FTE FM - Maximum Hit Level    = 2050
       FTE FM - Minimum Hit Level    = 1950
       FTE FM - Gaussian Noise Level = 0

```

Figure 8.25(a): An example 'multi FED test' report – Page 1.

```
101
102 ---------- Configurator settings for UniversalFedToolbox   1: ----------
103
104
105 *** Current FED settings:
106
107         FED - Read Route              = FED9U_ROUTE_VME
108         FED - DAQ Mode                = FED9U_MODE_SCOPE
109         FED - Scope Length            = 6
110         FED - Test Register           = NORMAL
111         FED - Strip Pedestal          = 0
112         FED - Strip High Threshold    = 50
113         FED - Strip Low Threshold     = 50
114         FED - Crate Number            = 1
115         FED - Slot Number             = 5
116         FED - Clock Source            = FED9U_CLOCK_TTC
117         FED - Trigger Source          = FED9U_TRIG_TTC
118
119
120 ---------- Configurator settings for UniversalFedToolbox   2: ----------
121
122
123 *** Current FED settings:
124
125         FED - Read Route              = FED9U_ROUTE_VME
126         FED - DAQ Mode                = FED9U_MODE_SCOPE
127         FED - Scope Length            = 6
128         FED - Test Register           = NORMAL
129         FED - Strip Pedestal          = 0
130         FED - Strip High Threshold    = 50
131         FED - Strip Low Threshold     = 50
132         FED - Crate Number            = 1
133         FED - Slot Number             = 9
134         FED - Clock Source            = FED9U_CLOCK_TTC
135         FED - Trigger Source          = FED9U_TRIG_TTC
136
137
138 ---------- Configurator settings for UniversalFedToolbox   3: ----------
139
140
141 *** Current FED settings:
142
143         FED - Read Route              = FED9U_ROUTE_VME
144         FED - DAQ Mode                = FED9U_MODE_SCOPE
145         FED - Scope Length            = 6
146         FED - Test Register           = NORMAL
147         FED - Strip Pedestal          = 0
148         FED - Strip High Threshold    = 50
149         FED - Strip Low Threshold     = 50
150         FED - Crate Number            = 1
151         FED - Slot Number             = 10
152         FED - Clock Source            = FED9U_CLOCK_TTC
153         FED - Trigger Source          = FED9U_TRIG_TTC
154
155
156 ---------- Configurator settings for UniversalFedToolbox   4: ----------
157
158
159 *** Current FED settings:
160
161         FED - Read Route              = FED9U_ROUTE_VME
162         FED - DAQ Mode                = FED9U_MODE_SCOPE
163         FED - Scope Length            = 6
164         FED - Test Register           = NORMAL
165         FED - Strip Pedestal          = 0
166         FED - Strip High Threshold    = 50
167         FED - Strip Low Threshold     = 50
168         FED - Crate Number            = 0
169         FED - Slot Number             = 16
170         FED - Clock Source            = FED9U_CLOCK_TTC
171         FED - Trigger Source          = FED9U_TRIG_TTC
172
173
174 ---------- Configurator settings for UniversalFedToolbox   5: ----------
175
176
177 *** Current FED settings:
178
179         FED - Read Route              = FED9U_ROUTE_VME
180         FED - DAQ Mode                = FED9U_MODE_SCOPE
181         FED - Scope Length            = 6
182         FED - Test Register           = NORMAL
183         FED - Strip Pedestal          = 0
184         FED - Strip High Threshold    = 50
185         FED - Strip Low Threshold     = 50
186         FED - Crate Number            = 1
187         FED - Slot Number             = 13
188         FED - Clock Source            = FED9U_CLOCK_TTC
189         FED - Trigger Source          = FED9U_TRIG_TTC
190
```

Figure 8.25(b): An example 'multi FED test' report – Page 2.

```
191
192 ---------- Configurator settings for UniversalFedToolbox   6: ----------
193
194
195 *** Current FED settings:
196
197        FED - Read Route               = FED9U_ROUTE_VME
198        FED - DAQ Mode                 = FED9U_MODE_SCOPE
199        FED - Scope Length             = 6
200        FED - Test Register            = NORMAL
201        FED - Strip Pedestal           = 0
202        FED - Strip High Threshold     = 50
203        FED - Strip Low Threshold      = 50
204        FED - Crate Number             = 1
205        FED - Slot Number              = 19
206        FED - Clock Source             = FED9U_CLOCK_TTC
207        FED - Trigger Source           = FED9U_TRIG_TTC
208
209
210 ---------- Configurator settings for UniversalFedToolbox   7: ----------
211
212
213 *** Current FED settings:
214
215        FED - Read Route               = FED9U_ROUTE_VME
216        FED - DAQ Mode                 = FED9U_MODE_SCOPE
217        FED - Scope Length             = 6
218        FED - Test Register            = NORMAL
219        FED - Strip Pedestal           = 0
220        FED - Strip High Threshold     = 50
221        FED - Strip Low Threshold      = 50
222        FED - Crate Number             = 1
223        FED - Slot Number              = 20
224        FED - Clock Source             = FED9U_CLOCK_TTC
225        FED - Trigger Source           = FED9U_TRIG_TTC
226
227
228 ################################################################################################
229 ################################################################################################
230
231 Test 1:
232
233   Initial settings:
234
235     Number of triggers to send from FTE TCS Master  = 1000
236     FED event comparison enabled
237     System status checking enabled
238
239     - UniversalFedToolbox   0:
240
241       Maximum readout rate (MB/s)  = 400
242       Tracker occupancy percentage = 5
243       Fed9UEvent error checking enabled
244
245     - UniversalFedToolbox   1:
246
247       Maximum readout rate (MB/s)  = 400
248       Fed9UEvent error checking enabled
249
250     - UniversalFedToolbox   2:
251
252       Maximum readout rate (MB/s)  = 400
253       Fed9UEvent error checking enabled
254
255     - UniversalFedToolbox   3:
256
257       Maximum readout rate (MB/s)  = 400
258       Fed9UEvent error checking enabled
259
260     - UniversalFedToolbox   4:
261
262       Maximum readout rate (MB/s)  = 400
263       Fed9UEvent error checking enabled
264
265     - UniversalFedToolbox   5:
266
267       Maximum readout rate (MB/s)  = 400
268       Fed9UEvent error checking enabled
269
270     - UniversalFedToolbox   6:
271
272       Maximum readout rate (MB/s)  = 400
273       Fed9UEvent error checking enabled
274
275     - UniversalFedToolbox   7:
276
277       Maximum readout rate (MB/s)  = 400
278       Fed9UEvent error checking enabled
279
```

Figure 8.25(c): An example ‘multi FED test’ report – Page 3.

```
**** Results For Test 1: ****

  Number of triggers sent                                  = 1000
  Time taken                                               = 53.42

  - FED 0:

    Number of events received                              = 1000
    Data rate (MB/sec)                                     = 0.0289923
    Fraction of samples for which throttle was enabled     = 0
    Errors:
      Number of events with Event Number errors            = 0
      Number of events with Event Length errors            = 0
      Number of events with Fed9U CRC errors               = 0
      Number of events with Fed9UEvent construction errors = 0

  - FED 1:

    Number of events received                              = 1000
    Data rate (MB/sec)                                     = 0.0289923
    Fraction of samples for which throttle was enabled     = 0
    Errors:
      Number of events with Event Number errors            = 0
      Number of events with Event Length errors            = 0
      Number of events with Fed9U CRC errors               = 0
      Number of events with Fed9UEvent construction errors = 0

  - FED 2:

    Number of events received                              = 1000
    Data rate (MB/sec)                                     = 0.0289923
    Fraction of samples for which throttle was enabled     = 0
    Errors:
      Number of events with Event Number errors            = 0
      Number of events with Event Length errors            = 0
      Number of events with Fed9U CRC errors               = 0
      Number of events with Fed9UEvent construction errors = 0

  - FED 3:

    Number of events received                              = 1000
    Data rate (MB/sec)                                     = 0.0289923
    Fraction of samples for which throttle was enabled     = 0
    Errors:
      Number of events with Event Number errors            = 0
      Number of events with Event Length errors            = 0
      Number of events with Fed9U CRC errors               = 0
      Number of events with Fed9UEvent construction errors = 0

  - FED 4:

    Number of events received                              = 1000
    Data rate (MB/sec)                                     = 0.0289923
    Fraction of samples for which throttle was enabled     = 0
    Errors:
      Number of events with Event Number errors            = 0
      Number of events with Event Length errors            = 0
      Number of events with Fed9U CRC errors               = 0
      Number of events with Fed9UEvent construction errors = 0

  - FED 5:

    Number of events received                              = 1000
    Data rate (MB/sec)                                     = 0.0289923
    Fraction of samples for which throttle was enabled     = 0
    Errors:
      Number of events with Event Number errors            = 0
      Number of events with Event Length errors            = 0
      Number of events with Fed9U CRC errors               = 0
      Number of events with Fed9UEvent construction errors = 0

  - FED 6:

    Number of events received                              = 1000
    Data rate (MB/sec)                                     = 0.0289923
    Fraction of samples for which throttle was enabled     = 0
    Errors:
      Number of events with Event Number errors            = 0
      Number of events with Event Length errors            = 0
      Number of events with Fed9U CRC errors               = 0
      Number of events with Fed9UEvent construction errors = 0

  - FED 7:

    Number of events received                              = 1000
    Data rate (MB/sec)                                     = 0.0289923
    Fraction of samples for which throttle was enabled     = 0
    Errors:
      Number of events with Event Number errors            = 0
      Number of events with Event Length errors            = 0
      Number of events with Fed9U CRC errors               = 0
      Number of events with Fed9UEvent construction errors = 0

  Event comparison errors:

