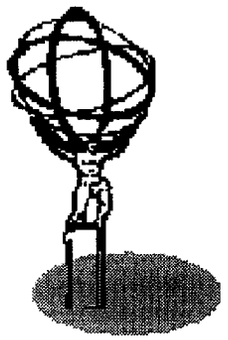
NIKHEF

NATIONAL INSTITUTE FOR NUCLEAR AND HIGH ENERGY PHYSICS



ATLAS Internal Note
MUON-NO-191
18 February 1998

ETR 96-09

NIMROD,
the ROD for the Monitored Drift Tubes in ATLAS

The NIKHEF MDT Read Out Driver collects data from the TDCs, mounted on the Monitored Drift Tubes. The data from several MDT chambers is merged and sent to the Read Out Buffer as complete events.

H.L.Groenstege
A.C.König
P.A.M.Rewiersma
T.A.M.Wijnen

h.groenstege@nikhef.nl

draft version 1.2

check for most recent version: http://www.nikhef.nl/pub/departments/et/atlas_mdt/nimrod/

NIKHEF, DEPARTMENT OF ELECTRONIC TECHNOLOGY

P.O.box 41882, NL 1009 DB AMSTERDAM

Table of contents

Figures

1. NIMROD

The NIMROD (NIKHEF MDT Read Out Driver) concentrates the data from a number of TDCs[1] (e.g. of a tower of muon chambers[2]) into a single output. The output drives the ROL (Read Out Link). The unit receives trigger and timing information from the CTP (Central Trigger Processor) and distributes this to the TDCs. The information is received by means of a TTCrx chip (Trigger Timing and Control receiver chip). The NIMROD is a VME module. The size of the modules probably will be conform the IEEE standard P1101.10, which describes a 9U * 400 mm board. It is foreseen that four NIMRODs may share a VME crate, controlled by a single master. This controller takes care of various settings in the NIMRODs, internal tests, spying on the data stream and possibly sending test data to ROB. The ROB (Read Out Buffer) is the interface to the ATLAS Data Acquisition System and the second level trigger. Each of the 96 towers in the barrel region has its own NIMROD, connected to 96 ROLs (Read Out Links). For the Forward MDT system the read out segmentation is still under study.

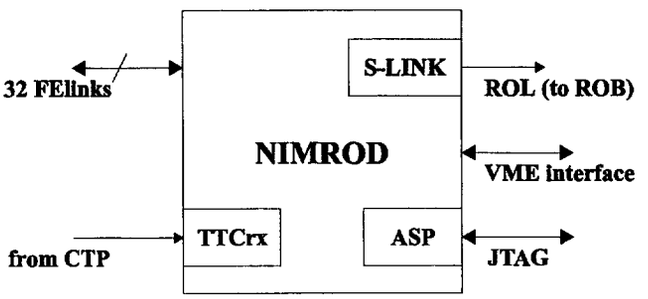


Figure 1: NIMROD connections

1.1 NIMROD connections

Each NIMROD accepts a maximum of 32 FELinks (Front-End Links). The average occupancy should not exceed ~40 %, for reasons to be explained later. The information from the TTC (Trigger Timing and Control) is received by a TTCrx mezzanine card. Each NIMROD has its own ROL (Read Out Link), which cannot be shared. The ROL is driven by an S-LINK mezzanine card. The VME interface is used for various settings at run-time, as a spy channel and to invoke clock phase calibration cycles (see page 9). The VME interface may also be used to insert test data that is sent to the ROB. The JTAG port is used to perform functional tests of NIMROD only and to set-up the TTCrx board. More about JTAG on page 6.

[1] A 24 channel TDC for the ATLAS precision muon chambers, Y. Arai (KEK) J. Christiansen (CERN).
[2] A tower is defined as a set of muon chambers within the same solid angle.

1.2 NIMROD crate (VME)

For the MDT barrel we propose a NIMROD crate that is located near the intersection of four towers. The FELink cable length will then be ~15 m. For the MDT barrel, consisting of 96 towers, 24 crates are needed in total. The NIMRODs in one crate are controlled by a single VME master. This VME master is connected to the DCS (Detector Control System).

If four NIMRODs share a crate, then there are 24 connections from the VME masters to the muon detector control system (barrel only). The crate is shown below. The various connections are described in the following sections.

