The Pre-Processor Module (PPM)

for the ATLAS Level-1 Calorimeter Trigger

Version 1.0

Apr.2009

The ATLAS group at the KIP Heidelberg

Contact: P. Hanke (hanke@kip.uni-heidelberg.de)

Abstract

The Pre-Processor of the ATLAS Level-1 Calorimeter Trigger prepares pipe-lined digital data as input to subsequent processors, which identify objects for the decision making on Level-1. The Pre-Processor system receives appr. 7200 analog signals from the entire calorimetry of the ATLAS experiment. The calorimetric detectors are divided into an electro-magnetic and a hadronic part along depth. Lateral division for the purpose of triggering gives cells of size 0.1 * 0.1 along azimuth and pseudo-rapidity. The signals represent deposits of 'transverse energy' in the electro-magnetic and the hadronic part respectively.

The Pre-Processor has to attribute digital values of transverse energy to the 'bunch-crossing', in which the proton-proton collision took place. The values are transmitted in 'real-time' as serial data-streams to the object-defining processors to obtain a trigger-decision without introduction of 'dead-time'. Furthermore, trigger data are made available for readout, since the signals of trigger cells are not accessible otherwise. Control and monitoring of the trigger performance is a major task in the experiment.

This document describes the most important entity of the Pre-Processor electronics - the Pre-Processor Module. The system comprises a large number (appr.128) of such identical modules covering the trigger space spanned by the ATLAS calorimeters.

1. Introduction

1.1 Overview.

The Pre-Processor Module (PPM) is the essential building block forming the Pre-Processor system. It receives analog signals from the ATLAS calorimeters on front-panel connectors. The differential signals are converted to uni-polar form and transmitted further for digitisation. In parallel, a discriminator checks the crossing of a threshold to determine a time-reference. This reference can be used later to relate an 'accepted event' to the 'bunch-crossing' in the LHC storage ring. 'Bunch crossings' occur repetitively at a rate of 40.08 MHz.

The actual signal processing is located on a Multi-Chip-Module (PPrMCM). Digitisation in a 'Flash Analog-to-Digital Converter' (FADC) to 10 bits with a sampling frequency of 40.08 MHz is the first step. Fine adjustment of the digitisation strobe to the peak of the analog signal is achieved with a chip, that allows a delay-setting in steps of 1 nsec across the 25 nsec LHC clock period. The following processing, which is very specific to the experiment, takes place in a self-designed 'Application Specific Integrated Circuit' (PPrASIC). Three data streams emanate as parallel bit-patterns from the PPrASIC:

1. Two streams represent energy deposits on a 0.1*0.1 space-grid defined in units of azimuthal angle and pseudo-rapidity. These values of energy-deposit are transmitted to the 'Cluster Processor' (CP) to identfy 'small' objects like 'photons', 'isolated electrons', 'tau-leptons' and others.
2. One other stream represents energy deposits on a 0.2*0.2 space-grid. Those data are transmitted to the 'Jet/Energy-Sum Processor' (JEP), where 'larger' objects like 'particle jets' are identified. In addition, energy sum-values are built to be used as e.g. 'missing transverse energy' signature.

Transmission of the data as parallel streams to the separate crate-systems (CP, JEP) is technically almost impossible due to the bulk-cabling involved. Hence, serialisation to high-frequency is implemented on the PPrMCM to maintain pipe-lined transfer in 'real-time'. These high-frequency data are transported on the PPM to the backplane periphery, where the cable connections are located.

The analog calorimeter trigger input signals are not accessible to the 'Data AQcuisition system' (DAQ) of the experiment. Hence, readout of the digitised values at the Pre-Processor is a mandatory task. Pipe-line memories store values at two points in the pre-processing chain. A dedicated system was developed to retrieve data to DAQ upon positive Level-1 decisions ('L1-Accept'). The system is implemented in a 'Field Programmable Gate Array', called 'Readout Merger FPGA'. It collects the relevant data from the distributed locations on the PPM, formats and provides parallel output to the backplane. An auxiliary module on the same slot, located in the rear of the crate, serialises the data stream for high-speed transmission to a ReadOut Driver (ROD). The method provides enough band-width to transport the data to the ATLAS DAQ system.

The Pre-Processor Module requires also a 'standard' interface to computer infrastructure for setting-up, debugging and control. For this purpose, the module is configured as a 'standard VME slave'. The entire address space of the module is accessible through VME.

1.2 Scope.

This document describes the implementation of the Pre-Processor Module. The modules built provide the full functionality as required for ATLAS.

The structure of the document is as follows:

- Section 2 outlines the functional blocks of the module as intended at the time of specification.
- Section 3 describes the corresponding implementation on the level of the 'Printed Circuit Board' (PCB) hardware.
- The 'software' view of the module, as seen from the VME and/or the DAQ(ROD)-side, is given in Section 4, where the content of the data space is defined.
- Technical details concerning the design process, the manufacturing and the testing of the modules are described in Section 5 followed by summary in Section 6.
- Tabular information, like connector pinnings, is given in the Appendices.

1.3 Related projects.

It should be noted, that the system described here, is embedded in a collaborative effort, which ranges from the calorimetry input across the Pre-Processor to the Level-1 object-finding Processors (CP and JEP) and finally to the decision-making 'Central Trigger Processor' (CTP) in the 'ATLAS Calorimeter Level-1 Trigger System'. The Pre-Processor module itself carries sub-components, where electronic functionality is integrated to a very high degree. Due to their complexity, these components (the PPrMCM and the PPrASIC) were specified and reviewed separately. Hence, their detailed description is given in separate documentation. However, only because of the high degree of integration, a system configuration could be achieved, which constitutes a viable solution in terms of size, reliability and modularity.

Related overview documentation comprises:

- The Technical Design Report on the First Level Trigger (TDR) **[Ref. 1]**.
- The Timing, Trigger and Control system for ATLAS (TTC) **[Ref. 2]**.

2. The functional blocks on the Pre-Processor Module.

The functional requirements and their allocation to building blocks on the PPM is outlined in this section. The realisation on the 'printed-cricuit boards'(PCBs) is described in the section following. The Pre-Processor module is designed to pre-process 64 analog input signals from the calorimeters. This entails, that the basic hardware-structure of the module is given by the processing chain for those 64 channels.

2.1 The 'real-time' signal path to the Trigger Processors.

The steps of processing for an individual channel are depicted in the figure below. The diagram reveals, that almost all 'algorithmic' functionality is concentrated on the PPrMCM (and the PPrASIC thereon). However, the analog signal handling as well as all control- and data-access is implemented on the PCB level. The following subsections address these issues.

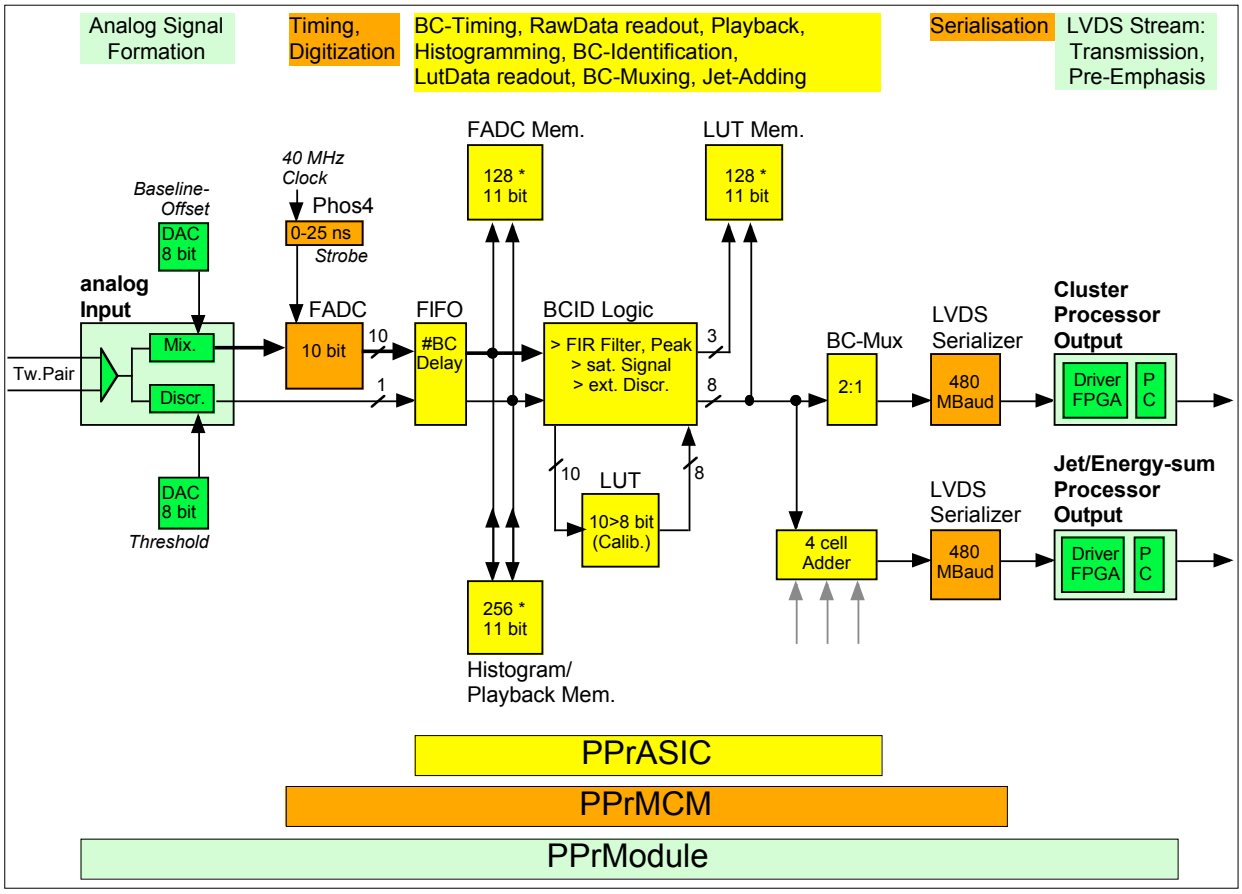

Figure 1 : Block diagram of pre-processing for one calorimeter trigger channel.

2.1.1 The input and conditioning of analog signals.

Every analog signal is taken from the front-panel connector to a differential line-receiver, where a single-ended signal is formed. Most signals are charge-balanced. Bipolar shaping gives a 'fast' half-wave used for triggering purposes, which is balanced by a 'slow' half-wave of opposite polarity but identical area.

All signals from 'Liquid Argon' calorimeters, which deliver about 75% of analog trigger inputs, are of this type. Only the 'Tile' calorimeter delivers unbalanced pulses typical for Photo-Multiplier tubes.

A voltage-level can be applied to any input-signal to map it into the digitisation window of the FADC. A programmable 'Digital-to-Analog Converter (DAC)' produces a DC-level output, which is added onto the signal. One DAC output is allocated to each signal-channel for individual adjustment. Also, the analog signal is rescaled an Operational Amplifier to match the linear amplitude-range to the FADC-window.

The conditioned analog signal is fanned out to two destinations:

1. The analog signal is routed directly to the PPrMCM, where it is digitised in the FADC.
2. In addition, the signal is viewed by a comparator, which marks the threshold crossing in time by a the 'rising edge' of its digital output. The threshold of the comparator is adjustable through another programmable DAC for each channel individually. The output signal goes also to the PPrMCM, where the PPrASIC marks the correct LHC clock-cycle. This constitutes one of the three implemented possibilities to perform 'Bunch-Crossing IDentification' (BCID).

The functions of the analog input stage are outlined below.

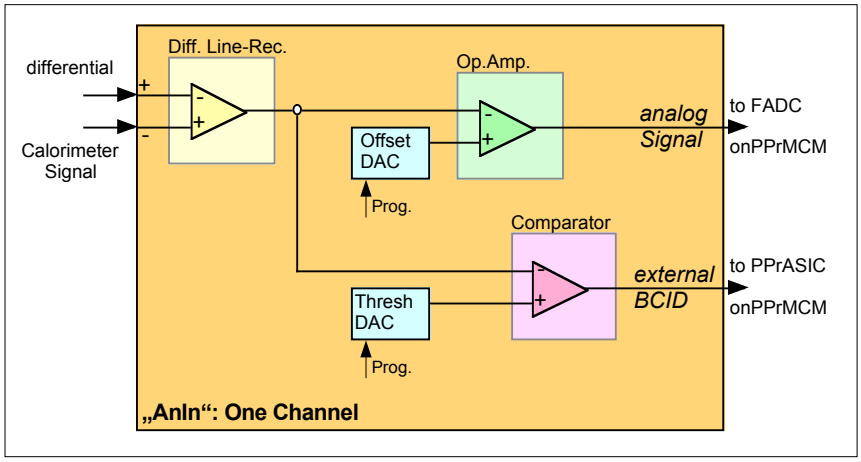

Figure 2 : Block Diagram of 'signal conditioning' and 'external BCID'.

The 'external BCID' comparator raises a digital output-level to logical '1' when the input crosses the programmable threshold in time. The other branch attenuates the signal to match the width of the FADC-window. Furthermore, the signal is 'lifted' to the FADC-input by addition of an appropriate DC-level. The result is shown below.

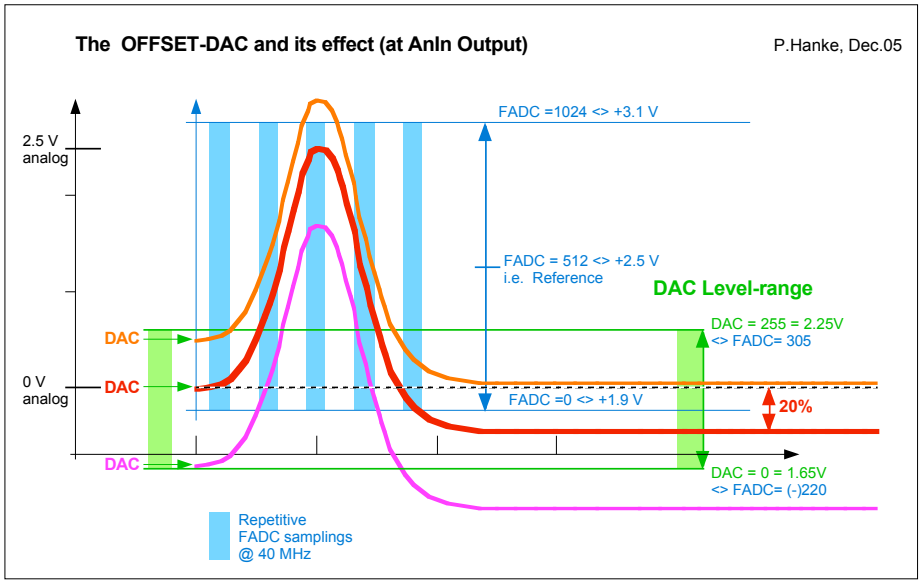

Figure 3 : Mapping an analog calorimeter signal into the FADC.

It should be noted, that the example depicted shows a 'Liquid-Argon' signal, which is charge-balanced. The first part shaped 'narrow' (50 nsec peaking-time) is used for triggering. It is followed by a 'wide' undershoot (400 nsec length) of equal area (balancing integral). Thus, the net charge-flow is nil on this signal-line.

The figure indicates the range of DC-levels that can be applied through the programmable DAC. Three cases are shown:

a. the level lies well below the bottom of the FADC-window. Hence, the digitisation applies an effective threshold. This setting suppresses excessive 'noise', if present on analog lines.
b. the level moves the signal's baseline just above the bottom of the FADC-window. This is a very likely setting at LHC-start, because it allows measurement of a noise-distribution on both sides of the base-line yet leaves also maximal digitisation-range for threshold-setting further downstream.
c. the DAC is set to its top-end. With such a setting the full swing of a LAr calorimeter signal can be digitised including the 20% undershoot amplitude. There are technical aspects, where this is useful (e.g. the shape of the undershoot tells about electron-absorption in the Liquid Argon).

An essential task in pre-processing is the identification of the proton-proton collision in time, i.e. on the clock-raster defined by the bunch-structure of the LHC. All-together, three methods for BCID are implemented in the design of the PPM. The purely 'digital' methods (A, B) are part of the PPrASIC functionality. For completeness sake, all three methods are briefly described here:

A. The FIR-Filter BCID method.

A pipe-line in the PPrASIC stores five consecutive FADC-samples, applies weights by multipliing the samples in the pipe with pre-defined coefficients and sums up the resulting values. Thus, the integral over a calorimeter signal is derived, improving transverse energy-resolution and minimising noise-contributions. A subsequent 'peak-finder' identifies the time-slice with the maximum value and attributes the 'pulse-integral' (energy value) to this LHC clock-cycle. The working range of the method spans from small signals (few GeV equivalent) over the linear signal range (up to ca. 250 GeV) into 'near' saturation. The limit is reached, when several consecutive samples are 'clipped' to the maximum FADC-value.

B. The 'digital BCID for saturated pulses'.

Two consecutive FADC-samples are compared to a 'low' and a 'high' threshold in the PPrASIC making use of the finite 'peaking-time' (50 nsec) of any analog input signal. Thus, detection of a 'leading edge' allows attribution of a 'virtual peak' to a certain LHC clock-cycle, which is the 'bunch crossing' for a saturated signal. The working range starts within the 'linear signal' range (ca. 200 GeV) reaching to 'infinitely' high energies.

C. The analog 'external BCID' on the AnIn daughter-board (see above).

The free-running (i.e. clock-independant) comparator indicates a threshold-crossing by its rising digital output. Given the shaper-defined peaking-time, BCID is performed by a properly delayed 'marking' of the correct time-slice in the PPrASIC. The working range starts well above the pre-defined comparator-threshold (to avoid 'glitches') and extends to 'infinity'. Hence, a large overlap with the other two mechanisms gives redundancy, which allows consistency-checking among the BCID-methods.

The signal formation is implemented on a daughterboard (AnIn). Each plug-on PCB holds 16 input channels, thus matching the modularity of the input connectors. This piece of hardware interfaces to remotely located electronics on the detectors. Exchangability of the part on the motherboard is useful allowing fast repair, if some external source has inflicted malfunction.

2.1.2 The actual pre-processing on a 'Multi-Chip Module'.

The PPrMCM carries all of the important pre-processing elements in the real-time data path. It is also realised as a plug-on unit. The argument for exchangability of this particular unit has been made elsewhere. Only its functional content and its interface to the motherboard is important in this context.

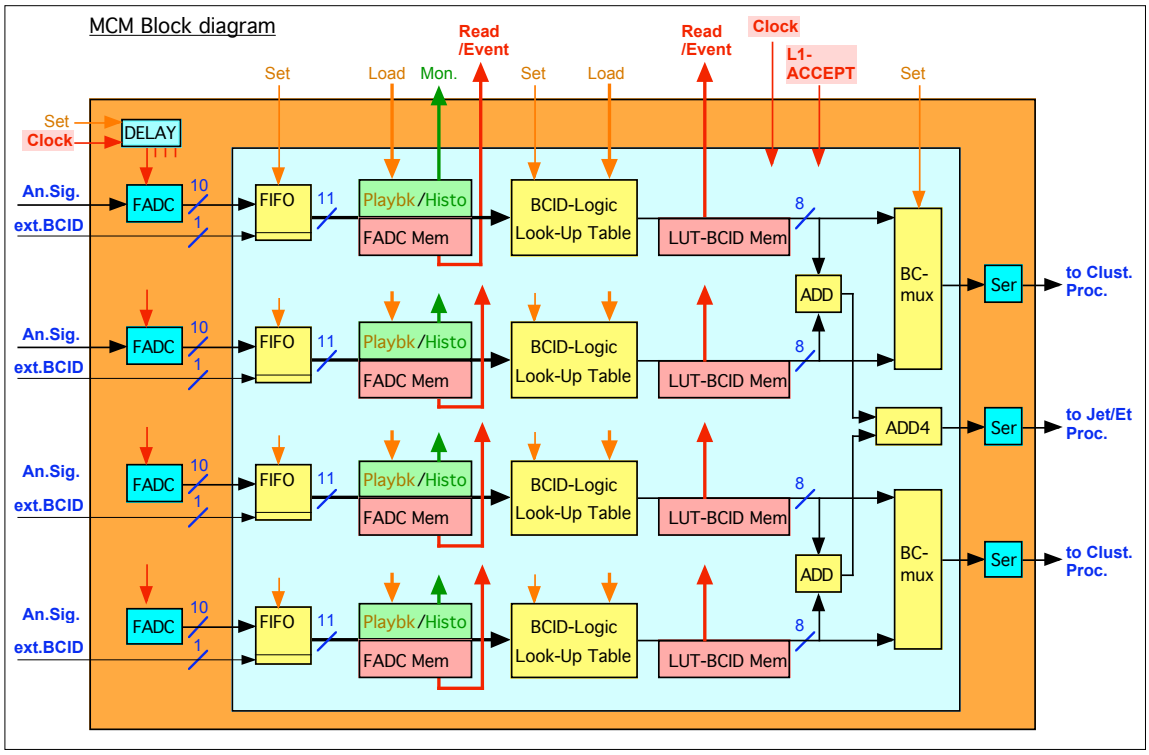

Figure 4 : The principal functions located on the Pre-Processor MCM.

Each PPrMCM handles 4 signal-channels. Hence, the PPM allocates mounting space and signal routing for 16 pieces of PPrMCMs. Because of the concentrated functionality, the consumed electrical power on each MCM is not negligible. A heat-exchanger mounted onto each PPrMCM ensures efficient removal of the heat produced.

Arrangement of signals from trigger towers in the calorimetry is not subject here. However, the main axis of all detectors runs along the pseudo-rapidity η. Architectur of the trigger demands the orthogonal direction, namely azimuth ϕ, to be the main axis. Reordering the tower-signals in the 'Receiver System' results in a 'circular' sequence of towers as shown in the figure below. The Pre-Processor produces 8-bit real-time data on the 0.1*0.1 granularity for the CP and 10-bit data on the 0.2*0.2 granularity for the JEP.

The 'Bunch Crossing Identification' forces an empty time-slice preceding and following the identified time-slice, because the seen energy is attributed to the identified slice in the processing pipe-line. Furthermore, the massive cabling, that would be needed for the 0.1-granularity, led to the scheme of multiplexing two towers onto one digital data-line linking the Pre-Processor to the Cluster-Processor. As the principal direction in the trigger is the azimuth, the multiplexing follows this axis as shown below.

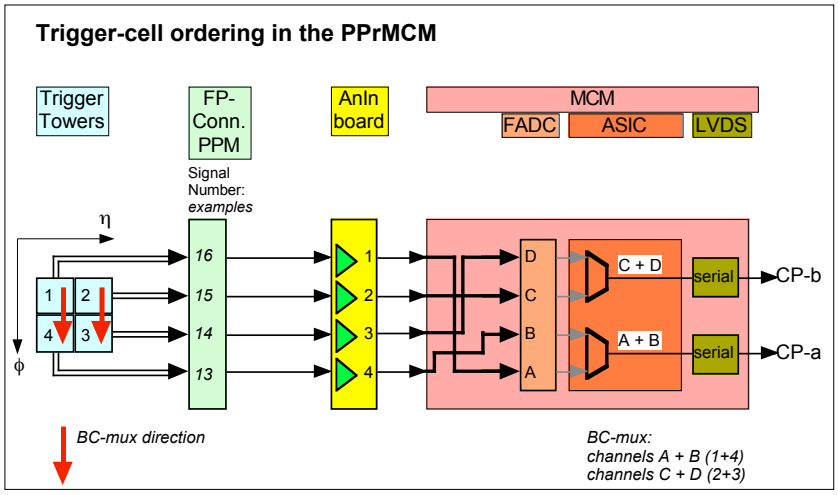

Figure 5 : Trigger -Towers and multiplexing of calorimeter data.

The 'BC-multiplexing scheme'.

The input signals follow 'linear order' at the MCM-input (channels 1,2,3,4). The trigger architecure requires

multiplexed pairs to be pair [A,B] and pair [C,D]. As example, only the pair [A,B] is considered, which is sent serially to the CP labelled 'CP-a'. For one pair, the cases of data sequencing coded into one 'Mux-bit' is visualized below.

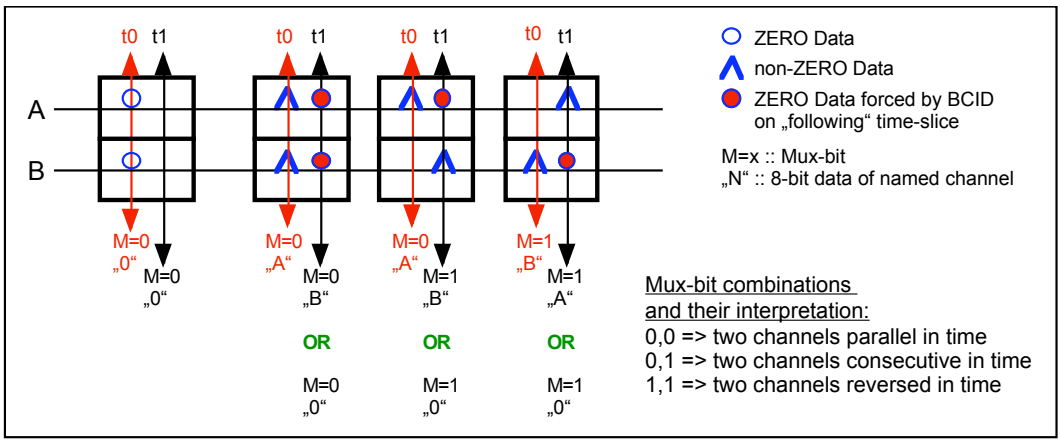

Figure 6 : The BC-multiplexing scheme.

Subsequent time-slices are labelled 't0' and 't1' respectively. 'M' denotes the Mux-flag-bit. The procedure is the same for the second channel-pair (C,D) on the MCM. Note, that summing 4 channels into a JEP-value is not at all affected by 'BC multiplexing'. The four 'BCIDed' values are summed as they emerge from the BCID-Decision Logic of the PPrASIC.

2.1.3 The serial data in real-time to the Cluster- and Jet/Energy-Processor.

Each of the PPrMCMs delivers three serial output streams at a rate of 480 MBaud/sec. Two of the streams contain data, which represent energy deposits on the 0.1*0.1 grid. They are input to the 'Cluster Processor'. The third stream contains energy deposits on the 0.2*0.2 grid, which are sent to the 'Jet / Energy Processor'.

In total, there are 16*3= 48 signal lines of high frequency routed to the PPM's board periphery. Special 'routing techniques' are applied for such signals. An impedance-matched strip-line together with 'pre-emphasis' of the digital signal at the source is used to fulfill the task.

An additional requirement comes from the architecture of the Level-1 calorimeter trigger system. The allocation of calorimeter signals to electronic modules demands, that certain signals (at the 'azimuthal' boundaries of each module) are duplicated and sent on separate cable links to the processors [see Ref. 1]. Furthermore, the PPM must have the capability to drive a cable link of significant length (appr. 11 m) to the processor-crates.

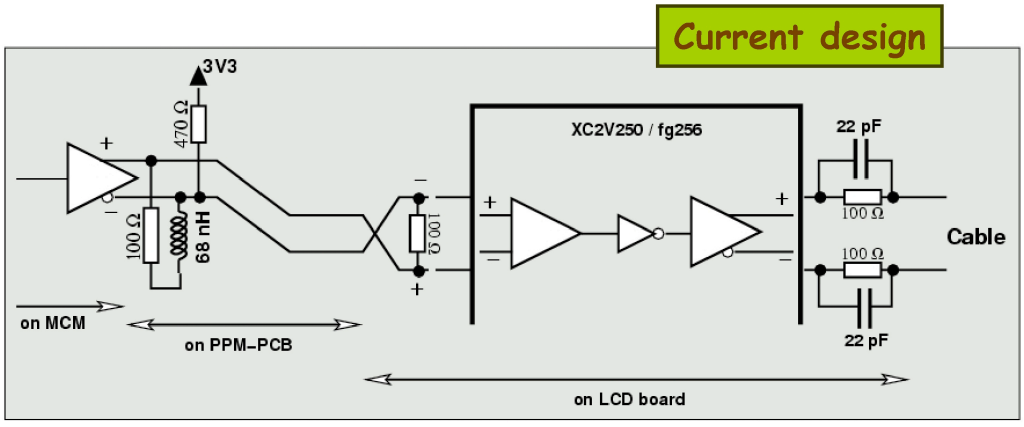

Figure 7 : Pre-Emphasis of LVDS across the PPM and over 11m of 'twin-pair' cables.

The task is attributed to a special driver-component called 'LVDS Cable Driver (LCD)', from which the signals are taken to a backplane connector. The connector is known by the brand-name 'CompactPCI' providing a high connection density combined with good grounding facilities. Its mechanical properties are adequate for a crate-to-crate cable installation.