    Number of Event Number mismatches                      = 0
    Number of Bx Number mismatches                         = 0
    Number of Event Length mismatches                      = 0
```

Figure 8.25(d): An example 'multi FED test' report – Page 4.

~ Chapter 9 ~

The FED Front End Firmware

9.1 The FED Front End Firmware Functionality

Most of the data processing performed by the FED is carried out within the 8 FE Units on each board. As described in Chapter 4, a FE Unit reads the optical output from 12 multiplexed APV pairs, amplifying, digitising and de-skewing the signals before passing them to a FE FPGA for analysis. The layout of the FE FPGA firmware is summarised in Figure 9.1. It was originally developed in Verilog by W. Gannon [61], but has since been converted to VHDL by O. Zorba using Mentor Graphics FPGA Advantage [97].

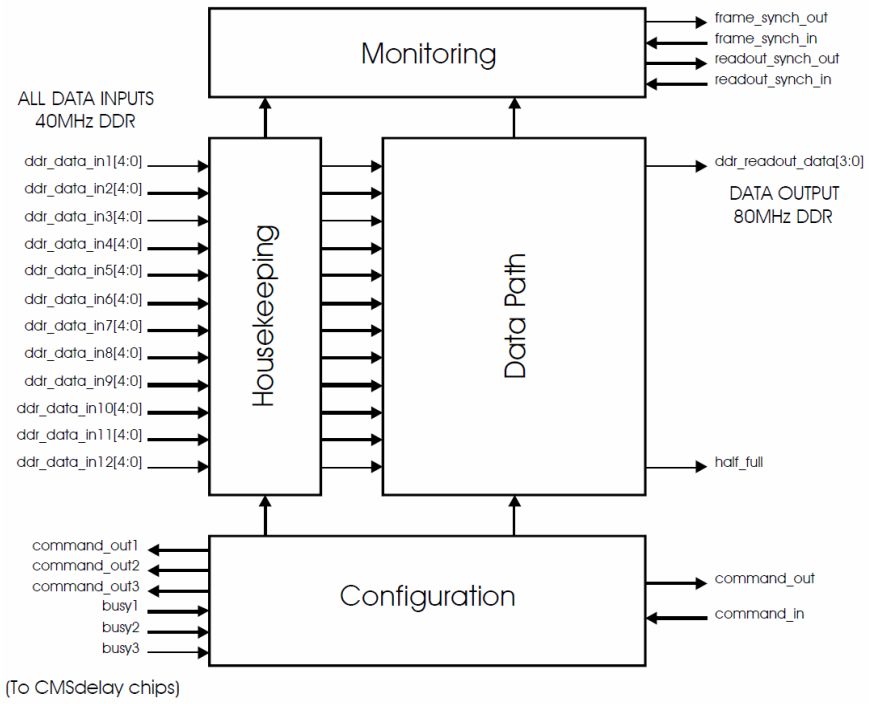


Figure 9.1: A block diagram showing the layout of the FED FE FPGA firmware. (Figure taken from Reference [61])

9.1.1 Housekeeping

Data initially enter a ‘Housekeeping’ block, which conducts the routine tasks of event detection and APV fault monitoring. A ‘ticker’ block examines the input from each of the 12 FE channels, comparing every ADC value with a programmable digital threshold. In the absence of L1 triggers, the signal from a multiplexed pair of APVs should consist of a two-clock-cycle wide logic high tick mark which repeats with a 70 clock cycle period. The ‘ticker’ block is able to lock to this mark, ensuring that it remains in synchronisation with the APVs.

Frame headers, signalling the start of an event, are identified by an additional four logic high values immediately after a tick mark. When these are detected, the ‘ticker’ block collects the interleaved 8bit pipeline addresses and error bits from the multiplexed APV headers and generates a start pulse to indicate that the following 256 ADC samples represent physics data.

A ‘tick master’ block monitors the start pulses output by each of the ‘tickers’. If the 12 FE channels are correctly synchronised, pulses from all ‘tickers’ will be generated at the same time. To account for synchronisation loss, the start of an APV event is in fact defined by the ‘tick master’ to be the clock cycle in which more than half of the enabled ‘tickers’ detect a frame. This ensures that the readout of all 12 channels remains in phase; ‘tickers’ that do not agree with the majority are marked with appropriate status bits.

The ‘tick master’ additionally reads the APV pipeline addresses output by the ‘tickers’, and calculates the majority (the most common value for each of the 8 bits). Pipeline addresses for all APVs should be the same, unless a ‘ticker’ becomes unsynchronised or an APV develops a fault. Provided that more than half of the ‘tickers’ are in synchronisation, the majority pipeline should give the correct value.

Whenever an event occurs, the ‘tick master’ sends a start event pulse to the ‘data path’ region of the FPGA, while the majority pipeline address and a number of status bits are transmitted to the BE FPGA for inclusion in the final event packet. The status bits enable the DAQ to determine which channels are unsynchronised, which pipeline addresses do not agree with the majority and which APVs have error flags. If a large number of channels lose synchronisation, a reset command may be sent to the Tracker.

9.1.2 Data Path

Most of the FE FPGA resources are applied to the task of processing the APV strip data, extracting the required hit information and buffering it prior to readout. This is accomplished by the ‘data path’ block. To maximise efficiency the 12 input channels are processed in parallel, using the same logic sequence. Functionally, the firmware is divided into 6 identical sections; a number of areas of memory are shared between pairs of inputs, to minimise logic requirements.

The first procedure carried out on data input to each channel of the ‘data path’ block is pedestal subtraction. Pedestals are components of the strip signals that should be invariant with time (over the timescale of a CMS experimental run), but may be different for every strip. Relative pedestal magnitudes can be estimated by taking the average of the strip heights in the absence of hits, and are loaded into a look up table in the FE FPGA when the FED is configured during initialisation.

Theoretically, pedestal subtracted data should consist of hit and random noise signals, superposed with a ‘DC’ Common Mode (CM) offset that may vary between events but is constant for each APV. The second stage of the ‘data path’ block calculates and subtracts this CM, a process described in detail in Section 9.2. Additionally, the APV strip data are reordered to match the physical strip layout (the

128 channels of an APV are arranged in a non-physical sequence when multiplexed at the output of the chip). While the CM is calculated, APV values are read into a Reordering RAM; they are read out in the correct order as the Common Mode offset is subtracted.

The ‘data path’ block then conducts the process of hit finding, or Zero Suppression (see Chapter 4). This is achieved by applying a two-layered threshold cut to the data. If multiple adjacent strips have a signal magnitude greater than threshold *A*, they are treated as hits and output. Any isolated strips with a magnitude greater than threshold *B* are also regarded as hits; threshold *B* is larger than threshold *A*, as a single high value is more likely to be caused by noise than two or more neighbouring ones. Data from all other strips are discarded.

In Zero Suppressed mode, hits are grouped into clusters. The starting address and length of each cluster is output, followed by the corresponding strip values. As one data word is transmitted each clock cycle, two free clock cycles are required for the cluster address and length before each set of hits. In order to meet these timing constraints, it is necessary to join any clusters that are separated by less than two strips (spanning the value in between).

Input Value	Output Value
1023	255
254 – 1022	254
0 – 253	0 – 253
< 0	0

Table 9.1: Rules for reducing Zero Suppressed data width from 10bit to 8bit.

Whilst hit finding, the data width of the strip values is additionally clipped from the 10bit range of the input ADC down to 8bit according to the scheme in Table 9.1, reducing the overall data volume output from the FED. This should not significantly affect the performance of the Tracker, as most normal physics data

should fit within the smaller digital range and positional information is the most important factor when reconstructing tracks; a 10bit ADC is initially used primarily to allow for large scale offsets in the data caused by pedestals and CM.

The reduced width cluster data are finally combined with event status information and stored in the FE memory buffer. When all the words of the event packet have been written, its size is transmitted to the BE FPGA; the data lengths from all FE FPGAs are transmitted simultaneously, and the BE FPGA determines whether there is currently enough space within the BE memory buffer to hold all of the information. If so, the BE FPGA sends a readout pulse to all of the FE FPGAs. Data from the 12 channels of each Front End are read out sequentially and multiplexed down from 8bit words at 80 MHz to 4bit words at 160 MHz, in order to reduce the number of required data line connections to the BE FPGA.

9.2 Common Mode Subtraction

The CM is that component of the signal offset which is common to all channels of an APV. If an event contains no hit information then a good approximation of the CM for an APV can be calculated by taking the average of the strip values (minus pedestals). In practice, events processed by the FED will contain hits, superimposed onto the Common Mode offset, which must be excluded from the calculation. A simple mean algorithm is therefore insufficient for the determination of CM values during the acquisition of normal physics data. The current FED CM subtraction algorithm and a potential alternative are discussed in the following sections.

9.2.1 The Median Method

Currently, hit information is effectively discarded during the CM calculation by taking the median of the APV channel values. The median is the central value when the strip data are sorted in numerical order, and as such is relatively unaffected by

outliers (i.e. hits). It should therefore provide a good estimate of the true mean offset (within statistical noise fluctuations) provided that less than half of the APV channels contain hits.

In a non-resource-critical environment, the median of a set of values may be found through the simple process of ordering them by size and selecting the value in the middle. However, it is difficult to implement an efficient sorting algorithm in the limited space of the FE FPGA. Typically, sorting a set of n data values requires at best $n \log(n)$ iterations, with each iteration involving some form of loop through all the values, and at worst n^2 iterations. If a single data operation were carried out every clock cycle in the FPGA, it would therefore take approximately 36,000 clocks to guarantee the correct reordering of 128 strip values. Alternatively, the strip data could be processed in parallel at each iteration, reducing the time requirements to ~270 clock cycles but necessitating simultaneous access to the 128 elements of an array containing all of the strip values at every clock; implementing such an array inside the FE FPGA uses of the order of 15% of the chip logic. As 12 Channels of data must be processed in parallel, this is clearly impractical.

In order to make best use of the FPGA resources, the Median Method algorithm (developed by M. French and W. Gannon) consists of a two stage histogram approach. At the first pass, a histogram is formed of the highest 5 bits of the 128 10bit values from each APV. By sequentially summing the contents of each histogram bin until the total is greater than or equal to 64 it is possible to find the bin in which the median value lies. This gives the first 5 bits of the median which are then used as a mask to filter out potential medians during a second pass through the data, where the lowest 5 bits of the APV values are histogrammed. Sequentially summing the bin contents of the second histogram again enables the central 5bit value to be found.

Combining the highest and lowest 5bit values from the two histogram passes provides the median of the 128 APV channels.

9.2.2 Median Method Timing

A diagram showing the flow of the Median Method algorithm is given in Figure 9.2. The median logic blocks are clocked at twice the LHC bunch crossing rate of 40 MHz in order to reduce the time required for the calculation; consequently, all timings are given in 80 MHz clock cycles (at 12.5 ns intervals).

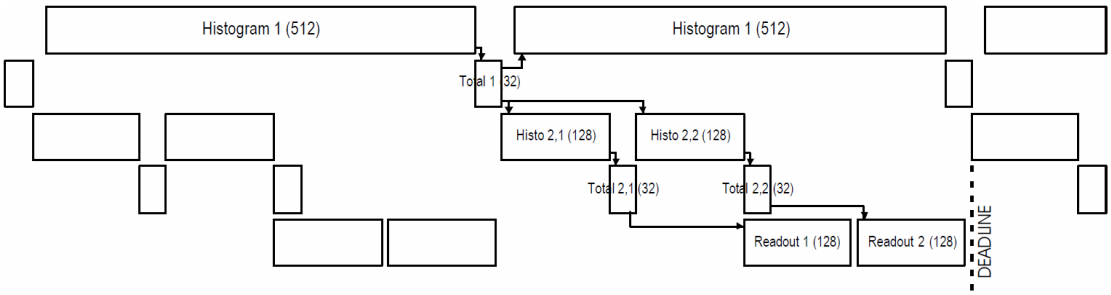


Figure 9.2: Median Method CM calculation timing diagram. All times (values in brackets) are given in 80 MHz clock periods. (Figure taken from Reference [61])

Strip data from each multiplexed pair of APVs simultaneously enter the Reordering RAM and the first median calculation block at 40 MHz. Storing the data for one complete event and building the first-pass histogram of the median algorithm therefore requires 512 clocks. Scanning through this histogram takes a further 32 clocks, after which the first stage median block is free to begin processing data from the next event (which may arrive within as little as 16 clocks).

Once the highest 5 bits of the median value have been found, the Reordering RAM is triggered to begin outputting de-multiplexed, reordered APV strip data at 80 MHz. These are read into the second median calculation block, which is implemented using a single RAM in order to reduce logic consumption. It must therefore operate sequentially on the data from each APV; each second-pass histogram takes 128 clocks to build, followed by another 32 clocks for scanning through the result. Readout of the CM subtracted data can begin as soon as the Reordering RAM is free, after both

second-pass histograms have been completed. This again occurs sequentially for each APV and requires a total of 256 clocks. To allow for consecutive input events with a minimum spacing of 16 clocks, the second histogramming stage of the median algorithm and the final CM subtracted readout must occur within 560 clocks; these processes currently require 544 clocks and thus fit well within the time limit.

9.2.3 Motivation for a New Common Mode Subtraction Algorithm

The Median Method for CM calculation has been tested extensively in simulation and hardware (for example, in Reference [106] and in the FED hit-finding verification of Chapter 8). While it has proved to be extremely reliable for analysis of all typical proton-proton collision event data, it has two main shortcomings.

Firstly, the median algorithm will fail catastrophically if more than half of the channels of an APV contain hits. In this case, a hit value will be incorrectly identified as the median. Subtracting this false median will introduce an unknown negative offset to the highest 64 hit-strip values and result in the loss of hit information on any other strips.

Events with greater than 50% occupancy are most likely to occur during heavy ion collisions and so under these conditions it is intended that the FED will either run in Processed Raw mode, or that the Zero Suppressed CM value will be shifted from the median of the APV channel data to somewhere between the lowest and the 63rd lowest values (effectively taking the median of a subset of the smallest channel values). However, neither of these solutions are ideal; running in Processed Raw mode will unnecessarily increase the overall FED data rate, leading to a reduced event trigger rate, and adjusting the level of the median to deal with particular occupancy ranges results in an algorithm that does not function well under general conditions.

Ultimately, it is desirable to run always in Zero Suppressed mode, but with a CM calculation that is able to deal with events which have arbitrary occupancy.