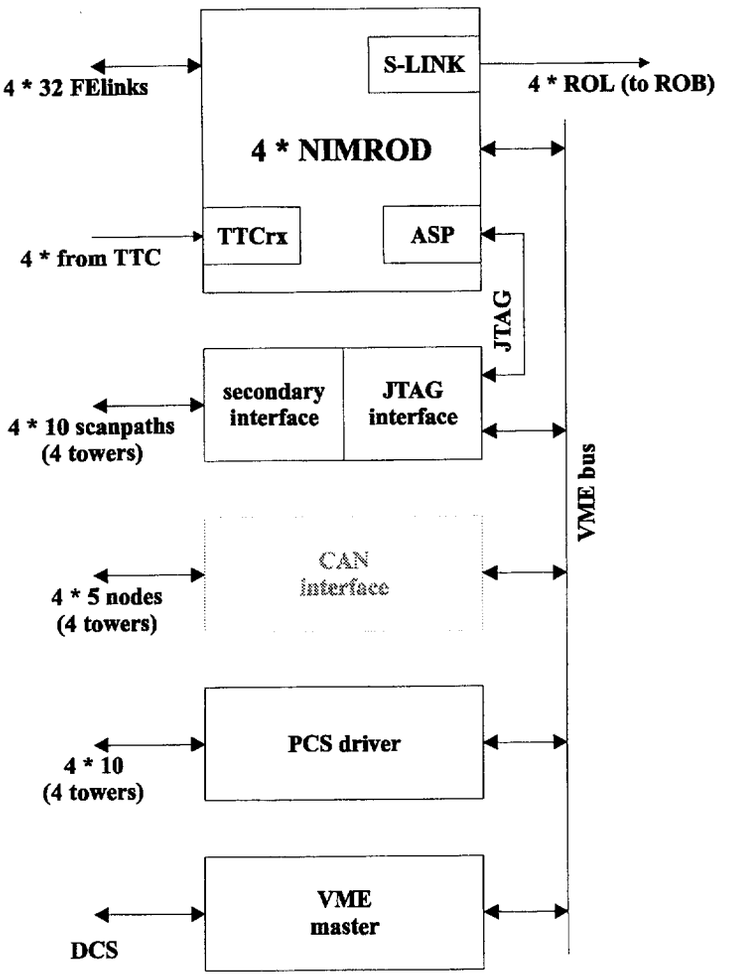


Figure 2: NIMROD crate, servicing 4 MDT towers

1.3 FELink

The link between the TDCs on the FEBoards and the NIMROD is called the FELink. The number of inputs required on the NIMROD is reduced by daisy chaining a number of TDCs. Simulations (by Christiansen) show that chaining more than 4 TDCs occasionally introduces large latencies. (The assumptions for these simulations are rather pessimistic however.) Allowing daisy chains of more than 4 TDCs does not substantially reduce the total number of FELinks for the MDT barrel. Moreover, daisy chaining reduces the redundancy in the system and increases the impact of failures. The TDC

offers the option to reduce the number of header and trailer words within the daisy chain in order to reduce the FELink occupancy.

We propose to:

- daisy chain no more than 4 TDCs.
- not interconnect TDCs on different multilayers (or even chambers) [3].

With these assumptions, no more than 26 FELinks are connected to one NIMROD for the towers currently defined for the ATLAS muon chamber barrel. For the design of the NIMROD a maximum of 32 FELink inputs is foreseen.

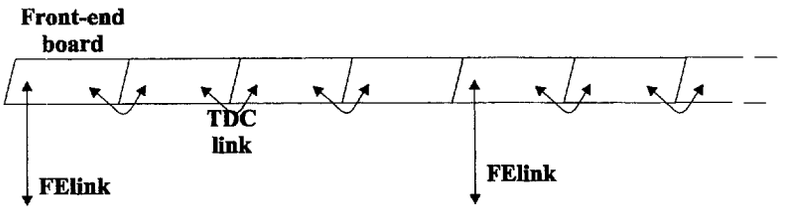


Figure 3: FELinks on the chamber

1.3.1 FELink signals

The serial data from the daisy chained TDCs is received via a differential line. The levels used are likely to be LVDS. Data is transmitted synchronous with the LHC clock at a rate of 80 Mbit/s. We propose to use two lines for the data transmission using the DS protocol. There is no handshaking between the TDCs and the FIFOs, so the NIMROD will have to keep up with the FELinks.

Two signals, generated by the TTCrx chip through the NIMROD, are distributed via differential lines:

- The 40 MHz LHC clock. All signals are synchronous to this clock.
- A combined signal, which length in clock-cycles determines its function [4]:
  - one clock-cycle: L1A is a timed pulse for each accepted level 1 (L1) trigger.
  - two clock-cycles: BCR is the synchronization signal of the LHC machine which resets the Bunch Counters in the FE electronics.
  - three clock-cycles: ECR is the Event Counter Reset which also resets all registers and memories in the TDC.