2.2 *The VME interconnection and on-board controls.*

The Pre-Processor Module is a standard VME 'slave module'. It is interfaced to the VME Bus on the crate backplane providing A32 / D32 access for a computing infrastructure.

VME serves mainly for the purpose of setting-up the modules for ATLAS operation (e.g. setting hardware-registers, downloading FPGA code, etc ...). Another VME application is debugging and/or testing of a module in a laboratory test environment. A laboratory crate does not necessarily have the full readout facilities (RGTM, ROD etc.), yet a module must be accessible for test - including data readout. The physical implementation of VME Bus follows the 'Standard'. The Pre-Processor crate is a 'off-the-shelf' VME-crate of 9U height equipped with a VME64xP backplane.

Design studies, made a long time ago, resulted in a board configuration for the PPM as shown in the figure below. This demonstrates, that the architectural considerations have not changed significantly since the early days of the ATLAS-project.

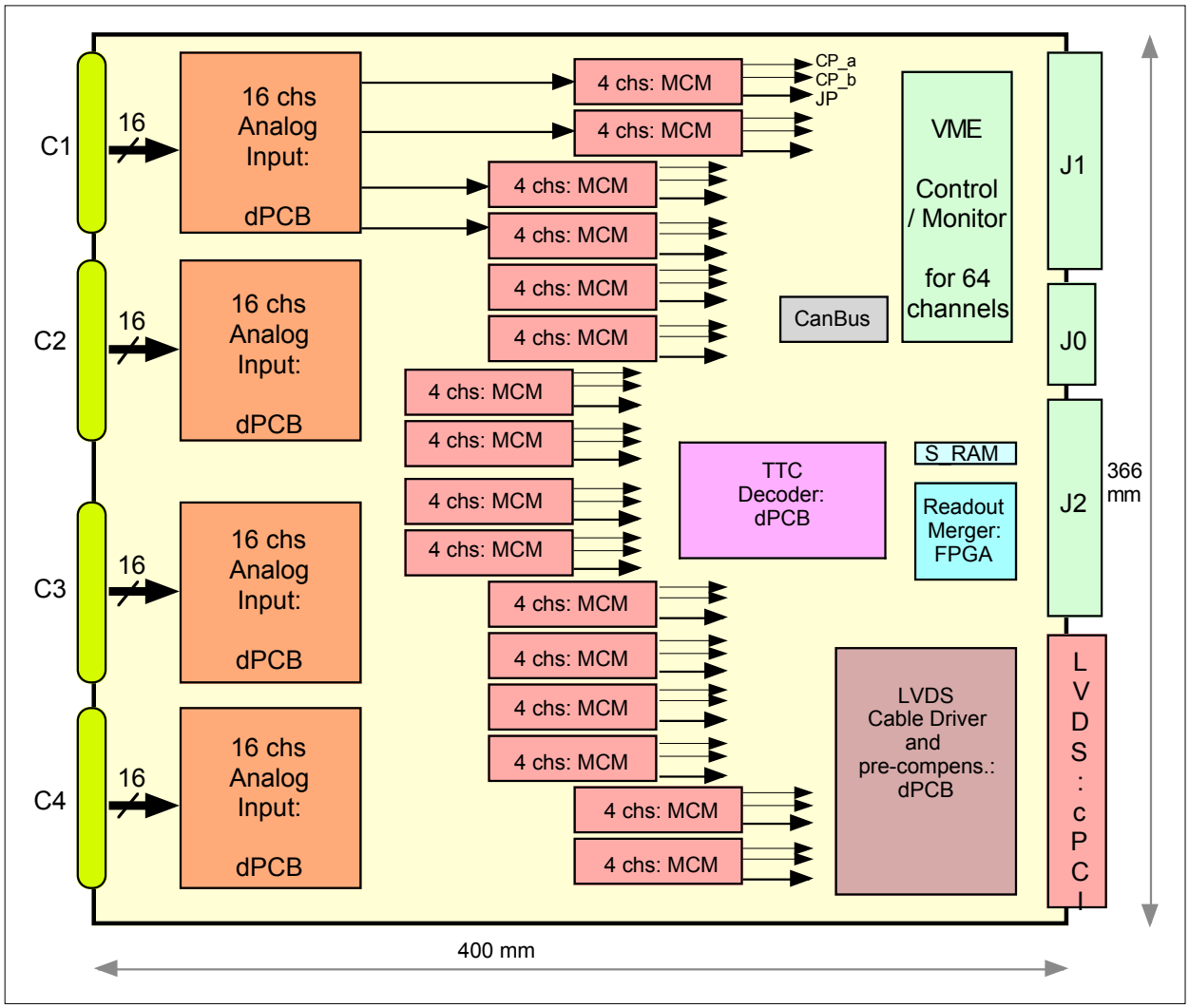

Figure 8 : Estimate of 'Real Estate' on a PPM-board.

2.3 *The data readout to DAQ.*

The 'data gate-way' into / out of the PPM is realised in the large, configurable device called ReM_FPGA. The VME Bus as well as the uni-directional G-Link (to ROD and DAQ) do interface to this central device on the PPModule.

Early work on the 'PipeLineBus' principle resulted in an ASIC-design to accomplish the task of 'merging data from the PPM sources for readout'. Due to the limited number of devices needed (1 per PPM) and due to the flexibility provided, a decision was made to use a 'Field Programmable Gate Array' instead of an ASIC.

The tasks of the ReM_FPGA comprise the following issues:

• Interface to I^2C busses for special devices (Phos4, TTCrx) on the PPM.

• Connect-up and serve the 2*16 ‘serial interfaces’ of the PPrASICs on the 16 MCMs.
• Interface to VME for control and low bandwidth readout.
• Interface the PPM data readout to the serialising RGTM-O (and to ROD and DAQ).

The device is a XiLinx Virtex XCV1000-e. The pin-layout at the package has been defined such, that FPGA-resources can be used optimally by firmware-code.

2.4 The Level-1 protocol via TTC: Clock, L1-Accept ...

The PPM, as a ‘pipelined device’, is driven by the LHC clock. The readout facilities require signals from the real-time protocol like the ‘L1-Accept’. Synchronisation of internal counters, such as the ‘Bunch-Crossing’ counter or the ‘Level-1 Event’ counter, is given by periodic RESET signals. Furthermore, a synchronous START signal is required to push data from ‘playback’ memories into the real-time data path for a stand-alone check of the trigger system.

These signals are provided by the TTC system on optical links. Each of the eight Pre-Processor crates will receive such a link. A ‘Timing Control Module’ (TCM) receives the optical signal-stream and provides electrical output for point-to-point distribution to 18 module-stations in a crate (16 are required for the Pre-Processor). The distribution is implemented as an auxiliary board plugged onto the crate's backplane along the J0-connectors. The PPM receives the TTC protocol through a TTCrx ASIC, which is located on yet another daughter-board called TTC Decoder (TTCdec). The output comprises the required signals ready for on-board use in the components concerned.

2.5 Slow Control.

The infrastructure of a crate (supply-voltages, supply-currents, fan-speed) is controlled via the standardised ‘slow-control’ system CAN-Bus implemented in the purchased VME crate.

Reliable system operation requires in addition monitoring of certain quantities on the module-level. Examples are the temperatures of PPrMCMs, the operating voltages and supply currents on each PPM board, the on-board produced supply-voltages for special components and more. This is achieved by a CAN-Bus interface, which transmits the module-information to a crate-level controller (implemented on the TCM). The PPM stations are connected to the TCM via a CAN-Bus realised on the same plug-on backplane as the TTC-distribution.

2.6 The Crate environment.

The architecture of the Pre-Processor system is matched to the calorimeter input as far as possible **[Ref. 3]**. For reasons of trigger latency, proximity of racks/crates is required for housing the following sub-systems:

1. Calorimeter Receiver Stations,
2. Pre-Processor,
3. the subsequent trigger processors (CP, JEP),
4. the Central Trigger Processor (CTP).

The proximity avoids unnecessary cable length. The ATLAS detector has two regions (the junctions between barrel and endcap on either side), where the bulk of trigger signals come from. Hence, the cables feed from the sides into a row of electronics racks lined up parallel to the detector. Signals pass first through the ‘Calorimeter Receiver Stations’, where - among other things - ‘transverse energy’ weighting is applied for the purpose of triggering.

Special attention is paid to the layout of cable routes, because the amount of cables is significant: appr. 4000 analog signals arrive at the trigger electronics for each hemisphere of pseudo-rapidity. The cables are rather massive as they carry 16 signals each on twisted pairs. For this reason, a scheme of ‘alternating sides for Input / Output’ has been adopted. Cables from the detector enter the ‘Receiver Stations’ at the rear (backplane-side). Output goes from the front-panel to the front of the respective Pre-Processor Modules, from where the ‘LVDS links’ leave from the rear (backplane-side) to reach the trigger processors. Coverage of the entire ATLAS calorimetry requires 2*4= 8 Pre-Processor crates. The layout of one Pre-Processor crate is detailed in the figure below.

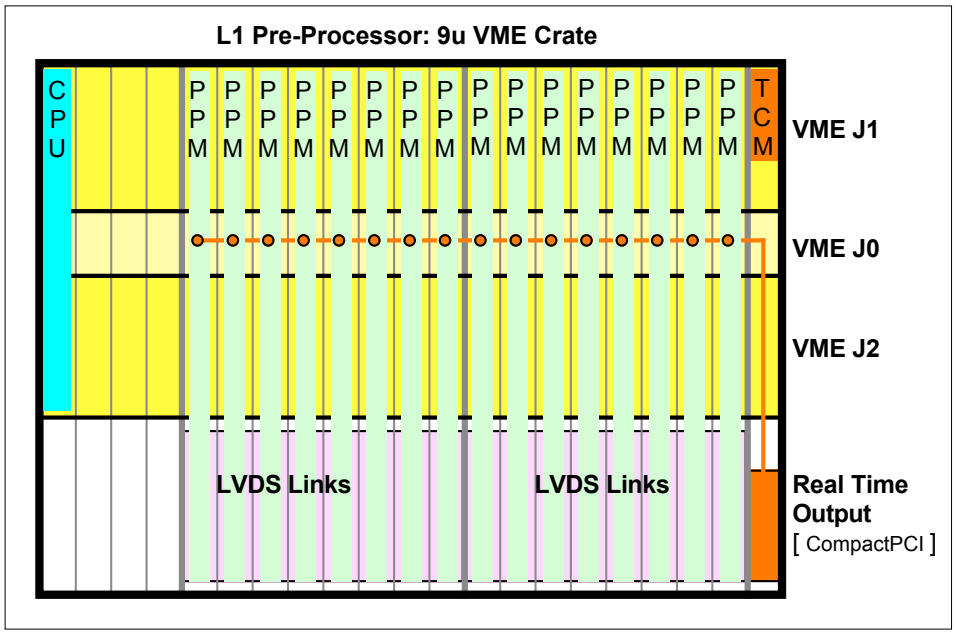

Figure 9 : A schematic allocation of PPMs in a Pre-Processor Crate.

As an example, a group of 8 PPMs is allocated to the electro-magnetic part and a group of 8 PPMs to the hadronic part of calorimetry. However, a particular allocation of modules is not mandatory, because access to control-parameters for each individual channel is given on the PPM (e.g. loading of BCID coefficients, LUT content etc...). Hence, other schemes of module allocation can be and have been adopted as long as the module-defined correspondence between input connector and LVDS-output is observed **[Ref. 4]**.

Important Note: All PPMs in the Pre-Processor system are identical in hardware.

A crate holds additional modules important for the system's operation.

A TCM **[Ref. 5]** receives TTC signals on an optical link and distributes them electrically on point-to-point links to the PPMs in the crate. In particular, the LHC clock, L1 'Accept' and other essential protocol-signals travel on this path across the experimental set-up. Another function on the TCM is the mastering of the afore mentioned CAN-Bus to/from the modules in the crate. A third role assumed by the TCM is a VME bus-display at its front-panel for visual diagnostics.

A network-connected CPU in the crate acts as local controller of the VME Bus. The tasks comprise the loading of parameters into the PPMs, local monitoring using non-triggered calorimeter signals recorded in ASIC-hardware and monitoring of event-data in 'spy-mode'. The number of stations available in a crate is nearly used up with this arrangement. With a CPU installed in the left-most slot (slot#1), only a three places (slot#2 to slot#4) are left free.

3. The Realisation of the Pre-Processor Module.

The system's architecture follows a sequence of modularities based on powers of 'two'. It starts with the detector input broken down into groups of four channels. This 'granularity'along with implementation-studies, made on small building blocks, showed the feasibility of the present, uniform board-layout. A detailed description of the module's periphery is given in the appendices. The connectivity is of particular importance for interfacing to systems 'upstream' and 'downstream'.

3.1 The physical implementation of the Pre-Processor Module.

Architectural considerations outlined earlier (see Ref. 1) led to a design of the Pre-Processor module as shown in the figure below **[Ref. 6]**.

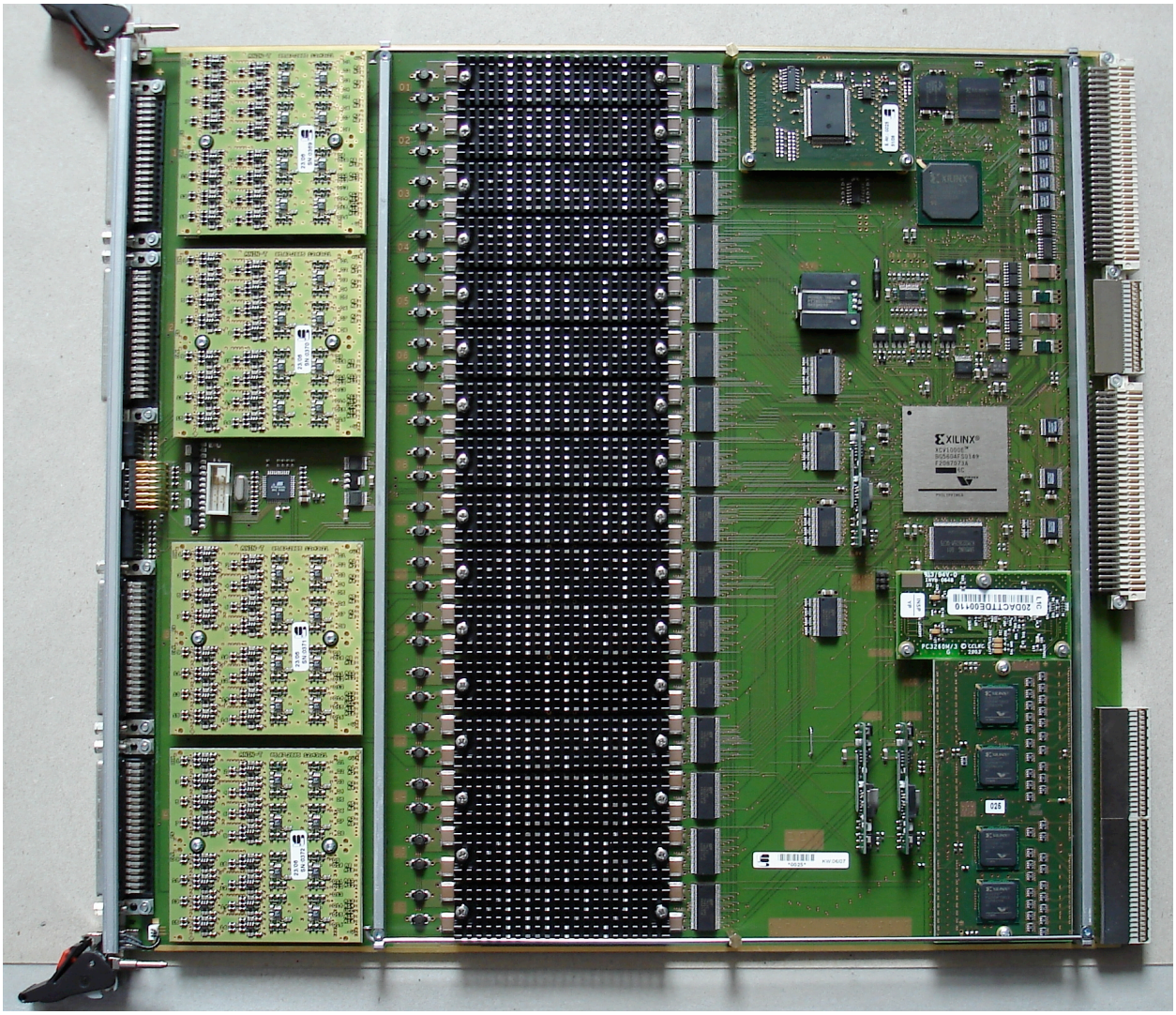

Figure 10 : The Pre-Processor Module (prod.-version 2.1).

The direction of flow for the real-time signals is from the left (front-panel connectors for analog input) to the bottom-right (backplane) side, where serial data streams (480 MBaud/sec) leave the module.

Easily visible are the many sub-components, which populate the motherboard:

- Four 'Analog Input boards (ANIN)' at the left.
- Sixteen 'Multi-Chip Modules (MCMs)' showing as 'column of heat-sinks' in the center.
- One 'Lvds Cable Driver (LCD)' at bottom-right.
- One 'TTC Decoder (TTCdec)' directly above the LCD.
- One CAN-module at top-right.
- One 'Readout Merger FPGA (ReM_FPGA)' at center-right (Xilinx).

Not only global heat-removal is important, but also a strong temperature-gradient between the 'bottom' and the 'top' of the PCB is not wanted. A high-flux air-stream from a cooling heat-exchanger is forced by fans from the bottom upwards through the crate. Temperature dependance of e.g. timing or reliability of components is not allowed to vary across the 64 channels on the motherboard.

3.2 Dimensions, Mechanics, visual Indicators.

The Pre-Processor module is configured as a PCB with a height of 9 NIM units (366 mm) and a depth of 400 mm. In fact, the depth has been reduced to 399.5 mm, because the strain on handles was too big when the board was inserted in the crate. Otherwise, the design follows the 'VME standard' **[Ref. 7]** set out for such modules. The number of layers is 8. Five layers are pure 'routing layers'. Three layers are used for full (or partial) power planes: +5.0V; +3.3V; GND (Analog-GND).

The PCB thickness is 2.0 mm required for sufficient rigidity along the board's lateral dimensions. The backplane

side is equipped with several connectors each having a fairly high pin-count. Any misalignment at insertion-time has catastrophic effects on the backplane in the crate. A 'macroscopic' addition is the implementation of mechanical stiffening for board-insertion/ extraction: the vertical stiffening rails are connected by two horizontal rods (top and bottom), which are supported midway. These rods add significant longitudinal strength to the board, when insertion forces are applied. Furthermore, the mechanical appliances (front-panel, extraction /insertion levers, 'grounding' etc.) comply with the IEEE standard 1101.10.

Reliability for long term operation also requires robust mechanics on the PCB itself. Vibrations from cooling fans and/or other influences to connection-stability are not allowed to compromise the performance. Hence, all daughterboards are fixed to the motherboard by screw-fixations (see picture of PPM).

Front-panel space is very limited due to the large number of input signals. Nevertheless, the free space in the center of the panel is allocated to visual indicators (LEDs).

The indicators inform on the status of:

- Firmware-loading (green = ok, red = loading failed):
  - for the ReM_FPGA (XCV1000e)
  - for the LVDS repeaters sending data to the CP
  - for the LVDS repeaters sending data to the JEP
- the crate-supply voltages (green = ok)
- the on-board generated voltages (green = ok)
- VME data traffic activity (yellow, 100 ms stretched DTACK)
- the arrival of 'L1-Accept' (yellow, 100 ms stretched)
- the TTC system clock (green, 100 ms stretched), i.e. PPM runs in DAQ mode.
- DAQ readout activity, i.e. ReM_FPGA output via G-Link is active (yellow).
- the local XTAL clock (yellow, 100 ms stretched), i.e. PPM runs in local check-out mode.
- OR of all ERRORs on the PPM (red = PPM not operational).

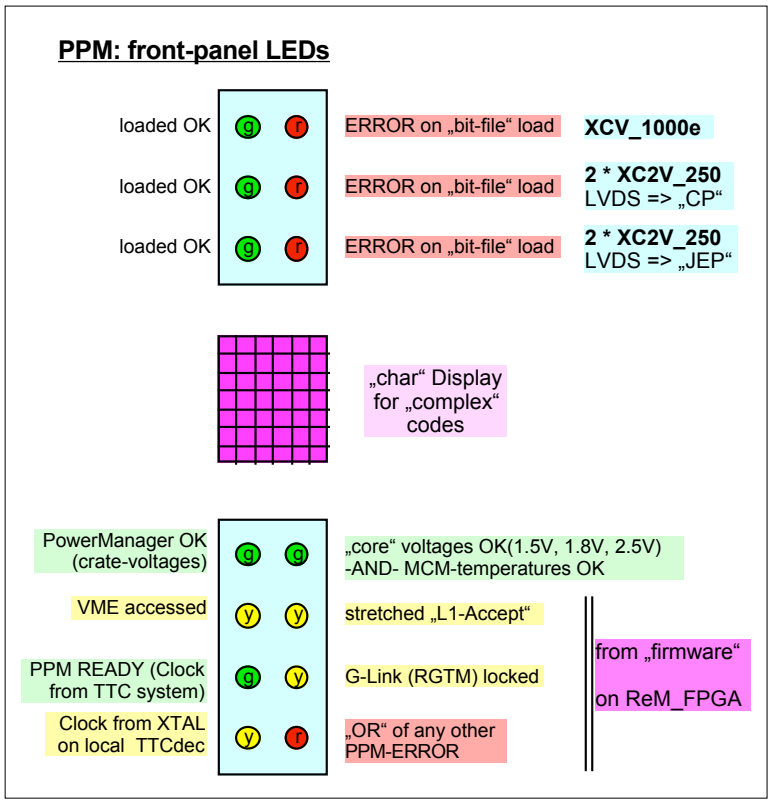

Figure 11 : The status indicators on the PPM front-panel.

The front-panel holds also a matrix-display element, which shows more complex information on e.g. error-states. These states are coded to 'characters', whose meanings point to detailed information. Naturally, a 'conversion table' is required for interpretion. The character-codes will be defined, when experience grows in operating large parts of the PP-system.

It was assumed, that rack-hardware allows connection of analog input cables from above and below at each crate, thus leaving a visible space in the center of the front-panel, where the LEDs are located. The final installation in ATLAS-USA15, unfortunately, brings cables from one direction only (either from top or from bottom) to a crate obscuring free view of the visual indicators.

3.3 Grouping of functional blocks, dissipated Power.

The following section describes the stages of signal processing implemented in hardware across the Pre-Processor Module starting from analog inputs to the real-time digital outputs. Data access for control as well as readout for checking purposes is given through VME. Readout data for DAQ are passed on to the RGTM by the ReM_FPGA upon receipt of a 'L1-Accept'. TTC input for real-time operation and 'slow control' of environmental parameters complete the module's functional content. An estimate of dissipated power together with practical considerations for efficient removal of produced heat ensures the viability of the PPM design.

3.3.1 AnIn

The Pre-Processor module receives 64 analog input signals from the calorimetry. The signals are grouped into four cables (16 channels each) with specially configured connectors for minimal cross-talk **[Ref. 8]**. The layout of the connector along with the allocation of calorimeter signals is given in **Appendix_A**. Each connector has a pair of fixation screws, which ensure proper electrical contact along with safety against accidental disconnection.

The incoming analog signals are routed directly to the 'AnIn' daughter-board **[Ref. 9]**. There are four such boards on the PPM matching up to the signal channels on the four input-connectors.

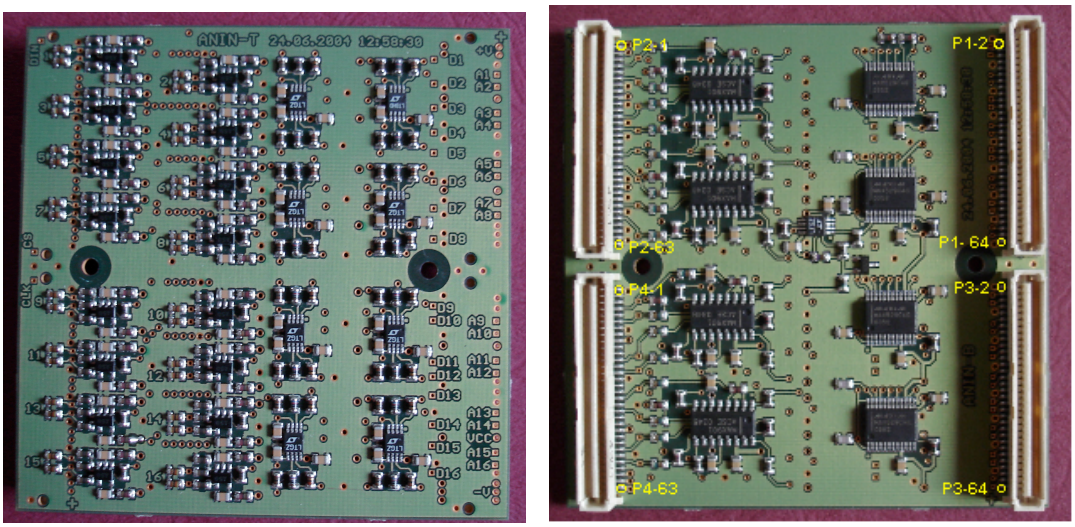

Figure 12 : The 'AnIn' daughterboard (left:top; right: bottom with connectors)

Connectors, known from the CMC standard (Common Mezzanine Card), are used to interface the daughterboard to the PPM. Layout-studies of the 'AnIn' PCB have shown, that the functionality can be implemented on daughterboards of size restricted by the vertical dimension of the motherboard.

Two kinds of outputs are transported directly across the motherboard to the PPrMCMs as input for each channel (trigger cell):

1. A uni-polar analog signal conditioned to the input of the digitising FADC.
2. A digital signal, whose rising edge represents the point in time when the analog signal crosses a pre-set voltage threshold. The signal is used further downstream for so-called 'external BCID'.

The list of connector-pins interfacing the AnIn-module to the motherboard is given in **Appendix_B**.

3.3.2 PPrMCM

The PPrMCM and its functionality are described in separate documentation **[Ref. 10]**. The pinning of the two connectors is given in **Appendix_C**. The connectors M1 (on the left in the figure below) and M2 (on the right) have 60 pins each with 0.5 mm pitch. This is a very sensitive electro-mechanical device requiring special care. Intrusion of dirt can impede the connection-quality in a severe way leading to a malfunction, which can be difficult to locate.

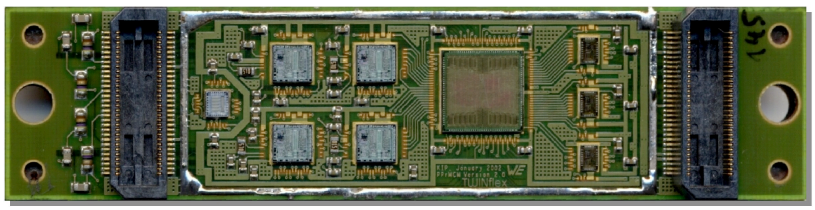

Figure 13 : The 'bonded' components on the PPrMCM (brass-lid removed).

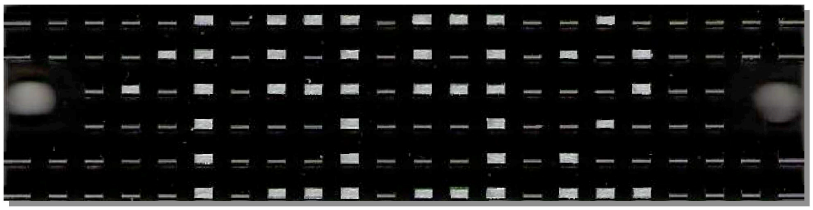

Figure 14 : The 'heat-exchanger' on the PPrMCM top-side showing a 'Serial Number'.

3.3.3 LCD

Two separate output streams (to CP and to JEP) are created on every PPrMCM. Each stream carries serial data of high speed (480 Mbit/ sec). Hence, careful routing of these signals via impedance-matched strip-lines has been implemented from the PPrMCM to the LCD daughterboard. The system architecture requires duplication (fanout) of certain data-streams on the Pre-Processor level [see Ref. 1]. Hence, capability is provided to drive the signals over a 'parallel pair' cable to its destination processor, which can be as far as 15 meters away.