Secondly, preliminary Test-beam data from Tracker Outer Barrel detector modules has revealed the phenomenon of non-uniform CM in a small number of APVs (an example can be found in Reference [111], also shown in Figure 9.8). These APVs typically demonstrate a splitting of the baseline into two distinct levels, superposed with non-linear continuous noise fluctuations. The effects are generally small, of the order of 10 FED input ADC counts over the entire range of a frame, but they can cause the Median Method algorithm to fail. If a baseline is split over two levels then the median for each will be different; it is therefore inappropriate to calculate and subtract just one median value.

Indeed, non-uniform baseline effects cannot be correctly described as Common Mode offsets since they are not common between channels, and so any standard CM algorithm will fail. However, as demonstrated by S. Paoletti [111], it is possible to minimise the impact of global APV baseline features by dividing the frame into subsections and calculating the CM locally for each. This deals effectively with baseline splitting (the individual levels are considered separately) and limits the impact of other non-uniform noise fluctuations to the section in which they occur (i.e. a noise feature in one section will not affect the CM calculation for another).

Unfortunately, applying a CM algorithm to specific regions of a frame requires the APV channels to be in order. The first stage of the current Median Method is carried out while the APV data are being read into the Reordering RAM. Analysing reordered data would require fitting both stages of the median algorithm in the time period currently occupied by the second histogramming pass. This is impractical to achieve in the limited space of the FE FPGA.

It would be beneficial to the performance of the CMS Tracker if the inherent limitations of the Median Method could be addressed by an alternative CM subtraction algorithm. The discovery of non-uniform CM during the prototyping of detector modules also highlights the fact that the true nature of CM effects will not be known until late in the Tracker assembly process; unexpected issues can and do arise when different items of hardware are used together for the first time. Consequently, it is important to investigate and develop a variety of CM subtraction techniques as early as possible, before CMS is fully operational. Algorithms capable of dealing with 'known' effects (such as non-uniform CM) must be prepared in advance, so that a fast response can be made if these phenomena are observed in the final Tracker system.

Even if these algorithms are never used (ultimately, non-uniform CM may be found only in the prototype systems), the implementation of original routines for the programmable logic devices on the FED encourages developers to exploit the full potential of the board. A good understanding of what the FED FPGAs are capable of will be an essential aid when attempting to solve any future problems that may be caused by as yet unobserved CM effects.

9.2.4 The Mean Method

The Mean Method algorithm is a feature-rich alternative to the Median Method. It has been designed to cope with events that have arbitrary strip occupancy (up to ~98%) and enables APV frames to be processed in subsections for greater compatibility with non-uniform CM effects.

Ideally, the CM for an APV (or section thereof) should always be found by taking the mean of all channels that do not contain hits; reliably distinguishing between strips that do and do not contain hit information is the only challenging aspect of this procedure. Whereas the original median-based algorithm automatically

‘discards’ hits (provided that there are less than 64), the Mean Method requires an additional stage of analysis. The simplest technique for filtering out hits is to establish a signal level (or ‘noise cap’) for each event, above which only hits should be present; the mean of all channel values below that level can then be determined. In previous mean algorithm studies (conducted by I. Tomalin [106]), this noise cap has been found by initially calculating the mean from all APV channels. Given an event with a small fraction of hit strips, the mean of all the values (CM and hits) will indeed be less than any one of the hits, but under other circumstances this method can prove inadequate.

If an event contains no or very few hits then the initial noise cap will be approximately equal to the CM itself. Averaging all of the values below the CM will result in a final CM estimate that has an artificial negative offset; subtracting an offset that is too small from a strip that contains only noise will lead to an artificially high final signal level, which could potentially be mistaken for a hit. Alternatively, if an event were to contain many hits over a wide range of levels then the overall mean could lie above the signal height of the smaller hit values. This would lead to an artificially high final CM offset, resulting in an unknown negative shift to the CM subtracted hits.

To increase the reliability of the algorithm and enable the support of high occupancy events, the Mean Method adopts a less conventional approach. During an initial pass though the strip data from each APV, the third lowest value is found; a predefined noise threshold (equal to some multiple of the APV channel noise) is then added to this to give the noise cap for the mean calculation pass. Starting from the effective minimum of the CM values and adding a fixed noise range ensures that the data region used in the mean calculation is relatively unaffected by the number of hits

an event contains. As a result, the algorithm remains effective even when events have a strip occupancy greater than 50%.

It may appear to be more sensible to use the lowest strip value as a starting point for the noise cap calculation, but this could lead to erroneous final CM values in the case of APV strip failure. If one of the 128 silicon microstrips connected to an APV (or the APV readout channel itself) fails, the output frames will contain one strip value that is set to zero. Consequently, a noise cap derived from the lowest strip value would be equal to the noise threshold. As the typical CM level will be of the order of one hundred ADC counts and the APV channel noise an order of magnitude lower, this noise cap would almost certainly be less than any of the values received from functioning strips; the mean of the values below the noise cap would therefore be zero and no CM would be subtracted. Selecting the third lowest value instead of the lowest minimises the impact of up to two APV strips failing during an experimental run.

9.2.5 Mean Method Timing

The implementation of the Mean Method is conceptually simpler than the Median Method and fewer clock cycles are required to process each event. A diagram of the timings for the main stages of the algorithm is shown in Figure 9.3; values are again given in 80 MHz clock cycles.

As with the Median Method, strip data from a pair of APVs are initially read into the Reordering RAM, taking 512 clocks. The Reordering RAM is then triggered to output de-multiplexed, reordered data from the first APV and then the second. For each APV the noise cap is calculated, requiring 128 clocks to find the third lowest value and an additional clock to add the noise threshold. Simultaneously, the strip data are stored in a 128 deep buffer, which is read out into the mean calculation blocks as soon as the noise cap is found. It takes 128 clocks to add all of the strip values below

the noise cap and then a further 12 to perform a binary division and obtain the mean, at which point the Reordering RAM is triggered for a final readout and the CM is subtracted. To minimise logic usage, the same blocks are used (sequentially) for each APV.

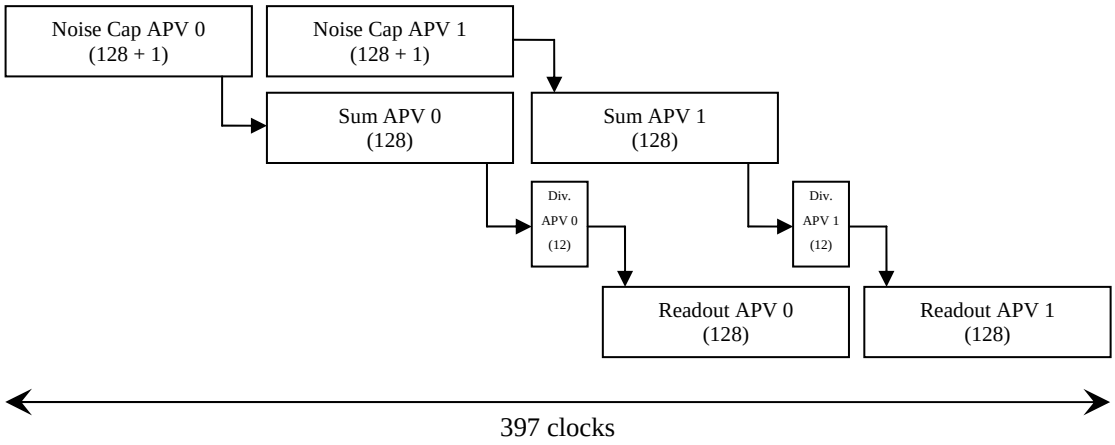


Figure 9.3: Mean Method CM calculation timing diagram. All times (values in brackets) are given in 80 MHz clock periods. 'Div.' refers to a binary division process.

The CM calculation and subtraction process requires 397 clock cycles, significantly less than the 560 clock post-reordering limit established in Section 9.2.2. It is this feature of the algorithm that enables the analysis of subsections of reordered APV frames, for the removal of non-uniform CM effects. A frame may be considered as a whole, or split into halves or quarters; the noise cap and summation blocks simply output a result after each set of 128, 64 or 32 strips, with extra binary divisions inserted as necessary.

9.2.6 Mean Method Logic Usage

Tables 9.2 and 9.3 show the logic requirements of each CM algorithm for a pair of FE channels. The implementation of the Mean Method uses approximately two and a half times the amount of logic required by the Median Method. This is primarily due to the constraint of having to run both passes of the Mean Method (finding the noise cap and calculating the mean) after the strip data have been reordered. It is therefore necessary

to buffer the reordered strip values as they are read out from the Reordering RAM during the first pass, so that they remain available for the second. No additional buffering is required for the Median Method since its first histogramming pass can operate on unordered data as they are initially read into the Reordering RAM.

Resource:	Used:	Utilisation:
Function Generators	1151	5.35%
CLB Slices	576	5.36%
Dffs or Latches	750	3.28%

Table 9.2: Mean Method logic usage for a pair of FE channels.

Resource:	Used:	Utilisation:
Function Generators	448	2.08%
CLB Slices	224	2.08%
Dffs or Latches	338	1.48%

Table 9.3: Median Method logic usage for a pair of FE channels.

The task of finding the third lowest value in an APV frame also has high resource demands, as each strip read from the Reordering RAM must be compared simultaneously with the current three lowest values every clock cycle. However, despite the increased requirements of the Mean Method, the algorithm fits easily within the FE FPGA; the logic usage for the complete FE design is shown in Table 9.4.

Resource:	Used:	Utilisation:
Function Generators	16060	74.68%
CLB Slices	8030	74.68%
Dffs or Latches	10576	46.24%
Block RAMs	24	42.86%

Table 9.4: Logic usage for entire FE FPGA design, including the Mean Method CM subtraction algorithm.

9.2.7 Experimental Evaluation of the Mean Method

Before any CM subtraction algorithm can be used at CMS, its functionality and reliability must be rigorously verified. Typically, this is done through computer

modelling; if the algorithm is implemented within the ORCA [112] event reconstruction framework then its performance can be tested with large quantities of simulated Tracker events and a quantitative evaluation of hit finding efficiency and data quality may be obtained (for example, a recent account of the simulated performance of the Median Method with heavy ion collisions, which highlights the need for an algorithm with improved high occupancy support, can be found in Reference [113]).

A version of the Mean Method that is compatible with ORCA has been written, but thus far time constraints have precluded a full simulation (this will be accomplished in the near future). However, using the FED Tester it is possible to supply a FED with APV frame data that imitates high occupancy Tracker output or contains non-uniform CM effects. Via this technique, an experimental evaluation of at least the basic functionality of the new algorithm has been performed.

The hardware setup for assessing CM subtraction is identical to that of the full FED system tests, described in Chapter 8, although the Mean Method requires additional FED configuration parameters. Each APV must be assigned a noise threshold (NT) and a ‘Common Mode Type’ (CMT) setting, the latter of which determines how many subsections each frame will be divided into (1, 2 or 4) during the CM calculation. Communication with the new registers is not supported by the current Fed9USoftware library and so an independent Mean Method register access class is used, enabling the noise thresholds and Common Mode Types to be loaded from a configuration file. This is combined with the UniversalFedToolbox class (described in Chapter 8) to produce a standard full system test, in which any custom APV frames may be input to the FED and compared with the Mean or Median Method processed output.

The basic functionality of the Mean Method is established by driving the FED with APV frames that contain a linearly increasing ramp. Plots of the input to the FED and the output after CM subtraction are shown in Figure 9.4. Each plot contains data from one of the pair of APVs that are read out by channel 0 of the FED.

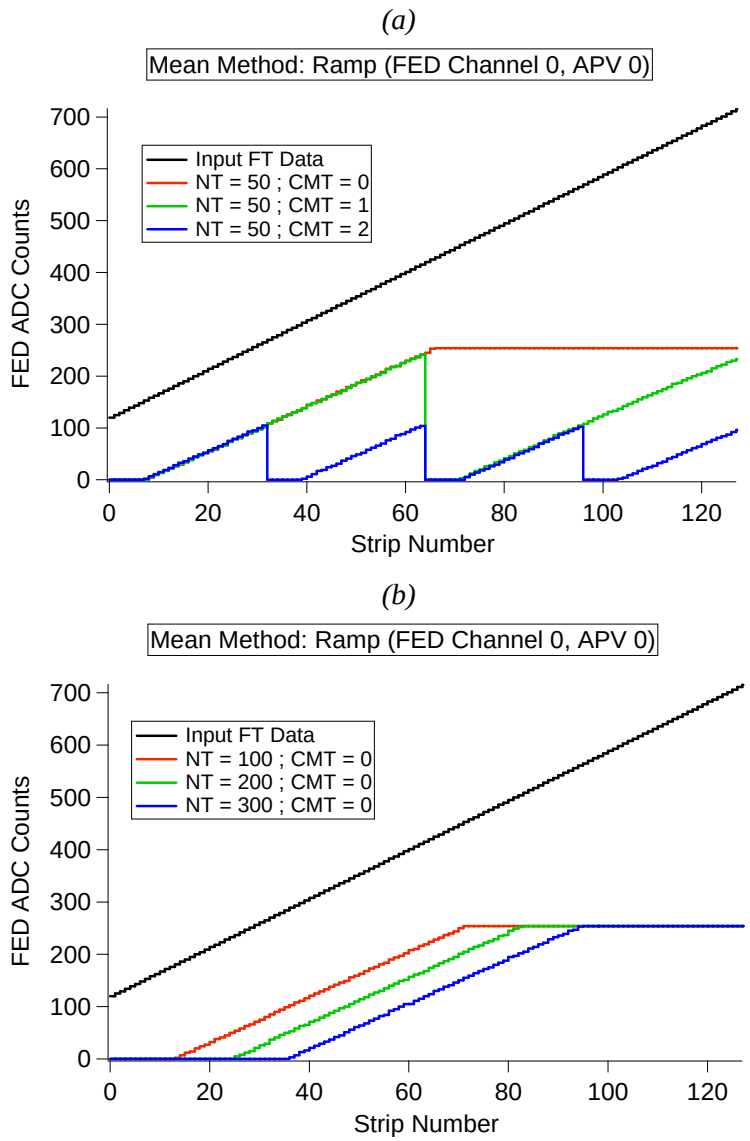


Figure 9.4: Plots showing a linearly increasing APV ramp frame input to the FED compared with the CM subtracted results when the Mean Method algorithm is used. NT is the noise threshold in FED ADC counts, while CMT is the 'Common Mode Type' setting (0 = frame processed in one section, 1 = frame divided into halves, 2 = frame divided into quarters). (a) demonstrates the performance of the Mean Method in all three CMT modes and (b) shows the effect of varying the noise threshold. It should be noted that frames were acquired from the FED in Zero suppressed mode; due to the 10bit to 8bit data width reduction, all output signals 'artificially' saturate above 254 FED ADC counts.

Figure 9.4(a) clearly demonstrates the splitting of the APV baseline into 1, 2 or 4 sections for the three modes of operation of the Mean Method algorithm. For a ramp signal, the calculated CM offset should be at approximately the midpoint between the minimum data value in the sample and the noise threshold. This is confirmed by studying the *x*-axis intercept and gradient of ramps output from the FED over a range of noise thresholds, such as those in Figure 9.4(b); as expected, it is found that the relative shift in the CM offset between noise threshold steps is approximately half the absolute noise threshold value.

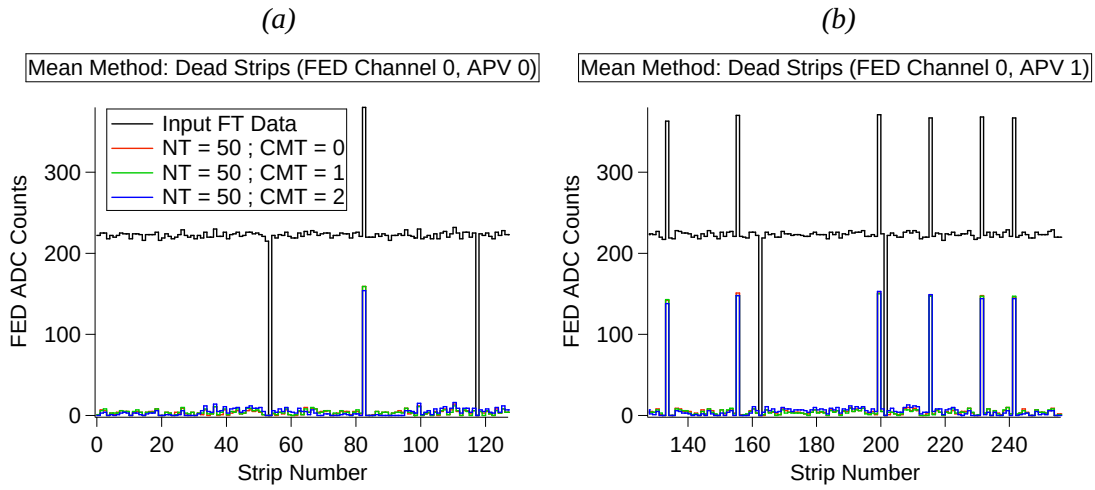


Figure 9.5: Randomly generated frames from the (simulated) pair of APVs read out by channel 0 of the FED. Each input frame contains two artificially disabled strips, and the CM subtracted results are shown for all of the CMT modes of the Mean Method algorithm. (a) corresponds to APV 0 and (b) to APV 1, where the same legend applies to both. Data from a pair of APVs are read out from the FED in a continuous block of 256 strips; hence the frame from APV1 starts at strip 128.