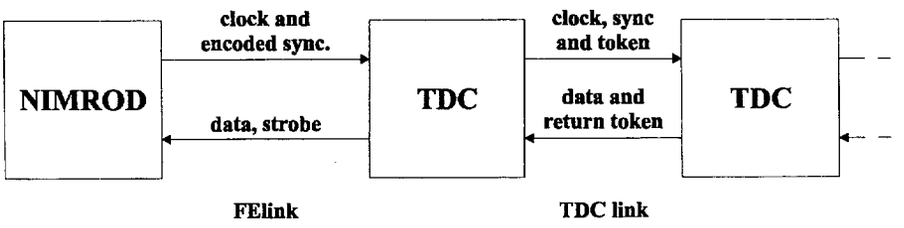


Figure 4: Signals on the FELink

[3] See: "The optimization of the tower structured read out of the MDT barrel" by A.C. König.

[4] At this moment it is not guaranteed that the different functions do not arrive at the same time.

1.3.2 FELink notes

The number of FELinks foreseen for the Barrel MDT system is order of 4000 if a maximum of 4 FEBoards are daisy chained into one FELink. Cable lengths in the order of 15 m are foreseen now. A low cost and effective solution for the physical part of the FELink is the use standard Telecom CAT5 FTP cable with RJ45 connectors. These local area network cables and connectors are used in extreme high quantities all over the world for high-speed networking and are well specified by the FCC, both components as well as the final cable connections. FCC Category 5 Network with Foiled Twisted Pair (shielded) cable, can be used for serial data rates up to 100Mbit/s. Specifications include limits on cable attenuation (22dB max. at 100MHz), cable and connector cross talk (NEXT <40dB) and radiated EMI.

A standard CAT5 FTP cable with RJ45 connectors provides 4 twisted-pairs for the FELink. This is enough for the signals needed:

- 40MHz LHC clock to TDC,
- TDC controls: L1A (Trigger), BCR and ECR resets,
- Data Out to NIMROD,
- Data Strobe to NIMROD.

Data Out and Data Strobe are combined to 'DS'-Link like signals -as used and defined by Thomson SGS for 100Mbit/s DS Transputer Links- to ensure a larger skew tolerance, using the same bandwidth. The TDC transmission clock is 80MHz

Using a 'more or less' standard like 'DS' has two advantages:

1. the implications and implementations are fully understood: see the Transputer Link specification.
2. Data Out and Data Strobe are both transmitted, giving uncomplicated transmit and receive circuits. The 80 MHz clock does not have to be recovered by a phase locked loop or an over-sampling circuit.

We propose to use LVDS (Low Voltage Differential Signaling) for all transmissions over the FELink. LVDS uses a switched 4mA current source to drive a 100 Ω terminated twisted pair (TWP) cable. This gives a voltage swing of only 400mV. LVDS is low power, can be easily implemented in ASICs (like the Arai/Christiansen TDC) and are also available as standard 26LS31/32-like devices. It may be possible to limit the slew rate of LVDS signals on the driver side in order to minimize the higher frequency harmonics and thereby reduce the radiated EMI.

LVDS drivers and receivers are available now from National Semiconductor in 26LS31/32 pin-compatible in both 5V and 3V versions. For example: DS90C031TM / DS90C032TM as 5V devices and DS90LV031TM / DS90LV032TM as 3V devices. These are Quad Line Driver/Receivers.

1.4 TTCrx

The information from the TTC is received by a TTCrx mezzanine card. It delivers the LHC clock and various fast control signals. The information is used in the NIMROD and distributed to the FEBoards. We may wish to use just one TTCrx board which is located on one of the NIMRODs in the VME crate. From there the information received should be distributed to the other NIMRODs in the same crate via a cable.

1.5 JTAG

The JTAG (IEEE 1149.1) ports that runs to the FEBoards are used to change various settings in the ASD chips, the TDCs and the phase calibration logic. It is implemented as a multidrop bus. JTAG will also be used for board level testing during production. Communication to and from the FEBoards goes via differential lines. If one bus is used for each of the two multilayers in one MDT chamber, then 10

JTAG ports are required for a tower of chambers in the barrel. Thus 4 * 10 = 40 JTAG ports are needed per NIMROD crate for the Front-End boards alone.

The JTAG Scan Ports can be implemented in a multidrop scheme using ASP (Addressable Scan Ports) on each FEBoard. These ASPs have a 10 bit address field with 1021 individual usable addresses and 3 reserved addresses.

The reserved addresses are allocated for:

- general broadcast (very useful for booting the system),
- general reset,
- general disconnect.

The ASP also has a "CON" pin [5], indicating that the ASP is connected, This pin can be used to enable an external output "TDO" line driver. Thus the Scan Path can be transformed into a four pair bus. Since a general reset address is implemented in the ASPs the optional JTAG TRST signal is not needed. The implication of implementing ASP is that each FEBoard needs an unique address.