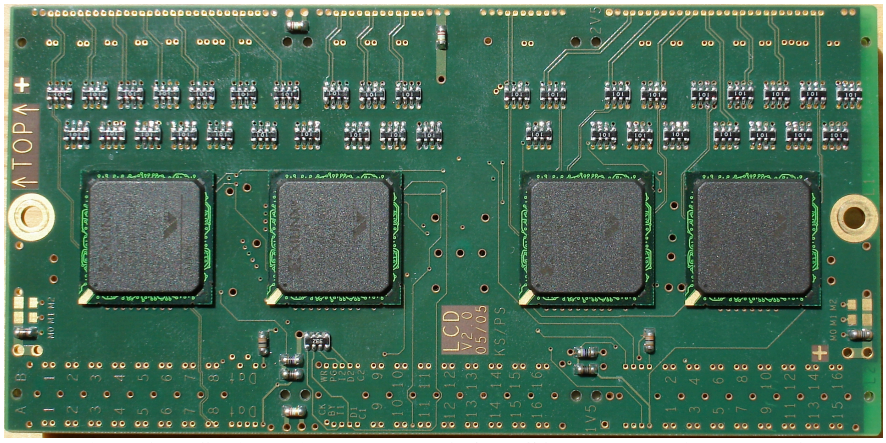

Figure 15 : The LCD board's top-side with 4 FPGAs and part of 'pre-emphasis' circuitry.

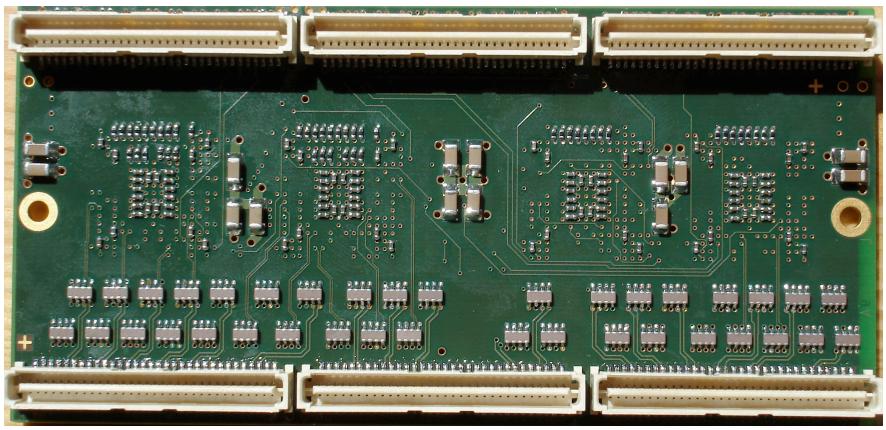

Figure 16 : The LCD board's bottom-side showing the other part of 'pre-emphasis' circuitry.

The duplication of signals is done inside the FPGAs (X2CV_250), where one input from the PPrMCM is routed to two output drivers. Studies of LVDS transmission **[Ref.11]** have shown, that it is necessary to 'pre-compensate' losses due to the cable's integration properties. This is achieved by passive components (R, C) placed close to the FPGA outputs. The exact dimensioning of the pre-compensation network (values) depends somewhat on the cable length. It is optimised for cables with a length of 10 to 15 meters.

3.3.4 TTCdec

Signals, distributed through the TTC protocol tree, are received on the PPM from the auxiliary backplane across the J0-connectors in a daughterboard called 'TTCdec' **[Ref. 12]**. The protocol-stream is decoded and individual signals are distributed to their destinations on the motherboard (PPrMCMs, ReM_FPGA, ...). Apart from reception and board-level distribution of 'real-time' protocol signals (e.g. 'LHC Clock', 'Level-1 Accept', 'Bunch-Counter Reset', 'Level-1 Event-Counter Reset' etc.), the TTC system is the only means to control synchronous system

activities.

Such system-wide applications in the Pre-Processor are:

1. Sending a 'global' RESET to the entire system.
2. 'Starting' and 'Stopping' data play-back from on-board memories for testing purposes.
3. 'Start' and 'Stop' of sending LVDS synchronisation patterns initiated by the 'LVDS_Sync1' signal on the PPrMCM.

Access to parameter settings on the TTCdec is given through a separate I^2C bus originating on the ReM_FPGA, which maps the registers to VME (see chapter 4 for details).

Reminder on I^2C adressing:

Observation: The TTCrx chip can lose its I^2C-address, which is generated from an on-board hard-wired I2C_ID. Only an immediate (within few msec) access after a 'reset' or a 'read/write' operation is always successful, i.e. acknowledged by the TTCrx.

Explanation: A standalone PPM uses the on-board crystal oscillator and not the encoded TTC signal with the embedded clock. The TTCrx has a 'watch-dog' circuit, that monitors whether its internal PLL has locked. But, without external TTC signals to the TTCrx, the PLL will not lock. As a consequence, the 'watch-dog' circuit detects the improper 'lock-state' and initiates an internal 'reset'. The procedure is permanently repeated as long as there is no external TTC signal to the TTCrx present. An 'odd state' of the TTCrx chip is the result, which gives rise to I^2C access problems. Tests with a TTC clock present never show any kind of I^2C access problems. i.e. I^2C cycles are always acknowledged. This is a 'bug' in the TTCrx design, which has to be kept in mind when operating the PPM in a test-rig without TTC input.

Conclusion: If externall TTC is missing, I^2C-access is only possible by 'cheating the watch-dog', e.g. by 'double' write-access within the watch-dog interval .

3.3.5 The 'special' I^2C bus to Phos4-chips on the MCMs.

The timer-chip called Phos4 was one of the first 'utility' devices developed at CERN for LHC-experiments in general. Time-adjustment with 1 nsec resolution over a LHC clock-cycle of 25 nsec is indeed a necessity in many parts of experimental apparatus. The Pre-Processor is one of these systems. The exact setting of the digitiser-strobe in time to the peak of an incoming analog signal is essential to obtain optimal energy-resolution. Hence, the Phos4-chip was implemented from the start of design as the 'fine-timing' component.

After the production of Phos4-wafers, tests were carried out to verify the performance of the device. Already here a problem showed up. The feedback of the 'delay-lock loop' can hang-up putting the step-size of Phos4-delays either to minimum or maximum values. The PPrASIC, under development at the time, took over the task to sense Phos4-activity by counting feedback pulses within a given time-interval. If no feedbacks occur, the PPrASIC signals an 'error-state' to the supervising software requiring corrective action.

In addition, the device was missing the usual 'reset' function, which is required for such a component to be used in a complex system. The only way to 'reset' the Phos4 was to cycle the supply-power to put it back into a defined mode of operation. However, the chips were already produced in the required quantity. Therefore, the system had to cope with this feature. There are several possibilities to cycle power on a Pre-Processor Module:

- 'switch off/on' the crate manually (not an option for ATLAS operation).
- 'switch off/on' via the hot-swap controller on the front-panel lever (almost as bad...).
- 'switch off/on' via the CanBus (that is, what CAN is for ...).
- 'switch off/on' via VME and the hot-swap controller (another possibility ...).

When the MCMs were built, testing of the production series revealed another problem, which only showed up rarely. It was not observed in tests of small MCM quantities. The I^2C serial bus transporting set-up data showed sensitivity to noise-glitches when the rise-time of the clock was 'slow'. The input to the Phos4 re-triggered on such glitches causing loss of the set-up data. It was cured by introduction of a 'pull-up' combined with termination resistors.

A pre-series of Pre-Processor Modules (PPM_1.0) was built with confidence, that all trouble had been eliminated. However, on the module-level yet another feature appeared. Again, this could only be observed on a small fraction of channels. Intensive debug work revealed, that the Phos4 requires a 'highly symmetric' reference clock (very nearly 50/50 duty-cycle) to ensure continuous operation of the DLL feedback. The clock provided by an XTAL oscillator (see top half of figure below) is not symmetric enough (45/55) causing Phos4 hang-up on some channels. Hence, the ReM_FPGA has to be used as 'facility' to provide a substitute XTAL-clock of required symmetry (see bottom half of figure below). The scheme implemented (PPM_2.0 and PPM_2,.1) has been tested extensively. The switching of clock-sources (phase-jump), when a firmware reload takes place, does not cause interuption of Phos4 operation.

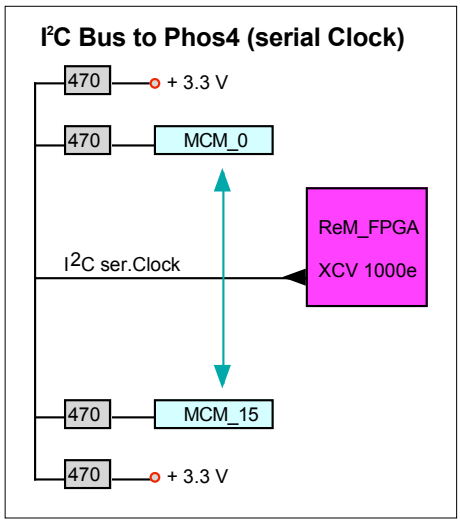

Figure 17 : Clock-distribution for the 'special' I²C bus to Phos4-chips on the MCMs.

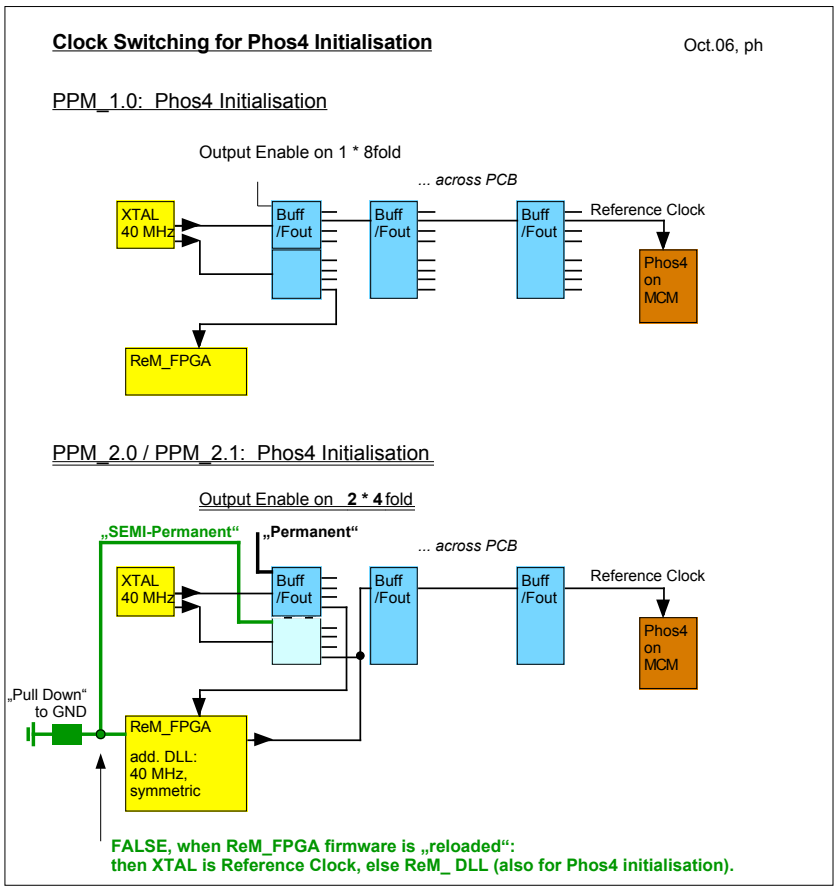

Figure 18 : Providing clocks on the PPM.

Note on latching FADC-data (and external BCID-bit).

The range of the Phos4-delays cover the full LHC-clock cycle. Depending on the Phos4-delay set (i.e. position of the FADC-strobe), the encoded data change. If data are latched for synchronous operation in the PPrASIC while changing, 'random data' appear in the processing downstream. To avoid this ambiguity, latching in the PPrASIC must be set according to the following rule (see also PPrASIC User Guide [10]).

• For Phos4-delays [0 nsec to 9 nsec, 22 nsec to 24 nsec], the data must be latched using the NEGATIVE clock-edge.

• For Phos4-delays [10 nsec to 21 nsec], data must be latched using the POSITIVE clock-edge. Note that, this causes the data to appear a whole latency-tick later in the processing pipe-line. Hence, this one tick-delay must be compensated by removing a delay-step in the PPrASIC delay-FIFO.

3.3.6 VME and Static RAM

The VME interface supports the slave-protocol as outlined in the corresponding specifications [see Ref. 6]. The VME implementation complies with A32 / D32 data-transfer. During development of the PPM, it was decided that wider data-access (D64) was not required.

The module recognises the following address-modifiers issued by a master-CPU in the crate:

- 0x09 Extended nonprivileged data access
- 0x0A Extended nonprivileged program access
- 0x0B Extended nonprivileged block transfer
- 0x0D Extended supervisory data access
- 0x0E Extended supervisory program access
- 0x0F Extended supervisory block transfer

The key-component for the readout interconnection of the PPM is the ReM_FPGA located near the backplane. It provides Input / Output functionality, which is all implemented in the FPGA code. A VME accessible Flash_RAM holds the FPGA code for a fast 'loading' locally **[Ref. 13]**.

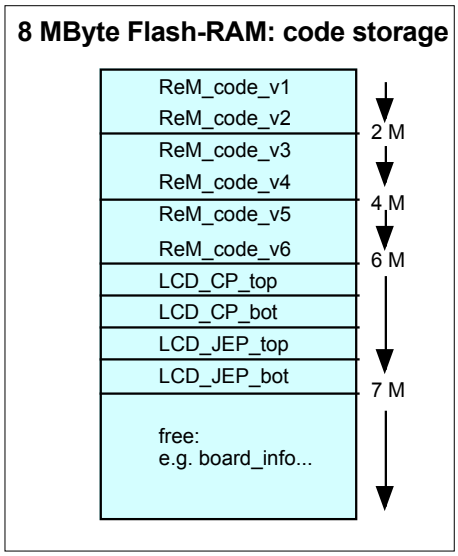

Figure 19 : Storage of firmware binaries in Flash_RAM.

The generous Flash-RAM space is attributed to firmware versions as shown in the figure above. Since the LCD FPGAs contain only fixed 'routing / fanout' firmware for signal-distribution, only one version is foreseen to be held here. The ReM_FPGA is quite a different case: up to 6 versions can be stored. Examples could be:

1. Speed-optimised 'production version' for DAQ
2. 'debug version' for front-end (SPI, I^2C, writing to Ser. Interfaces)
3. 'debug version' for back-end (DAQ readout from Ser. Interfaces)
4. 'debug version' for checking 'read-back' of settings
5. ...
6. ...

The functionality of the ReM_FPGA comprises the following 'hardware' tasks:

- A 'Serial Programming Interface' (SPI) is provided to set DACs on the four AnIn daughterboards. They define analog values for thresholds (external BCID comparator) and analog offsets (baseline shift of analog input-signal).
- The timing elements (Phos4) on the PPrMCMs are set by means of a I^2C bus. The bus master is implemented as FPGA code.
- Another I^2C bus master assures communication with the TTCdec daughterboard.
- 'Write' and 'read' access to each PPrASIC is possible through 'serial interfaces'. As there are many (64) channels on a PPM, 'serial interface'-ports are implemented in a number to serve 'doublets' of channels, namely 32.

'Historical' Note: It is explicitly stated, that 32 serial interfaces are provided to serve the 'doublets' of PPrASIC-channels. Only, if severe technical problems would have arisen (e.g. excessive routing density on the main board), the 'fall-back' of daisy-chaining the two channel-doublets on each ASIC would have been considered. As a consequence, all four PPrASIC-channels on a PPrMCM would be served by one serial interface only. This would have reduced the number of serial ports to 16 on the PPM's ReM_FPGA, but would have demanded a sacrifice on

readout band-width.

- The ReM_FPGA is the communicating partner to VME bus. To facilitate data-transport to VME, a synchronous static RAM (S-RAM) is implemented, where blocks of data can be stored intermediately. The size of the RAM is 1M locations * 36 bit width, of which 32-bit words are 'seen' by VME. The data arrive as 'pieces' from the different sources on the module and can be accumulated in this 'intermediate" store . Examples are
  - ASIC-register contents read back for verification,
  - 'rate-meter' results from individual channels,
  - 'unbiased' histograms of FADC digitisations from individual channels. ...

The accumulated and arranged results can be transfered as 'block data' via VME to the analysing CPU in the crate.

'Historical' Note: A decision on the type of FPGA had been taken some long time ago for the 'XiLinx Virtex XCV1000e'. The chosen chip-package took into account aspects of PCB manufacturing. It is a 'Ball Grid Array' (BGA) with 1.27 mm pitch. The number of input-output connections (560 pin-balls) is sufficiently big to cover the requirements of the PPM. Verilog code has once been developed for a 'demonstrator ReM_ASIC'. After the decision for a FPGA, a complete re-writing of the code as 'firmware' was necessary for today's implementation on the PPM.

3.3.7 DAQ readout

Data readout in so-called 'physics runs' of the experiment (ATLAS-DAQ) uses a different path with higher band-width. The event-related data ('readout') are separated from monitoring data ('read-back') inside the ReM_FPGA. The readout data are extracted for all 64 channels, formatted with header and trailer and are propagated through the VME-J2 connector to a 'Rear G-Link Transmission Module' (RGTM). The current version of this module with optical G-Link output (RGTM-O) is specified elsewhere **[Ref. 14]**. The RGTM-O sends the data over a fiber to a ROD **[Ref. 15]** for collection and/or compression.

3.3.8 Dissipated Power, Cooling.

The consumption of electrical power is far from uniform across the PPM. In fact, it has a strong maximum at the places taken up by PPrMCMs. Placement of these components did at first try to distribute the produced heat over the largest possible cross-section with the cooling air-stream. A 'chevron-like' placement of the PPrMCMs was attempted to achieve this (see Figure 5). However, the realisation in a PCB layout showed constraints, which did not allow this. Hence, a fully 'vertical' arrangement was implemented (see Figure 7).

The estimated power consumption on the board is distributed as follows:

- The measured power for the AnIn daughter-board is around 3000 mW each. The four 'plug-on' units contribute **12.0 Watt** to the power-bill of the PPM.
- The power consumption on each PPrMCM is composed as follows according to the specifications of the various manufacturers:

4* AD9042 (FADC) 4 * 600 mW = 2400 mW (on +5V)
1* PHOS4 (Timer) = 200 mW (on +3.3V)
1* PPrASIC = 2300 mW (on +3.3V)
3* DS92LV1021 (ser. LVDS transmitter) = 500 mW (on +3.3V)

Hence, the summed values (rounded upwards) for a PPrMCM are 2.5 W on +5V and 3.0 W on +3.3V giving a total of 5.5 Watt. It has been shown on 'demonstrators' that the exchange of produced heat with the fan-forced air-flow is guaranteed by the heat-sink mounted on the PPrMCM. The sixteen PPrMCMs add up to a maximum of **88.0 Watt** on the module, which is concentrated at the PCB center (see Figure 7).

- The remaining logic (VME interface, TTCdec, LCDs and Rem_FPGA ...) is estimated to consume no more than **40.0 Watt** together.

Hence, the dissipated power of the PPM will not exceed the limit of **140 Watt**. The value is managable in a standard VME64x crate. The following table compiles measured consumptions (for some items) and estimates (for the other items) and relates them to the supply-voltages of the crate.

Supply *[Volt]*	**SubDevice**	**Current** *[Ampere]*	**Power** *[Watt]*	**Comment**
+ 5.0	AnIn (* 4)	1.5		measured
	MCM (* 16)	7.7		FADC specs

	Ancill. Logic (VME...)	6.0		estimated
	Current-Sum	15.2 A	76 W	
[- 5.0]	AnIn (* 4)	0.92 A	4.5 W	on-board from -12V, measured
+ 3.3	MCM (*16)	14.5		ASIC simulation
[+ 2.5]	FPGA-, LCD- I/O	1.0		on-board from +3.3V, estimated from specs
[+ 1.8]	FPGA-, LCD Core	1.0		on-board from +3.3 V, estimated from specs
	Current-Sum	16.5 A	55.5 W	
				[Total: 136 Watt]

Table 1 : Estimated Power Consumption of the PPM

Breaking down the requirements of the PPM to supplied power from the backplane results in:

- 50 W to be supplied on +3.3V (ca. 16 Amp)
- 80 W to be supplied on +5.0V (ca. 16 Amp).

This is within the limits set by VME64xP standard. Each pin on the backplane can safely transfer 2 Amp. There are 10 pins allocated to +3.3 Volt supply and 12 pins are allocated to the +5.0 Volt supply (6*VCC + additional 6*VCC on J0: see VME64x-**VIPA**).

A fully equipped Pre-Processor VME crate holds 16 PPMs and some auxiliary modules (e.g. CPU, TCM). The total power amounts to max. 260 A on +5.0 Volts and max. 260 A on +3.3 Volts. The crates and the corresponding power-supply units purchased have the ability to provide this.

A **verification of power-consumption** and a verification of the cooling capability of a fully equipped crate (16 PPMs, 1 TCM, 1 CPU) has been performed in the KIP-Laboratory. The consumption observed on the crate's current-meter is:

175 Ampere on +3.3 Volt, 150 Ampere on +5.0 Volt.

These values translate into appr. **84 Watt comsumed by each PPM** – well below the 'worst case' estimate. The crate has been operated without problems using fan-cooling with ambient air in the laboratory (no water heat-exchanger). Temperature measurements on MCMs show values well within operational limits (max. 60° C).

3.3.9 Control facilities (CAN).

Local Control of operating conditions.

Each PPM is 'fused' on-board for each supply-voltage in order to prevent damage to the backplane in case of a fault on the module itself. This is defined policy within the 'Level-1 Calorimeter Trigger' project. The PPM follows these guide-lines. Passive fuses produce a high voltage-drop at the currents consumed. Therefore an 'active' solution is chosen. Power ON/OFF is handled by a 'hot-swap' controller 'TPS2345', which is connected to a switch in the handle of the front-panel, but also connected to CAN. Furthermore, the controller monitors the 'ramping' of each voltage. Unforeseen conditions lead to 'switching OFF'.

NOTE.

There are two reasons for the following remarks on 'hot-swapping':

A. 'bent pins', specially in the LVDS-area, are not catastrophic, but certainly very bad when a full system-installation of 124 PPMs is expected to operate with highest efficiency.

B. swapping modules in a crate under power is meant to increase system-availability, but the opposite is achieved when neighbouring modules are damaged in the process.

Removal and Re-insertion of a module (brief guide-lines).

1. When a faulty module is identified, switch it OFF by pressing the red button in the lower handle of the front-panel. Also the left/right neighbours must be switched OFF.

Reasoning: The moving module in the centre could touch neighbours under power with bad consequences.

2. Insert a new module in the slot and slide it back until gentle resistance can be felt.

NEVER apply thumb-force FIRST to the BOTTOM half of the PPM front-panel !

Always push at the top until ,catching' is felt (area of upper analog connectors - they are solid and well bolted to the PCB). The alignment due to the two, fairly rugged' VME-DIN-connectors (J1, J2) with their casing lifts the

board to the correct height, because the PCB has up to 1 mm ‚play' in the guide-rails.

Reasoning: Normally, the VME connectors catch first. The ‚flimsy' CompactPCI has shorter pins (KIP changed this explicitly). By pushing at the bottom, one could still manage for the CompactPCI pins to touch first. With 2 mm pin-raster and 1mm PCB play, a desaster can occur. The insert/extract action is a safe procedure, if executed correctly.

3. When done correctly, the PPM 'sits' in the pin-holes. Now, the two levers can push it fully in. Apply force SIMULTANEOUSLY to top-bottom levers with courage. It is necessary to acquire the 'feeling'.

The 'ATMEGA16' micro-controller on the PPM senses several critical quantities on the module and converts them to digital values. These are:

- the voltage across a temperature-measuring diode on the ReM_FPGA (XCV1000e),
- the voltages across sixteen temperature-measuring diodes on the 16 MCMs.

In addition, the device drives a 7*5 matrix display to show more complex codes as 'characters' on the front-panel.

List of coded error-states as of Apr.2009 (see 3.3.5):

- 'V' alternating '1' => +5.0 V on board deviates ≥ 10%
- 'V' alternating '2' => +3.3 V on board deviates ≥ 10%
- 'V' alternating '3' => +2.5 V core-supply deviates ≥ 10%
- 'V' alternating '4' => +1.8 V core-supply deviates ≥ 10%
- 'V' alternating '5' => +1.5 V core-supply deviates ≥ 10%
- 'V' alternating '6' => -5.0 V analog-supply deviates ≥ 10%
- 'T' alternating 'n' ('n'=1, 2, 3 … 9, A, … G= dec.16) => Temp. of MCM_'n' exceeds 80°C.

=> 'alternating' display freezes to 'checkered matrix', when the alarm-condition disappears.

- 'checkered matrix' => Module is powered-up only

=> status displays defined by on-line software:

- 'L' => Module is busy with firmware loading
- 'C' => Module is being configured (parameters are loaded)
- 'I' => Module is initialised, ready for operation
- ... list to be extended

Remote Control (CAN).

Conditions for the operation of electronics are continously monitored in the ATLAS experiment. Equipment like rack-cooling, crate power-supplies, fan-units etc. are connected to the Detector Control System (DCS) on the level of electronics racks. The back-bone of this 'slow control' system is the serial CAN-Bus **[Ref. 16]**, which allows measurements on the time-scale of minutes or longer. Rack- and crate-level supervision is 'default' in DCS. The Level-1 Trigger system has extended this supervision to the board-level.

The PPr sub-system has a TCM in VME-slot#21 as CAN-master in every crate. The TCM interfaces the external CAN-Bus (DCS) with the internal CAN in the crate, where the individual modules are identified by their slot-number (geographical address #1 to #20). The TCM uses the crate-number (PP-crate#0 to PP-crate#7) as coded on an auxiliary backplane (see **Appendix F**) for CAN-node identification to the external CAN. The CAN-node ID has to be ≥1, because ID=0 is used to address all nodes simultaneously. Hence, the PP-system nodes are

PP-crate	#0	#1	#2	#3	#4	#5	#6	#7
CAN-node	ID = 1	ID = 2	ID = 3	ID = 4	ID = 5	ID = 6	ID = 7	ID = 8

The PPM-CAN interface in form of a daughterboard (see **Ref. 16**) consists basically of a 'chip-size' controller (Fujitsu MB90F594) running CAN-Bus software ('CanOpen' compatible). It receives some quantities directly in digital form (e.g. MCM temperatures converted to ºC). Others are digitised on the CAN-board itself (e.g. supply-voltages). All data are transmitted to the CAN-master over bus-lines on the auxiliary back-plane, which is plugged onto the VME64xP-J0 connectors,

The table below summarises the PPM–CAN interface data by listing the quantities to be monitored or controlled.

Quantity to 'read'	**nom. Val.**	**Resol.**	**dig. Val.**	**dig. Resol.**	**Range**	**Remark**
supply +5.0 V	5.0 V	10 mV	500	1	8 bit	digitised on CAN-board

supply +3.3 V	3.3 V	10 mV	330	1	8 bit	digitised on CAN-board
-5.0 V from supply–12 V		10 mV	500	1	8 bit	shifted into pos. Voltages; digitised on CAN-board
on-board +2.5 V	2.5 V	10 mV	250	1	8 bit	digitised on CAN-board
on-board +1.8 V	1.8 V	10 mV	180	1	8 bit	digitised on CAN-board
on-board +1.5 V	1.5 V	10 mV	150	1	8 bit	digitised on CAN-board
XCV1000e temp.	appr. 40 °C	1 °C	40	1	8 bit	Op.-limits: ok < 60°C; warn < 64°C; switch-off > 68°C;
MCM temp.	appr. 50 °C	1 °C	50	1	8 bit	**Calibrated to** INTeger by ATMEGA; Limits as on XCV1000e; 16 Values from 16 MCM-Diodes.
Quantity to 'write'	**nom. Val.**	**Resol.**				**Remark**
'RESET' to PPM	0	-	-	-	binary	equiv. to 'Power-On' Reset of PPM
'Pow. ON/OFF' to PPM	0	-	-	-	binary	Power ON/OFF to PPM, if enabled by missing jumper on Service-Module
VMEtoCAN RESET	0	-	-	-	binary	reset the CAN MicroContr. from VME
VMEtoCAN Program_Select	0	-	-	-	binary	VME enables CAN Module to load CAN firmware to the CAN MicroController

Table 2 : CAN-Interface on PPM.