The ability of the Mean Method to cope with strip hardware failures is verified by the plots in Figure 9.5, where two strips at random locations in each of the pair of simulated APVs are artificially disabled. If the algorithm selected the lowest data sample as the minimum of the data range used by the mean calculation, the CM value for any noise threshold less than the APV pedestal would be incorrectly found to be zero. Choosing the third lowest value means that the CM offset is merely underestimated by typically 0.8% of the APV pedestal per dead strip; this is

comparable with the channel noise, and represents an acceptable loss of accuracy given that a critical hardware error has occurred.

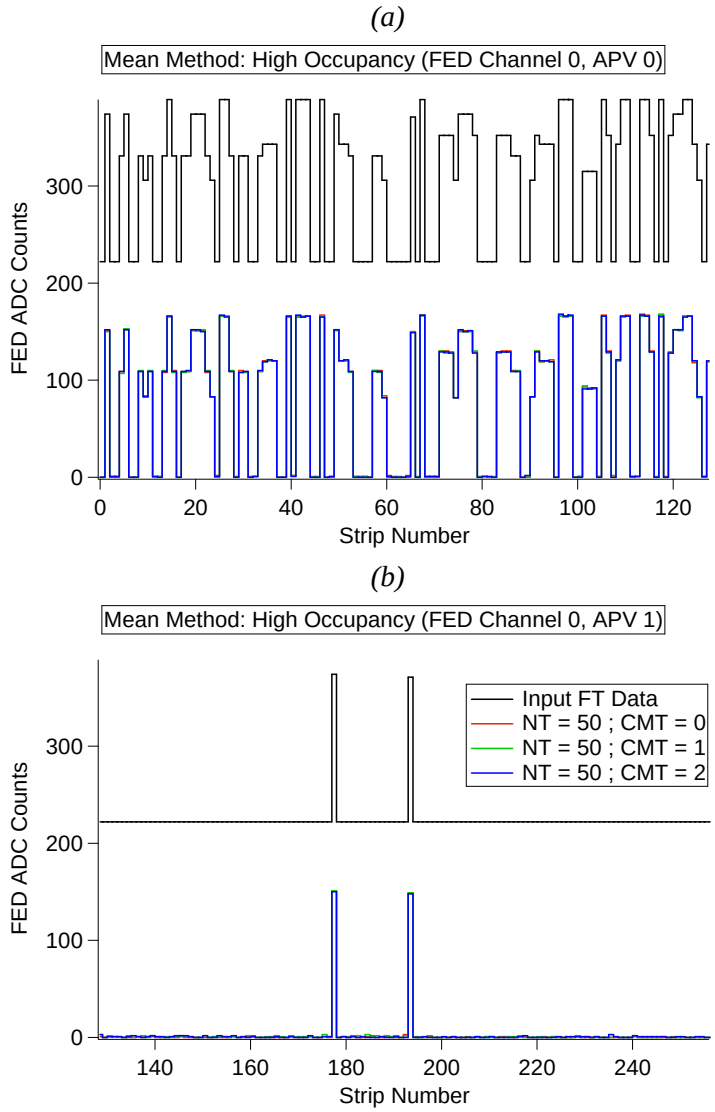


Figure 9.6: Simulated high (a) and low (b) occupancy APV frames processed within a single event by the Mean Method algorithm (the legend in (b) applies to both plots). CM subtracted results are shown for all 3 CMT modes.

Figure 9.6 demonstrates the high occupancy event handling capability of the Mean Method. For this test, simulated multiplexed pairs of APV frames are transmitted to the FED, where the output from APV 0 has a strip occupancy of ~62.5% while that from APV 1 has an occupancy of ~1.6%. Both the high and low occupancy frames are processed in exactly the same way by the Mean Method

algorithm. In each case (regardless of whether the frames are divided into 1, 2 or 4 sections), all of the input hits are correctly identified.

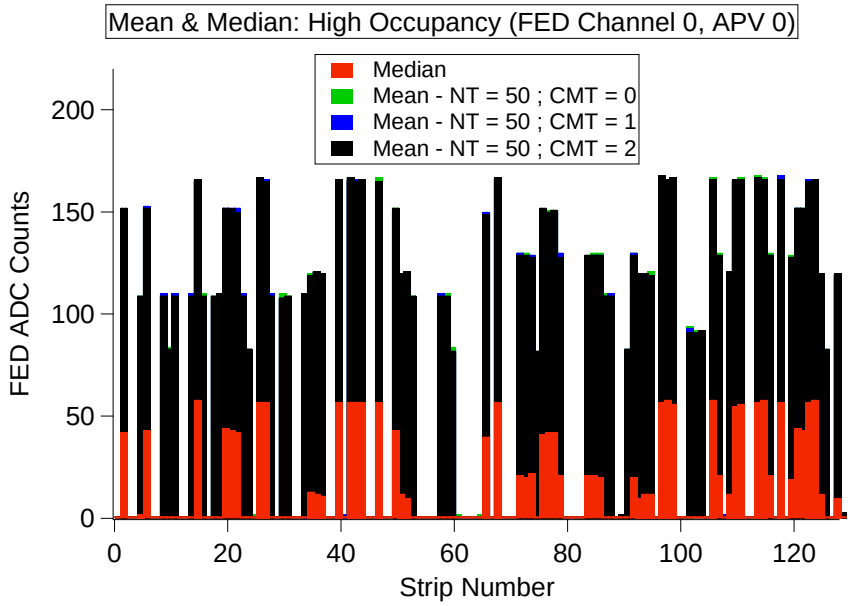


Figure 9.7: A comparison between CM subtracted results generated by the Median and Mean Method algorithms for a frame with a strip occupancy of ~62.5%.

Mean Method results for the high occupancy frame are compared with those produced by the Median Method in Figure 9.7. It can be seen that the median-based algorithm fails catastrophically, with ~30% of the input hits lost and an average level reduction of ~70% for the hits that remain. The performance of the Median Method degrades further as Tracker occupancy increases, whereas the Mean Method preserves all hits provided that at least three strips in each APV subsection are unoccupied.

An example of the phenomenon of non-uniform CM is given in Figure 9.8. This shows test beam results taken from one problematic APV on a TOB sensor module [111]; the raw frame data have been processed in software (hence the one negative value that would have been clipped to zero inside the FED), and the CM level calculated using the Median Method is shown as a dashed red line. Counting from 0 (the strip number axis in Figure 9.8 starts at 1), strips 38 and 39 exhibit unusually high noise, but the main feature of the event is that the baseline is split into

two distinct levels. The average background level for strips 0 to 63 is ~4 ADC counts higher than that for strips 64 to 127.

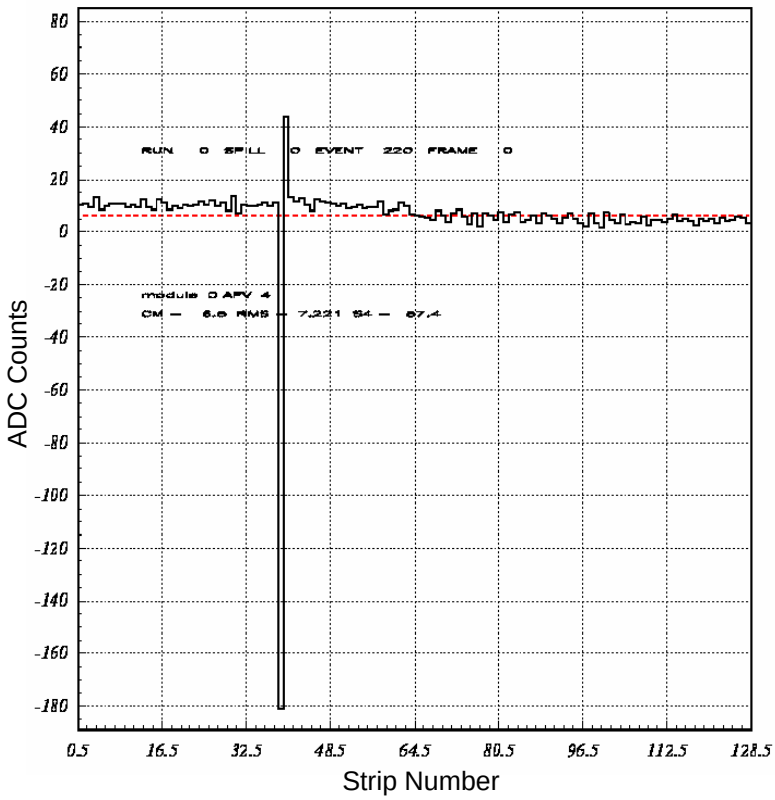


Figure 9.8: TOB test beam data for one problematic APV, demonstrating the phenomenon of non-uniform CM. The dashed red line shows the Median Method estimate of the CM offset. (Figure taken from Reference [111])

Using the computer program DataThief [114], the event data in Figure 9.8 were extracted and converted into an APV frame file, which was loaded into the FED Testers and transmitted to the FED. CM subtracted results obtained using both the Mean and Median Method algorithms are presented in Figure 9.9; to reduce statistical fluctuations and increase clarity, each data set is the average of the values read from all 96 FED channels.

When the APV frame is processed in one 128-strip section, using either the Mean or the Median Method, the calculated CM offset lies at approximately the midpoint of the two baseline levels. Consequently, the offset is underestimated for channels 0 to 63 and overestimated for channels 64 to 127. The first half of the CM

subtracted event is therefore artificially shifted up, which could potentially raise the noise fluctuations above the threshold for hit detection, while the second half is artificially shifted down. In Figure 9.9, a large proportion of the values from strips 64 and above are subtracted completely, which in extreme cases could lead to the loss of hit information.

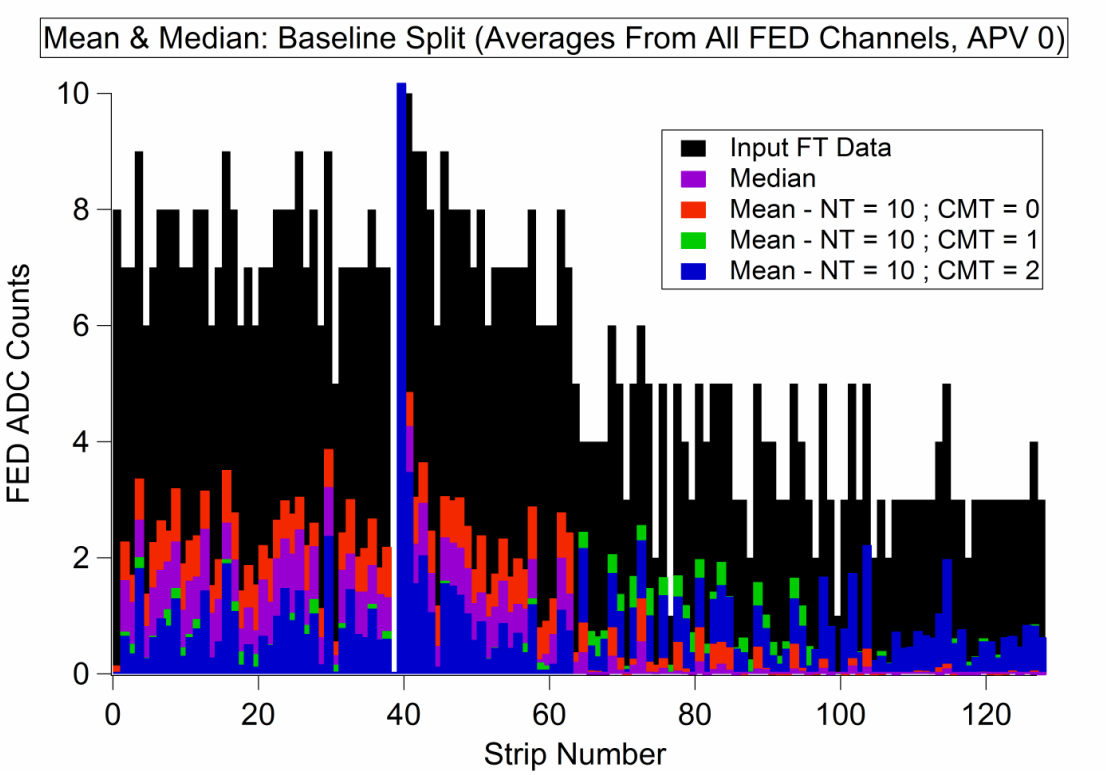


Figure 9.9: A comparison between CM subtracted results obtained when the Median and Mean Method algorithms are applied to an APV frame exhibiting non-uniform CM. The data sets (with the exception of the input FED Tester data) are the averages from all 96 FED channels. A noise threshold of 10 ADC counts is used for the Mean Method, and results for all CMT modes are shown.

However, if the APV baseline is processed in two halves (or four quarters), the calculated CM offsets correctly represent the average signal levels across all strips; in Figure 9.9, the CM subtracted results for the Mean Method with a CMT setting of 1 or 2 are approximately uniform, maintaining random noise fluctuations of approximately the same amplitude as those in the original data. The ability of the Mean Method to

divide APV frames into subsections should therefore make it a more adaptable solution for dealing with non-uniform CM effects.

9.3 Cluster Finding

In the current FE firmware, clusters of hits within APV frames are found using a simple threshold comparison on reordered pedestal and CM subtracted data. The threshold testing method of cluster finding requires the baseline of each APV frame to be approximately uniform. APV splitting and relatively small-scale CM irregularities can be dealt with by performing the CM calculation on subsections of each frame, but the current algorithms will fail if the APV baseline exhibits extremely large-scale non-uniform behaviour.

Such behaviour has not yet been observed during current hardware tests but, as stated in Section 9.2.3, it is important to prepare for as many eventualities as possible. If, for example, interference between hardware modules in the final Tracker system led to significant distortion of APV baselines (beyond the handing capability of the Mean Method algorithm), the FED would be unable to operate correctly in Zero Suppressed mode. All events would have to be read out in Processed Raw mode, reducing the maximum Level 1 trigger rate to 10 kHz, and the Tracker online software would require the addition of pattern recognition algorithms to enable the subtraction of the non-uniform background.

9.3.1 A New Baseline Correction Algorithm

In an attempt to make cluster finding in the FED compatible with APV frames that have non-uniform backgrounds, a baseline correction algorithm was developed that could be run in the FE FPGAs themselves. There are many accepted baseline correction techniques (Reference [115] provides an excellent summary) but they generally require numerically intensive procedures that run far more effectively off-

line on high performance computers than in real-time in the limited resources of an FPGA. Each FE FPGA must be able to perform simultaneous baseline subtraction on events from 12 APVs at a time, in addition to maintaining all of the standard housekeeping, frame recognition, pedestal subtraction and data reordering functionality. Consequently, only a few hundred logic gates are available for each instance of the baseline correction algorithm. An entirely new, highly efficient approach to the baseline subtraction problem was therefore devised.

An APV frame may be thought of as an electromagnetic spectrum, and the principles of applied spectroscopy can be used in its analysis. Like any spectrum, it may be considered to consist of signals of interest (in this case hits) superimposed onto a structured background, with random noise added to the result. The background can usually be assumed to be a low frequency signal with any amplitude, whereas the random noise is normally of high frequency and low amplitude. The ‘interesting’ signals are expected to comprise of higher frequency spectral components than the background, with greater amplitudes than the noise. Taking this into account, the baseline of the spectrum or frame may be thought of as a slowly varying signal with local fluctuations that are within the random noise amplitude; that is, the baseline should be continuous relative to the hit signals. Baseline correction therefore becomes a problem of identifying discontinuities in the frame and subtracting data that are not ‘interestingly’ bound by discontinuities.

The baseline correction algorithm operates in the following manner. It takes reordered frame data and initially finds and subtracts the lowest value from all 128 channels. A ‘discontinuity threshold’ is then defined; this represents a difference between adjacent channels greater than would be expected were the data to consist purely of background and noise (in practice, it is a multiple of the APV channel

noise). It is therefore the limit on the discontinuity between adjacent strips above which only hits will be found. An iterative processing of the frame data begins. At each step, the discontinuity threshold is subtracted from all of the APV channels that are ‘unlocked’; if a channel value is less than the discontinuity threshold, it is set to zero. All channels that are set to zero after the subtraction are considered to be ‘locked’. A three phase process then examines the locked statuses from the current and previous iterations in order to identify the data discontinuities.

The first phase compares the locked statuses before and after the subtraction. Important regions of the ‘strip locked’ profile are wherever a locked strip is adjacent to an unlocked one; at each of these points, the locked strip represents baseline that has been subtracted away, and the unlocked strip is either the continuation of the baseline or a data discontinuity (hit). If one of these locked/unlocked pairs is examined before and after the discontinuity threshold subtraction then either the unlocked channel will become locked, signifying that it was part of the continuous baseline, or it will remain unlocked, indicating that it is a true discontinuity. When a hit is found in this way, the corresponding channel is automatically locked to ensure that nothing is subtracted from the hit information at the next iteration.

The subsequent two phases are used to identify hit clusters that are bound by discontinuities. If both ends of a continuous block of unlocked channels become locked between the start and the end of the first phase of discontinuity finding, then it signifies that either end of the block is a hit. As the channels between the two hits are further from the baseline than the discontinuity threshold, they must also be hits; consequently, they are also automatically locked. The updated ‘locked status’ profile is then input to the next iteration of discontinuity threshold subtraction and the process is repeated, until all of the APV channels have become locked and only the hit

information remains. A more detailed description of the underlying logic of the algorithm is given in the following section.

9.3.2 Baseline Correction Algorithm Implementation and Logic Utilisation

A prototype of the new baseline correction algorithm has been implemented in VHDL. Figure 9.10 demonstrates the flow of data from a single APV through the baseline subtraction logic path; timings are once again given in 80 MHz clock cycles.

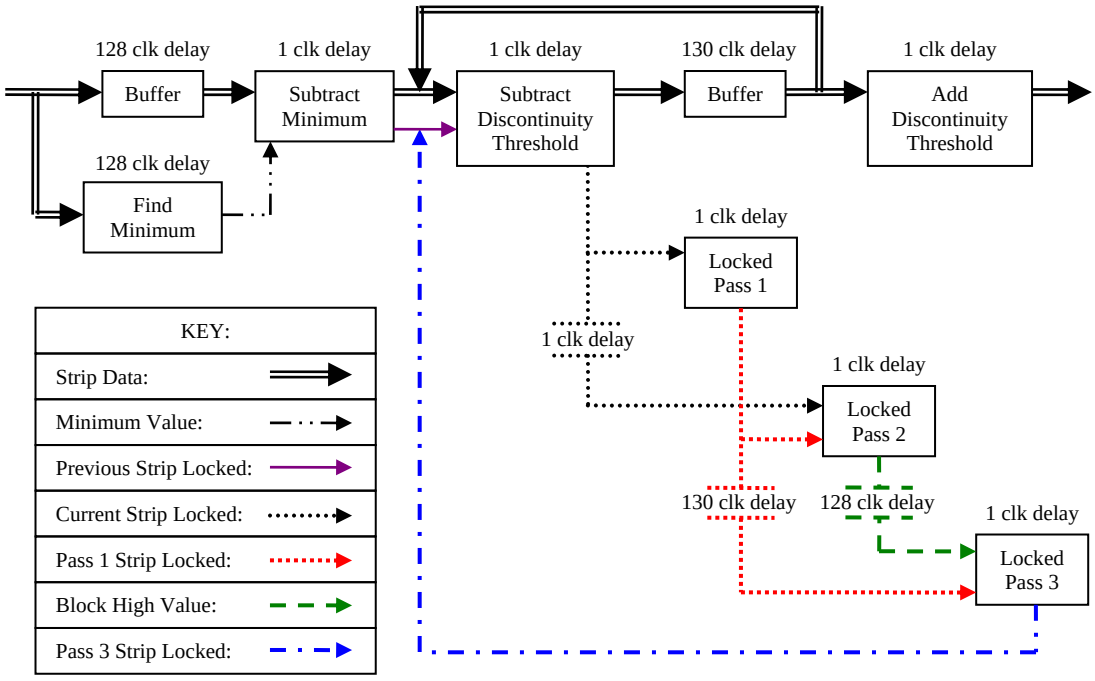


Figure 9.10: A diagram showing the flow of data through the main functional logic blocks of the baseline correction algorithm. Timing values (relative delays introduced by logic processes or buffers) are given in 80MHz clock cycles.

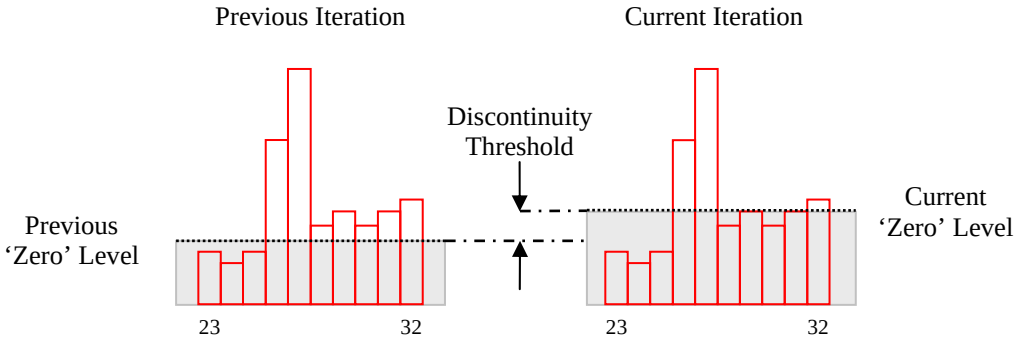
Initially the minimum strip value is found whilst the data are read into a buffer, which would correspond to a Reordering RAM in the FE FPGA. Once the buffer is full, strip values are read out one at a time; the minimum is subtracted and a 1bit ‘strip locked’ signal is generated (a strip is locked if it has a value of zero after subtraction, represented by a locked signal of ‘0’).

Each shifted value and locked bit are fed into the first iteration of the discontinuity threshold subtraction loop. If a strip is unlocked then the discontinuity

threshold is subtracted (with any negative values clipped to zero) and a new locked status bit is assigned. Whilst the strip data are passed into a second buffer, ready for the next iteration of the loop or final readout, the current strip locked status signals are analysed. This represents the core of the process, where the actual baseline identification is performed. Due to the simplicity of the underlying algorithm, it has been possible to implement each of the three locking passes with almost trivially simple logic operations. These are described in the following sections.

9.3.2.1 Locked Pass 1

pass_1_locked(x) = **curr_locked**(x) AND (**prev_locked**(x – 1) AND **prev_locked**(x + 1))



Strip Number:	23	24	25	26	27	28	29	30	31	32
prev_locked	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
curr_locked	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**1**
pass_1_locked	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**1**

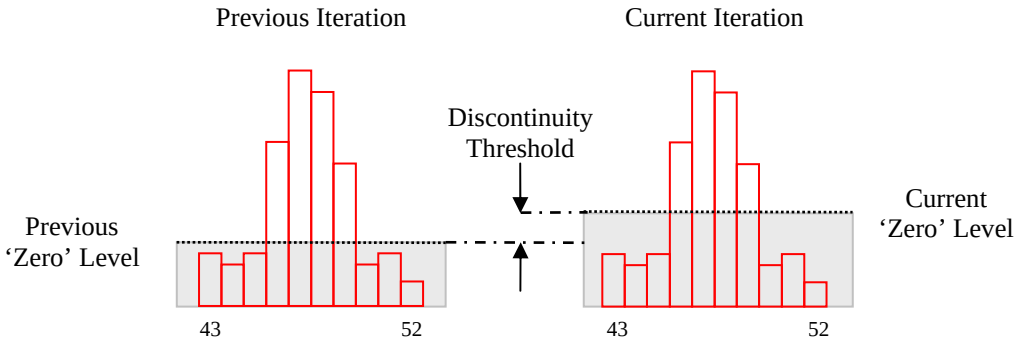
Figure 9.11: A diagram showing the logic operation performed by the 'Locked Pass 1' block of the baseline subtraction algorithm. pass_1_locked, curr_locked and prev_locked are the 'Pass 1', current and previous locked status signals, shown in Figure 9.10, and 'x' refers to strip number. As an example, the logic is applied to a small section of an artificial APV frame after baseline subtraction has occurred; strip 26 corresponds to a hit, and becomes locked. Strip 27 is also a hit, and would become locked at the next iteration.

In order to find discontinuities that have become apparent between successive discontinuity threshold subtractions, the 'Locked Pass 1' block compares the current strip locked signal with the result from the previous loop iteration. At each clock cycle, it merely performs a logical AND between the current locked signal for strip *x*

and the previous locked signals for strips ($x - 1$) and ($x + 1$). This generates an output signal that is automatically set to '0' when a strip next to a locked one remains unlocked between iterations of the loop. Pass 1 additionally generates a done signal if every strip during the current iteration is locked, which triggers the loop to stop. A diagram illustrating an example of a 'Locked Pass 1' logic operation is given in Figure 9.11.

9.3.2.2 Locked Pass 2