The Scan Path Bus will be implemented as a four pair differential bus (RS422 or perhaps LVDS-alike signal levels). All JTAG signals are unidirectional and can be terminated properly.

The use of CAT5 FTP cable with RJ45 connectors seems an optimal choice for connecting the JTAG Controller with the ASPs on the Chambers. The JTAG serial bit rate is limited by the cable length used. For example: if the Secondary Scan Path Cable is 20m (100ns) the "TCK" clock rate is limited to 5MHz. Between the boards a flat cable is more efficient.

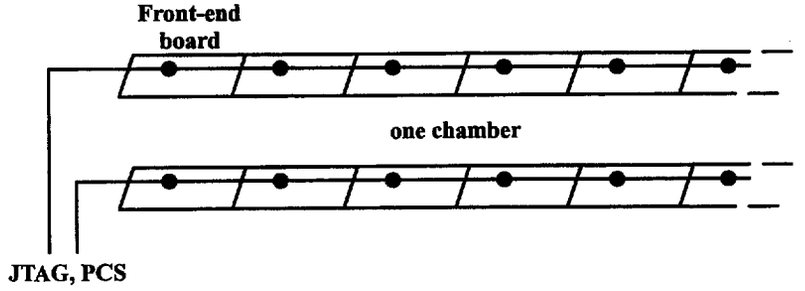


Figure 5: JTAG on the chamber

Methods are under study how to introduce redundancy in these connections, since JTAG is essential for the operation of the chips on the FEBoards. It may be possible to use the two JTAG busses on the chamber as each others back up. Separate interface chips should be used, since these are the most vulnerable parts.

[5] TI SN74ABT8996, 10-bit Addressable Scan Port, Multidrop-Addressable IEEE1149.1 (JTAG) TAP Transceiver

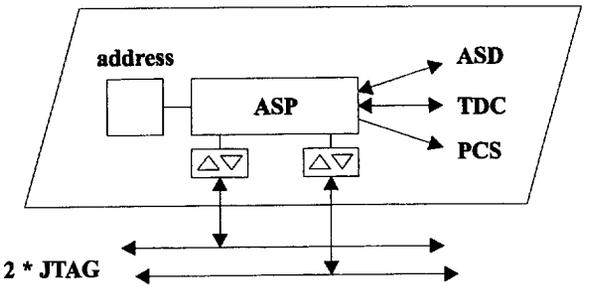


Figure 6: JTAG redundancy

The VME-JTAG interface (commercially available now) may contain six primary scan paths. Four of these can be used e.g. to service the MDT tower via 40 (multiplexed) secondary scan paths. The multiplexer will be a separate module in the VME crate. It distributes messages and collects the returned results. It also performs the electrical conversion to differential lines, to allow the use of long cables.

Locally, a fifth primary scan path is used to connect to the NIMRODs in the crate. Also for this short distance we plan to use LVDS signals. There will be a provision on the NIMROD to connect to a standard JTAG interface for testing purposes. Inside the NIMRODs, JTAG is distributed to the internal BST (Boundary Scan Test) ports and to the TTCrx mezzanine board. An ASP (Addressable Scan Path) interface is used for this internal distribution.

1.6 CAN

The CAN field-bus is used for various slow controls. To be able to perform stand alone tests and for clock phase calibrations, one needs to be able to control the high voltage for the MDT chambers. One CAN node per chamber is foreseen to handle the high voltage supplies. Thus, for a set of four towers in the barrel, a total of 4 * 5 nodes = 20 nodes are required. This may be done from the NIMROD crate, which is connected to the DCS. However, the HV control may also reside in a separate slow control crate since other detector controls probably require a DCS crate anyway. The detector controls for temperature and magnetic field measurements do not influence the read out system itself and are not needed to test the chambers.

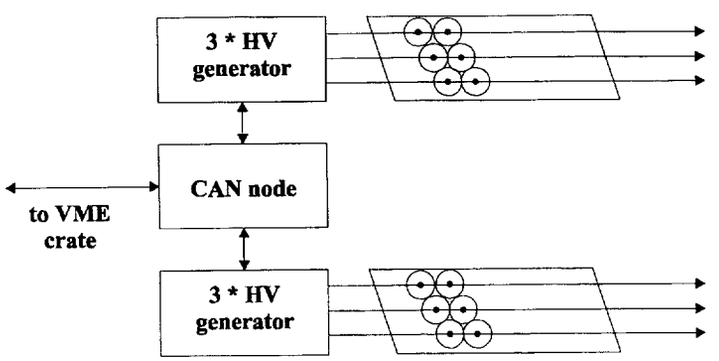


Figure 7: High voltage control

1.7 PCS

The Phase Calibration System [6] calibrates the phase of the LHC clock at the Front-End level. During a calibration run the following happens: Via JTAG the ASD chips, the TDCs and the PCS selectors are prepared. Then the PCS driver distributes a series of pulses. The results are read out via the normal data channel from the FELinks to the ROB. The Level 2 trigger should not act on this data. All data is needed for the off line analysis.