3.4 Latency across the PPM.

The propagation time of a given analog input signal (counted from the signal's peak) has been measured from the input-connector to the LVDS output of the PPrMCM (part 1 and 2 below). The third part (LCD) has been determined separately. The total latency breaks down as follows given in units of LHC clock-ticks:

1. From Input connector across AnIn-board to input of PPrMCM: 2
2. Processing on PPrMCM: 13 (to JEP)
3. LVDS through LCD to the PPM backplane connector : 1

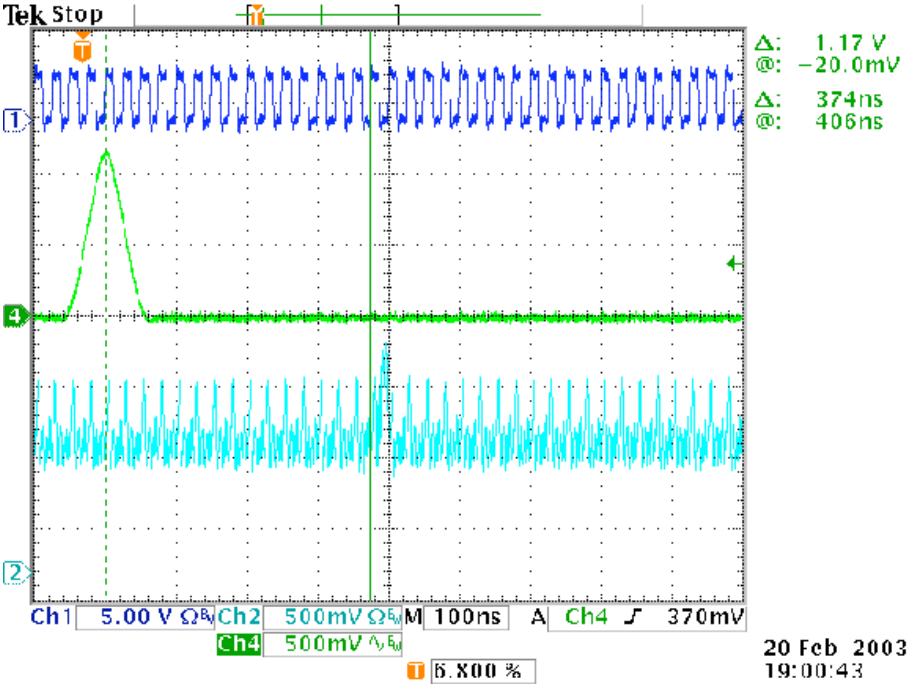

Figure 20 : Measured latency (375 nsec) across the AnIn and PPrMCM for JEP-output.

Hence, the total latency contribution of the Pre-Processor Module to the Level-1 pipe-lined system is **16 LHC clock cycles** as compared to 17 estimated in the TDR [see Ref. 1]. The data to the CP are not subject to the 4-cell summing in the ASIC hardware. Hence, the CP data emanate ONE clock-tick earlier, verified by measurement.

The latency measurement has been repeated in 2008 using final production modules. This means, that no more 'estimates' are needed. The measurement covers signal propagation from the front-panel to the backplane connector of the PPM including the LCD daughterboard in the real-time data-path. The results given above have been verified and confirmed.

3.5 The Backplane connectivity.

The implementation of VME Bus follows the VME64x standard. Hence, the upper 6 NIM-Units in the crate are occupied by a VME64xP-VIPA backplane. The allocation of signals specific to the Pre-Processor on 'user-defined' pins of the standardised VME connectors is described later in this section.

Only the remaining lower 3 NIM-Units are available for full-custom installations. The Pre-Processor crate passes the real-time signals (LVDS) through this space for distribution to the subsequent trigger processors. A custom-made backplane fragment holds the 2 mm CompactPCI connectors. There is no signal routing on the backplane fragment, only feed-through of pins extending to the rear of the crate, where a shroud guides the cables plugged in. The pinning of real-time signals is outlined in **Appendix_D**. Grounding the LVDS-Links is an issue common to the sending (Pre-Processor) and the receiving end (trigger processors). The handling of grounds on the sending side, i.e. on the PPM, is also shown in Appendix_D. Signal grounds are grouped according to their destination in the subsequent logic. Optional passive components can be inserted to ensure 'cleanliness' of the ground potentials.

The connection to the 'Rear G-Link Transmission Module' for event-readout is implemented on 'user-defined' pins of the VME-connector J2. The allocation of the data- and control-lines is outlined in **Appendix_E.**

Distribution of TTC protocol-signals from the TCM in slot#21 is realised by point-to-point lines routed on an auxiliary backplane attached to the VME-connector J0. CAN-Bus requires two serial lines across the slots on the backplane. These lines are also realised on this printed-circuit board.

All pins of VME-J0 reach through the auxiliary printed circuit board. Unoccupied pins are used by a small (ca. 4*4 cm^{*2}) plug-on board, which carries adapters for 'servicing' the PPM. 'Servicing' means dedicated data I/O for e.g. CPLD re-loading. The pinning of J0 is given in a table of **Appendix_F**.

4. Data transfer to/from the Pre-Processor Module

The Pre-Processor Module can be accessed for full 'read' and 'write ' cycles through the VME bus. VME has all data-sources and all data-destinations mapped to address-space. Hence, everything (also readout data) can be accessed with the speed given by the bus.

A second, equally important, but uni-directional data-path delivers 'readout data', i.e. event-related results from the pre-processing (FADC results, BCID results) to the ATLAS-DAQ. The data are serialised on the auxiliary module (RGTM-O) for transmission on a high-bandwidth link to a ROD and further on to the ATLAS Read-Out System. Both data-paths originate on the ReM_FPGA of the PPM.

4.1 The ReM_FPGA on the PPM

The resources of the ReM_FPGA are divided into three major parts: the VME-port (VME-ReM Interface), the actual Readout Merger (ReM) and the Data-I/O interfaces to the Pre-Processor hardware.

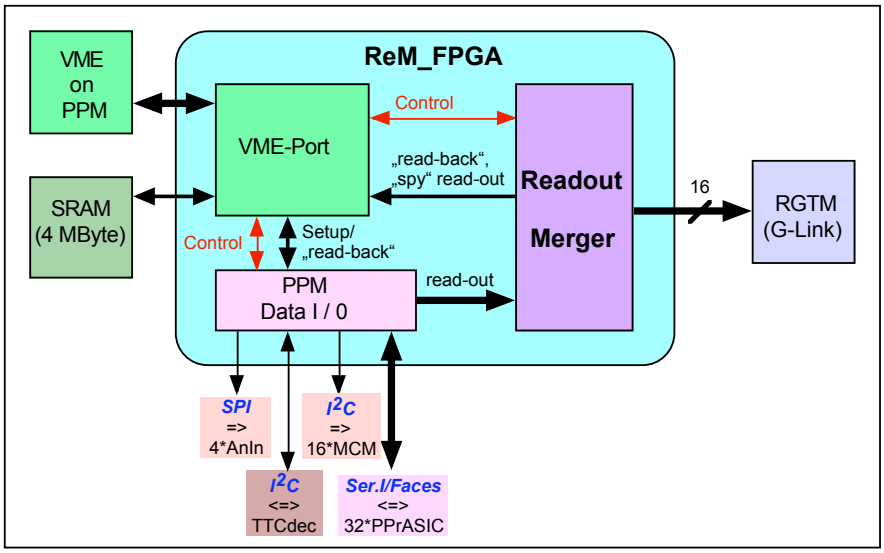

Figure 21 : The functional blocks on the ReM_FPGA (Virtex XCV1000-e).

The **VME-Port** contains the functionality to write to all data-destinations (configuration registers, 'playback' memories and many others). It is also the 'receiver' of data, which are read back from those locations for verification. In addition, there are other data to be collected through VME, like 'monitoring' memories holding unfiltered trigger input data, 'event-data' memories for spy-analysis etc.. The programming model gives a map of addresses with their data-content .

The **Readout Merger** has to assume several tasks, which can be summarised as 'data routing' and 'data formatting'. The main issues are:

- Control data (read back from configuration registers via Data I/O) are separated off and passed on to VME.
- Monitoring data (read back from accumulating memories), which are only of local interest to the Level-1 trigger (e.g. 'rate-meter' content, 'raw transverse energy' histograms) are made accessible to VME.
- Local (PPr system) usage of event-related data (e.g. detector / trigger cross-calibration) requires the full rate (at the expense of dead-time) to be available via VME. Other applications (e.g. 'spy' analysis) can do with a scaled-down rate of event-data copies. Any local read-out shall not slow-down DAQ.
- Data-acquisition mode for physics as 'top-priority' task requires, that event-related readout data from multiple sources (64 trigger tower channels from 16 PPrASICs collected on the 32 serial interfaces) are merged into a formatted data-record complete with 'header' and 'trailer' for transmission to a ROD.

The **Data I/O** firmware is an assembly of different bus-masters passing data to and fro under implementation of the correct protocol and data-format.

4.1.1 The VME-Port and the 'intermediate' S-RAM store.

The VME bus is directly connected to the ReM_FPGA. The FPGA, in turn, communicates with a S-RAM. which has 1M locations of 36-bit width (i.e. 4 MByte). The S-RAM control on the ReM_FPGA uses the memory as 32-bit data fields by truncating VME-addresses accordingly.

The **first** task of the VME-port interface is to exchange data with VME bus across the crate. The module is ready to receive VME 'read / write ' after 'power-up' and 'firmware-load' from the Flash-RAM into the ReM_FPGA.

A **second** task of the VME-ReM interface is the supervision of data-flow through the Readout Merger. Set-up data are passed directly over the corresponding Data I/O port to the destinations. After loading the firmware, the 'operation default' puts the ReM into DAQ operating mode, where 'Level-1 accepted' event data are collected. 'Read-back' data are filtered out for local use. Similarly, sub-sets of event-data (spy-mode) are copied.

A **third** task, handled by the VME-port interface, is the data-exchange with the peripheral S-RAM. 'Read-back' data from each ASIC-channel arrive as a 'single word' or 'block' carried along with the 'Level-1 accepted' event-data. It is sensible to pre-collect such data from all channels in the S-RAM. Similarly, a copy of event-data from all 64 channels can be assembled in the S-RAM. VME 'block read' is an efficient way to move these data to a local crate CPU for monitoring and/or local analysis.

Examples of data in the S-RAM are:

- register content, read from subdevices on the PPM (PPrASICs, I^2C etc.)
- event-data for local monitoring (spy-mode).
- statistical data accumulated in hardware (FADC histograms, 'rate-meter' results).

4.1.2 The Readout Merger.

The Readout Merger (ReM) manages the data-transfer from the distributed sources on the PPM. The ReM separates out 'read-back' data from the data collected on the module (see above), it assembles the event-data for DAQ into formatted records and sends these records via the RGTM to a ROD. Furthermore, the ReM has to follow the DAQ-protocol as transmitted by the TTC system. The important signals are:

- The LHC Bunch crossing Clock
- The Level-1 Accept
- The Bunch crossing Counter Reset before each LHC-turn
- The periodic Event Counter (i.e. 'Level-1 Accept' Counter) Reset.

4.1.3 The Data Input/ Output ports.

The ports provide the appropriate bus-protocol for shipment of set-up data and reception of read-back as well as event-related data. Three distinct kinds of ports are implemented:

- A 'Serial Programming Interface', which downloads data to DACs without reading-back capability from the hardware devices.
- Two separate I^2C busses to Phos4s and the TTCdec. The devices can be loaded, but only the latter provides local storage for reading back.
- Thirty-two custom-designed serial interfaces to pairs of PPrASIC channels.

4.2 VME access to the PPM.

The VME bus-protocol is maintained by permanent code residing in a CPLD. Apart from serving 'normal' VME data traffic, the protocol CPLD also holds some registers for operation of the PPM (e.g. Control- and Status Register, FPGA-loading of firmware ....).

The PPM's space of VME-addresses is shown in the figure below. The four 'top bits' <A31:A28> are used for crate-identification (8 Pre-Processor crates) in the Level-1 trigger system. However, they are only used for addressing a crate/slot via the TTC-system and via the CAN-Bus. They are ignored by VME, i.e. are set to the fixed value of 0xC or binary '1100', because each crate contains a local CPU as VME bus-master.

Five bits <A27:A23> above the module's internal address-space identify the module in one of the 21 slots of the VME-crate (geographical address).

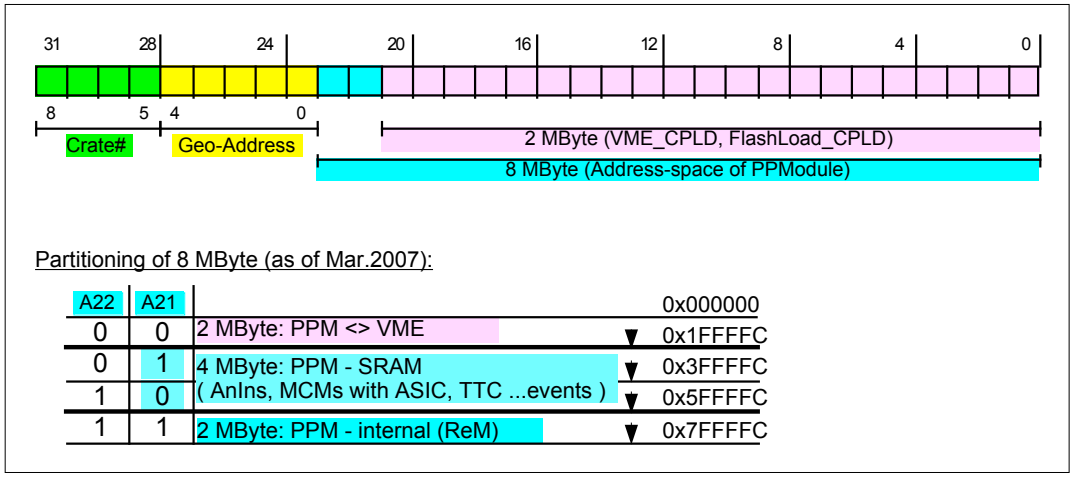

Figure 22 : VME address-space for a Pre-Processor Module.

The PPM requires **8 MegaByte** of address-space, i.e. **0x000000 to 0x7FFFFF**. The two bits at the top <A22: A21> are used for subdivision of the internal module-space. The first 2 MByte (0x000000 to 0x1FFFFF) are allocated to 'common' module-registers in the VME-CPLD. The remaning 6 MByte (0x200000 to 0x7FFFFF) are used for direct access to other on-board registers and/ or memories.

The VME-CPLD passes certain addresses to yet another CPLD, which in turn administrates the 'Flash-Memory'

store of firmware-code. It controls firmware-load from the Flash-RAM into the respective devices. Only then, the PPM is ready for operation. The Flash-Memory has its own capacity of 8 MByte. A subdivision here is made for storage of binary code of several firmware-versions ('DAQ', "debug' ...) for the XCV1000e ReM_FPGA (6 MByte) as well as firmware for the 'routing and LVDS-driving' X2CV-250 FPGAs on the LCD-daughterboard (2 MByte).

4.3 Data model of the PPM.

Testing-software in the laboratory and, more so, DAQ-software requires a full description of the PPM's data-space. The following gives the mapping for control- and readout purposes starting with the PPM as a whole. The module is then be 'broken down' to the sub-module level and, finally, to the channel-level, where the data-space structures are repetitive.

4.3.1 General issues.

- All registers are readable over the VME bus, i.e. there are no 'write-only' registers. Status registers are 'read-only' registers; Control registers are 'read-write' registers.
- The 'power-up' condition of all VME registers is 'all zeros', unless stated otherwise.
- A 'byte' is an 8-bit field; a 'word' is a 32-bit field.
- 'Setting' a bit field means setting all of its bits to '1';
- 'Clearing' a bit field means writing '0' into it.
- Bit-setting to '1' is equivalent logical 'true'; '0' is 'false'.
- Each PPM is uniquely identified (e.g. to a local CPU) by its position in a crate. The slot-position is given by coded pins (geographical address: <A27:A23>) implemented in the VME64xP standard backplane.

4.3.2 VME addressing.

The VME implementation has been developed for the 'pre-production' PPMs (20 boards of version PPM_2.0). Those are fully functional modules. The production version PPM_2.1 has hardware improvements implemented on the PCB-level, which were identified in the final reviews of the design. There are no differences to the pre-production version from the operational point of view.

The following sections describe the VME programming model for the PPM. VME addressing allows byte-wise access on the bus. However, almost all data-locations on the PPM are 4 byte wide (32-bit words). The VME implementation of register addresses is outlined below.

- The addressing starts with an 'Identification Register' at the **base-address = 0x000000**. The on-board store of firmware bit-files (Flash RAM) is administrated through registers at the addresses following. Configuration of the module is achieved and reflected in registers at addresses above **0x100000** ranging to 0x170000, where the ATMEGA micro-controller is accessible.
- The sixteen MCMs occupy each an address-range of 8192 bytes (0x2000) starting at address **0x200000**.
- Operation of the module requires settings in the TTCdecoder module. The TTC parameters are set through registers at addresses above **0x240000**. They are passed via VME through the Data I/O on the ReM_FPGA. This I^2C-path is bi-directional, i.e. not only the VME-registers are readable, but also the final data-location on the TTCdec.
- Driven by the TTC trigger protocol (e.g. LHC-Clock, L1-Accept), event-data like FADC samples, LUT-BCID results are extracted from scrolling memories into 'derandomizer buffers'. The data are fetched into the ReM_FPGA via the 32 serial interfaces to the PPrASICs. The corresponding Readout 'status registers' and 'data-FIFOs' are accessible at addresses **0x600000** to 0x6000FC.
- A range above address **0x620000** holds a map of 'flag-bits' used by the ReM_FPGA's firmware to administrate the 'read back' from PPrASICS. VME can only read those bits. The 'flag-bits' are the four spare bits on each SRAM word (bit#32-35).
- The operational status of Phos4 chips on MCMs (feedback to Delay-Lock-Loop alive) is held at **0x7FFEFC**.
- A set of eight registers (> **0x7FFF00**) is defined for quick-loading of parameters used for the conditioning of analog input signals. Note, that the same values can be accessed on a channel-by-channel basis as laid out in the MCM-related registers.
- A register (at address **0x7FFF80**) marks channels, from whose discriminator a 'Trigger L1-Accept' can be derived and fed into an 'OR' to trigger the module locally.
- Another register (**0x7FFF84**) allows to generate a set of 'pseudo-protocol' pulses internally for

technical purposes.

- VME can read the ReM_FPGA firmware version at **0x7FFFD0**.
- Registers of the kind 'Control/ Status' common to the PPM as module-entity are allocated to addresses **≥ 0x7FFFD4**.

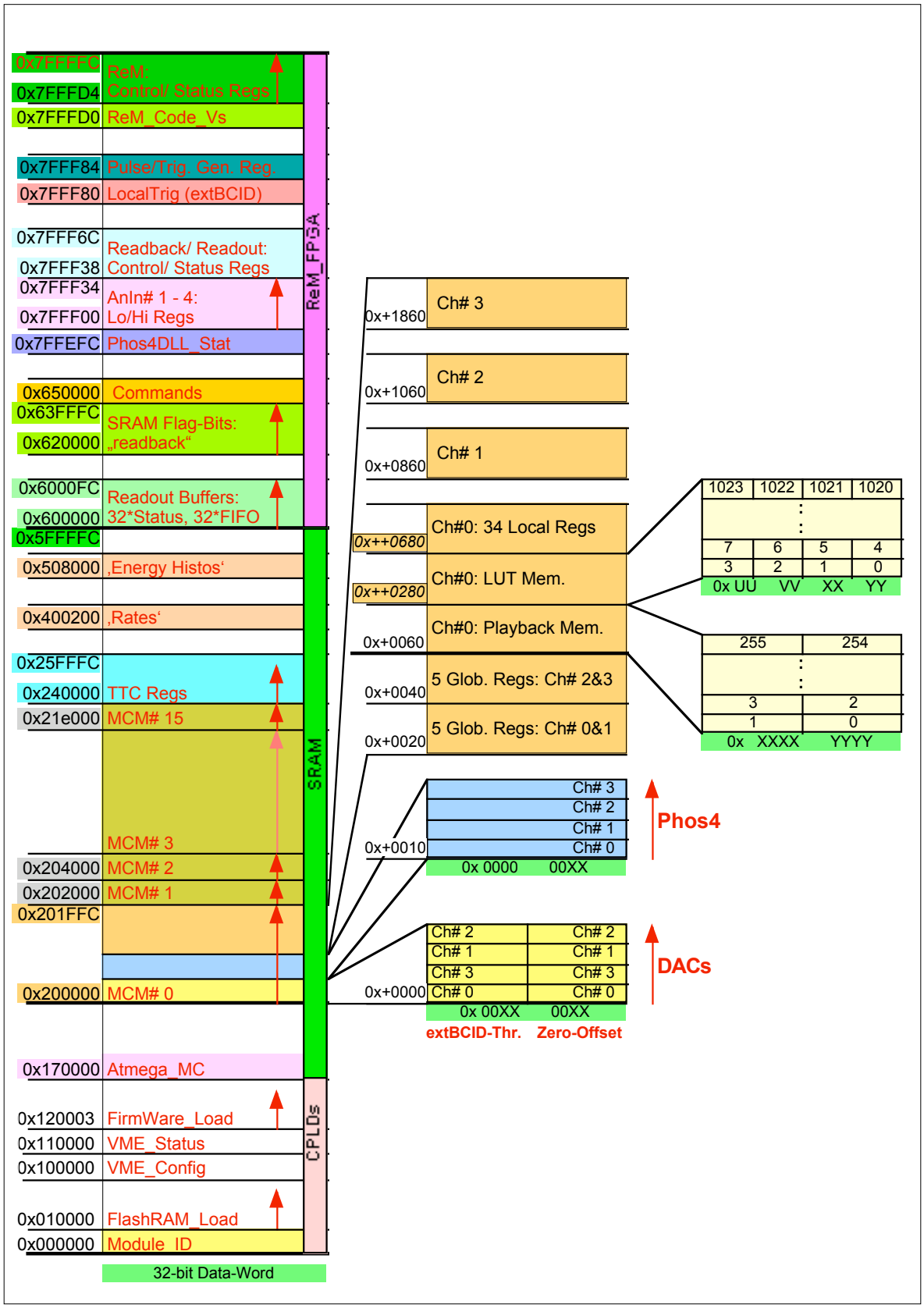

Figure 23 : The PPM as seen from VME.

The module is supervised by a VME bus-master, the CPU in the crate. Registers in the CPLDs controlling bus-traffic and firmware-loading provide the means to configure the PPM into the desired operational mode. These registers are listed in the table below with their 'absolute' allocation in address-space of VME.

The programming package HDMC (Hardware-Diagnostic-Monitoring-Control) uses addresses for 32-bit words to access module-level registers. Those addresses are also listed in the tables below. Daughterboards on the PPM are 'sub-modules' (sub-classes in HDMC). The corresponding hardware-items are PPrMCMs (as well as PPrASICs thereon) and the TTCdecoder. Their register-addresses are defined as 'byte-address' and used as incremental address upon the sub-modules base-address.

Register Name	Size / byte	Byte ADDRESS	Location	Register Type	Word32 ADDRESS (HDMC)	Register Definition	Description	Acc.
CPLD Registers		abs. Address						
Module_ID	4	0x000000	VME_CPLD	ModReg32	0x000000	PPM.Base	Birth-certificate of Module	R
VME_Config	4	0x100000	VME_CPLD	ModReg32	0x040000	PPM.VMEConf	VME Configuration	R/W
VME_Status	4	0x110000	VME_CPLD	ModReg32	0x044000	PPM.VMEStat	VME Status	R/W
FlashRAM_Addr	4	0x010000	FLSH_CPLD	ModReg32	0x004000	PPM.FRAddr	FlashRAM address	R/W
FlashRAM_Data	1	0x020003	FLSH_CPLD	ModReg32	0x008000	PPM.FRDat	FlashRAM Data	R/W
FlashRAM_Reset	1	0x030003	FLSH_CPLD	ModReg32	0x00C000	PPM.FRRes	FlashRAM Reset	R/W
FlashLoadXilinx	4	0x040000	FLSH_CPLD	ModReg32	0x010000	PPM.FRLoad	"Flash" Control for loading	R/W
Flash_Status	4	0x050000	FLSH_CPLD	ModReg32	0x014000	PPM.FRStat	Loading status from "Flash"	R/W
ReM_FPGA_Reprog	1	0x120003	FLSH_CPLD	byte only		PPM.XCVEReprog		R/W
ReM_FPGA_PrgDat	1	0x130003	FLSH_CPLD	byte only		PPM.XCVEProgDat		R/W
LCD_Reprog	1	0x140003	FLSH_CPLD	byte only		PPM.XC2VReprog		R/W
LCD_XC2V_PrgDat_1	1	0x150003	FLSH_CPLD	byte only		PPM.XC2VProgDat1		R/W
LCD_XC2V_PrgDat_2	1	0x160003	FLSH_CPLD	byte only		PPM.XC2VProgDat2		R/W
ATMEGA_MC	4	0x170000	VME_CPLD	ModReg32	0x05C000	PPM.ATMEGA		R/W
Common Registers		abs. Address						
SRAM_Update_Flags	32768	0x620000	S-RAM	ModReg32	0x188000	PPM.(hdmc?)	up to 0x63FFFC	R
SRAM_TTCrx_Flags	124	0x640100	S-RAM	ModReg32	0x190040	PPM.(hdmc?)	up to 0x64017C	R
RateMeter_Enable	4	0x650000	ReM_FPGA	ModReg32	0x194000	PPM. SIFs	Enable rate acquisition on 64chs (SerIF channel-pairs)	W
RateMeter_Disable	4	0x650004	ReM_FPGA	ModReg32	0x194001	PPM. SIFs	Disable rate acquisition on 64chs (SerIF channel-pairs)	W
Histo_Enable	4	0x650008	ReM_FPGA	ModReg32	0x194002	PPM.SIFs	Enable histogramming on 64chs (SerIF channel-pairs)	W
Histo_Disable	4	0x65000C	ReM_FPGA	ModReg32	0x194003	PPM. SIFs	Disable histogramming on 64chs (SerIF channel-pairs)	W
PHOS4DLL_Stat	4	0x7FFEFC	ReM_FPGA	ModReg32	0x1FFFBF	PPM.P4DLL first 16 bit : what is different to 7FFF5C ?	Status of Phos4_DLLs	R/W
AnIn1_lo	4	0x7FFF00	ReM_FPGA	ModReg32	0x1FFFC0	PPM.DEFAULT	direct block-load via SPI:	R/W
AnIn1_hi	4	0x7FFF04	ReM_FPGA	ModReg32	0x1FFFC1	PPM.DEFAULT	threshold and offset	R/W
AnIn2_lo	4	0x7FFF10	ReM_FPGA	ModReg32	0x1FFFC4	PPM.DEFAULT	"	R/W
AnIn2_hi	4	0x7FFF14	ReM_FPGA	ModReg32	0x1FFFC5	PPM.DEFAULT	"	R/W
AnIn3_lo	4	0x7FFF20	ReM_FPGA	ModReg32	ox1FFFC8	PPM.DEFAULT	"	R/W
AnIn3_hi	4	0x7FFF24	ReM_FPGA	ModReg32	0x1FFFC9	PPM.DEFAULT	"	R/W
AnIn4_lo	4	0x7FFF30	ReM_FPGA	ModReg32	0x1FFFCC	PPM.DEFAULT	"	R/W
AnIn4_hi	4	0x7FFF34	ReM_FPGA	ModReg32	0x1FFFCD	PPM.DEFAULT	"	R/W
RateMeter_Status_1	4	0x7FFF38	ReM_FPGA	ModReg32	0x1FFFCE	PPM.CHAN_1	Ratemeter status of digital Ch_1 to Ch_32	R
RateMeter_Status_2	4	0x7FFF3C	ReM_FPGA	ModReg32	0x1FFFCF	PPM.CHAN_2	Ratemeter status of digital Ch_33 to Ch_64	R
Histo_Status_1	4	0x7FFF40	ReM_FPGA	ModReg32	0x1FFFD0	PPM.CHAN_1	Energy-histo status of digital Ch_1 to Ch_32	R
Histo_Status_2	4	0x7FFF44	ReM_FPGA	ModReg32	0x1FFFD1	PPM.CHAN_2	Energy-histo status of digital	R

Register Name	Size / byte	Byte ADDRESS	Location	Register Type	Word32 ADDRESS (HDMC)	Register Definition	Description	Acc.
							Ch_33 to Ch_64	
FADCpipe_Status_1	4	0x7FFF48	ReM_FPGA	ModReg32	0x1FFFD2	PPM.CHAN_1	FADC pipeline memory status of Ch_1 to Ch_32	R
FADCpipe_Status_2	4	0x7FFF4C	ReM_FPGA	ModReg32	0x1FFFD3	PPM.CHAN_2	FADC pipeline memory status of Ch_33 to Ch_64	R
LUTpipe_Status_1	4	0x7FFF50	ReM_FPGA	ModReg32	0x1FFFD4	PPM.CHAN_1	LUT/BCID pipeline memory status of Ch_1 to Ch_32	R
LUTpipe_Status_2	4	0x7FFF54	ReM_FPGA	ModReg32	0x1FFFD5	PPM.CHAN_2	LUT/BCID pipeline memory status of Ch_33 to Ch_64	R
Playback_Status	4	0x7FFF58	ReM_FPGA	ModReg32	0x1FFFD6	PPM.SIFs	Playback memory status (SerIF channel-pairs ORed) 64 Chans :: 32 bits	R
Phos4_ASIC_Status	4	0x7FFF5C	ReM_FPGA	ModReg32	0x1FFFD7	PPM.SIFs see 0x1FFFBF ?	Phos4 status (Each Phos4 mapped onto odd+even bit) 16 Phos4s :: 32 bits	R
ReadOut_FADC_LUT	4	0x7FFF60	ReM_FPGA	ModReg32	0x1FFFD8	PPM.RO_Config	Configure PPr Readout: (#FADC, #LUT/BCID)	R/W
GLink_DAV	4	0x7FFF64	ReM_FPGA	ModReg32	0x1FFFD9	PPM.GLink	Extend 'Data Avail.' on GLink protocol	R/W
Chan_Disable_1	4	0x7FFF68	ReM_FPGA	ModReg32	0x1FFFDA	PPM.CHAN_1	VME writable mask of disabled Chans (eg. LUT="0") to set ERROR bits to ROD (digital Ch_1 ... Ch_32)	R/W
Chan_Disable_2	4	0x7FFF6C	ReM_FPGA	ModReg32	0x1FFFDB	PPM.CHAN_2	VME writable mask of disabled Chans (eg. LUT="0") to set ERROR bits to ROD (digital Ch_33 ... Ch_64)	R/W
MCM_Control	4	0x7FFF70	ReM_FPGA	ModReg32	0x1FFFDC	PPM.MCM_Control		
LocalTrig_Timing	4	0x7FFF80	ReM_FPGA	ModReg32	0x1FFFE0	PPM.GENL1A_T	'self-triggering' operation	R/W
LocalTrig_Config	4	0x7FFF84	ReM_FPGA	ModReg32	0x1FFFE1	PPM.GENL1A_C	'self-triggering' operation	R/W
LocalCounterReset	4	0x7FFF88	ReM_FPGA	ModReg32	0x1FFFE2	PPM.Counter_Reset	'self-operation' Resets	R/W
ReM_Fware_Vers	4	0x7FFFD0	ReM_FPGA	ModReg32	0x1FFFF4	PPM.ReM_Version	Firmware-Version loaded into in ReM	R/W
VME_ReM_Status	4	0x7FFFD4	ReM_FPGA	ModReg32	0x1FFFF5	PPM.ReM_Status	VME-ReM_Interface: Status Register	R
VME_ReM_ DAQControl	4	0x7FFFD8	ReM_FPGA	ModReg32	0x1FFFF6	PPM.DAQ_Control	VME-ReM_Interface: DAQ-Control Register	R/W
VME_ReM_Control	4	0x7FFFDC	ReM_FPGA	ModReg32	0x1FFFF7	PPM.ReM_Control	VME-ReM_Interface: Control Register	R/W
VME_ReM_Command	4	0x7FFFE0	ReM_FPGA	ModReg32	0x1FFFF8	PPM.ReM_Command	VME-ReM_Interface: Command Register	R/W
VME_ReM_Error	4	0x7FFFE4	ReM_FPGA	ModReg32	0x1FFFF9	PPM.ReM_Error	VME-ReM_Interface: Error Register	R

Table 3 : Addressing VME Registers for control/set-up of the Pre-Processor Module.