```
IF ( (cluster_started == false) AND (pass_1_locked(x) == 1) )

        block_high_value[index] = NOT curr_locked(x - 1)
        cluster_started = true

ELSE IF ( (cluster_started == true) AND (pass_1_locked(x) == 0) )

        block_high_value[index] = block_high_value[index] OR ( NOT curr_locked(x) )
        cluster_started = false
        index++ AT NEXT ITERATION
```



Strip Number:	43	44	45	46	47	48	49	50	51	52
curr_locked	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**0**	**0**	**0**
pass_1_locked	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**
cluster_started	**F**	**F**	**F**	**F**	**T**	**T**	**F**	**F**	**F**	**F**
block_high_value	**U**	**U**	**U**	**U**	**0**	**0**	**0**	**U**	**U**	**U**
index	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**4**	**4**	**4**

Figure 9.12: A diagram showing the logic operation performed by the 'Locked Pass 2' block of the baseline subtraction algorithm. pass_1_locked and block_high_value are the 'Pass 1' locked status and 'Block High Value' signals shown in Figure 9.10, and 'x' refers to strip number. block_high_value is a 64 deep array, with each element referenced by 'index'. As an example, the logic is applied to a small section of an artificial APV frame; the cluster from strip 46 to strip 49 is assigned a block_high_value of 0 ('U' means unassigned), since it is bound on either side by discontinuities.

‘Locked Pass 2’ identifies each cluster of unlocked strips within a frame and assigns them ‘block high’ values of ‘0’ or ‘1’ depending upon whether or not they are bounded on either side by discontinuities. The start of a cluster is found by checking the locked signal from Pass 1 for a transition between a ‘0’ and a ‘1’. If such a transition is found at strip x then the ‘block high’ value for that cluster is initially set to the inverse of the current locked signal for strip $(x - 1)$; it is therefore equal to zero if a discontinuity was found at strip $(x - 1)$ during the current iteration. Pass 1 locked signals are then checked until the next strip with a ‘0’ is found, signifying the end of the group; the ‘block high’ value is set to the logical OR between itself and the inverse of the current locked signal for that strip. This is a (logically) simple method for ensuring that the ‘block high’ associated with a cluster is equal to ‘0’ if and only if the strips at both ends of the cluster were previously locked as discontinuities. The ‘block high’ value for the nth cluster is stored at address n in a 1bit wide, 64 deep memory array. An example of a ‘Locked Pass 2’ logic operation is shown in Figure 9.12.

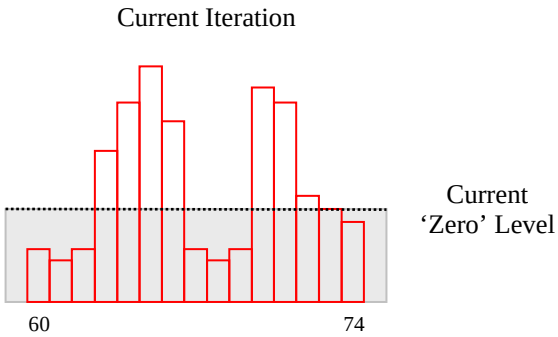
9.3.2.3 Locked Pass 3

Once Pass 2 is complete, the next iteration of the loop begins. ‘Locked Pass 3’ simply performs the logical AND between the (buffered) Pass 1 locked signals and the ‘block high’ value for each set of unlocked strips; an example is shown in Figure 9.13. This automatically locks any strips that are bounded by discontinuities, and provides the previous strip locked values for the current iteration.

After all of the strips have been locked, strip values stored in the second APV data buffer are diverted away from the baseline subtraction loop. It is a characteristic of the algorithm that the discontinuity threshold is subtracted from the non-zero strips

once too often, and so a final logic block re-adds the threshold as required before the data are finally read out.