It is not clear yet how this process should be invoked. In principle the Front-End Boards and the NIMROD need a Level 1 trigger to start the read out. If this BCID actually comes from the CTP, the phase calibration pulses from the driver must be carefully timed. However, the TDC has an option to disable trigger matching. All data is passed directly to the read-out FIFO. This may be useful during the calibration run, to avoid complicated communication with the CTP.

Though the data should be read out via the ROB, it may be convenient to be able to do so via the VME bus. The VME master in the crate allows for stand alone tests of the NIMRODs and various parts of the Front-End electronics. The PCS can then be used to inject signals at the test inputs of the ASDs.

[6] See: "A possible solution for the phase calibration of the MDTs in Atlas" by H.L. Groenstege

2. Data rates in the NIMROD

The data rates and occupancy in the NIMROD are determined by several external constraints.

- The data can be sent to the ROB at 1 Gbit/s (33 Mw/s). The ROB may generate a *busy* however. The total data rate from the event built from data from the FELinks must be less than 32 Mw/s.
- The FELinks send the TDC data at 80 Mbit/s, which gives a maximum data rate on each link of approximately 2.3 Mw/s. For the barrel we estimate that approximately 60 % of the incoming data must be stored in the memory where it is to be used by the event builder and sent to the ROB. (The other 40 % are headers and trailers which are checked but not sent to the ROB.) When all 32 FELinks are used, actually used data may come in at a maximum rate of 0.6 * 32 * 2.3 Mw/s ≈ 44 Mw/s. Therefor the average occupancy of the FELinks should not exceed ~32 Mw/s / 44 Mw/s ≈ 70 %. When, for testing purposes, the data reduction in the NIMROD is disabled, all incoming data is to be sent to the ROB. This would give 32 * 2.3 Mw/s ≈ 74 Mw/s. In that case the average occupancy of the FELinks should not exceed ~32 Mw/s / 74 Mw/s ≈ 43 %.

Though the average L1A trigger rate is 100 kHz, it may go up to 1 MHz for short periods (max. 16 L1A triggers in 16 μs). Only after simulations the required buffer capacity can be determined. For a first approximation, we can calculate the upper and lower limit. For the lower limit we use the speed of the ROL and the maximum allowed latency. Thus we find 100 μs × 33 Mw/s ≈ 3300 words. This is only true if the data can be randomly accessed. But each RAM is used as a 16-fold FIFO. Therefor the upper limit is determined by the inputs: 32 channels × 2.3 Mw/s × 100 μs ≈ 7360 words. This becomes even more when the data from the individual links has to be derandomized.

The readout scheme is shown in Figure 8. It ensures that the internal bandwidth exceeds the total of the input and output bandwidth.