The following table lists the locations for event-data readout assembled in the ReM_FPGA.

Register Name	Size / byte	ADDRESS	Location	Register Type	Word32 ADDRESS (HDMC)	Register Definition	Description	Acc.
Readout		abs. Address						
ROB_Stat_ **MCM0**_SIF_AB	4	0x600000	ReM_FPGA	ModReg32	0x180000	PPM.ROBStat		R/W
ROB_ **MCM0**_FIFO_AB	4	0x600004	ReM_FPGA	ModFIFO32	0x180001	PPM.Default		R/W
ROB_Stat_ **MCM0**_SIF_CD	4	0x600008	ReM_FPGA	ModReg32	0x180002	PPM.ROBStat		R/W
ROB_	4	0x60000C	ReM_FPGA	ModFIFO32	0x180003	PPM.Default		R/W

Register Name	Size / byte	ADDRESS	Location	Register Type	Word32 ADDRESS (HDMC)	Register Definition	Description	Acc.
MCM0_FIFO_CD								
ROB_Stat_ **MCM1**_SIF_AB	4	0x600010	ReM_FPGA	ModReg32	0x180004	PPM.ROBStat		R/W
ROB_ **MCM1**_FIFO_AB	4	0x600014	ReM_FPGA	ModFIFO32	0x180005	PPM.Default		R/W
ROB_Stat_ **MCM1**_SIF_CD	4	0x600018	ReM_FPGA	ModReg32	0x180006	PPM.ROBStat		R/W
ROB_ **MCM1**_FIFO_CD	4	0x60001C	ReM_FPGA	ModFIFO32	0x180007	PPM.Default		R/W
ROB_Stat_ **MCM2**_SIF_AB	4	0x600020	ReM_FPGA	ModReg32	0x180008	PPM.ROBStat		R/W
ROB_ **MCM2**_FIFO_AB	4	0x600024	ReM_FPGA	ModFIFO32	0x180009	PPM.Default		R/W
ROB_Stat_ **MCM2**_SIF_CD	4	0x600028	ReM_FPGA	ModReg32	0x18000A	PPM.ROBStat		R/W
ROB_ **MCM2**_FIFO_CD	4	0x60002C	ReM_FPGA	ModFIFO32	0x18000B	PPM.Default		R/W
ROB_Stat_ **MCM3**_SIF_AB	4	0x600030	ReM_FPGA	ModReg32	0x18000C	PPM.ROBStat		R/W
ROB_ **MCM3**_FIFO_AB	4	0x600034	ReM_FPGA	ModFIFO32	0x18000D	PPM.Default		R/W
ROB_Stat_ **MCM3**_SIF_CD	4	0x600038	ReM_FPGA	ModReg32	0x18000E	PPM.ROBStat		R/W
ROB_ **MCM3**_FIFO_CD	4	0x60003C	ReM_FPGA	ModFIFO32	0x18000F	PPM.Default		R/W
ROB_Stat_ **MCM4**_SIF_AB	4	0x600040	ReM_FPGA	ModReg32	0x180010	PPM.ROBStat		R/W
ROB_ **MCM4**_FIFO_AB	4	0x600044	ReM_FPGA	ModFIFO32	0x180011	PPM.Default		R/W
ROB_Stat_ **MCM4**_SIF_CD	4	0x600048	ReM_FPGA	ModReg32	0x180012	PPM.ROBStat		R/W
ROB_ **MCM4**_FIFO_CD	4	0x60004C	ReM_FPGA	ModFIFO32	0x180013	PPM.Default		R/W
MCM5 **MCM6** **MCM7**		**... ...**						
ROB_Stat_ **MCM8**_SIF_AB	4	0x600080	ReM_FPGA	ModReg32	0x180020	PPM.ROBStat		R/W
ROB_ **MCM8**_FIFO_AB	4	0x600084	ReM_FPGA	ModFIFO32	0x180021	PPM.Default		R/W
ROB_Stat_ **MCM8**_SIF_CD	4	0x600088	ReM_FPGA	ModReg32	0x180022	PPM.ROBStat		R/W
ROB_ **MCM8**_FIFO_CD	4	0x60008C	ReM_FPGA	ModFIFO32	0x180023	PPM.Default		R/W
MCM9 **MCM10** **MCM11**		**... ...**						
ROB_Stat_ **MCM12**_SIF_AB	4	0x6000C0	ReM_FPGA	ModReg32	0x180030	PPM.ROBStat		R/W
ROB_ **MCM12**_FIFO_AB	4	0x6000C4	ReM_FPGA	ModFIFO32	0x180031	PPM.Default		R/W
ROB_Stat_ **MCM12**_SIF_CD	4	0x6000C8	ReM_FPGA	ModReg32	0x180032	PPM.ROBStat		R/W
ROB_ **MCM12**_FIFO_CD	4	0x6000CC	ReM_FPGA	ModFIFO32	0x180033	PPM.Default		R/W
ROB_Stat_ **MCM13**_SIF_AB	4	0x6000D0	ReM_FPGA	ModReg32	0x180034	PPM.ROBStat		R/W
ROB_ **MCM13**_FIFO_AB	4	0x6000D4	ReM_FPGA	ModFIFO32	0x180035	PPM.Default		R/W
ROB_Stat_ **MCM13**_SIF_CD	4	0x6000D8	ReM_FPGA	ModReg32	0x180036	PPM.ROBStat		R/W
ROB_ **MCM13**_FIFO_CD	4	0x6000DC	ReM_FPGA	ModFIFO32	0x180037	PPM.Default		R/W
ROB_Stat_ **MCM14**_SIF_AB	4	0x6000E0	ReM_FPGA	ModReg32	0x180038	PPM.ROBStat		R/W

Register Name	Size / byte	ADDRESS	Location	Register Type	Word32 ADDRESS (HDMC)	Register Definition	Description	Acc.
ROB_ **MCM14**_FIFO_AB	4	0x6000E4	ReM_FPGA	ModFIFO32	0x180039	PPM.Default		R/W
ROB_Stat_ **MCM14**_SIF_CD	4	0x6000E8	ReM_FPGA	ModReg32	0x18003A	PPM.ROBStat		R/W
ROB_ **MCM14**_FIFO_CD	4	0x6000EC	ReM_FPGA	ModFIFO32	0x18003B	PPM.Default		R/W
ROB_Stat_ **MCM15**_SIF_AB	4	0x6000F0	ReM_FPGA	ModReg32	0x18003C	PPM.ROBStat		R/W
ROB_ **MCM15**_FIFO_AB	4	0x6000F4	ReM_FPGA	ModFIFO32	0x18003D	PPM.Default		R/W
ROB_Stat_ **MCM15**_SIF_CD	4	0x6000F8	ReM_FPGA	ModReg32	0x18003E	PPM.ROBStat		R/W
ROB_ **MCM15**_FIFO_CD	4	0x6000FC	ReM_FPGA	ModFIFO32	0x18003F	PPM.Default		R/W

Table 4 : Addressing Data-Readout from ASICs on PPM.

The PPM follows the DAQ-protocol delivered by the TTC system to the TTC-decoder module. The TTCdec has an on-board I^2C-bus access, which is mapped to VME as shown in the table below.

Register Name	Size / byte	ADDRESS	Location	Register Type	Word32 ADDRESS (HDMC)	Register Definition	Description	Acc.
TTCdec Registers		abs. Address					see: TTCrx Manual	
TTCFineDelay1	4	0x240000	S-RAM	ModReg32	0x090000	PPM.TTCDelay		R/W
TTCFineDelay2	4	0x240004	S-RAM	ModReg32	0x090001	PPM.TTCDelay		R/W
TTCCoarseDelay	4	0x240008	S-RAM	ModReg32	0x090002	PPM.TTCDelay		R/W
TTCControl	4	0x24000C	S-RAM	ModReg32	0x090003	PPM.TTCControl		R/W
TTCDummy1	4	0x240010	S-RAM	ModReg32	0x090004	PPM.TTCDummy	NOT accessible	
TTCDummy2	4	0x240014	S-RAM	ModReg32	0x090005	PPM.TTCDummy	NOT accessible	
TTCDummy3	4	0x240018	S-RAM	ModReg32	0x090006	PPM.TTCDummy	NOT accessible	
TTCDummy4	4	0x24001C	S-RAM	ModReg32	0x090007	PPM.TTCDummy	NOT accessible	
TTCSnglErrCnt0_7	4	0x240020	S-RAM	ModReg32	0x090008	PPM.TTCErrCnt	Error Counter (volatile)	R/W
TTCSnglErrCnt8_15	4	0x240024	S-RAM	ModReg32	0x090009	PPM.TTCErrCnt	Error Counter (volatile)	R/W
TTCDblErrCnt0_7	4	0x240028	S-RAM	ModReg32	0x09000A	PPM.TTCErrCnt	Error Counter (volatile)	R/W
TTCSEUErrCnt8_15	4	0x24002C	S-RAM	ModReg32	0x09000B	PPM.TTCErrCnt	Error Counter (volatile)	R/W
TTCDummy5	4	0x240030	S-RAM	ModReg32	0x09000C	PPM.TTCDummy	NOT accessible	
TTCDummy6	4	0x240034	S-RAM	ModReg32	0x09000D	PPM.TTCDummy	NOT accessible	
TTCDummy7	4	0x240038	S-RAM	ModReg32	0x09000E	PPM.TTCDummy	NOT accessible	
TTCDummy8	4	0x24003C	S-RAM	ModReg32	0x09000F	PPM.TTCDummy	NOT accessible	
TTCID0_7	4	0x240040	S-RAM	ModReg32	0x090010	PPM.TTCID1		R/W
TTCID6_13MMA	4	0x240044	S-RAM	ModReg32	0x090011	PPM.TTCID2		R/W
TTCI2CID_MMB	4	0x240048	S-RAM	ModReg32	0x090012	PPM.TTCID3		R/W
TTCConfig1	4	0x24004C	S-RAM	ModReg32	0x090013	PPM.TTCConfig		R/W
TTCConfig2	4	0x240050	S-RAM	ModReg32	0x090014	PPM.TTCConfig		R/W
TTCConfig3	4	0x240054	S-RAM	ModReg32	0x090015	PPM.TTCConfig		R/W
TTCStatus	4	0x240058	S-RAM	ModReg32	0x090016	PPM.TTCStatus		R/W
TTCDummy9	4	0x24005C	S-RAM	ModReg32	0x090017	PPM.TTCDummy	NOT accessible	
TTCBC0_7	4	0x240060	S-RAM	ModReg32	0x090018	PPM.TTCBC	BC Counter (volatile)	R/W
TTCBC8-15	4	0x240064	S-RAM	ModReg32	0x090019	PPM.TTCBC	BC Counter (volatile)	R/W
TTCEC0_7	4	0x240068	S-RAM	ModReg32	0x09001A	PPM.TTCEC	Event Counter (volatile)	R/W
TTCEC8_15	4	0x24006C	S-RAM	ModReg32	0x09001B	PPM.TTCEC	Event Counter (volatile)	R/W
TTCEC16_23	4	0x240070	S-RAM	ModReg32	0x09001C	PPM.TTCEC	Event Counter (volatile)	R/W

Table 5 : Addressing VME Registers on the PPM-wide TTCdec module.

The handling of parameters on the PPM makes intensive use of the 'intermediate' S-RAM store. ReM-firmware operates the on-board 'bus-systems' (SPI, I^2C, ser. Interfaces). A scheme, described below, is implemented to facilitate the distribution and verification of the multitude of parameters loaded to PPM-destinations.

The controlling program (e.g. online DAQ software) writes a data-set via VME to the S-RAM. The set is called

'Reference' (see 'S-RAM Content' below and 4.4.2. 'Verfication of settings'). Firmware transports these parameters to the destinations. Simultaneously, an exact copy is placed in the 'Readback' section of the S-RAM. Each readable parameter here is flagged as 'obsolete' (flag-bit#32 = 1).

Data words in the 'Readback' section are updated by the ReM-firmware, when a request for 'read back' from 'source' is made by software. The flag-bit is set to '0' indicating that the parameter is 'up-to-date'.

Comparison of the 'Reference' and 'Readback' sections allows easy verification of parameter loading. The scheme is used for all destinations, where special on-board buses are involved (e.g. DACs, Phos4s, TTCdec, PPrASICs). Note, that DACs and Phos4s cannot be read back from the actual device. Hence, the 'Readback' shows just a copy of the 'Reference'.

The base-addresses of the register-space for each of the 16 MCM-submodules are given below.

Register Name	Size / byte	ADDRESS	Location	Register Type	Word32 ADDRESS (HDMC)	Register Definition	Description	Acc.
MCMs		abs. Address						
PPrMCM [0]	8192	0x200000	S-RAM	PprMCM	0x080000	see Table 7	Block of MCM Registers	R/W
PPrMCM [1]	8192	0x202000	S-RAM	PprMCM	0x080800		Block of MCM Registers	
PPrMCM [2]	8192	0x204000	S-RAM	PprMCM	0x081000		Block of MCM Registers	
PPrMCM [3]	8192	0x206000	S-RAM	PprMCM	0x081800		Block of MCM Registers	
PPrMCM [4]	8192	0x208000	S-RAM	PprMCM	0x082000		Block of MCM Registers	
PPrMCM [5]	8192	0x20A000	S-RAM	PprMCM	0x082800		Block of MCM Registers	
PPrMCM [6]	8192	0x20C000	S-RAM	PprMCM	0x083000		Block of MCM Registers	
PPrMCM [7]	8192	0x20E000	S-RAM	PprMCM	0x083800		Block of MCM Registers	
PPrMCM [8]	8192	0x210000	S-RAM	PprMCM	0x084000		Block of MCM Registers	
PPrMCM [9]	8192	0x212000	S-RAM	PprMCM	0x084800		Block of MCM Registers	
PPrMCM [10]	8192	0x214000	S-RAM	PprMCM	0x085000		Block of MCM Registers	
PPrMCM [11]	8192	0x216000	S-RAM	PprMCM	0x085800		Block of MCM Registers	
PPrMCM [12]	8192	0x218000	S-RAM	PprMCM	0x086000		Block of MCM Registers	
PPrMCM [13]	8192	0x21A000	S-RAM	PprMCM	0x086800		Block of MCM Registers	
PPrMCM [14]	8192	0x21C000	S-RAM	PprMCM	0x080800		Block of MCM Registers	
PPrMCM [15]	8192	0x21E000	S-RAM	PprMCM	0x087800		Block of MCM Registers	

Table 6 : Addressing MCMs on a PPM.

The hierarchical description of registers descends to the level of the 16 indexed Pre-Processor Multi-Chip Modules installed on the PPM. The set for each **PPrMCM** - [#0 to #15] - is composed of data for:

- analog pulse-conditioning on the AnIn board,
- timing of the strobe for digitisation in the Phos4,
- GLOBAL registers controlling pairs of serial interfaces,
- and, finally, the set of control-registers for each of four PPrASIC-channels.

Every MCM holds four Pre-Processor channels. The data-volume required to serve a MCM amounts to 8192 bytes or 2048 'words of 32 bit', i.e. 0x2000 bytes = 0x800 words. Addresses given in the table below are **relative** to the start-address of the respective PPrMCM (see above).

Register Name	Size / byte	ADDRESS-Offset	Location	Register Type	Word32 ADDRESS (HDMC)	Register Definition	Description	Acc.
NonAsic-/ SIF-Regs		**MCM[n] n=0,1, ... 15**						
DAC_Channel_1	4	+0x0000		PpmReg32	+0x0000	PPM.DAC	For 'reorder sequence':	R/W
DAC_Channel_4	4	+0x0004		PpmReg32	+0x0001	PPM.DAC	see: MCM Ref.Man.	R/W
DAC_Channel_2	4	+0x0008		PpmReg32	+0x0002	PPM.DAC	For 'reorder sequence':	R/W
DAC_Channel_3	4	+0x000C		PpmReg32	+0x0003	PPM.DAC	see: MCM Ref.Man.	R/W
PHOS4_ChannelA	4	+0x0010		PpmReg32	+0x0004	PPM.PHOS4		R/W
PHOS4_ChannelB	4	+0x0014		PpmReg32	+0x0005	PPM.PHOS4		R/W
PHOS4_ChannelC	4	+0x0018		PpmReg32	+0x0006	PPM.PHOS4		R/W
PHOS4_ChannelD	4	+0x001C		PpmReg32	+0x0007	PPM.PHOS4		R/W
SIF0_ChannelAB	4	+0x0020		PpmReg32	+0x0008	PPM.SIF0	ASIC:global Reg	R/W

Register Name	Size / byte	ADDRESS-Offset	Location	Register Type	Word32 ADDRESS (HDMC)	Register Definition	Description	Acc.
SIF1_ChannelAB	4	+0x0024		PpmReg32	+0x0009	PPM.SIF1	ASIC:global Reg	R/W
SIF2_ChannelAB	4	+0x0028		PpmReg32	+0x000A	PPM.SIF2	ASIC:global Reg	R/W
SIF3_ChannelAB	4	+0x002C		PpmReg32	+0x000B	PPM.SIF3	ASIC:global Reg	R/W
SIF4_ChannelAB	4	+0x0030		PpmReg32	+0x000C	PPM.SIF4	ASIC:global Reg	R/W
SIF0_ChannelCD	4	+0x0040		PpmReg32	+0x0010	PPM.SIF0	ASIC:global Reg	R/W
SIF1_ChannelCD	4	+0x0044		PpmReg32	+0x0011	PPM.SIF1	ASIC:global Reg	R/W
SIF2_ChannelCD	4	+0x0048		PpmReg32	+0x0012	PPM.SIF2	ASIC:global Reg	R/W
SIF3_ChannelCD	4	+0x004C		PpmReg32	+0x0013	PPM.SIF3	ASIC:global Reg	R/W
SIF4_ChannelCD	4	+0x0050		PpmReg32	+0x0014	PPM.SIF4	ASIC:global Reg	R/W
ASIC Channels								
Chan[A]	2048	+0x0060		PprChan	**+0x0018**		ASIC: Channel Regs	
Chan[B]	2048	+0x0860		PprChan	**+0x0218**		ASIC: Channel Regs	
Chan[C]	2048	+0x1060		PprChan	**+0x0418**		ASIC: Channel Regs	
Chan[D]	2048	+0x1860		PprChan	**+0x0618**		ASIC: Channel Regs	

Table 7 : Addressing Registers for VME-access to a MCM [1 of 16] on a Pre-Processor Module.

Each of the four **PPrASIC channels** - [#A to #D] - is controlled by registers defined in the following table. The data-length for a single ASIC channel is 2048 bytes (0x800). Again, addresses given are **relative** to the start-address of the respective PPrASIC channel (see above).

Register Name	Size / byte	ADDRESS-Offset	Location	Register Type	Word32 ADDRESS (HDMC)	Register Definition	Description	Acc.
ASIC ChannelRegisters		**Chan [m] m=A,B,C,D**						
Playback_BT		++0x0000		ModReg32	0x0000	PPM.Default	Byte Transfer to Plbk ?	
Playback_Mem	512	++0x0004	PPrASIC	ModMem32 [≤ 0x7F]	++0x0001 => ++0x0080	**PPM.PBM**	256*11bit Playback/Histo Memory: 2*16bit packed, i.e. [127 words of 32 bit]	R/W
LUT_Load	4	++0x021C		ModReg32	+0x0087	PPM.Default	God only knows ?	R/W
LUT_BT		++0x0220		ModReg32	+0x0088	PPM.Default	Byte Transfer to LUT ?	
LUT_Mem	1024	++0x0224	PPrASIC	ModMem32 [≤0xFF]	++0x0089 => ++0x0188	**PPM.LUT**	1024*8bit Look-upTable Memory: 4*8bit packed. i.e. [255 words of 32 bit]	R/W
ChannelControlReg[00]	4	++ 0x0624	PPrASIC	PpmReg32	++0x0189	PPM.CR0	Control Register	R/W
ChannelControlReg[01]	4	++ 0x0628	PPrASIC	PpmReg32	++0x018A	PPM.CR1	Control Register	R/W
ChannelControlReg[02]	4	++ 0x062C	PPrASIC	PpmReg32	++0x018B	PPM.CR2	Control Register	R/W
ChannelControlReg[03]	4	++ 0x0630	PPrASIC	PpmReg32	++0x018C	PPM.CR3	Control Register	R/W
ChannelControlReg[04]	4	++ 0x0634	PPrASIC	PpmReg32	++0x018D	PPM.CR4	Control Register	R/W
ChannelControlReg[05]	4	++ 0x0638	PPrASIC	PpmReg32	++0x018E	PPM.CR5	Control Register	R/W
ChannelControlReg[06]	4	++ 0x063C	PPrASIC	PpmReg32	++0x018F	PPM.CR6	Control Register	R/W
ChannelControlReg[07]	4	++ 0x0640	PPrASIC	PpmReg32	++0x0190	PPM.CR7	Control Register	R/W
ChannelControlReg[08]	4	++ 0x0644	PPrASIC	PpmReg32	++0x0191	PPM.CR8	Control Register	R/W
ChannelControlReg[09]	4	++ 0x0648	PPrASIC	PpmReg32	++0x0192	PPM.CR9	Control Register	R/W
ChannelControlReg[10]	4	++ 0x064C	PPrASIC	PpmReg32	++0x0193	PPM.CR10	Control Register	R/W
ChannelControlReg[11]	4	++ 0x0650	PPrASIC	PpmReg32	++0x0194	PPM.CR11	Control Register	R/W
ChannelControlReg[12]	4	++ 0x0654	PPrASIC	PpmReg32	++0x0195	PPM.CR12	Control Register	R/W
ChannelControlReg[13]	4	++ 0x0658	PPrASIC	PpmReg32	++0x0196	PPM.CR13	Control Register	R/W
ChannelControlReg[14]	4	++ 0x065C	PPrASIC	PpmReg32	++0x0197	PPM.CR14	Control Register	R/W
ChannelControlReg[15]	4	++ 0x0660	PPrASIC	PpmReg32	++0x0198	PPM.CR15	Control Register	R/W
ChannelControlReg[16]	4	++ 0x0664	PPrASIC	PpmReg32	++0x0199	PPM.CR16	Control Register	R/W
ChannelControlReg[17]	4	++ 0x0668	PPrASIC	PpmReg32	++0x019A	PPM.CR17	Control Register	R/W
ChannelControlReg[18]	4	++ 0x066C	PPrASIC	PpmReg32	++0x019B	PPM.CR18	Control Register	R/W
ChannelControlReg[19]	4	++ 0x0670	PPrASIC	PpmReg32	++0x019C	PPM.CR19	Control Register	R/W
ChannelControlReg[20]	4	++ 0x0674	PPrASIC	PpmReg32	++0x019D	PPM.CR20	Control Register	R/W

Register Name	Size / byte	ADDRESS-Offset	Location	Register Type	Word32 ADDRESS (HDMC)	Register Definition	Description	Acc.
ChannelControlReg[21]	4	++ 0x0678	PPrASIC	PpmReg32	++0x019E	PPM.CR21	Control Register	R/W
ChannelControlReg[22]	4	++ 0x067C	PPrASIC	PpmReg32	++0x019F	PPM.CR22	Control Register	R/W
ChannelControlReg[23]	4	++ 0x0680	PPrASIC	PpmReg32	++0x01A0	PPM.CR23	Control Register	R/W
ChannelControlReg[24]	4	++ 0x0684	PPrASIC	PpmReg32	++0x01A1	PPM.CR24	Control Register	R/W
ChannelControlReg[25]	4	++ 0x0688	PPrASIC	PpmReg32	++0x01A2	PPM.CR25	Control Register	R/W
ChannelControlReg[26]	4	++ 0x068C	PPrASIC	PpmReg32	++0x01A3	PPM.CR26	Control Register	R/W
ChannelControlReg[27]	4	++ 0x0690	PPrASIC	PpmReg32	++0x01A4	PPM.CR27	Control Register	R/W
ChannelControlReg[28]	4	++ 0x0694	PPrASIC	PpmReg32	++0x01A5	PPM.CR28	Control Register	R/W
ChannelControlReg[29]	4	++ 0x0698	PPrASIC	PpmReg32	++0x01A6	PPM.CR29	Control Register	R/W
ChannelControlReg[30]	4	++ 0x069C	PPrASIC	PpmReg32	++0x01A7	PPM.CR30	Control Register	R/W
ChannelControlReg[31]	4	++ 0x06A0	PPrASIC	PpmReg32	++0x01A8	PPM.CR31	Control Register	R/W
ChannelControlReg[32]	4	++ 0x06A4	PPrASIC	PpmReg32	++0x01A9	PPM.CR32	Control Register	R/W
ChannelControlReg[33]	4	++ 0x06A8	PPrASIC	PpmReg32	++0x01AA	PPM.CR33	Control Register	R/W

Table 8 : Addressing Registers for ONE ASIC_channel [A,B,C or D] on a PPrMCM.

A good part of the PPM's address-space (50% = 4 Mbyte) is allocated to the S-RAM. The usage of the S-RAM space is outlined below. It is directly accessible through VME.