```
IF (pass_1_locked(x) == 1)

        pass_3_locked(x) = pass_1_locked(x) AND block_high_value[index]
        cluster_started = true

ELSE

        pass_3_locked(x) = pass_1_locked(x)

        IF (cluster_started == true)

                cluster_started = false
                index++
```



Strip Number:	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74
curr_locked	0	0	0	1	1	1	1	0	0	0	1	1	1	0	0
pass_1_locked	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0
cluster_started	F	F	F	F	T	T	F	F	F	F	F	T	T	F	F
block_high_value	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0
index	5	5	5	5	5	5	6	6	6	6	6	6	6	7	7
pass_3_locked	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

Figure 9.13: A diagram showing the logic operation performed by the 'Locked Pass 3' block of the baseline subtraction algorithm. curr_locked, pass_1_locked, pass_3_locked and block_high_value are the current, 'Pass 1', 'Pass 3' locked status and 'Block High Value' signals shown in Figure 9.10 ('x' refers to strip number). block_high_value is a 64 deep array, with each element referenced by 'index'. As an example, the logic is applied to a small section of an artificial APV frame; the first cluster becomes locked, whereas the second does not since it is bound by a discontinuity on only one side. Note that pass_3_locked is equivalent to the previous locked status signal for the next loop iteration.

9.3.3 Baseline Correction Algorithm Logic Utilisation

The logic utilisation for the baseline subtraction algorithm, operating on a single APV, is shown in Table 9.5. As the algorithm effectively replaces Common Mode

subtraction, which frees approximately 30% of the FE FPGA resources in the case of the Median Method, it would be quite feasible to include this hit finding technique in the FED firmware.

Resource:	Used:	Utilisation:
Function Generators	431	2.00%
CLB Slices	216	2.01%
Dffs or Latches	284	1.24%

Table 9.5: Baseline correction algorithm logic usage for one APV.

9.3.4 Simulated Performance of the Baseline Correction Algorithm

The new baseline subtraction algorithm has not yet been implemented in the FE FPGA, as this would require significant changes to the way in which the BE FPGA handles data and throttle signals (denoting the FE buffer status) from the 8 FE Units. With the current cluster finding algorithm, a constant number of clock cycles is required to process the data from each APV; each FE buffer therefore has the same occupancy and event data from each FE FPGA are ready for read out at the same time. In the case of the baseline subtraction algorithm, processing each APV frame may require a different number of iterations. Consequently, the BE FPGA would have to independently monitor the throttle signals from all FE Units and take measures to ensure that the data arriving at different times from each FE FPGA are correctly ordered in the final output event packet. Sufficient FPGA resources are available for these additional tasks, but it was not feasible to modify the firmware in the time available.

Instead, a test bench for the stand alone single APV implementation of the algorithm has been developed, which enables the logic to be simulated with Mentor Graphics ModelSim [97]. Using the simulation it is possible to run the baseline subtraction on arbitrary APV frame data, examples of which are shown in Figure 9.14.

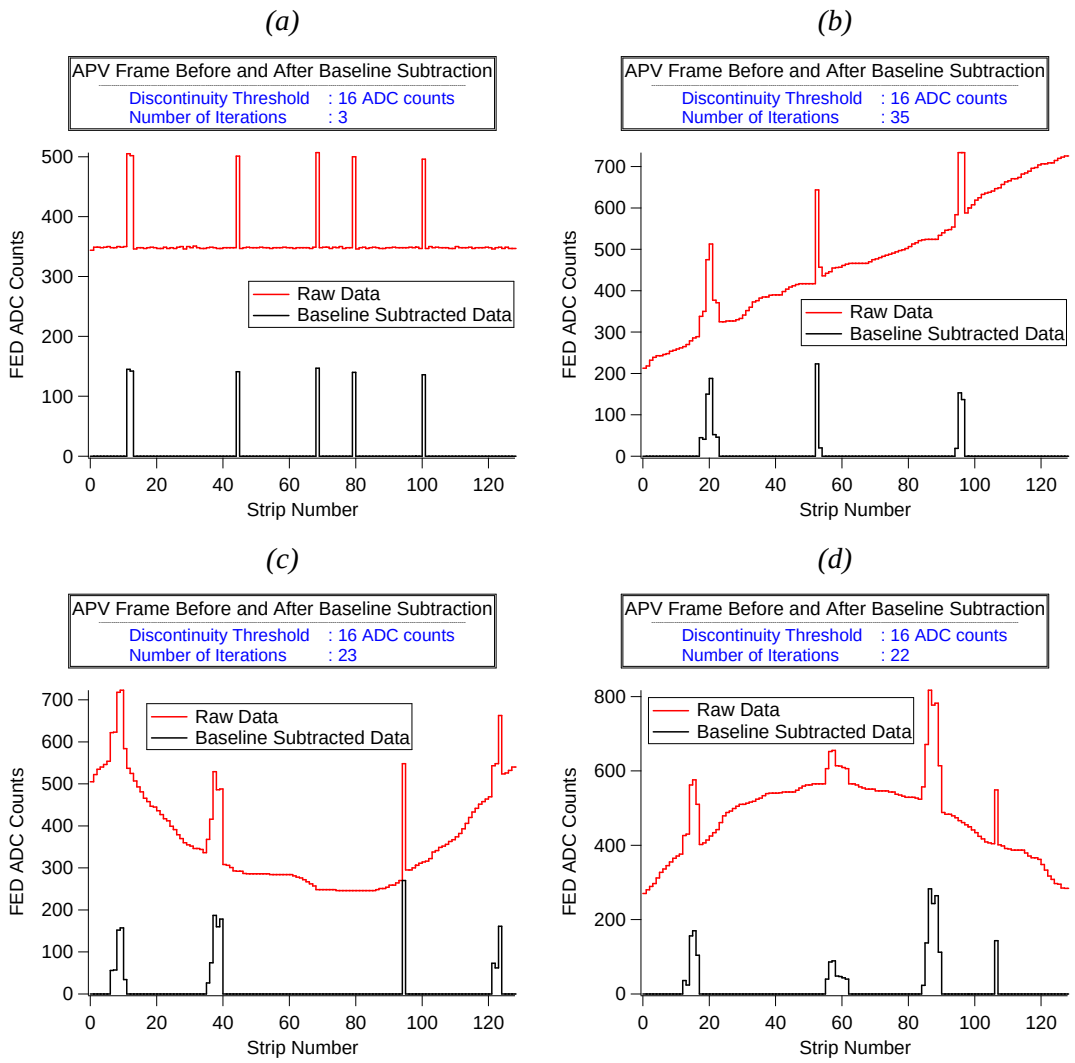


Figure 9.14: Example APV frames, before and after the application of the baseline subtraction algorithm.

All of the plots in Figure 9.14 are generated with a discontinuity threshold of 16 ADC counts. The raw data in Figure 9.14(a) are from a randomly generated APV frame sent to the FED by the FED Tester and read out in Virgin Raw mode. When the background is flat, with a random noise amplitude less than the discontinuity threshold, the algorithm requires approximately 3 loop iterations to subtract the entire baseline. More challenging artificial frames with entirely non-uniform baselines are shown in Figures 9.14(b) to 9.14(c). In each case, the background is successfully removed and the hits are easily identified.

Although the baseline subtraction algorithm functions correctly, the current implementation is in prototype form and a number of issues remain. Currently, there is little support for disabled strips (or strip hardware failures during experimental runs), and large scale baseline splitting may cause the algorithm to fail. However, it is fairly straightforward to add checks for these conditions to the firmware and deal with them as special cases.

Of greater importance is the number of clock cycles required to process an APV frame. If the FED is to operate with 100 kHz L1 triggers, a maximum average of six baseline subtraction iterations are allowed per event. The number of iterations required for a frame is proportional to the range of the APV baseline divided by the discontinuity threshold. For the grossly distorted frames in Figures 9.14(b) to 9.14(d), over 20 discontinuity threshold subtraction loops are necessary. At CMS, it is unlikely that APV baselines will ever exhibit such extreme behaviour and most normal events should be processed within the 6 iteration limit. If they are not, the algorithm can easily be modified to cease after a fixed number of loops (asserting a status bit if the baseline subtraction is incomplete); however, it may be acceptable to run at a reduced trigger rate if baseline distortion were ever to become a significant problem at CMS.

Nevertheless, this new baseline subtraction algorithm provides an effective means for identifying hits in the worst cases of APV baseline distortion, which could easily be adapted for use in any circumstances where a continuous background must be quickly and efficiently removed from a signal.

Conclusions

The Large Hadron Collider is a particle accelerator designed for the study of new physics at the TeV energy scale. As one of two general purpose detectors at the LHC, the Compact Muon Solenoid is designed to identify cleanly the signatures of a diverse range of new physics processes, but is optimised for the discovery of the Higgs boson.

A vital element of CMS is the Silicon Tracker, which provides the information necessary for the measurement of particle momentum and charge, and enables the reconstruction of the physical layout of collision events. The LHC represents a challenging experimental environment, requiring a high performance Tracker with very fine granularity and a fast response. It must survive a hostile radiation environment for the 10 year lifetime of the experiment and the Tracker readout system has to cope with unprecedented output data rates of ~1.4 TB/s.

This thesis presents work related to the testing and development of the Tracker front end readout electronics and in particular the Front End Driver, an essential component which interfaces the detector to the CMS Data Acquisition system and reduces the enormous volume of data from the Tracker to a manageable level.

The APV25 Readout Chip

The APV25 provides the initial on-detector analog readout stage of the CMS Tracker silicon microstrips. Chosen primarily for its high intrinsic radiation tolerance, the chip is implemented using a deep sub-micron Complementary Metal-Oxide-Semiconductor process. Each APV samples, amplifies and shapes the signals from 128 detector channels, internally buffering data samples to accommodate the CMS Level 1 trigger latency and readout delays.

To ensure a Tracker of the highest quality and efficiency, all APVs must be rigorously verified before they are integrated into the final system. An APV25 wafer probing station enables the efficient screening of all 360 chip dies on each wafer received from the manufacturer, with a comprehensive range of functionality and performance tests. After investigating and identifying an initial flaw in the APV manufacturing process, an excellent production yield has been achieved. 131,745 chips have selected for use at CMS, meeting the full requirement of the Tracker. Results obtained from the wafer probing tests demonstrate a high level of APV performance and uniformity.

The APV Emulator

Data can be read out from an APV25 at a maximum rate of one collision event every 7 μs. L1 triggers, prompting readout, occur at a maximum average frequency of 100 kHz, but random fluctuations can lead to trigger separations of less than 7 μs. To accommodate these fluctuations, event data are buffered within each APV. However, if an increased trigger rate persists for an extended period, the APV buffers will overflow. Due to the transmission time between the detector and the Trigger Control System in the CMS counting room, it is impossible to monitor the pipeline occupancy of APVs in the Tracker itself with a fast enough response.

An APV Emulator has been developed to precisely simulate the buffer of an APV and determine its status in less than 50 ns; it will be placed in the CMS counting room such that the transmission time for status signals will be negligible. A “golden” pipeline address for each L1 trigger is also provided by the APVE, enabling synchronisation checks between all of the APVs in the Tracker.

The APVE is implemented as a 6U VME card and the board logic is contained within a Field Programmable Gate Array. Either a real APV25 chip or an FPGA-

based simulation of the APV pipeline may be used to monitor the APV buffer occupancy; the simulation provides greater buffer efficiency, as the internal status of the APV can be accessed directly. A ‘virtual’ TCS and Fast Merging Module are also included in the APVE firmware, enabling the functionality of the board and the magnitude of the APVE-TCS control loop to be evaluated without additional hardware requirements.

The real APV25 chip on the APVE is vulnerable to damage from physical contact, static discharge and moisture in the air; it is therefore necessary to protect it via encapsulation. Glob top was chosen as the most cost effective solution. Investigations revealed that potentially harmful air vacancies are introduced to the glob top when it is initially mixed. Degassing the epoxy resin in a vacuum sealing device prior to application was found to remove all vacancies in the vicinity of the APV chip. A temperature cycling test proved that the encapsulated APV is tolerant of thermal fluctuations.

A software package has been written to drive the APVE, using the Hardware Access Library for VME access. The APVE software provides complete control of the board, with full logging and graceful exception handling. It will soon be integrated into the XDAQ framework for use in the final system at CMS.

Measurements of the fraction of L1 triggers vetoed as a function of the buffer occupancy threshold at which ‘Busy’ is asserted, combined with a C code model of the APV pipeline logic, revealed that the Tracker deadtime introduced by the APVE during normal operation at CMS will be of the order of 0.06%. This should have no significant effect on the overall Tracker deadtime.

The Front End Driver

During normal operation, the Tracker generates data at a rate of ~1.4 TB/s. This is greatly in excess of the processing speed of the CMS DAQ. The Front End Driver reduces the volume of data read out from the Tracker to a level that is compatible with the DAQ through the process of Zero Suppression; identifying strips that contain hit information and discarding the remainder. It also converts analog optical signals from the APVs into the digital electronic format required for the subsequent processing stages.

Each FED accepts ~3 GB/s and outputs ~50 MB/s per percent of Tracker occupancy. Although the average Tracker occupancy is expected to be low, random increases can lead to an output data rate from the FED which exceeds the capacity of the subsequent readout hardware. To absorb occupancy fluctuations, the processed event data are buffered at the Front End and Back End of the FED. However, if the Tracker occupancy remains high for an extended period of time, the buffers will become full and events will be lost.

Simulations have shown that the FE buffers should be able to cope with an occupancy of 9% before L1 triggers are vetoed, while the occupancy supported by the BE buffer is heavily dependent upon the FED readout rate. At the nominal maximum sustained readout rate of 200 MB/s, the simulation predicts that the FED should be able to cope with an occupancy of 4% before events are lost. This is in excess of the highest occupancy predictions for the Tracker and so it should not be necessary for the FED to veto L1 triggers during normal operation; the occupancy limits of real FED hardware are established during the FED performance tests.

FED Industry Testing

The FED is a complex device; manufacturing a board of this complexity is a technically difficult task, and there is great potential for assembly errors. To reduce any possible delay to the FED production schedule, it is important to identify production faults as early as possible.