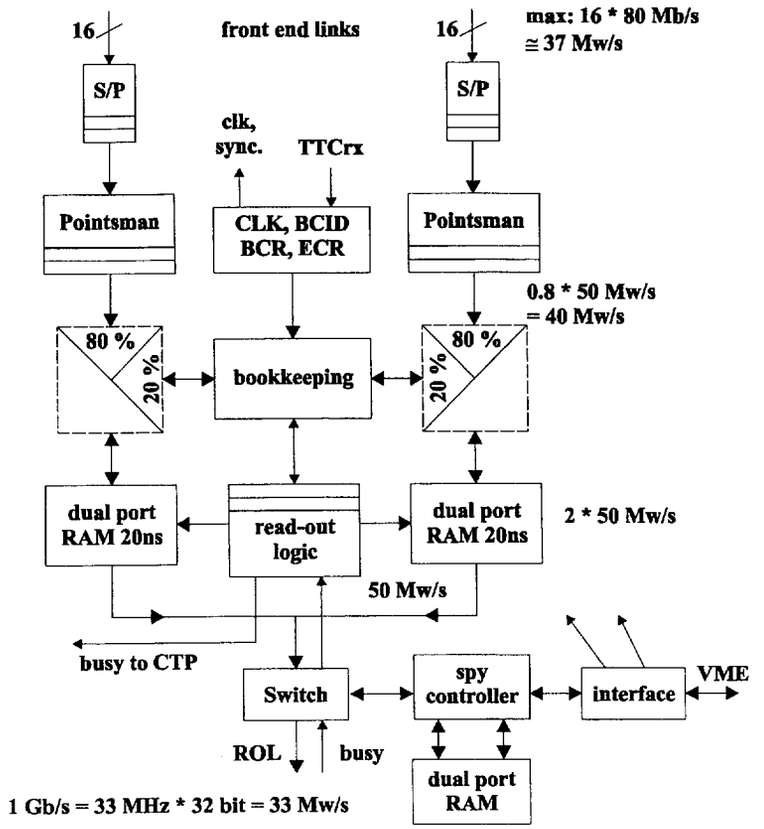


Figure 8: Data rates in the NIMROD

3. General description

3.1 Block diagram

The trigger and timing information from the central trigger processor is received via the TTCrx. The signals are distributed to the FEBoards. The information is used internally to check whether the correct event blocks are received from the TDCs. The data from the FEBoards are converted from serial to parallel (32 bit) and stored into a two words deep FIFO. The pointsman uses 80 % of the available time to store the data into the dual ported RAM. The bookkeeping handles the addressing of the RAM devices, which are used as 16-fold FIFOs. These Front-End FIFOs (FE_FIFO) will store the data coming from one FELink. The pointsman uses a double registered priority scheme. This means that the requests are handled in a fixed order and new requests are clocked into the priority encoder when it is completely empty. The bookkeeping recognizes the *end of group* words when they are stored in RAM. When these words have been received from all the TDCs, the event is complete. This is signaled to the read-out logic. This logic reads the event from RAM independently from the bookkeeping and sends the data to the ROB via the Slink. Both the bookkeeping an the read-out logic have a list of FELink and TDC numbers currently in use.

The read-out logic checks whether the complete event is available in time. This maximum time is programmable via the VME interface. If the event is not complete in time, the bookkeeping is informed and error words are written in the FIFOs of FELinks not finished in time or not written at all. Then the bookkeeping informs the read-out logic that the event is complete. Data from unfinished FELinks may optionally be skipped by the read-out logic.

The spy memory is a bi-directional FIFO. Via VME it can be instructed to spy on the event data, sent to the ROL. This can be continuously until full, every Nth event or when a specific trigger identifier is recognized. The memory can also be used to send test data from VME to the ROB. The read-out logic is disabled in this mode. A complete event (with the correct headers etc.) is loaded via the VME interface. It can then be sent to the ROB at full ROL speed. Ideas to use the spy memory for statistics are under study. The spy controller still looks at the event data, but does not store the data in the memory. Instead it will control counters, residing in the memory, depending upon the event data contents.

3.2 Event memory

When the NIMROD receives a L1 trigger from the TTCrx, the current value of the event counter (ROD_L1ID) and the BCID time stamp (ROD_BCID) are stored in the L1ID_FIFO. The L1A signal is also sent to the TDCs via the FELink and after some time the data belonging to this trigger will arrive - via the FELink - in the FE_FIFOs. When the NIMROD sequencer has collected the correct FE_BCID, the TDC data and a word count from all FE_FIFOs, the event data for this tower of MDT chambers is complete. The NIMROD then sends the event data including a header and a trailer to the ROB via the ROLink. The header will contain the L1ID and the identification of the NIMROD and the trailer includes flags, error detection or correction bits and a word count for the total event block.

The sequencer checks the current local time with the ROD_BCID time stamp of the L1 trigger being processed. It will use a time-out to ensure that data will be sent within the allowed latency. If the sequencer does not recognize the correct BCID or L1ID before the time-out, an empty event block for that FELink is generated. (More about the FIFO handling on page 14.)

When the L1ID_FIFO becomes half-full, a warning flag will be set in the next ROD header word. The ROB monitor may detect this condition and send a warning to the CTP to reduce the trigger rate. A more direct approach, where the NIMROD directly sends a busy signal to the CTP, will - for the MDT barrel - require close to one hundred signal lines. To date it is not clear whether this CTP input exists.

When the almost-full flag of the L1ID_FIFO becomes active, the event blocks from the FELinks are no longer processed by the sequencer. Empty events are generated and the corresponding data in the FE_FIFOs is flushed. Only the word counts are transmitted for off line analysis. This continues until the BCID_FIFO becomes half empty again or an FEReset is received. The situation where the L1ID_FIFO is almost-full may arise when the trigger rate is high or a lot of data has to be processed or when the Read Out Buffer is no longer capable of accepting data and sends a XOFF signal via the ROLink to the ROD [7].

In the worst case, the L1ID_FIFO contains 1 MHz * 100 μs (maximum trigger rate * maximum allowed data latency) ≈ 100 BCIDs. The actual required total depth of the L1ID_FIFO depends on the period that the trigger rate increases to higher rates. A better figure for this can only be found after simulations. However, since the memory size is very small, one can simply put enough memory (e.g. 128 words).