ADDRESS	Size /byte	Word32 ADDRESS (HDMC)	**S-RAM Content**	Description	Location Cap: 4 MB	Acc.
abs. Address		abs. Address				
0x200000	8192	0x080000	MCM Register settings	**Reference** Data to load into MCM_0	S-RAM	R/W
0x202000	8192	0x080800	MCM Register settings	**Reference** Data to load into MCM_1	S-RAM	R/W
...			...	...		
0x21E000	8192	0x087800	MCM Register settings	**Reference** Data to load into MCM_15	S-RAM	R/W
0x220000	8192	0x088000	MCM Register readings	Readback Data from MCM_0	S-RAM	R/W
0x222000	8192	0x088800	MCM Register readings	Readback Data from MCM_1	S-RAM	R/W
...		…	...	...		
0x23E000	8192	0x08F800	MCM Register readings	Readback Data from MCM_15	S-RAM	R/W
0x240000	128	0x090000	TTC_Decoder settings	**Reference** TTC settings (32 Regs foreseen)	S-RAM	R/W
0x240100	128	0x090040	TTC_Decoder readings	Readback TTC settings (32 Regs foreseen)	S-RAM	R/W
0x260000	32768	0x098000	Plbck_Data_1 (11/16 bit used)	Play-back Data Set_1: (2*16bit)* 256 samples* 64 chs	S-RAM	R/W
0x268000	32768	0x09A000	Plbck_Data_2 (11/16 bit used)	Play-back Data Set_2: (2*16bit)* 256 samples* 64 chs	S-RAM	R/W
0x270000	32768	0x09C000	Plbck_Data_3 (11/16 bit used)	"	S-RAM	R/W
0x278000	32768	0x09E000	Plbck_Data_4 (11/16 bit used)	"	S-RAM	R/W
0x280000	32768	0x0A0000	Plbck_Data_5 (11/16 bit used)	"	S-RAM	R/W
0x288000	32768	0x0A2000	Plbck_Data_6 (11/16 bit used)	"	S-RAM	R/W
0x290000	32768	0x0A4000	Plbck_Data_7 (11/16 bit used)	"	S-RAM	R/W
0x298000	32768	0x0A6000	Plbck_Data_8 (11/16 bit used)	"	S-RAM	R/W
0x300000 => 0x3001FC	512	0x0C0000	Reference_Rates	**Reference** Store for Delta-Time / Rate-Count (from DAQ)	S-RAM	R/W
0x300200 => 0x3003FC	512	0x0C0080	Readback_Rates	Readback Store for Delta-Time / Rate-Count	S-RAM	R/W
0x310000 => 0x317FFC	32768	0x0C4000	Reference_Histos	**Reference** Store for FADC energy-spectra (from DAQ)	S-RAM	R/W
0x318000 => 0x31FFFC	32768	0x0C6000	Readback_Histos	Readback Store for FADC energy-spectra	S-RAM	R/W
0x320000 => 0x5FFFFC = End of Mem.			Spy_Event_Buffer for PPM (header+3raw+1bcid) = Readout	Spy-Event Data Buffer for local Monitoring: 5*32bit *64 chs +empty word /event => to be implemented	S-RAM	R/W

Table 9 : Address Mapping for the 'peripheral' S-RAM (4 MByte).

Data-Format for ‘rate’(2 words) /channel.

24	16	8	0
xxxx xxxx	xxxx xxxx	DeltaTime_H	DeltaTime_L

24	20	10	0
xxxx xxxx	xxxx	Rate_H	Rate_L

Data-Format for ‘energy histogram’(128 words) /channel

27	16	11	0
xxxx x	#Counts / bin#1	xxxx x	#Counts / bin#0
… … …			
… … …			
… … …			
xxxx x	#Counts / bin#255	xxxx x	#Counts / bin#254

4.3.3 Register Definition

Register Definitions (i.e. bit-wise content) for all PPM-registers referenced in the tables above are given below.

HDMC Register Definition	Bit#	Default Value (dec.)	Name	Description
PPM Registers on VME_CPLD				
PPM.Base	0-15 16-23 24-31	**1845** 0 **13**	KIP_Identifier PPM_Serial_Number CPLD_Fware_Vers	when Gustav established electrical laws SerNr-range: 001 to 160 Version loaded in VME-protocol CPLD
PPM.VMEConf	0-7 8-14 15 16-18 19 20 21 22 23 24 25 26 27 28 29 30 31		Intrpt Intrpt_RQ Intrpt_EN Intrpt_Encode SOFT_Intrpt_RQ Flash_Load_SEL CAN_VME_RESET_EN VMECAN_RESET VMECAN_PrgSEL JTAG_TCK JTAG_TDO JTAG_TMS JTAG_TRST JTAG_VME_EN ATMEGA_RESET XCVE_RESET PWR_OFFON	Interrupt Vector asserted by Crate-Master CPU Interrupt Level set by PPM as request (7 bit binary) Enable PPM to set Interrupt Request Encoded Interrupt Level (3 bits) Interrupt Request generated by PPM-algorithm Load FPGAs from FlashMem, else from VME Enable CAN to reset PPM Reset CAN micro-controller from VME Select CAN controller to load CAN-firmware from RS232 JTAG Test Clock JTAG Test Data Out JTAG Test Mode State JTAG Test Reset Enable JTAG via VME Reset ATMEGA micro-controller Reset ReM_FPGA “Soft” Power Off/On on PPM for Reset; HSwap-Ctrl re-powers
PPM.VMEStat	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15		XC2V_Prg_Done2_0 XC2V_Prg_Done1_0 W_XCVE_CPLD ATMEGA_CPLD_DIS ATMEGA_POWER_OK JTAG_TDI XCVE_VME_Intrpt XCVE_PHOS4_CLK_EN XC2V_Prg_Done2 XC2V_Prg_Init2 XC2V_Prg_Done1 XC2V_Prg_Init1 XCVE_Prg_Done XCVE_Prg_Init XILINX_Prg_BUSY CAN_RESET	“done” flag for first JEP XC2V on LCD; ready to load second “done” flag for first CP XC2V on LCD; ready to load second “reserved” Disable 4bit comm. from VME_CPLD, while prog.loading. ALL Voltages, Temperatures on PPM are OK; “big OR” JTAG Test Data In Interrupt from Rem_FPGA; e.g. Monit. Data are available Enable Phos4-Clock from Rem_FPGA after fw loaded Loading of firmware into LCD_FPGAs (JEP) finished Start Loading of firmware into LCD_FPGAs (JEP) Loading of firmware into LCD_FPGAs (CP) finished Start Loading of firmware into LCD_FPGAs (CP) Loading of firmware into ReM_FPGA finished Start Loading of firmware into ReM_FPGA Firmware Load on PPM in progress CAN resets the PPM

HDMC Register Definition	Bit#	Default Value (dec.)	Name	Description
	16-31		yet unused	
PPM.ATMEGA	0-31		32 bit data	
PPM.GENL1A_T	0-14	0 ?	delay	set delay for first local L1-A
	15	0	enable	enable generation of L1-A as configured
unit ?	16-30	0	delay_external	set delay for external pulse to trigger a pulse generator
	31	0	enable_external	enable generation of external pulse
PPM.GENL1A_C	0-7	0 ?	L1A_count	set number of L1-As to be generated
electrical output ? unit ?	8-11	0	length_external	set length of external pulse to trigger a pulse-generator
16 AnIn-chs already ORed?	12	0	AnIn1_mask	select AnIn (1,2,3,4) for local triggering (one or more ORed)
	13	0	AnIn2_mask	"
	14	0	AnIn3_mask	"
	15	0	AnIn4_mask	"
	16-31	0	L1A_gap	Time distance between L1-As (in LHC clock ticks)
PPM.Counter_Reset	0	0	BcCntRst	Reset BunchCrossing counter locally on PPM (self op. only)
	1	0	EcCntRst	Reset Event counter locally on PPM (self operation only)
	2-31		Unused	
PPM Registers on FlashLoad_CPLD				
PPM.FRAddr	0-31		32bit data	
PPM.FRDat	0-31		32bit data	
PPM.FRRes	0-31		32bit data	
PPM.FRLoad	0-7		yet unused	
	8-19		yet unused	
	20		XCVE_Vers_LSB	select XCVE-version to load from FlashRam (1-6)
	21	1	XCVE_Vers_1	select XCVE-version to load from FlashRam (1-6)
	22	0	XCVE_Vers_MSB	select XCVE-version to load from FlashRam (1-6)
	23	0		
	24		FLASH_SEL_XCVE	select ReM_FPGA (XCV1000e) for firmware load
	25		FLASH_SEL_XC2V_1	select LCD_CP_ FPGAs (XC2V_250) for firmware load
	26		FLASH_SEL_XC2V_2	select LCD_JEP_ FPGAs (XC2V_250) for firmware load
	27		unused	...
	28		unused	...
	29		unused	...
	30		FLASH_RESET_XC2V	Reset Flash-Loader for LCD_FPGAs
	31		FLASH_RESET_XCVE	Reset Flash-Loader for ReM_FPGA
PPM.FRStat	0		FLASHRAM_BUSY	
	1		FLASH_Prg_XIL_BUSY	FlashRAM BUSY transferring firmware bit-files
	2		FLASH_FREE3	
	3		FLASH_FREE2	
	4		FLASH_XIL_Prg_BUSY	? see bit 1.
	5		FLASH_XC2V2_OK	
	6		FLASH_XC2V1_OK	
	7		FLASH_XCVE_OK	
	8		FLASH_XC2V_Done2	
	9		FLASH_XC2V_Done2_0	
	10		FLASH_XC2V_Init2	
	11		FLASH_XC2V_Done1	
	12		FLASH_XC2V_Done1_0	
	13		FLASH_XC2V_Init1	
	14		FLASH_XCVE_Done	
	15		FLASH_XCVE_Init	
	16-31		yet unused	
PPM.XCVReprog	0-7		byte-val	
PPM.XCVProgDat	0-7		byte-val	
PPM.XC2VReprog	0-7		byte-val	
PPM.XC2VProgDat1	0-7		byte-val	
PPM.XC2VProgDat2	0-7		byte-val	

HDMC Register Definition	Bit#	Default Value (dec.)	Name	Description
Common PPM Registers on ReM_FPGA				
PPM.DEFAULT	0-31		32bit value	wherever needed
PPM.ReM_Version	0-15 16-31		subversion number mainversion number	Version-number read from loaded ReM-bitfile: MainVersion . SubVersion
PPM.ReM_Status	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 Unused	DAQ_active PPM_Load SRAM_Init_Busy VME_SRAM_Read_Status VME_SRAM_Write_Status Rdbk_SRAM_Write_Status ASIC_rip ASIC_wip ReadBack_Active ASIC_Cfg_Rdbk_Active TTCrx_ Rdbk_Active Rate_Rdbk_Active Histo_Rdbk_Active Ratemeter_Enabled Histograms_Enabled Local_Trig_Busy MCM_Lvds_Sync MCM_Lvds_Tclk_RF RGTM_LinkRdy RGTM_Optic_TXfault Empty_Readout_FIFOs Full_Readout_FIFOs ROD_Event_Overflow AnIn_1_SPI_Busy AnIn_2_SPI_Busy AnIn_3_SPI_Busy AnIn_4_SPI_Busy I2C_Phos4_Write_Busy I2C_TTCrx_Write_Busy TTCdec_ClockStat_1 TTCdec_ClockStat_2	PPM in **DAQ mode** PPM in Configuration mode (**/DAQ**) initialise 'Rdbk block' in SRAM Busy: read data from SRAM, transfer to VME Busy: write data from VME to SRAM Busy: write 'readback data' to SRAM read-back from ASIC-registers (ie. send read-back commands) write to ASIC-registers in progress gobal 'readback in progress', i.e. OR of bits #9, 10, 11 Readback of ASIC_Config registers in progress Readback of TTCdec_Config registers in progress Readback of 'tower rates' in progress Readback of 'tower energy histograms' in progress acquisition of 'rates' enabled in all channels via ASIC_reg_17 acquisition of 'histos' enabled in all channels via ASIC_reg_17 **... ??** "GLink Ready" from RGTM-module 1: Lvds Xmitter send Sync.-pattern 1: Asic Data latched at Lvds on 'falling edge' Optical Transmitter FAULT on RGTM ReM_Readout_FIFOs empty due to missing L1A ReM_Readout_FIFOs full ("high water"): too high L1A-rate ReM_Readout_FIFOs overflow: ASIC_RO stopped till emptied Busy writing DACs on AnIn_1 (offsets, thresholds) via SPI Busy writing DACs on AnIn_2 (offsets, thresholds) via SPI Busy writing DACs on AnIn_3 (offsets, thresholds) via SPI Busy writing DACs on AnIn_4 (offsets, thresholds) via SPI Busy setting Phos4-Delays via I2C-bus Busy setting TTCrx registers via I2C-bus Indicator: TTCrx-LHC Clock is in use (default ??) Indicator: TTCrx-XTAL is in use
PPM.DAQ_Control	0 1-31	1	**set DAQ_Mode** Unused	writing here ONLY sets bit#0 in ReM_Control to DAQ !!!
PPM.ReM_Control	0 1 2 3 4 5 6 7-31	0 0 0 0 0 0 0 	DAQ_mode force_ASIC_Rdbk Spy_Buffer_Reset Spy_Buffer_Enable Spy_Single_Event Spy_after_Event Spy_SerIFace Unused	PPM (ReM) IS in DAQ mode (Read only) *with spy-copies ?!* Force Readback of ASIC-regs (by overwriting SRAM-flags) Clear the spy-buffer in SRAM Enable copying of event-data to spy-buffer in SRAM copy a single event (start event) into spy-buffer for VME keep up copying subsequent events into spy-buffer copy all serIF data (rdbk, event data) into spy-buffer
PPM.ReM_Command	0 1 2-31	0 0 	collect_Rates collect_Histos Unused	collect rates from all 64 PPM-channels collect energy-spectra from all 64 PPM-channels
PPM.ReM_Error	0 1 2 3 4 5 6 7	0 0 0 0 0 0 0 0	deny_ASIC_write_DAQ deny_ASIC_write_Rdbk deny_RateEnable deny_RateDisable deny_CollectRates deny_HistoEnable deny_HistoDisable deny_CollectHistos	VME-write to ASIC-regs denied while DAQ=ON VME-write to ASIC-regs denied while 'Readback-in-Progress' VME-'rate enable' denied while 'Readback-in-Progress' VME-'rate disable' denied while 'Readback-in-Progress' VME-'collect rates' denied while 'Readback-in-Progress' VME-'histo enable' denied while 'Readback-in-Progress' VME-'histo disable' denied while 'Readback-in-Progress' VME-'collect histos' denied while 'Readback-in-Progress'

HDMC Register Definition	Bit#	Default Value (dec.)	Name	Description
	8	0	deny_SRAM_DataRead	VME-read from SRAM denied while 'Readback-in-Progress'
	9	0	deny_SRAM_DataWrite	VME-write to SRAM denied while 'Readback-in-Progress'
	10	0	deny_LocalTrig_Delay	VME-write denied to set LocalTrig generation
	11	0	deny_LocalTrig_Config	VME-write denied to set LocalTrig generation
	12	0	deny_LocalCounter_Reset	VME-write denied to RESET the local L1A-Counter
	13	0	deny_ROD_RO_samples	VME-write denied to set #readout samples (FADC+LUT)
	14	0	deny_GLink_DAV	VME-write denied to extend DAV-length for Glink
	15	0	deny_ASIC_Ch_Disable_1	VME-write denied to disable PP-channels
	16	0	deny_ASIC_Ch_Disable_2	VME-write denied to disable PP-channels
	17	0	deny_AnIn_DAC	VME_ReM_SPI-write denied to set AnIn-DACs
	18	0	deny_AnIn_Rdbk	VME_ReM_SPI-write denied due to 'Readback-in-Progress'
	19	0	deny_Phos4	VME_ReM_I2C-write denied to set Phos4-delay
	20	0	deny_Phos4_Rdbk	VME_ReM_I2C-write denied due to 'Readback-in-Progress'
	21	0	deny_MCM_Control	VME-write denied to change MCM-settings
	22	0	deny_TTCrx	VME-write denied to change TTCrx-settings
	23	0	deny_TTCrx_Rdbk	VME-read from TTC-regs denied while 'Readback-in-Progress'
	24	0	deny_TTCrx_nonReg	access to unused/undef. TTC-reg.
	25	0	Invalid_ASIC_Rdbk_Request	Request to an 'invalid'(i.e. nonexistent) ASIC Rdbk-address
	26	0	Invalid_TTCrx_Rdbk_Request	Request to an 'invalid'(i.e. nonexistent) TTCrx Rdbk-address
	27-31		Unused	
PPM.P4DLL	0	0	Phos4_Static_MCM0	Phos4 Delay-Lock-Loop without feedback (detected on ASIC)
	1	0	Phos4_Static_MCM1	"
Phos4 DLL ::	2	0	Phos4_Static_MCM2	"
error =1, ie 1=BAD	3	0	Phos4_Static_MCM3	"
	4	0	Phos4_Static_MCM4	"
	5	0	Phos4_Static_MCM5	"
	6	0	Phos4_Static_MCM6	"
	7	0	Phos4_Static_MCM7	"
	8	0	Phos4_Static_MCM8	"
	9	0	Phos4_Static_MCM9	"
	10	0	Phos4_Static_MCM10	"
	11	0	Phos4_Static_MCM11	"
	12	0	Phos4_Static_MCM12	"
	13	0	Phos4_Static_MCM13	"
	14	0	Phos4_Static_MCM14	"
	15	0	Phos4_Static_MCM15	"
Clocks from ReM_DLLs::	16	0	Sys_ClkDLL_Lock	ReM System clock generation: DLL locked
lock ok =1, ie 1=GOOD	17	0	GLink_ ClkDLL_Lock	External GLink clock generation: DLL locked
	18	0	Mcm_ClkDLL_Lock	MCM (Phos4, ASIC) clock generation: DLL locked
	19	0	McmSer_ClkDLL_Lock	MCM LVDS-serialiser clock generation: DLL locked
	20	0	SRam_ClkDLL_Lock	SRAM clock generation: DLL locked
	21	0	SerIF_ClkDLL_Lock	
	22-31		Unused	
PPM.SIFs	0-31	0	32 bit mask	Bit mask for 32 channel-pairs (SerIFaces on ASICs)
PPM. CHAN_1	0-31	0	32-bit mask	Bit mask for 32 channela on ASIC (#0 to #31)
PPM. CHAN_2	0-31	0	32-bit mask	Bit mask for 32 channela on ASIC (#32 to #63)
PPM.MCM_Control	0	0	VME_Sync_Playback	1: Start Playback synchron. on all 64 PPM-channels (from VME)
	1	0	MCM_Lvds_Sync	1: Enable LVDS Xmitters to send SYNC pattern
	2	0	MCM_Lvds_Tclk_RF	1: Data from ASIC is latched on 'falling edge'
	3	1	MCM_Lvds_D_En	READ ONLY (perm. =1): Lvds Data Enable
	4	0	MCM_Sync_Readout	READ ONLY (perm. =0): Start/Stop ASIC readout synchron.
	5-31		Unused	

HDMC Register Definition	Bit#	Default Value (dec.)	Name	Description
PPM.RO_Config : WRITE	0-2 6-31	1	b000 : RO_mode_31 b001 ; RO_mode_51 b010 : RO_mode_71 b011 ; RO_mode_93 b100 : RO_mode_115 b101 : RO_mode_151 Unused	3 FADC plus 1 LUT/BCID slice to read out 5 FADC plus 1 LUT/BCID slice to read out (default) 7 FADC plus 1 LUT/BCID slice to read out 9 FADC plus 3 LUT/BCID slice to read out 11 FADC plus 5 LUT/BCID slice to read out 15 FADC plus 1 LUT/BCID slice to read out
PPM.RO_Config : READ	0-2 3 4-6 7 8-11 12-31		CfgNumRoSamples Unused NumBcid Unused NumFadc Unused	readback; see WRITE Number of LUT/BCID samples set by ReM-firmware Number of FADC samples set by ReM-firmware
PPM.GLink	0-3 4-31		GapLength Unused	Default=8; extra #ticks to extend DAV Gap (i.e. time needed for ReM to send next RO block)
PPM.ROBStat	0-8 9-11 12 13 14 15 16-31		SIFROB_Content Unused SIFROB_Overflow SIFROB_Underflow SIFROB_Full SIFROB_Empty Unused	
ASIC Ser. Interface Global Registers				
PPM.SIF0	0 1 2-31	0 0	ReadMainEnable ReadoutEnable Unused	Global ASIC: Main enable for ASIC readout Global ASIC:
PPM.SIF1	0-1 2-3 4-31	0 0	ReadoutSyncRaw ReadoutSyncBcid Unused	Global ASIC: Enable external start of FADC data readout Global ASIC: Enable external start of LUT data readout
PPM.SIF2	0-4 5-31	3	GenOut Unused	Global ASIC: General Outputs (LVDS-Sync etc.)
PPM.SIF3	0-1 2 3 4-31	2 0 0	JetSumMode BypassBcMux ChannelSelect Unused	Global ASIC: Jet-cell Summing Mode Global ASIC: Control for bypassing BC-Multiplexing Global ASIC: Select output channel in BCMux bypass mode
PPM.SIF4	0 1-31	0	DaisyChainEnable Unused	Global ASIC: Enable internal daisy-chaining for pair of SerIFs
Non-ASIC Channel Registers				
PPM.DAC	0-7 8-15 16-31		threshold offset Unused	threshold for ext.BCID Comparator DC-Level for AnalogPulse Offset --
PPM.PHOS4	0-7 8-31		delay Unused	Phos4 delay setting: 1 to 24 nsec
ASIC Channel Registers				
PPM.PBM	0-15 16-31			cell_0 (16 bit, only 11 bit data) cell_1 : ... repeat (127*32-bit word) : cell_254 cell_255
PPM.LUT	0-7 8-15 16-23 24-31			cell_0 (8 bit) cell_1 cell_3 cell_4 : ... repeat (255*32-bit word) : cell_1020 cell_1021 cell_1022

HDMC Register Definition	Bit#	Default Value (dec.)	Name	Description
				cell_1023
PPM.CR0	0 1 2 3 4 5 6-31	1 1 0 1 0 1	InBcidNegedge InDataNegegde ReadoutSource BypassLut InvMsbDisable ExtBcidEdgeEnable Unused	Latch external BCID-bit with negative clock-edge Latch FADC data (10 bit) with negative clock-edge Source of readout data Bypass the LUT Disable inversion of MSB for FADC data Enable Edge Detection for ext. BCID (i.e. shape to 1 tick)
PPM.CR1	0-6 7-31	10	PipeDelayBcid Unused	Delay in pipeline memory of LUT result data
PPM.CR2	0-6 7-31	10	PipeDelayRaw Unused	Delay in pipeline memory of raw FADC readout data
PPM.CR3	0-3 4-7 8 9 10-31	0 0 0 0	SyncDelayBcid SyncDelayData SyncBypassBcid SyncBypassData Unused	Input delay of external BCID bit (FIFO depth) Input delay of FADC data (FIFO depth) Bypass the synchronisation FIFO for ext. BCID signal Bypass the synchronisation FIFO for FADC data
PPM.CR4	0-3 4-7 8-31	0 0	FIRCoeff1 FIRCoeff2 Unused	FIR filter coefficient#1 FIR filter coefficient#2
PPM.CR5	0-3 4-31	1	FIRCoeff3 Unused	FIR filter coefficient#3
PPM.CR6	0-3 4-7 8-31	0 0	FIRCoeff4 FIRCoeff5 Unused	FIR filter coefficient#4 FIR filter coefficient#5
PPM.CR7	0-9 10-31	767	SatHigh Unused	Upper threshold for BCID of saturated pulses
PPM.CR8	0-9 10-31	255	SatLow Unused	Lower threshold for BCID of saturated pulses
PPM.CR9	0-9 10-31	1023	SatLevel Unused	Energy level for marking as saturation
PPM.CR10	0-9 10-31	511	EnergylevelLow Unused	Lower energy threshold (window) for BCID decision logic
PPM.CR11	0-9 10-31	895	EnergylevelHigh Unused	Upper energy threshold (window) for BCID decision logic
PPM.CR12	0-7 8 9-31	254 0	BcidDecision1 SatOverride1 Unused	BCID decision logic LUT for high energy region (0xFE) BCID result override flag for high energy region
PPM.CR13	0-7 8 9-31	250 0	BcidDecision2 SatOverride2 Unused	BCID decision logic LUT for middle energy region (0xFA) BCID result override flag for middle energy region
PPM.CR14	0-7 8 9-31	240 0	BcidDecision3 SatOverride3 Unused	BCID decision logic LUT for low energy region (0xF0) BCUD result override flag for low energy region
PPM.CR15	0-2 3 4-6 7 8-9 10-31	0 0 6 0 2	StartBit PeakFinderCond DelayExtBcid DecisionSource DelaySatBcid Unused	Start bit, from where the FIR filter result is clipped Condition for backward/forward comparison in peak finder Delay of external BCID-bit before decision logic Decision source in BCID decision logic (FADC / FIRresult) Delay of saturated BCID result before decision logic
PPM.CR16	0-6 7-9 10-31	3 1	NumBcRaw NumBcBcid Unused	Number of raw FADC data time-slices to be read out Number of LUT result time-slices to be read out
PPM.CR17	0 1 2 3 4-5 6-31	0 0 0 0 1	RateEnable RateSource HisEnable HisSource HisOpMode Unused	Enable Rate-Metering Select Source (FADC / LUT) for Rate-Metering Enable Histogramming (when playback disabled) Select Source (FADC / LUT) for Histogramming Histogramming operation mode
PPM.CR18	0-9 10-31	32	RateEtThresh Unused	Signal threshold for Rate-Metering
PPM.CR19	0-7 8-31	20	RateDelTimeL Unused	Rate calculation time span (lower byte)
PPM.CR20	0-7 8-31	5	RateDelTimeH Unused	Rate calculation time span (upper byte)
PPM.CR21	0-7 8-31	32	HisEtThresh Unused	Signal threshold for Histogramming
PPM.CR22	0-5 6-31	1	HisLowerBcL Unused	Lower bunch number into Histogramming (lower 6 bits)
PPM.CR23	0-5 6-31	0	HisLowerBcH Unused	Lower bunch number into Histogramming (upper 6 bits)

HDMC Register Definition	Bit#	Default Value (dec.)	Name	Description
PPM.CR24	0-5 6-31	37	HisUpperBcL Unused	Upper bunch number into Histogramming (lower 6 bits)
PPM.CR25	0-5 6-31	53	HisUpperBcH Unused	Upper bunch number into Histogramming (upper 6 bits)
PPM.CR26	0-9 10-31	0	LutPedestal Unused	Pedestal of ramp pre-loaded into LUT
PPM.CR27	0-10 11-31	256	LutSlope Unused	Slope of ramp pre-loaded into LUT
PPM.CR28	0 1 2 3-31	0 0 0	PlaybackEnable PlaybackSync PlaybackOneshot Unused	Enable playback mode Enable synchronisation of playback by external input Enable one-shot mode of playback (inject memory-content once only)
PPM.CR29	0-7 8-31	0	PlaybackDelayH Unused	Playback delay for multi-shot mode (upper byte)
PPM.CR30	0-7 8-31	0	PlaybackDelayL Unused	Playback delay for multi-shot mode (lower byte)
PPM.CR31	0-6 7-31	16	AlmostFullRaw Unused	'Almost full' mark in raw FADC-data derandomizer
PPM.CR32	0-6 7-31	16	AlmostFullBcid Unused	'Almost full' mark in LUT-result derandomizer
PPM.CR33	0-7 8-31	255	SaturationValue Unused	Value inserted as LUT-result, when in 'override' mode
PPM Protocol Registers on TTCrx				
PPM.TTCDelay	0-7 8-31		Delay Unused	see: TTCrx Manual
PPM.TTCControl	0-7 8-31		Control Unused	see: TTCrx Manual
PPM.TTCDummy	0-31		any data	real dummy; register is NOT accessible
PPM.TTCErrCnt	0-7 8-31		Error Count Unused	TTC-driven Counter; 'write' generally means 'reset'
PPM.TTCID1	0-7 8-31		ID Unused	see: TTCrx Manual
PPM.TTCID2	0-5 6-7 8-31		ID MasterModeA Unused	see: TTCrx Manual see: TTCrx Manual
PPM.TTCID3	0-5 6-7 8-31		I2C_ID MasterModeB Unused	see: TTCrx Manual see: TTCrx Manual
PPM.TTCConfig	0-7 8-31		Config Unused	see: TTCrx Manual
PPM.TTCStatus	0-7 8-31		Status Unused	see: TTCrx Manual
PPM.TTCBC	0-7 8-31		Bunch Counter Bits Unused	TTC-driven Counter; 'write' generally means 'reset'
PPM.TTCEC	0-7 8-31		Event Counter Bits Unused	TTC-driven Counter; 'write' generally means 'reset'

Table 10 : Definitions of Registers used in VME-access.

4.4 Operation of the Pre-Processor.

The following section gives a summary of PPM-internal actions, which are all initiated via VME. An action as such is performed by firmware residing in the ReM_FPGA [see **Ref. 13**]. It should be noted, that the on-board S-RAM plays a central role as interface between local components on the PPM and controlling software.

- The S-RAM is on one hand the store for all parameters downloaded via VME --- generally called 'Reference'.
- The S-RAM is on the other hand also the store for parameters read-back from the hardware locations --- generally called 'Readback', e.g. a register in the PPrASIC.