A custom Acceptance Test has been designed to verify each manufactured board before it leaves the assembly plant. The Acceptance Test software provides a framework for entering the results of any manual procedures and assesses the performance of the FED through a comprehensive series of automated digital and analog functionality checks. All test results are automatically entered into an XML/HTML database. To enable relatively inexperienced plant operatives to test boards effectively, the interface to the Acceptance Test software is as user-friendly as possible; it has simple push-button controls and a clear pass/fail approach to testing, with detailed diagnostics available on demand.

Over 100 FEDs have undergone the Acceptance Test procedure without encountering a single software problem. The Acceptance Test has proved to be highly effective at screening out faulty boards and will be used for the remainder of the FED production cycle.

The FED Tester

It is essential to verify the performance of the FED under realistic CMS operating conditions before it is used for data taking in the final system. A FED Tester has been designed which can precisely simulate the shape and timing of optical signals from the Tracker, and also provide the FED with trigger and control signals identical to those from the final TCS. It can be loaded with any APV frame data or test patterns

and is therefore able to test all data processing aspects of the FED hardware and firmware. 4 FTs are required to drive all 96 channels of a FED.

The FT makes use of CMS Tracker Outer Barrel Analog Optohybrids for the generation of optical signals. As the semiconductor laser diodes in an AOH are temperature sensitive, thermal fluctuations in the laboratory lead to a variation in output optical intensity. Testing the FED requires well-defined optical inputs and so it is necessary to employ an AOH temperature control system. The AOH temperature is stabilised by using a Proportional-Integral-Differential feedback algorithm to apply a controlled heating level to the device in order to maintain it at a set point a few degrees above room temperature. A prototype system was used to quantify the thermal sensitivity of an AOH; it was found that the rate of change of optical output with temperature is -0.60 ± 0.02 MIP/°C (quoted error: standard deviation). The final temperature control system implemented on the FT can maintain temperature stability to ±0.18 °C (maximum deviation) over a range of 20 °C above room temperature.

A complete software package for controlling the FT has been written. This enables 1-5 FTs to be used transparently as a single FT Ensemble when driving the FED, and provides a well-structured ‘description’ object system to organise the multiple FTE configuration parameters.

S-Link Testing

S-Link is a standardised specification for communication between a front end module and the next stage of its readout system. The FED Kit is a prototype of the S-Link64 connection that will be used to readout the FED at CMS.

An experimental setup was developed to enable S-Link data integrity checks using real FED hardware under (approximately) realistic CMS operating conditions. 1 TB of test pattern data were transmitted through a 1.5 m S-Link connection without

receiving errors, corresponding to a probability of ~2.4 x 10^{-11} @ 95% CL that an S-Link word will be transmitted incorrectly. With the current setup, a data verification test lasting ~5 months would be required to guarantee no more than one S-Link transmission error per week of normal CMS operation (assuming no errors were observed). This time could be reduced by operating multiple FEDs in parallel, perhaps using the final CMS DAQ.

A new S-Link Transition Card has also been designed. It provides a buffered interface between the FED and S-Link Transmitter, and routes FED throttle signals to a standard Ethernet connector for transmission to the FMM/TCS. The Transition Card has been used extensively, for all of the FED performance tests, and no data integrity problems have ever been observed.

FED Performance Testing

A complete hardware and software environment for full FED system tests has been developed. To enable efficient control of the system, a unified software architecture has been adopted. All device drivers and utility objects are arranged in a well-organised nested class structure; system configuration, calibration and initialisation can be achieved with a single function call from the UniversalFedToolbox class.

A number of FED performance tests have been implemented using the UniversalFedToolbox control and access framework. The mean FED channel noise was measured at 0.70 ADC counts (an equivalent noise charge of ~220 electrons) and, for an isolated hit (or cluster) received by a single FED channel, the induced crosstalk on the nearest neighbouring channels was found to have an amplitude equal to ~1% of the hit magnitude. FED noise and crosstalk should therefore have no significant effect on the performance of the Tracker.

The ability of FED to correctly identify all input hits has been verified and its performance at high data rates has been assessed. With the current firmware, the FED can cope with a sustained Tracker occupancy of 2.8% and temporary fluctuations up to 7%. However, once the Zero Suppressed Lite mode of operation is in place, the output data rate from the FED at an occupancy of 2.9% should decrease from 217.8 to ~139 MB/s. Consequently, the FED should be able to handle Tracker occupancies in excess of the predicted maximum average levels at CMS (~3%). It has also been demonstrated that the FED is a viable option for the readout of the TOTEM tracker, where the average occupancy will be ~16%, provided that the TOTEM DAQ can support a sustained data rate of 400 MB/s and the 192 APVs normally read out by a single FED are instead distributed among four boards.

The procedures developed to verify the fundamental design and performance of the FED are also used for the routine post production acceptance testing of all boards received from the manufacturer. An optical ramp test verifies the photodiodes in the FED Opto RX components, and a universal readout test checks the functionality of the S-Link connection and the FED throttle; with the latter, between 1×10^{11} and 1×10^{12} FED events have been analysed from over 30 different boards without receiving errors.

Finally, the UniversalFedToolbox has been expanded upon to enable full crate tests, effectively simulating the entire Tracker readout system. An extension of the universal readout test, supporting an arbitrary number of FEDs, has been written. Although this has yet to be exploited to its full potential, simple tests involving 2 FEDs driven in Zero Suppressed mode by 2 FTEs with mixed S-Link and VME readout, and 8 FEDs all read out in Scope mode via VME, have been conducted and no event errors or synchronisation problems have been observed.

The FED Front End Firmware

In Zero Suppressed mode, the FED Front End firmware performs APV frame finding, pedestal subtraction, strip reordering, Common Mode subtraction, data width reduction and cluster (hit) finding.

The CM is that component of the signal offset which is common to all channels of an APV. It is currently estimated using the Median Method, where the CM is simply the median of the 128 APV strip values. However, the Median Method algorithm fails when the strip occupancy is greater than 50% and it has no support for the recently discovered phenomenon of non-uniform CM.

An alternative Mean Method has been developed, which can cope with arbitrary strip occupancies up to ~98%. It also enables the CM calculation to be performed on subsections of an APV frame, effectively dealing with baseline splitting and small-scale non-uniform CM effects. The performance of the Mean Method has been verified in hardware, and a full ORCA simulation is forthcoming.

In anticipation of potential large-scale non-uniform CM effects, an extremely efficient baseline correction algorithm has been developed. It is capable of subtracting any continuous background from an APV frame and a prototype system has proved to be highly effective in simulation. A potential downside is that the algorithm relies upon an iterative process, and if more than 6 iterations were required for baseline subtraction then it would reduce the maximum L1 trigger rate supported by the FED to below 100 kHz; however, this may be acceptable if baseline distortion were to become a significant issue at CMS. The baseline correction algorithm can easily be adapted for use in any circumstances where a continuous background must be quickly and efficiently removed from a signal.

Acronyms

ACE *Advanced Configuration Environment*
ADC *Analogue-to-Digital Converter*
ALICE *A Large Ion Collider Experiment*
AOH *Analog Optohybrid*
AOI *Automated Optical Inspection*
APSP *Analogue Pulse Shape Processor*
APV *Analogue Pipeline (Voltage Mode)*
APV25 *APV in 0.25 μm silicon CMOS technology*
APVE *APV Emulator*
APVMUX *APV Multiplexer*
ATLAS *A Toroidal LHC Apparatus*

BE *Back End*
BGA *Ball Grid Array*

CCU *Communication Control Unit*
CDF *Common Data Format*
CERN *The European Laboratory for Particle Physics Research*
CF *Compact Flash*
CL *Confidence Limit*
CLB *Configurable Logic Block*
CM *Common Mode*
CMOS *Complementary Metal-Oxide-Semiconductor*
CMS *Compact Muon Solenoid*
CMT *Common Mode Type*
CP *Charge-Parity*
CPU *Central Processing Unit*
CRC *Cyclic Redundancy Check*
CR-RC *Capacitor-Resistor Resistor-Capacitor*
CTE *Coefficient of Thermal Expansion*

DAC *Digital-to-Analogue Converter*
DAQ *Data Acquisition (System)*
DC *Direct Current (constant level)*
Dff *Differential Flip Flop*

ECAL *Electromagnetic Calorimeter*
EM *Electromagnetic*
EPROM *Erasable Programmable Read Only Memory*

FE_____*Front End*
FEC_____*Front End Controller*
FED_____*Front End Driver*
FEH_____*Front End Hybrid*
FEM_____*Front End Module*
FEMB_____*Front-end Motherboard*
FIFO_____*First In First Out*
FMM_____*Fast Merging Module*
FPGA_____*Field Programmable Gate Array*
FT_____*Fed Tester*
FTE_____*Fed Tester Ensemble*

GEM_____*Gas Electron Multiplier*
GIII_____*Generic III*
GTCS_____*Global TCS*

HAL_____*Hardware Access Library*
HCAL_____*Hadronic Calorimeter*
HEP_____*High Energy Physics*
HLT_____*High Level Triggers*
HTML_____*Hypertext Markup Language*

IC_____*Integrated Circuit* or *Imperial College*
I^2C_____*Inter-IC*
ILD_____*Inter-Level Dielectric*

JTAG_____*Joint Test Action Group*

L1_____*Level 1*
LDC_____*Link Destination Card*
LED_____*Light Emitting Diode*
LEP_____*Large Electron-Positron Collider*
LHC_____*Large Hadron Collider*
LHCb_____*LHC Beauty Experiment*
LSC_____*Link Source Card*
LTCS_____*Local TCS*
LVCMOS____*Low Voltage CMOS*
LVDS_____*Low Voltage Differential Signalling*

MIP_____*Minimum Ionising Particle*
MUX_____*Multiplexer*

NT_____*Noise Threshold*

ORCA_____*Object-orientated Reconstruction for CMS Analysis*

PC_____*Personal Computer*
PCB_____*Printed Circuit Board*
PCI_____*Peripheral Component Interconnect*
PCI-X_____*PCI Extended*
PID_____*Proportional-Integral-Differential*

QGP_____*Quark Gluon Plasma*

RAL_____*Rutherford Appleton Laboratory*
RAM_____*Random Access Memory*
RJ-45_____*Registered Jack-45*
ROMB_____*Read-out Motherboard*
RUWG_____*Readout Unit Working Group*
RX_____*Receiver*

SEQSI_____*Sequencer for use in Silicon Readout Investigation*
S-Link _____*Simple Link*
S-Link64____*Simple Link – 64bit extension*
SM_____*Standard Model*
SUSY_____*Supersymmetry*

TAP_____*Test Access Port*
TCS_____*Trigger Control System*
TDI_____*Test Data In*
TDO_____*Test Data Out*
TEC_____*Tracker End Cap*
TIB_____*Tracker Inner Barrel*
TID_____*Tracker Inner Disks*
TOB_____*Tracker Outer Barrel*
TOTEM_____*Total and Elastic Measurement*
TTC_____*Trigger, Timing and Control*
TTC*ci*_____*TTC Common Interface*
TTC*ex*_____*TTC Laser Encoder/Transmitter*
TTC*rx*_____*TTC Receiver*
TTC*tx*_____*TTC Laser Transmitter*
TTC*vi*_____*TTC VME Bus Interface*

USB_____*Universal Serial Bus*
UTP5_____*Unshielded Twisted Pair 5*

VHDL_____*VHSIC Hardware Description Language*
VHSIC_____*Very High Speed Integrated Circuit*
VI_____*Virtual Instrument*
VME_____*Versa Module Europa*
VME64X____*Versa Module Europa – 64bit extension*

XDAQ_____*Cross Platform DAQ Framework*
XML_____*Extensible Markup Language*
XSL_____*Extensible Stylesheet Language*

References

[1] **The Large Hadron Collider Homepage**, http://lhc.web.cern.ch/lhc/

[2] The ALEPH, DELPHI, L3 and OPAL Collaborations, **Search for the Standard Model Higgs Boson at LEP**, Phys. Lett., B 565 (2003) 61-75

[3] R. Kinnunen, **Higgs Physics at LHC**, CERN/CMS/CR/2002/020

[4] K. Lassila-Perini, **Higgs Physics at the LHC**, CERN/CMS/CR/2001/018

[5] G. Wrochna, **Physics at LHC**, Acta Physica Polonica B, Vol. 33, No. 11 (2002)

[6] M. Dittmar, **Searching for the Higgs and other Exotic Objects**, CERN/CMS/CR/1999/009

[7] C. E. Wulz, **CMS Physics Overview**, CERN/CMS/CR/2001/016

[8] J. W. Rohlf, **Physics Reach with CMS at High and Super-High Luminosities**, Eur. Phys. J., C 34 (2004) s221-s239

[9] D. Denegri, V. Drollinger, R. Kinnunen, K. Lassila-Perini, S. Lehti, F. Moortgat, A. Nikitenko, S. Slabospitsky, N. Stepanov, **Summary of the CMS Discovery Potential for the MSSM SUSY Higgses**, CERN/CMS/NOTE/2001/032

[10] M. Konecki, **Prospects for CP Violation Measurements with ATLAS and CMS**, Nucl. Phys. B, Proc. Suppl. 99 (2001) 220-7