[7] Using a link with a handshake

3.3 TDC communication

How the data is processed in NIMROD also depends on the format used in the TDC chips and the actions taken in the daisy chain (FELink). The data format should also have a convenient format for further processing, after it has been sent to the ROB (see chapter 4).

For the moment we assume the following:

- The TDC data is preceded by a header word (Begin Of Group) containing the FE_BCID (12 bits), the FE_ECNT (12 bit event counter, which should match the least significant part of the L1ID) and some flags. The BOG word is needed by the NIMROD to be able to check whether the event blocks of each trigger are in synchronization with blocks of other TDCs.
- The TDC keeps track of lost data or lost events (e.g. because of a trigger buffer overflow) and pass this information to the NIMROD. Flag bits are set in a separate error word that is sent after all data words and before the trailer word. The NIMROD will copy the error words in the data stream to the ROB.

  The NIMROD will detect lost events, which can be caused by an overflow in the TDC or in the NIMROD or by a synchronization error. For these events the NIMROD will mark the overflow in the data send to the ROB.
- If a TDC finds no data in its internal memory for the current L1 trigger, it will send no data words to the NIMROD. It may still send header and/ or trailer words (TDC option).
- The event data is followed by a trailer word (End Of Group) containing the same FE_ECNT (12 bit event counter), a word count specifying the total number of words in this event block (including the BOG word and the EOG word). If a TDC finds errors it will report those in a separate word that is transmitted (before the EOG word) to the NIMROD.
- The word count and the flags in the EOG word will be checked by the NIMROD and the EOG may be transmitted to the ROB optionally.

  The NIMROD will normally transmit only the Front-End event blocks (containing the TDC data) to the ROB. Empty event blocks may be skipped since the TDC data words contain both a TDC number and a TDC channel number. The FELink number is passed in a ROD data word preceding the TDC data. For debugging purposes the NIMROD may be switched into transparent mode, in which all header, data and trailer words are passed unmodified to the ROB.
- Each word, transmitted serially by the TDC, contains a start bit, 32 data bits, a parity bit and one or two stop bits. In case of a parity error, the data is still included in the event and the error will be flagged in the trailer generated by the NIMROD for off line analysis.
- There is no handshake between the NIMROD and the TDC because this would cost time and require more wires in the FELink. A FIFO is needed on both sides of the FELink to decouple the serial data transmission from data generation and data reception. A large input FIFO on the NIMROD inputs is not needed if the input bandwidth is big enough and the central storage capacity is sufficient to store data for the maximum allowed latency.

3.4 FE_FIFO handling

During normal processing the NIMROD sequencer the FE_FIFOs will have some data and the event blocks can be copied to the output buffer. However one or more of the FE_FIFOs may be empty, half full or almost full. In each of these conditions the sequencer will have to take special actions.

1A. FE_FIFO empty option A:

When a FE_FIFO is empty the sequencer will wait until data arrives in the FE_FIFO. After a programmable time-out the sequencer will insert an *empty event* into the output buffer and continue with the next FE_FIFO. When time-outs occur this will cause long latencies in the arrival of the complete event in the ROB. In this way the FELink event blocks are always in the same order in the total event block.

1B. FE_FIFO empty option B:

When a FE_FIFO is empty the sequencer will continue with the next FE_FIFO. If there is data available, it is copied to the output buffer. If it is also empty, the next unprocessed FE_FIFO is checked until all FE_FIFOs are done. The sequencer handles the required bookkeeping. When there are still unprocessed FE_FIFOs after a programmable time-out the sequencer will flag the *missing events* for each unprocessed FE_FIFO.

In this case it is likely that a TDC did not generate data for this L1A only, because of an internal error or a header may have been corrupted. The readout logic of the FIFO will use the next BCID from the BCID_FIFO to check this. If again L1A is not correct, synchronization is lost. For the following events send to the ROB, error words are inserted to indicate the loss of synchronization and absence of data from that TDC. The TDC data entering the FE_FIFO is flushed or inserted in the data stream for debugging purposes (NIMROD option). Synchronization can be established again at ECReset.

2. FE_FIFO half full.

When a FE_FIFO is half full this flag is added to the header of the next event transmitted to the ROB. The amount of unprocessed data inside the FE_FIFO may (optionally) be reduced by not storing the data words of subsequent Front-End blocks coming from that FELink. Only the header and trailer word (with a flag *killed data* inserted) are stored. This should prevent the NIMROD memory from overflowing while the front-ends continue to run without interference. Through the flag in the ROD header the ROB monitor may detect this condition and send a warning to the CTP to reduce the trigger rate. A more direct approach, where the NIMROD directly sends a busy to the CTP, will take close to a hundred lines (for the barrel). At this moment it is not clear whether this CTP input exists.