This method is an economic way to handle the parameter exchange with the PPM, because there are many different destinations combined with different bus-protocols required to 'poke/peek' data on the PPM-board. This multitude of tasks is left to the ReM-firmware, hence screened from the controlling software in a CPU. The instructions to move data are implemented using 'top-level ' VME registers called 'REM_Command/ Control/ Status'.

4.4.1 Setting-up the PPM.

The ReM_FPGA is loaded with firmware either from VME or from the on-board Flash-RAM. Either is initiated by controlling software setting the VME configuration-register appropriately. Only then, the module becomes 'fully accessible' for set-up.

The actual setting-up phase of the PPM consists of transferring configuration data to the on-board S-RAM. The data are written via VME directly to the ReM_FPGA, which places a 'Reference' of the data-set in the S-RAM. VME instructs the ReM_firmware to fetch the data from S-RAM and put them under the appropriate bus-protocol to the on-board destinations.

4.4.2 Verification of settings in the Pre-Processor.

The local CPU can access **settings** for **reading only while DAQ** is going on. Online-software must ensure this by checking the corresponding 'control /status bits'.

Verification of parameters proceeds by instructing ReM-firmware to collect those data back from the on-board locations (see also 4.3.2 'VME addressing'). They are stored in the S-RAM 'Read-back' section in the same format as the 'Reference'. The procedure can take some time, because many ports (e.g. 32 serial interfaces to PPrASICs) are involved. Furthermore, some readback data have to be filtered out from serial data streams. The ReM-firmware signals when the requested 'readback' is completed and changes the S-RAM flag-bits accordingly (reflected in the VME readable registers).

The I^2C-bus to the TTC-Decoder is bi-directional, therefore readable. However, locations on SPI-bus as well as the Phos4-I^2C-bus cannot be read back. Hence, the ReM-firmware has placed a direct copy of the 'Reference' into the 'Readback' area for identical block-format. The crate-CPU can fetch the data-blocks for direct comparison, i.e. verification of hardware settings.

Outside DAQ mode, the local CPU has full freedom to execute 'read / write / read-back' cycles to any data location in the Pre-Processor system. PPrASIC readback requires local generation of pseudo 'L1-Accepts', which is possible on each PP-module. The data can be analysed directly via VME.

4.4.3 Data-Taking at the Pre-Processor.

The data-taking mode of operation requires absolute priority for readout data to be sent to the ROD (and DAQ). Write-access to any registers or memories of on-board components is blocked (see above).

The ReM_FPGA assembles readout data-records and transmits them in parallel to the RGTM (see below: Event Readout to the ROD). Composition of data is defined by the setting of appropriate channel-registers on the PPrASICs. Numbers of time-slices (FADC samples, LUT-BCID-results) could be set for each channel individually – only the ROD would have to cope with different data-volumes from different channels. This is a possible, but unlikely scenario for DAQ.

The relevant tasks in the ATLAS on-line 'run-control' shall define the readout configuration uniformly across the PPModule (or even across the entire Pre-Processor system) for DAQ running.

4.4.4 Monitoring at the Pre-Processor.

In DAQ mode, there are different levels of monitoring determined by the actual needs in the experiment.

After an LHC-fill there must be 'Reading back' of PPr-specific monitoring data (rates in calorimeter towers, unbiased FADC-histograms of energy in calorimeter towers). The acquisition of those data is done by the ReM-firmware. Results are stored in the S-RAM. Control over acquisition and availability of data is given by means of VME register-bits, e.g. the PPM signals by 'REM_Status bit', that data are taken (rate counting-interval completed, FADC-histogram filled).

The scheme of 'Reference' and 'Readback' sections in S-RAM is also implemented for monitoring with a slight difference of purpose. The 'Reference' shall reflect 'good conditions' to which the 'Readback' is compared. Deviations signal problems, e.g. excessive noise, beam background etc.

Optionally, the ReM_FPGA produces copies of Pre-Processor readout data, which are of interest to the system locally, if DAQ-priority allows to do so. These 'slice-data' (FADC-samples, LUT-BCID-result) are stored to S-RAM. The copy mechanism can be steered by the 'trigger type', i.e. 'selective readout' could be implemented in firmware. A 'fill-up mechanism' communicates data-availability via VME to the analysing crate-CPU performing 'local monitoring'.

Outside DAQ mode, there has to be a possibility (stand-alone taks) to operate the Pre-Processor acquiring rates and histograms – at least. Those data are needed with frequent update-intervals to optimise beam conditions in

ATLAS.

4.4.5 Serial Programming Interface to DACs.

The SPI serves exclusively for transfer of set-up data to the 'front-end' DACs (analog signal offset, discriminator threshold for 'external BCID) on the AnIn daughterboards. The VME register formats are defined above. Read-back from those devices is not possible. A 'real' verification of DAC settings can only be achieved by

- digitisation of the DAC-level in the FADC and reading those data.
- scanning across the threshold of the discriminator and reading the output-bit.

4.4.6 I^2C Interfaces to Phos4 and TTCdec.

VME data for the I^2C Interfaces to the Phos4-chips as well as the TTCdecoder are arranged as shown in the table below: Bit#16-17 identify the I^2C port (Phos4 bus, TTCdec bus); Bit#08-15 specify the I^2C address on that bus along with the data-byte to be sent in Bit#00-07.

For addressing the Phos4 timer-chips, the valid address in Bit#08-15 is limited to a 5 bit sub-mask with other bits required at fixed values. The I^2C bus to the Phos4s is 'write only', because those chips do not provide data for read-back. Furthermore, the bus-implementation across the large board requires special shaping of the serial clock, which prevents reverse flow of data.

31	18	16	8	0	**VME data word (Bits)**
X	XXX XXXX XXXX XX	I^2C Port	I^2C Address	Data	**Content**

Table 11 : Device addressing on the I^2C busses (Phos4, TTCdec).

4.4.7 The Serial Interfaces to the PPrASIC.

The 32 Serial Interfaces to channel-pairs on the PPrASICs are implemented as ports in the Data-I/O firmware. The input/output is based on 13-bit words, which are used for configuration-write, read-back and readout of event-data . A detailed description of 'command-exchange' for the required actions is given in a separate document [see **Ref.13**].

Two types of words can be sent to the PPrASICs: ***'Command words'***, which toggle action on the chip and ***'Data words'***, which contain data to be written to memories and/or registers.

All configuration data stored in registers and all memory contents can be read back through the serial interfaces of the PPrASIC. These read-back data are interleaved with 'event' data in a fixed way: One word of read-back data is transmitted along with every 'event' readout following a 'L1-Accept'. After removing the event data from the stream only the read-back data remain for local use.

There are three types of read-back words:

status words, sent when no specific read-back is requested,

header words, which start a block of read-back data and

data words containing the actual data retrieved by the 'read-back'.

A detailed description of the PPrASIC can be found in 'User Manual of the PPrASIC' [**see Ref. 10**]. The format of these words is defined there.

4.5 Event Readout via RGTM to the ROD (PPr).

The task of highest priority in the Readout Merger is the collection of event-related data. Each serial interface sends data from a pair of channels. For the sake of simplicity the format, defined in the 'PPrASIC User Manual '[see Ref. 10], is repeated here. The data-fields have a width of 13 bits. The example below shows the case of 3 FADC samples plus 1 LUT_BCID result per event for DAQ. This set is the minimal requirement for verification of BCID.

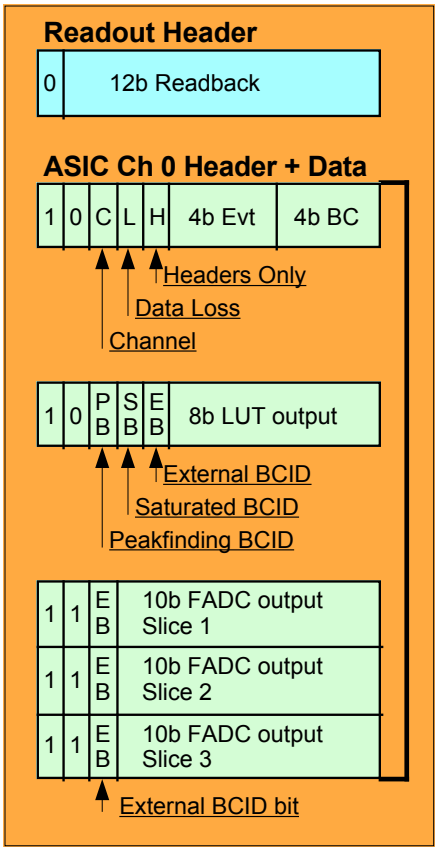

Figure 24 : Serial Data stream from ONE ASIC-channel.

The Readout Merger strips off the data in the 'Read-back Channel' for each of the 32 channel-pairs. Note, that the 'default' set in the PPrASIC is 5 FADC-slices and 1 LUT-BCID-slice. The number of readout slices per event can vary from 0 to 127 for the FADC samples and from 0 to 7 for the LUT-BCID results.

4.5.1 Definition of the G-Link Data Format.

A 'natural channeling' is given by the presence of 16 MCMs, whose readout results are mapped serially onto 16 lines to one G-Link transmitter in the slot. The one-to-one correspondence allows straight-forward transmission of data from pairs of serial interfaces on each PPrASIC. The tasks left for the ReM_FPGA are the following two:

- strip-off 'read-back' data for local monitoring in a PPr CPU.
- sequence the 4 ASIC-channels ('A' to 'D', i.e. the two Ser. Interfaces per MCM) onto the G-Link bit-stream.

Important for the ATLAS performance is the maximum 'dead-time free' speed of readout over the ROD-ROS path. Assuming a ROD (with16 G-Link inputs from 16 PPMs in a crate) can receive all data sent, the Rem_FPGA / G-Link assembly determines the readout speed. The data volume on the readout in turn is determined by the number of FADC-samples and the number of LUT-BCID results to be recorded around the time-slice of the collision.

Three FADC samples around the 'Bunch Crossing IDentified' time-slice are the minimal requirement (see above). A '3+1' readout scheme requires 188 bit /'L1-Accept'to be sent to the G-Link at 40 MHz. Thus, the readout time is 188 bit * 25 nsec = 4.7 μsec.

Five FADC samples cover the analog input signal to make a reasonably accurate energy-measurement for the setting of thresholds in the trigger logic downstream. A '5+1' readout scheme requires 276 bit /'L1-Accept'to be sent to the G-Link at 40 MHz giving a readout time of 276 bit * 25 nsec = 6.9 μsec.

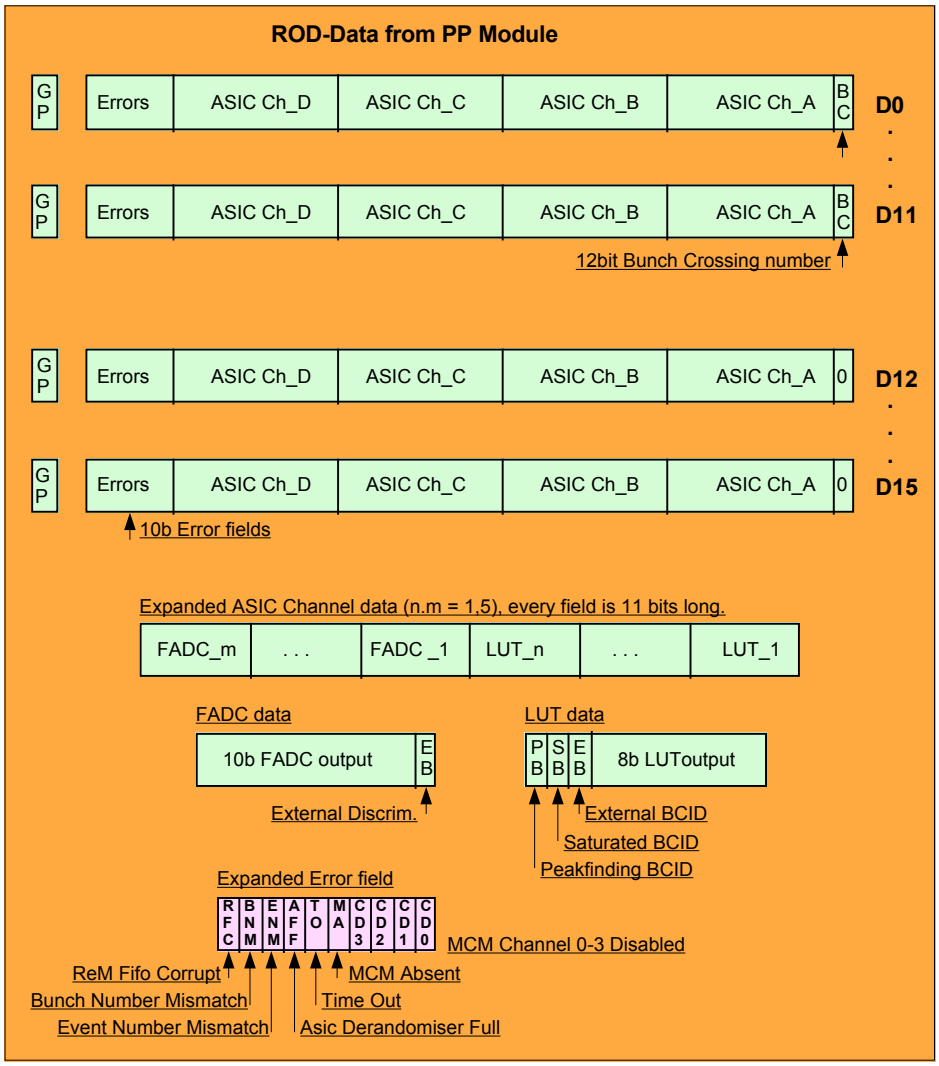

Figure 25 : G-Link data format.

ATLAS requires to read out at a maximum rate of 100 kHz, giving a 10 μsec time-interval for data-transfer. Even a '5+3' readout scheme requires only 364 bit /'L1-Accept' to be sent across each G-Link line. Thus, the readout time is 364 bit * 25 nsec = 9.1 μsec, which just about matches up with the limitation. In general, the length of a G-Link frame is: [4*(11*(n+m)) +11 + 1] clock-ticks, where n, m are the number of time slices.

The table below explains the data-fields.

Data	Field	Width (bits)	Content
BC	BC	1	Bunch Crossing Nunber: 12 bits contained in D0-D11; to be checked in ROD against BC Number from TTC for L1A
LUT	LUT Output	8	calibrated Et value from LUT for this clock cycle
	EB	1	External BCID was active for this clock cycle
	SB	1	Saturated BCID was active for this clock cycle
	PB	1	Peak-finding BCID was active for this clock cycle
FADC	FADC Output	10	Digitised Output from FADC for this clock cycle
	EB	1	Ext. BCID Discriminator output for this clock cycle
Errors	CD0-3	1	MCM Channel 0-3 is Disabled in software. How? setting all LUT output to 'zero' ?
	MA	1	This particular MCM is absent from PPM. Unlikely, because PPMs are always exchanged as a 'WHOLE'.

	TO	1	Time-Out: One or more channels did not produce data in response to L1A.
	AFF	1	L1As are too frequent. A derandomising memory on the PPrASIC is full.
	ENM	1	Event Number Mismatch between ASIC-counter and ReM-FPGA reference.
	BNM	1	Bunch Number Mismatch between ASIC-counter and ReM-FPGA reference.
	RFC	1	Rem_FPGA Readout -FIFO corrupt.
GP	GP	1	Longitud. Parity for bit-stream on G-Link

Table 12 : Data-Fields on G-Link stream.

No other 'protocoling' is needed for the data-streams. The G-Link bandwidth is sufficiently big to push data in a 'systolic' manner to the ROD. If RODs get too 'busy', the ATLAS DAQ slows down the rate of 'L1-Accept'. A PPM G-Link is identified at the ROD by a single optical input fiber.

5. Design, Manufacturing and Testing of the Pre-Series/Production Series.

Development and prototype work is done. Four fully functional 'prototype' PPMs (Version PPM_1.0) were used for testing with other Level-1 system parts as well as for commisioning work in ATLAS-USA15. The work was mainly concerned with check-out of signals from the calorimetry installed at the final positions inside the ATLAS detector.

The assembly of sub-devices (AnIn, LCD) and 'prototype' PPMs constituted a trial for relating to suitable industry, where PCB manufacturing, component-loading and soldering for the large number of modules was done.

Preparatory work on schematics and PCB layout for the FINAL PPModule-version has been progressing up to Dec.2005. The design of the printed-circuit boards was done at our institute using the 'Cadence Design Suite'. Manufacturing of the PCB was outsourced to a company known to have the required technology for PCBs of this size. Facilities for all steps of assembly-work are available in house, e.g. component placement and soldering for 'surface mount' up to 'fine-pitch', tooling for BGA placement as well as soldering/reworking. These assembly-steps on the prototypes were performed at KIP to have knowledge on technical issues involved before outsourcing.

The pre-series of PCB manufacturing delivered 20 boards (Version PPM_2.0). Some boards were fully assembled in house and tested. With satisfactory results, the other modules were outsouced to an assembly company. In parallel (or rather before), the assembly of the necessary plug-on devices in the required numbers was done as well (4*20 AnIn daughter-boards, 20 LCD daughterboards). The 16*20 MCMs were already available.

The larger number of modules also brought the advantage, that more data-sinks/-sources were available to carry out tests covering a large 'area' of trigger-space. The aspect was important for test-work done in conjunction with upstream calorimeters as well as downstream trigger processors.

The pre-series provided 16 modules required for a 'full crate' operational test in the laboratory - a pre-requisite for the 'Production Readiness Review (PRR)'. Some smaller technical modifications were identified. They were implemented as 'wire-modifications' on PPMs_2.0. These modules are a functional equivalent to the production version. Inclusion in the layout resulted in yet another PPM-version (PPM_2.1). The PRR was held in October 2006.

The test of fully assembled PPMs uses a 64-channel wide, analog signal source. VME access via the ReM_FPGA is a pre-requisite to set-up every PPM for operation on 'real-time' signals. Hence, the initial test environment comprises a local crate-controller (e.g. 'home-brew') using VME only. Capture of real-time data on LVDS-outputs from the PPM is achieved by a unit, which receives the full quantity of cables. But, recording of all LVDS channels in parallel at full speed (40 MHz) is not possible in the laboratory. Rather, a group of four channels is selected and routed to four LVDS deserialisers. The data of those 4 channels@40 MHz are captured in a fast memory for a limited duration. The memory is big enough, however, to allow checks on data-integrity. PPMs passing this test-bench go to a 'full crate test' in the test-rig of the KIP-ATLAS laboratory.

Test-Results for all components are stored in a dedicated SQL-database, which is accessible at:
http://www.kip.uni-heidelberg.de/atlas/db/DbPPr/welcome.html

The database holds tables showing:

- the installation status of the system in ATLAS-USA15.
- the status (and test results) of PPM motherboards fully equipped with daughterboards.

- the status (and test results) of AnIn daughterboards.
- the status (and test results) of Multi-Chip-Modules.
- the status (and test results) of LVDS Cable Driver daughterboards.
- other peripheral modules (TTCdec, CAN, RGTM).

One of the commercial VME crates (the Pre-Processor test-rig) is furnished with an additional 'power-brick' providing +48Volt on the backplane. Thus, the crate can house and supply a 9U ROD along with 16 PPMs allowing full checks of the DAQ output. The DAQ readout via G-Link to a ROD will be installed, when hardware and appropriate software are available.

6. Summary

This document outlines the specification for the Pre-Processor Module, which was intended to be a full-channel and full-function module. It has been shown, that the functionality was reached,. Reviewing and testing of sub-components as well as the full assembly has been successfully completed. Also, Level-1 system parts 'upstream' and 'downstream' were subject to similar procedures. Thus, the design fulfills its tasks as part of the ATLAS Level-1 Calorimeter Trigger System.

Project management details are omitted as the delivery of a working Pre-Processor system to the ATLAS experiment is the responsibility of the ATLAS group at KIP Heidelberg.

Appendices

A. Front-panel connectors.

The front-panel connector of the PPM is of the same type as used in the calorimetry for analog signal transport. Its pinning is shown in the figure below together with the additional pins connected to ground-shields.

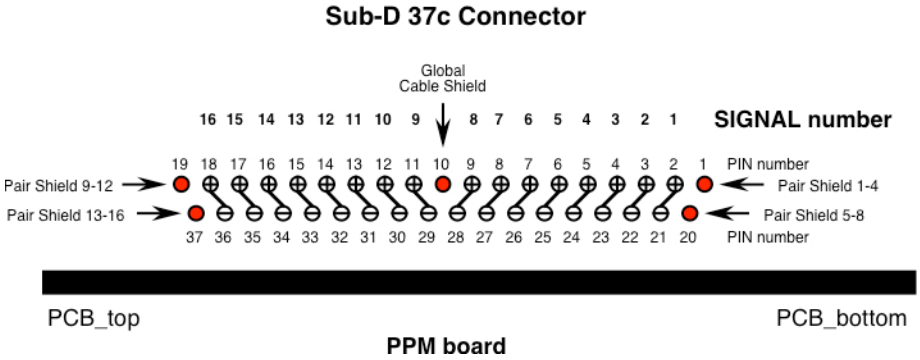

Figure 26 : Pinning of the PPM front-panel connector from LAr calorimeters.

The pseudo-connections drawn from '+' pins to '-' pins indicate the partnership of differential pairs, each of which form an analog calorimeter signal

B. AnIn connectors.

- Two of the connectors (P1, P3) take differential input-signals arriving from the front-panel (see also Figure 8).
- The other connectors (P2, P4) carry two kinds of outputs:
  - uni-polar, analog signals (A*), which are taken to a FADC on a MCM.
  - a 'digital' TTL-level (D*), which is taken as 11th bit directly to the input-FIFO of the PPrASIC marking 'external BCID'.

The surrounding connectivity shows, that care has been taken to screen the 'analog domain' from the 'digital domain'. However, the PCB-layout together with the absence of cross-talk of any kind is the real proof for good shielding.

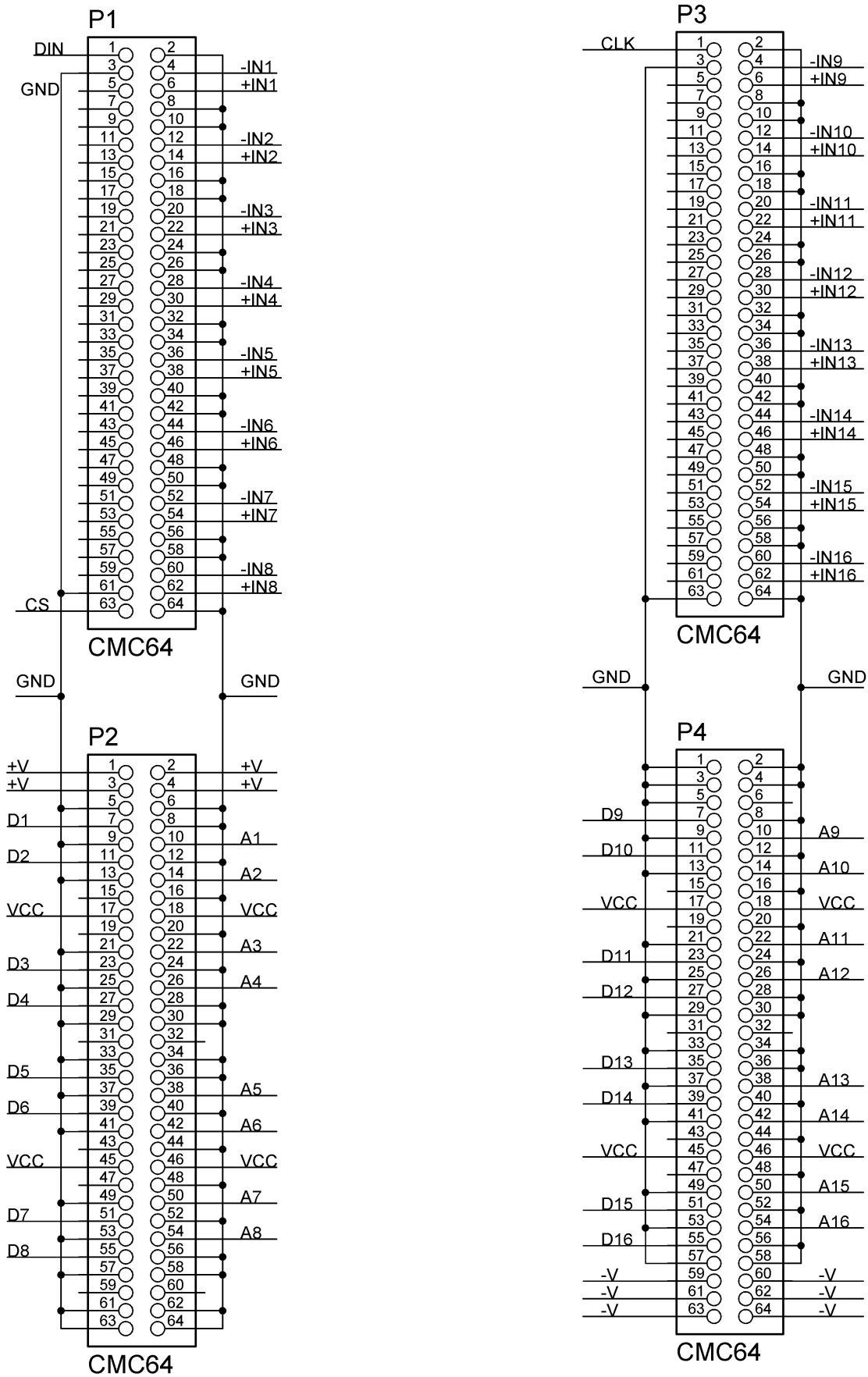

Figure 27 : Signals in/out of the AnIn daughterboard.

C. MCM connectors.

Socket M1:

PIN	SIGNAL-NAME	DESCRIPTION	VALUE
1	DVCC	Power Supply for digital Logic	+3.3 V
2	DVCC	Power Supply for digital Logic	+3.3 V
3	DVCC	Power Supply for digital Logic	+3.3 V
4	DVCC	Power Supply for digital Logic	+3.3 V
5	AVDD	Power Supply for analog Logic	+5 V :FADC
6	AVDD	Power Supply for analog Logic	+5 V :FADC
7	AVDD	Power Supply for analog Logic	+5 V :FADC
8	AVDD	Power Supply for analog Logic	+5 V :FADC
9	AGND	Ground for analog Logic	0 V :FADC
10	AGND	Ground for analog Logic	0 V :FADC
11	AGND	Ground for analog Logic	0 V :FADC
12	AGND	Ground for analog Logic	0 V :FADC
13	EXTBCID0	External BCID input Channel 1	Digital IN
14	ADC1_IN	Analog Input to ADC1	Analog IN
15	DGND	Ground for digital Logic	0 V
16	AGND	Ground for analog Logic	0 V :FADC
17	PHOS4_ADD5	I^2C address Phos4	Control IN
18	PHOS4_ADD2	I^2C address Phos4	Control IN
19	JTAGTDI	JTAG test Data input	Test IN
20	PHOS4_ADD3	I^2C address Phos4	Control IN
21	JTAGTDO	JTAG test Data output	Test OUT
22	PHOS4_ADD4	I^2C address Phos4	Control IN
23	DGND	Ground for digital Logic	0 V
24	AGND	Ground for analog Logic	0 V :FADC
25	EXTBCID2	External BCID input Channel 3	Digital IN
26	ADC3_IN	Analog Input to ADC3	Analog IN
27	DGND	Ground for digital Logic	0 V
28	AGND	Ground for analog Logic	0 V :FADC
29	ADC_C LK	LHC Bunch Crossing Clock	Digital IN
30	JTAGTRST	JTAG test Reset	Test IN
31	CLK_PHOS4	Crystal Oscillator for Phos4 (40 MHz)	Digital IN
32	SDA	I^2C data Phos4	Control IN
33	DGND	Ground for digital Logic	0 V
34	AGND	Ground for analog Logic	0 V :FADC
35	EXTBCID3	External BCID input Channel 4	Digital IN
36	ADC4_IN	Analog Input to ADC4	Analog IN
37	DGND	Ground for digital Logic	0 V
38	AGND	Ground for analog Logic	0 V :FADC
39	SCL	I^2C Clock	Control IN
40	JTAGTMS	JTAG test mode Select	Test IN
41	GENOUT1	Gen. Output bit from PPrASIC register	Control OUT
42	PHOS4ERROR	Phos4 Delay-Feedback halted (from PPrASIC counter) ??	Control OUT
43	TEMPMEASURE	Temperature Sensor on PPrASIC	an. Control OUT
44	JTAGTCK	JTAG test Clock	Test IN
45	DGND	Ground for digital Logic	0 V
46	AGND	Ground for analog Logic	0 V :FADC
47	EXTBCID1	External BCID input Channel 2	Digital IN
48	ADC2_IN	Analog Input to ADC2	Analog IN
49	AGND	Ground for analog Logic	0 V :FADC
50	AGND	Ground for analog Logic	0 V :FADC
51	AGND	Ground for analog Logic	0 V :FADC
52	AGND	Ground for analog Logic	0 V :FADC
53	AVDD	Power Supply for analog Logic	+5 V :FADC
54	AVDD	Power Supply for analog Logic	+5 V :FADC
55	AVDD	Power Supply for analog Logic	+5 V :FADC
56	AVDD	Power Supply for analog Logic	+5 V :FADC
57	DVCC	Power Supply for digital Logic	+3.3 V
58	DVCC	Power Supply for digital Logic	+3.3 V
59	DVCC	Power Supply for digital Logic	+3.3 V
60	DVCC	Power Supply for digital Logic	+3.3 V

Table 13 : The Input Connector of the PPrMCM (left).