[11] G. Wrochna, **Heavy Ion Physics with CMS Detector**, CERN/CMS/CR/2002/012

[12] CMS Collaboration, **CMS, The Compact Muon Solenoid: Technical Proposal**, CERN/LHCC/94/38

[13] CMS Collaboration, **CMS, The Tracker System Project: Technical Design Report**, CERN/LHCC/98/006

[14] CMS Collaboration, **Addendum to the CMS Tracker TDR**, CERN/LHCC 2000/016

[15] CMS Collaboration, **CMS, The Electromagnetic Calorimeter Project: Technical Design Report**, CERN/LHCC/97/033

[16] CMS Collaboration, **CMS, The Hadron Calorimeter Project: Technical Design Report**, CERN/LHCC/97/031

[17] CMS Collaboration, **CMS, The Muon Project: Technical Design Report**, CERN/LHCC/97/032

[18] CMS Collaboration, **CMS: The Magnet Project, Technical Design Report**, CERN/LHCC/97/010

[19] LHCC LHC Experiments Committee, **The Trigger and Data Acquisition Project, vol. II: Data Acquisition and High Level Trigger**, CERN/LHCC/2002/026

[20] J. Varela, **CMS L1 Trigger Control System**, CERN/CMS/NOTE/2002/033

[21] C. Seez, **The CMS Trigger System**, Eur. Phys. J., C 34 (2004) s151-s159

[22] L. L. Jones, **APV25 User-Guide Manual**, Version 2.2, 2001 http://www.te.rl.ac.uk/med/projects

[23] M. Raymond et al., **The CMS Tracker APV25 0.25 µm CMOS Readout Chip**, Proceedings of the 6th Workshop on Electronics for LHC Experiments, CERN/LHCC/2000/041

[24] M. French et al., **Design and Results from the APV25, a Deep Submicron CMOS Front-End Chip for the CMS Tracker**, Nucl. Instrum. Methods Phys. Res., A 466 (2001) 359-65

[25] P. Murray, **APVMUX User Guide Version 1.0**, 2000 http://www.te.rl.ac.uk/med/projects

[26] **The Analog Optohybrid Homepage**, http://wwwhephy.oeaw.ac.at/u3w/f/friedl/www/aoh/

[27] F. Vasey, **CMS Tracker Optical Readout Link Specification**, http://tilde-vasey.home.cern.ch/~vasey/specs/readout/readout_system.pdf

[28] J. Coughlan et al., **The Front-End Driver card for the CMS Silicon Strip Tracker Readout**, Proceedings of the 8th Workshop on Electronics for LHC Experiments, CERN/LHCC/2002/034

[29] J. Coughlan et al., **The CMS Tracker Front-End Driver**, Proceedings of the 9th Workshop on Electronics for LHC Experiments, CERN/LHCC/2003/055

[30] **CERN S-Link Homepage**, http://hsi.web.cern.ch/HSI/s-link/

[31] G. Anelli et al., **Total Dose Behaviour of Submicron and Deep Submicron CMOS Technologies**, Proceedings of the 3rd Workshop on Electronics for LHC Experiments, CERN/LHCC/97/060

[32] E. Noah et al., **Total Dose Irradiation of a 0.25 μm Process**, Proceedings of the 6th Workshop on Electronics for LHC Experiments, CERN/LHCC/2000/041

[33] T. Cornwell, A. Bridle, **Deconvolution Tutorial**, http://www.cv.nrao.edu/~abridle/deconvol/deconvol.html

[34] S. Gadomski et al., **The Deconvolution Method of Fast Pulse Shaping at Hadron Colliders**, Nucl. Instrum. Methods Phys. Res., A 320 (1992) 217-227

[35] N. Bingefors et al., **A Novel Technique for Fast Pulse-Shaping using a Slow Amplifier at LHC**, Proceedings of the 6th European Symposium on Semiconductor Detectors, CERN/PPE/92/070

[36] Geoff. Hall, Imperial College London, **The Deconvolution Method of Pulse Shaping**, March 2000 (unpublished)

[37] Philips Semiconductors, **The I^2C-Bus Specification**, http://www.semiconductors.philips.com/markets/mms/protocols/i2c/

[38] **National Instruments LabVIEW Homepage**, http://www.ni.com/labview/

[39] R. Bainbridge, **Influence of Highly Ionising Events on the CMS APV25 Readout Chip**, CERN/THESIS/2004/032

[40] R. Kawahara et al., **The Effectiveness of IDDQ and High Voltage Stress for Burn-in Elimination [CMOS Production]**, IEEE International Workshop on IDDQ Testing, 1996

[41] M. Raymond et al., **APV25 Production Testing and Quality Assurance**, Proceedings of the 8th Workshop on Electronics for LHC Experiments, CERN/LHCC/2002/034

[42] P. Barrillon et al., **Production Testing and Quality Assurance of CMS Silicon Microstrip Tracker Readout Chips**, CERN/CMS/NOTE/2004/016

[43] P. Barrillon et al., **Final Results from the APV25 Production Wafer Testing**, Proceedings of the 11th Workshop on Electronic For LHC and Future Experiments, IC/HEP/05/9, http://www.imperial.ac.uk/research/hep/preprints/05-9.pdf

[44] G. Iles et al., **The APVE Emulator to Prevent Front-End Buffer Overflows Within the CMS Silicon Strip Tracker**, Proceedings of the 8th Workshop on Electronics for LHC Experiments, CERN/LHCC/2002/034

[45] A. Racz et al., **Trigger Throttling System for CMS DAQ**, Proceedings of the 6th Workshop on Electronics for LHC Experiments, CERN/LHCC/2000/041

[46] Ph. Farthouat, P. Gällnö, **TTC-VMEbus Interface**, http://ttc.web.cern.ch/TTC/TTCviSpec.pdf

[47] B. Taylor, **TTC laser transmitter (TTCex, TTCtx, TTCmx) User Manual**, http://ttc.web.cern.ch/TTC/TTCtxManual.pdf

[48] F. Vasey, **CMS Tracker Optical Control Link Specification**, http://tilde-vasey.home.cern.ch/~vasey/specs/control/control_system.pdf

[49] C. Paillard et al., **The CCU25: A Network Oriented Communication and Control Unit Integrated Circuit in a 0.25 μm CMOS Technology**, Proceedings of the 8th Workshop on Electronics for LHC experiments, CERN/LHCC/2002/034

[50] A. Racz et al., **The Final Prototype of the Fast Merging Module (FMM) for Readout Status Processing in CMS DAQ**, Proceedings of the 10th Workshop on Electronics for LHC and Future Experiments, CERN/LHCC/2004/030

[51] **VMEbus International Trade Association Homepage**, http://www.vita.com/

[52] Xilinx, **Virtex-II Platform FPGA Handbook**, http://www.xilinx.com

[53] **Epoxy Technology Homepage**, http://www.epotek.com/

[54] C. Schwick, **Hardware Access Library User's Manual**, http://cmsdoc.cern.ch/~cschwick/software/documentation/HAL/

[55] **World Wide Web Consortium Homepage**, http://www.w3.org/

[56] **log4cplus Homepage**, http://log4cplus.sourceforge.net/

[57] **XDAQ Homepage**, http://xdaq.web.cern.ch/xdaq/

[58] N. Marinelli, **APV Logic Simulations**, CERN/CMS/NOTE/1999/028

[59] A. Caner et al., **On Balancing Data Flow from the Silicon Tracker**, Draft Note (2001) http://cern.ch/tomalini/readout.pdf

[60] I. Tomalin et al., **Expected Data Rates from the Silicon Strip Tracker**, CERN/CMS/NOTE/2002/047

[61] B. Gannon, **Compact Muon Solenoid (CMS) Front End Driver (FED) Front-End FPGA Technical Description**, Version 1.3, http://www.te.rl.ac.uk/esdg/cms-fed/qa_web/firmware.html

[62] S. Taghavi, **Compact Muon Solenoid (CMS) Front End Driver (FED) Back End FPGA Technical Description**, Version 1.6, http://www.te.rl.ac.uk/esdg/cms-fed/qa_web/firmware.html

[63] Xilinx, **System ACE CompactFlash Solution**, Datasheet, Version 1.5, http://direct.xilinx.com/bvdocs/publications/ds080.pdf

[64] E. Freeman, **Notes on CMSdelay chip**, http://www.te.rl.ac.uk/esdg/cms-fed/qa_web/firmware.html

[65] E. Freeman, **CMS FED VME interface chip**, http://www.te.rl.ac.uk/esdg/cms-fed/qa_web/firmware.html

[66] CMS Collaboration, **CMS Tracker Optical Readout Link Specification Part 5: Analogue Opto-Receiver Module**, Version 1.7, CERN/CMS/TK/ES/0008, http://www.te.rl.ac.uk/esdg/cms-fed/hardware/datasheets/datasheets.html

[67] CMS Collaboration, **CMS Tracker Optical Readout Link Specification Part 5.1: Receiving Amplifier**, Version 4.9, CERN/CMS/TK/ES/0009, http://www.te.rl.ac.uk/esdg/cms-fed/hardware/datasheets/datasheets.html

[68] Elantec, **EL2140C/2141C 150 MHz Differential Twisted Pair Driver**, Datasheet, http://www.te.rl.ac.uk/esdg/cms-fed/hardware/datasheets/datasheets.html

[69] Analog Devices, **10-Bit, 40/65/80/105 MSPS 3 V Dual A/D Converter AD9218**, Datasheet, http://www.te.rl.ac.uk/esdg/cms-fed/hardware/datasheets/datasheets.html

[70] Analog Devices, **12 Channel, 8-Bit TrimDACs with Power Shutdown AD8802/AD8804**, Datasheet, http://www.ortodoxism.ro/datasheets/analogdevices/AD8804ARU.pdf

[71] E. Corrin, **Development of Digital Readout Electronics for the CMS Tracker**, PhD thesis for the University of London (2002), http://corrin.home.cern.ch/corrin/downloads/thesis.pdf

[72] National Semiconductor, **BGA (Ball Grid Array)**, Application Note 1126, http://www.national.com/an/AN/AN-1126.pdf

[73] J. Coughlan et al., **The Manufacture of the CMS Tracker Front-End Driver**, Proceedings of the 10th Workshop on Electronics for LHC and Future Experiments, CERN/LHCC/2004/030

[74] **Itochu Homepage**,
http://itochu.metacanvas.com/EN/

[75] **ERSA Homepage**,
http://www.ersa.de/en/

[76] **JTAG Technologies Homepage**,
http://www.jtag.com/

[77] Corelis, **Boundary-Scan Tutorial**,
http://www.corelis.com/products/Boundary-Scan_Tutorial.htm

[78] **Fed 9U Software Homepage**,
http://fed9u.web.cern.ch/fed9u/

[79] National Semiconductor, **LM82 Remote Diode and Local Digital Temperature Sensor with Two-Wire Interface**, Datasheet,
http://cache.national.com/ds/LM/LM82.pdf

[80] Maxim, **Low-Cost, µP Supervisory Circuits**, Datasheet,
http://pdfserv.maxim-ic.com/en/ds/MAX705-MAX813L.pdf

[81] G. Iles et al., **A Testing Device for the CMS Tracker Front End Driver Cards**, Proceedings of the 9th Workshop on Electronics for LHC Experiments, CERN/LHCC/2003/055

[82] M. Friedl, **Analog Optohybrids CMS Tracker TOB/TEC Technical Specification**, Version 1.11,
http://wwwhephy.oeaw.ac.at/u3w/f/friedl/www/aoh/aoh_tobtec_technical_111.pdf

[83] Analog Devices, **12-Bit, 300 MSPS High Speed TxDAC+® D/A Converter AD9753**, Datasheet,
http://www.analog.com/UploadedFiles/Data_Sheets/54507129241804AD9753_b.pdf

[84] Analog Devices, **325 MHz, 8 × 8 Buffered Video Crosspoint Switches AD8108/AD8109**, Datasheet,
http://www.analog.com/UploadedFiles/Data_Sheets/21801249AD8108_9_b.pdf

[85] OpenCores Organization**, Specification for the: WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores**, Revision B.3,
http://www.opencores.org/projects.cgi/web/wishbone/wbspec_b3.pdf

[86] G. Cervelli et al., **Radiation Tolerant Linear Laser Driver IC Reference and Technical Manual**, Version 4.1,
http://proj-lld.web.cern.ch/proj-lld/Manual4.3.pdf

[87] C. Williams, **Feedback and Temperature Control**, http://newton.ex.ac.uk/teaching/CDHW/Feedback/

[88] H. van der Bij et al., **S-LINK, a Data Link Interface Specification for the LHC Era**, IEEE Trans. Nucl. Sci. 44 (1997)

[89] R. McLaren et al., **The S-LINK Interface Specification**, CERN/ALICE/INT/1995/35 http://hsi.web.cern.ch/HSI/s-link/spec/spec/s-link.pdf

[90] R. McLaren et al., **The S-LINK 64 bit Extension Specification: S-LINK64**, https://edms.cern.ch/file/249683/2/slink64_v20.pdf

[91] **CMS DAQ Horizontal Pages**, http://cmsdoc.cern.ch/cms/TRIDAS/horizontal/

[92] V. Brigljevic et al., **FEDkit: a Design Reference for CMS Data Acquisition Inputs**, Proceedings of the 9th Workshop on Electronics for LHC Experiments, CERN/LHCC/2003/055

[93] D. Gigi, **Universal PCI board**, http://dgigi.home.cern.ch/dgigi/fed_kit/uni_pci_doc.pdf

[94] **PCI-SIG Homepage**, http://www.pcisig.com/specifications

[95] E. Cano, D. Gigi, **FEDKIT User's Manual and Programmer's Manual**, Version beta 0.9, http://cano.home.cern.ch/cano/fedkit/fedkit-0.9.pdf

[96] **SBS Technologies Homepage**, http://www.sbs.com/

[97] **Mentor Graphics Homepage**, http://www.mentor.com/

[98] Fairchild Semiconductor, **74LVT16245 • 74LVTH16245 Low Voltage 16-Bit Transceiver with 3-STATE Outputs**, Datasheet, http://www.fairchildsemi.com/ds/74/74LVTH16245.pdf

[99] National Semiconductor, **DS90LV018A 3V LVDS Single CMOS Differential Line Receiver**, Datasheet, http://cache.national.com/ds/DS/DS90LV018A.pdf

[100] National Semiconductor, **DS90LV047A 3V LVDS Quad CMOS Differential Line Driver**, Datasheet, http://cache.national.com/ds/DS/DS90LV047A.pdf

[101] National Semiconductor, **LM1117/LM1117I 800mA Low-Dropout Linear Regulator**, Datasheet, http://cache.national.com/ds/LM/LM1117.pdf

[102] **Express Circuits Ltd Homepage**, http://www.express-circuits.co.uk/

[103] **Cemgraft Electronic Manufacturing Homepage**, http://www.cemgraft.co.uk/

[104] P. Moreira et al., **TTCrx Reference Manual**, Version 3.10, http://ttc.web.cern.ch/TTC/TTCrx_manual3.10.pdf

[105] J. Coughlan et al., **User Requirements Document for the Final FED of the CMS Silicon Strip Tracker**, Version 0.50, CERN/CMS/NOTE/2001/043

[106] I. Tomalin, **On Calibration, Zero Suppression Algorithms and Data Format for the Silicon Tracker FEDs**, CMS/IN/2001/025, http://hep.physics.ucsb.edu/people/affolder/in01_025.pdf

[107] **TOTEM Homepage**, http://totem.web.cern.ch/Totem/

[108] LHCC LHC Experiments Committee, **TOTEM Technical Design Report**, CERN/LHCC/2004/002

[109] **WaveMetrics Homepage**, http://www.wavemetrics.com/

[110] M. Pearson, **FED VME in USC55**, Version 0.92, http://mrp.home.cern.ch/mrp/documents/FED_Crates_USC55.pdf

[111] S. Paoletti, **Study of One CMN TOB Module Using Testbeam Software**, Tracker Electronics & Hybrids Meeting Report, February 2004, http://agenda.cern.ch/askArchive.php?base=agenda&categ=a04328&id=a04328s1t10%2Fdocuments%2Fcmn-apv3.pdf

[112] **ORCA Homepage**, http://cmsdoc.cern.ch/orca/

[113] C. Roland, **Track Reconstruction in Heavy Ion Events using the CMS Tracker**, CERN/CMS/NOTE/2006/031

[114] **DataThief Homepage**, http://www.datathief.org/

[115] F. Turner et al., **Investigation of Selected Baseline Removal Techniques as Candidates for Automated Implementation**, Applied Spectroscopy, Volume 59, Number 5, May 2005, pp. 545-574(30)