Socket M2:

PIN	SIGNAL-NAME	DESCRIPTION	VALUE
1	DVCC	Power Supply for digital Logic	+3.3 V
2	DVCC	Power Supply for digital Logic	+3.3 V
3	DVCC	Power Supply for digital Logic	+3.3 V
4	DVCC	Power Supply for digital Logic	+3.3 V
5	DGND	Ground for digital Logic	0 V
6	LVDS_SYNC1	Send Synch. pattern on LVDS_1 for duration of LVDS_SYNC1	CMOS IN
7	LVDS1_DO	Serial Output from LVDS_1 (to CP)	LVDS Out
8	SEROUT1	Output of PPrASIC serial interface 1	CMOS OUT
9	LVDS1_DOBAR	Complement of LVDS_1_DO	LVDS OutBar
10	DGND	Ground for digital Logic	0 V
11	DGND	Ground for digital Logic	0 V
12	L1ACCEPT	Level 1 'Accept' signal	CMOS IN
13	LVDS1_DEN	Data Output Enable on LVDS_1 (level after power-up)	CMOS IN
14	PPRASICCLK	PPrASIC clock (LHC clock via TTC)	CMOS IN
15	AGND	Ground for analog Logic	0 V :LVDS
16	AGND	Ground for analog Logic	0 V :LVDS
17	GENOUT3	Gen. Output bit from PPrASIC register	Control OUT
18	LVDS1_TCLK	Transmit Clock for LVDS1	CMOS IN
19	AVCC	Power Supply for analog Logic: LVDS	+3.3 V :LVDS
20	AVCC	Power Supply for analog Logic: LVDS	+3.3 V :LVDS
21	SER IN2	Input of PPrASIC serial interface 2	CMOS IN
22	LVDS2_SYNC1	Send Synch. pattern on LVDS_2 for duration of LVDS_SYNC1	CMOS IN
23	SERFRAME	Frame select for PPrASIC serial interfaces	Control IN
24	LVDS1_TCK_R_F	Select clock-edge (rise/fall) for input to LVDS transmitter	Control IN (Lvl)
25	SERCLK	Clock for PPrASIC serial interfaces	CMOS IN
26	SYNCPLAYBACK	Start synchronous PlayBack from memories	CMOS IN
27	DGND	Ground for digital Logic	0 V
28	SYNCREADOUT	Start synchronous Memory readout	CMOS IN
29	LVDS2_DO	Serial Output from LVDS_2 (to CP)	LVDS Out
30	SEROUT2	Output of PPrASIC serial interface 2	CMOS OUT
31	LVDS2_DOBAR	Complement of LVDS_2_DO	LVDS OutBar
32	DGND	Ground for digital Logic	0 V
33	DGND	Ground for digital Logic	0 V
34	GENOUT4	Gen. Output bit from PPrASIC register	Control OUT
35	LVDS3_SYNC1	Send Synch. pattern on LVDS_3 for duration of LVDS_SYNC1	Control IN
36	LVDS2_DEN	Data Output Enable on LVDS_2 (level after power-up)	CMOS IN
37	EVCNTRES	Level 1 Event counter Reset	CMOS IN
38	SERIN1	Input of PPrASIC serial interface 1	CMOS IN
39	BCCNTRES	Bunch-Crossing counter Reset	CMOS IN
40	LVDS2_TCK_R_F	Select clock-edge (rise/fall) for input to LVDS transmitter	Control IN (Lvl)
41	GENOUT2	Gen. Output bit from PPrASIC register	Control OUT
42	LVDS2_TCLK	Transmit Clock for LVDS2	CMOS IN
43	AVCC	Power Supply for analog Logic: LVDS	+3.3 V :LVDS
44	AVCC	Power Supply for analog Logic: LVDS	+3.3 V :LVDS
45	RESETBAR	PPrASIC soft reset ??	Control IN
46	LVDS3_DEN	Data Output Enable on LVDS_3 (level after power-up)	CMOS IN
47	AGND	Ground for analog Logic	0 V :LVDS
48	AGND	Ground for analog Logic	0 V :LVDS
49	DGND	Ground for digital Logic	0 V
50	LVDS3_TCK_R_F	Select clock-edge (rise/fall) for input to LVDS transmitter	Control IN (Lvl)
51	LVDS3_DO	Serial Output from LVDS_3 (to JEP)	LVDS Out
52	DGND	Ground for digital Logic	0 V
53	LVDS3_DOBAR	Complement of LVDS_3_DO	LVDS OutBar
54	DGND	Ground for digital Logic	0 V
55	DGND	Ground for digital Logic	0 V
56	LVDS3_TCLK	Transmit Clock for LVDS3	CMOS IN
57	DVCC	Power Supply for digital Logic	+3.3 V
58	DVCC	Power Supply for digital Logic	+3.3 V
59	DVCC	Power Supply for digital Logic	+3.3 V
60	DVCC	Power Supply for digital Logic	+3.3 V

Table 14 : The Output Connector of the PPrMCM (right).

D. LVDS signals through the PPM backplane onto cable-connectors.

The following gives the connector pinning, to transport the real-time LVDS signals through the backplane and from there across cables to the respective trigger processors. The connector on the PCB and its associated counterpart are of the CompactPCI family housing five rows of pins. The connector height (#columns) is chosen to cover the needs, i.e. 22 rows for the CP-path and 25 rows for the JEP-path. The shaded areas in the following table

indicate the grouping of signals onto AMP ‘twin-pair’ cables for transfer to the respective trigger processors.

FP: Conn-PAIR No	MCM No. ProcOut	CPCI: ColRow	Signal		
		Row‘c’ = GND			
To Cluster-Processor					
	MCM1-CP_a_Fout	1a, 1b	LVDS (+, -)	W	
	MCM1-CP_b_Fout	1d, 1e	“		
	MCM2-CP_a_Fout	2a, 2b	“		
	MCM2-CP_b_Fout	2d, 2e	“		
C1-16, C1-13	MCM1-CP_a	3a, 3b	LVDS (+, -)	A	
C1-15, C1-14	MCM1-CP_b	3d, 3e	“		
C1-12, C1-9	MCM2-CP_a	4a, 4b	“		
C1-11, C1-10	MCM2-CP_b	4d, 4e	“		
C1-8, C1-5	MCM3-CP_a	6a, 6b	LVDS (+, -)	B	
C1-7, C1-6	MCM3-CP_b	6d, 6e	“		
C1-4, C1-1	MCM4-CP_a	5a, 5b	“		
C1-3, C1-2	MCM4-CP_b	5d, 5e	“		
C2-16, C2-13	MCM5-CP_a	7a, 7b	LVDS (+, -)	C	
C2-15, C2-14	MCM5-CP_b	7d, 7e	“		
C2-12, C2-9	MCM6-CP_a	8a, 8b	“		
C2-11, C2-10	MCM6-CP_b	8d, 8e	“		
C2-8, C2-5	MCM7-CP_a	10a, 10b	LVDS (+, -)	D	
C2-7, C2-6	MCM7-CP_b	10d, 10e	“		
C2-4, C2-1	MCM8-CP_a	9a, 9b	“		
C2-3, C2-2	MCM8-CP_b	9d, 9e	“		
C3-16, C3-13	MCM9-CP_a	11a, 11b	LVDS (+, -)	E	
C3-15, C3-14	MCM9-CP_b	11d, 11e	“		
C3-12, C3-9	MCM10-CP_a	12a, 12b	“		
C3-11, C3-10	MCM10-CP_b	12d, 12e	“		
C3-8, C3-5	MCM11-CP_a	14a, 14b	LVDS (+, -)	F	
C3-7, C3-6	MCM11-CP_b	14d, 14e	“		
C3-4, C3-1	MCM12-CP_a	13a, 13b	“		
C3-3, C3-2	MCM12-CP_b	13d, 13e	“		
C4-16, C4-13	MCM13-CP_a	15a, 15b	LVDS (+, -)	G	
C4-15, C4-14	MCM13-CP_b	15d, 15e	“		
C4-12, C4-9	MCM14-CP_a	16a, 16b	“		
C4-11, C4-10	MCM14-CP_b	16d, 16e	“		
C4-8, C4-5	MCM15-CP_a	18a, 18b	LVDS (+, -)	H	
C4-7, C4-6	MCM15-CP_b	18d, 18e	“		
C4-4, C4-1	MCM16-CP_a	17a, 17b	“		
C4-3, C4-2	MCM16-CP_b	17d, 17e	“		
	MCM15-CP_a_Fout	20a, 20b	LVDS (+, -)	V	
	MCM15-CP_b_Fout	20d, 20e	“		
	MCM16-CP_a_Fout	19a, 19b	“		
	MCM16-CP_b_Fout	19d, 19e	“		

FP: Conn-PAIR No	MCM No. ProcOut	CPCI: ColRow	Signal		
			Col.21: empty		
Connector: end			Col.22: empty		
To Jet/Energy-Processor					
			Col.1: empty		
	MCM16-JEP_Fout_1	2a, 2b	Spec.PPM_8a (+, -)	F1	
	MCM5_JEP_Fout	2d, 2e	Spec.PPM_8a (+, -)		
			3a, 3b: empty		
			3d, 3e: empty		
	MCM4-JEP_Fout_1	4a, 4b	Spec.PPM_8a (+, -)	F2	
	MCM9-JEP_Fout_1	4d, 4e	Spec.PPM_8a (+, -)		
			5a, 5b: empty		
			5d, 5e: empty		
			6a, 6b: empty	V8	
			6d, 6e: block@JEM		
	MCM8-JEP_Fout_1	7a, 7b	Spec.PPM_8 (+, -)		
			7d, 7e: block@JEM		
	MCM1-JEP_Fout_1	9a, 9b	LVDS (+, -)	WX, WX	
	MCM2-JEP_Fout	9d, 9e	"		
	MCM3-JEP_Fout	8d, 8e	"		
	MCM4-JEP_Fout_2	8a, 8b	"		
C1-(16+15+14+13)	MCM1-JEP	11a, 11b	LVDS (+, -)	AB, AD, AH	
C1-(12+11+10+9)	MCM2-JEP	11d, 11e	"		
C1-(8+7+6+5)	MCM3-JEP	10d, 10e	"		
C1-(4+3+2+1)	MCM4-JEP	10a, 10b	"		
C2-(16+15+14+13)	MCM5-JEP	13a, 13b	LVDS (+, -)	CD, EH, AH	
C2-(12+11+10+9)	MCM6-JEP	13d, 13e	"		
C2-(8+7+6+5)	MCM7-JEP	12d, 12e	"		
C2-(4+3+2+1)	MCM8-JEP	12a, 12b	"		
C3-(16+15+14+13)	MCM9-JEP	15a, 15b	LVDS (+, -)	EF, AD, AH	
C3-(12+11+10+9)	MCM10-JEP	15d, 15e	"		
C3-(8+7+6+5)	MCM11-JEP	14d, 14e	"		
C3-(4+3+2+1)	MCM12-JEP	14a, 14b	"		
C4-(16+15+14+13)	MCM13-JEP	17a, 17b	LVDS (+, -)	GH, EH, AH	
C4-(12+11+10+9)	MCM14-JEP	17d, 17e	"		
C4-(8+7+6+5)	MCM15-JEP	16d, 16e	"		
C4-(4+3+2+1)	MCM16-JEP	16a, 16b	"		
	MCM15-JEP_Fout	18a, 18b	LVDS (+, -)	VJ, VJ	
			18d,18e:block@JEM		
	MCM16-JEP_Fout_2	19a, 19b	LVDS (+, -)		
			19d,19e:block@JEM		
			20a, 20b: empty	W8	
			20d, 20e: empty		

FP: Conn-PAIR No	MCM No. ProcOut	CPCI: ColRow	Signal		
	MCM9-JEP_Fout_2	21a, 21b	Spec.PPM_8 (+, -)		
	MCM10-JEP_Fout	21d, 21e	Spec.PPM_8 (+, -)		
	MCM8-JEP_Fout_2	22a, 22b	Spec.PPM_8a (+, -)	F3	
	MCM13-JEP_Fout	22d, 22e	Spec.PPM_8a (+, -)		
			23a, 23b: empty		
			23d, 23e: empty		
	MCM12-JEP_Fout	24a, 24b	Spec.PPM_8a (+, -)	F4	
	MCM1-JEP_Fout_2	24d, 24e	Spec.PPM_8a (+, -)		
			25a, 25b: empty		
Connector: end			25d, 25e: empty		

Table 15 : LVDS signal allocation at PPM backplane; cables to Processors.

Special arrangements are necessary to accomodate the signal-fanout in the so-called 'forward region' (i.e. PPM_8 and PPM_8a covering part of the HEC and the entire FCAL). This is required for a 'jet-trigger' in the forward region. For fully efficient jet-finding algorithms, additional signals must be fanned-out at azimuthal boundaries in a few of the PPMs. Yet another document gives the correspondence between 'analog input' and 'digital LVDS output' **[Ref. 17]**, The center row of pins on the CompactPCI connector is dedicated to the 'Ground' potential. To ensure minimal noise induced onto these fast signal connections, the cable 'grounds' are grouped on the PPM according to their destination. A scheme, outlined in the figure below, is implemented allowing additional protection by optional insertion of appropriate passive components.

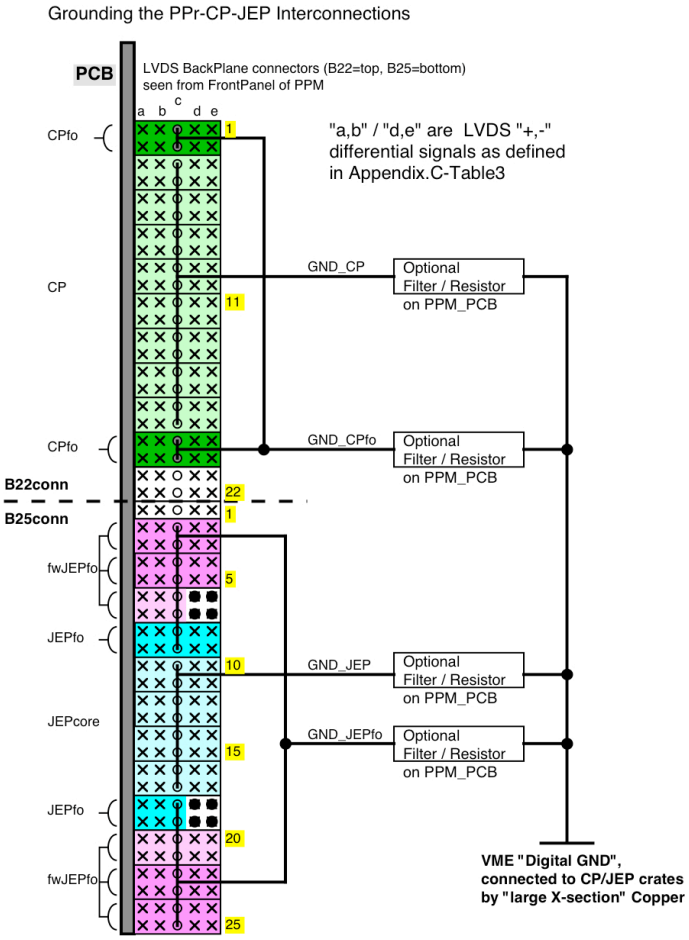

Figure 28 : Grounding scheme for the 'inter-Processor' connection.

E. Trigger data Readout across VME-J2 to DAQ (ROD).

The fast readout of Level-1 data is allocated to so-called 'user-defined' pins on the J2-Connector. It is part of the

standardised VME backplane. 'GND' and '+5V power' are taken from 'standard VME pins' located on 'Row Z and B' (shaded fields). The pin allocation, using only 'user-defined' pins in Rows C and D to the RGTM, is given in the table below.

VME64xP: J2 (extended)					
Col. No.	Row Z	Row A	Row B	Row C	Row D
01			+5V	LINKRDY	
02	GND		GND	DAV*	
03			Reserved	RESET*	Tx_Fault
04	GND		A24		
05			A25	DATA<19>	Laser_Dis*
06	GND		A26	DATA<18>	
07			A27	DATA<17>	
08	GND		A28	DATA<16>	
09			A29		
10	GND		A30	DATA<15>	
11			A31	DATA<14>	
12	GND		GND	DATA<13>	
13			+5V	DATA<12>	PPM+5V
14	GND		D16		PPM+5V
15			D17	DATA<11>	PPM+5V
16	GND		D18	DATA<10>	PPM+5V
17			D19	DATA<9>	
18	GND		D20	DATA<8>	
19			D21		
20	GND		D22	DATA<7>	
21			D23	DATA<6>	
22	GND		GND	DATA<5>	
23			D24	DATA<4>	
24	GND		D25		
25			D26	DATA<3.>	
26	GND		D27	DATA<2>	
27			D28	DATA<1>	
28	GND		D29	DATA<0>	
29			D30		
30	GND		D31		
31			GND	STRBIN40	
32	GND		+5V		

Table 16 : The pin allocation for the RGTM on the J2 Connector of VME64xP.

F. The Auxiliary Backplane across VME-J0: TTC, CAN, PPM-maintenance.

The distribution of the TTC signals is taken over by the TCM positioned in slot#21 of every Pre-Processor crate. The differential TTC signals **(TTCrx+, TTCrx-)** are carried by point-to-point routing on a auxiliary printed-circuit board across the backplane to the 16 PPMs in slot#5 to slot#20.

The TCM also plays the role of CAN-Master for the modules in the crate. Hence, the CAN-Bus connection **(CAN+, CAN-)** is realised on the same hardware.

The table below gives the pinning of the VME-J0 connector at a PPM-position. The pins used are 'user-definable' (double line-frame) on the standard VME64x backplane as well as on the standard VME64xP-VIPA backplane.

A small plug-on PCB called 'Service Module' allows field-connection for 'loading' purposes. It is basically a translator from the backplane 'compact PCI' to individual connections for computer periphery. The 'services' are concerned with:

- the VME-protocol device (CPLD_1)
- the Flash-RAM controlling device (CPLD_2)
- a JTAG-chain, useful to check some component-connectivity on the module.
- some LED-indicators (e.g TTC clock) for 'expert diagnosis'.

Col. No.	**Row f**	**Row e**	**Row d**	**Row c**	**Row b**	**Row a**	**Row z**
01		+5 V	+5 V	+5 V	+5 V	+5 V	
02				+5 V			
03							
04		CPLD_1_TMS	CPLD_2_TMS	JTAG_TMS			

Col. No.	Row f	Row e	Row d	Row c	Row b	Row a	Row z
05		CPLD_1_TCK	CPLD_2_TCK	JTAG_TCK			
06		CPLD_1_TDI	CPLD_2_TDI	JTAG_TDI			
07		CPLD_1_TDO	CPLD_2_TDO	JTAG_TDO			
08			GND	JTAG_TRST			
09		GND	TTCrx+	GND			
10		GND	TTCrx-	GND			
11			GND	CAN_Pwr_Off			
12		GAddr <8>	CAN_RxD	LED_CAN_Off			
13		GAddr <7>	CAN_TxD	LED_CAN+5V			
14		GAddr <6>	CAN_DTR	GND			
15		GAddr <5>	GND	SYSCLK			
16		CAN-	CAN+	GND			
17							
18							
19							

Table 17 : TTC distribution on VME64xP backplane (J0/P0).

References

[1] The Technical Design Report on the First-Level trigger, ATLAS Collaboration, June 1998
http://atlasinfo.cern.ch/Atlas/GROUPS/DAQTRIG/TDR/tdr.html

[2] Timing, Trigger and Control for LHC detectors.
http://ttc.web.cern.ch/TTC/intro.html
http://ttc.web.cern.ch/TTC/TTCrx_manual3.11.pdf

[3] The Pre-Processor system in the ATLAS Level-1 Calorimeter Trigger
(Overview, Architecture), P.Hanke, KIP Heidelberg, Dez.2005,
http://www.kip.uni-heidelberg.de/atlas/projects/construction/PP_System/PPSys_Wrup.pdf

[4] Level-1 Calorimeter Trigger: Cable Mappings and Crate Layouts from Analogue Inputs
to Processors, L1Calo group, Dec.05
http://hepwww.ph.qmul.ac.uk/l1calo/doc/out/CableMappings.pdf

[5] Module Specification: Timing and Control Module (TCM), N.Gee, Nov.2005
http://hepwww.rl.ac.uk/Atlas-L1/Modules/TCM/TCM_Draft_08_2.pdf

[6] Reference to the schematics of the Pre-Processor Module. KIP Heidelberg, K.Schmitt, 2005
http://www.kip.uni-heidelberg.de/atlas/projects/construction/PP_Module/PPM21_schematics.pdf

[7] • ANSI/VITA 1-1994: VME64 Specification
• ANSI/VITA 1.1-1997: VME64 Extension (VME64x)
• ANSI/VITA 1.3-1997: VME64x*9U*400mm Format (VME64x-9U)
• ANSI/VITA 23-1998: VME64 Extensions for Physics (VME64xP)
http://www.vita.com/vme64x.html
http://esone.web.cern.ch/ESONE/Syscomms98/

[8] PRR for Atlas Calorimeter Level-1 Trigger Cables (with specification)
P.Borgeaud, J.Pascual, LAr Calorimeter group, ATL-AL-ES-0009 in ATLAS-EDMS

[9] • A daughter-board with line-receivers on the Pre-Processor Module, P.Hanke,
http://www.kip.uni-heidelberg.de/atlas/projects/construction/PP_Module/AnIn_Spec_postPDR_Apr02.pdf

• Schematics of the 'AnIn' daughterboard for the PPM, D.Weiskat, KIP Heidelberg, 2004
http://www.kip.uni-heidelberg.de/atlas/projects/construction/PP_Module/AnIn20_schematics.pdf

[10] • Specification of the PPrASIC
http://www.kip.uni-heidelberg.de/atlas/projects/construction/PP_ASIC/PPrASICspec_Jul99.pdf

• User- and Reference Manual for the PPrASIC
http://www.kip.uni-heidelberg.de/atlas/projects/construction/PP_ASIC/pprasicman.pdf

• Prel. Design Review of the PPrMCM... (and specification)
http://www.kip.uni-heidelberg.de/atlas/projects/construction/PP_MCM/PPrMCM_Spec_Mar01.pdf

• Final Design Review of the PPrASIC.
http://hepwww.rl.ac.uk/Atlas-L1/Modules/PPr/PPrASIC_FDR.pdf

• Final Design Review for the PPrMCM
http://hepwww.rl.ac.uk/Atlas-L1/Modules/PPr/pprmcmfdrApr01.pdf

• Production Readiness Review for the PPr ASIC/MCM assembly, KIP Heidelberg, 2005,
ATC-RD-ER-0025 (id=573194) in ATLAS-EDMS

• The Pre-Processor 'Multi-Chip Module' (PPrMCM) assembly
in the ATLAS Level-1 Calorimeter Trigger Pre-Processor,
P.Hanke, ATLAS group at KIP Heidelberg, July 2005
http://www.kip.uni-heidelberg.de/atlas/projects/construction/PP_MCM/PPMCM_Wrup.pdf

[11] • Measurement of Bit Error Rate in LVDS Transmission test,
P.Meshkov, KIP Heidelberg, July 2003
http://www.kip.uni-heidelberg.de/atlas/projects/construction/PP_System/LVDS_Links/LVDS_BERtest_Jul03.pdf

• Schematics of the LCD daughterboard for the PPM (by K.Schmitt)
http://www.kip.uni-heidelberg.de/atlas/projects/construction/PP_Module/LCD20_schematics.pdf

[12] Project Specification: ATLAS Level-1 Cal. Trigger – TTC Decoder Card, W.Qian, Nov.05
http://hepwww.rl.ac.uk/Atlas-L1/Modules/TTC/TTCDec_Version1.1.pdf

[13] ReM_FPGA: firmware implementation.... (V.Andrei, K.Mahboubi et al. ...)
... ... document to come !

[14] Specification of the Rear G-Link Transmission Module (RGTM), P.Hanke, V.Perera
KIP Heidelberg, Rutherford Laboratory, 2003-05
http://www.te.rl.ac.uk/esdg/atlas-flt/specs/RGTM/RGTM-Optical/RGTM-Ov1.pdf

[15] Project Specification: ATLAS Level-1 Cal. Trigger Read-Out Driver. N.Gee, Jan.07
http://hepwww.rl.ac.uk/Atlas-L1/Modules/ROD/ROD-spec-version1_1.pdf

[16] • Introduction to CAN-Bus, ATLAS DCS group
http://atlasinfo.cern.ch/ATLAS/GROUPS/DAQTRIG/DCS/CANbus.html

• Schematics of the CAN-controller for the PPM, D.Weiskat, KIP Heidelberg, Nov.2005
http://www.kip.uni-heidelberg.de/atlas/projects/construction/PP_Module/CAN_MCTRL_Schem.pdf

[17] Channel Mapping on the PPM, K. Mahboubi, KIP Heidelberg, Feb.2006-02-10
http://www.kip.uni-heidelberg.de/atlas/projects/construction/PP_Module/ppm_channel_mapping.pdf

Contents

Figures and Tables

Glossary

- AnIn — PPM daughter-board for analog Input and external BCID
- ASIC — Application Specific Integrated Circuit
- BCID — Bunch Crossing IDentification in the LHC intersection region
- FCAL — Liquid Argon, Forward Calorimeter (e.m. and hadronic)
- CAN-Bus — System and data-bus for monitoring hardware operating conditions
- CP — Cluster Processor
- CPLD — Complex Programmable Logic Device
- CPM — Cluster Processor Module
- CPU — Central Processor Unit (here a 'standard' computer as crate module)
- CSR — Control and Status Register
- CTP — Central Trigger Processor
- DAC — Digital-to-Analog Converter
- DAQ — Data AcQuisition system (the ATLAS 'on-line' computing system)
- DCS — Detector Contol System of the ATLAS experiment
- DLL — Delay Lock Loop
- FADC — Flash Analog-to-Digital Converter
- FIFO — First IN - First Out
- FPGA — Field Programmable Gate Array
- G-Link — Gigabit Rate serial Transmit/Receive chip-set
- I^2C — … serial Bus
- JEP — Jet / Energy-Sum Processor
- JEM — Jet / Energy-Sum Module
- LAr_emEC — Liquid Argon, electromagnetic calorimeter EndCap
- LAr_emBar — Liquid Argon, electromagnetic calorimeter Barrel
- LAr_HEC — Liquid Argon, Hadronic calorimeter EndCap
- LCD — Lvds Cable Driver
- Level-1 — First Level Trigger in the ATLAS experiment: 'Level-1'
- L1 — abbreviation for 'Level-1' trigger
- LHC — Large Hadron Collider
- LVDS — Low Voltage Differential Signalling
- LUT — Look-Up Table
- PCB — Printed Circuit Board
- PPM — Pre-Processor Module
- PPr — abbreviation for 'Pre-Processor'
- PPrASIC — Pre-Processor Application Specific Integrated Circuit
- PPrMCM — Pre-Processor Multi-Chip Module (plugged onto PPM)
- RAM (SRAM) — (Static) Random Access Memory
- ReM_FPGA — Readout Merger Field Programmable Gate Array
- RGTM — Rear G-Link Transmission Module, electrical G-Link output
- RGTM-O — Rear G-Link Transmission Module, Optical G-Link output
- ROB — ReadOut Buffer (the ATLAS interface to the DAQ)
- ROD, PPROD — ReadOut Driver module, PreProcessor ROD
- ROS — ReadOut System (the NEW ATLAS DAQ system, replacing ROB)
- S-Link — 160 MBit/sec 'Simple Link' standard developed at CERN
- SPI — Serial Programming Interface
- TCM — Timing and Control Module (TTC distribution in crate, VME display)
- TileCal — scintillator Tile, hadronic Calorimeter (in the barrel region)
- TTC — Timing, Trigger and Control system
- TTCdec — TTC decoder (a PPM daughterboard)
- VME — Versa Module Euro 'Data Bus Standard'