3. FE_FIFO almost full.

When the FE_FIFO is almost full, all further event blocks (only headers and trailers, as dictated by the half full condition) from that FELink will be discarded, i.e. not stored in the FE_FIFO. The cutoff will be made after the trailer of the current event block has arrived.

A flag instructs the sequencer to insert *empty events* in the output buffer when it arrives at this position. The headers, still coming in from the FELink, are stored in a fixed location and overwritten by each new event. This will tell the sequencer which events (up to this header) have been discarded when it reaches this data.

This mode will be turned off when the half full condition of the same FE_FIFO disappears, giving some hysteresis between turn on and turn off.

3.5 Data handling and reduction

When the NIMROD receives a L1 trigger from the TTCrx, the current value of the event counter (ROD_L1ID) and the BCID time stamp (ROD_BCID) are stored in the L1ID_FIFO. The data from the FELinks are stored in the FE_FIFOs. They are packed together with the ROD header and trailer words in one block, thus comprising the data of one tower of MDT chambers, that is sent to the ROB. When errors are detected by the NIMROD, extra flags are set in the bookkeeping. These flags are added to the trailer words. The trailer also contains error correction / detection data and a word count for the total event block.

The order of the FELink data blocks in the outgoing stream, is determined by the NIMROD's input connectors, preferably in the order of how the TDCs are mounted on the chamber. NIMROD will not change the order of the words send by the TDCs via the FELinks.

The header and trailer words from the TDCs may be merged or skipped provided that the TDC data words carry sufficient address and channel information. For debugging purposes a transparent mode will be implemented in which all data from the FELinks will be passed unmodified to the ROB. This implies that the data format in both modes (transparent or reduced) must be compatible.

Several methods of data reduction are under study.

1. Since the event building process can only start after the event block from the FELink is complete (after the EOG word), we may not need the header word. This may function if all information needed for the synchronization and error checking can be contained in the trailer word.
2. Some flags from the FELink event blocks may be gathered into a few status words that are added to the ROD trailer words.
3. If TDCs are chained together and share one FELink, the master TDC sends one header word and one trailer word for the total FELink. A scheme on the FELink takes care of proper numbering of the TDCs.
4. For empty events the TDCs generate no data. They pass on the token to the next TDC in the chain. The *begin of group* and *end of group* words are always sent by the master TDC. These are needed for internal bookkeeping in the NIMROD. Optionally each TDC may send a *begin of TDC* and a *end of TDC* word that can be used for debugging purposes.
5. Discarding all header and trailer words coming from the FELinks in the NIMROD. This may substantially reduce the data volume. The data from each TDC may contain a word count in the trailer. If this number is correct, it may safely be discarded.

   For example: let us assume that for the barrel every L1A each TDC generates 3 data words. When a header and a trailer word are added, the event block contains 5 words in total. When 20 FELinks with each a chain of 4 TDCs are connected to a NIMROD the total number of words equals 80 * 5 words = 400 words. The NIMROD will only add ~5 words as header and trailer words. Since the true TDC data words consist of only 80 * 3 words, skipping all headers and trailers reduces the data volume to ~245 words or some ~60 %.

4. Data format

The description below gives a summary of the data format that the MDT TDC and the NIMROD use. This write-up is still preliminary and subject to changes without notice. Also it should be clear that not all of the described header and trailer words will always be present in the data. However, the format is chosen such that every type of header or data word may uniquely be distinguished without sending a special notification to the ROB handler. The NIMROD may optionally strip off TDC header and trailer words after checking to reduce the total data volume that is sent to the ROB.

4.1 Header types

The upper four bits of a 32 bit word are used as a type identifier. The header type is denoted by a one in bit 31 and the eight possible header types are shared by the ROD and the TDC:

Type 31...28	Description
1 000	Begin Of Event (for ROD)
1 001	Error flags (for ROD)
1 010	Begin Of Group (for TDC)
1 011	Begin Of TDC (for TDC)
1 100	End Of TDC (for TDC)
1 101	End Of Group (for TDC)
1 110	Checksum (for ROD)
1 111	End Of Event (for ROD)

4.1.1 ROD header types

ROD BOE (Begin Of Event) word:

Type	Description	
31...28	27...24	23...00
1000	BOEF	L1ID

L1ID: 24 bits L1 identification (event counter)
BOEF: 4 bits BOE Flags:

27...24	Function
0001	reserved
0010	reserved
0100	reserved
1000	reserved

ROD ERF (Error Flags) word:

Type	Description			
31...28	27...24	23...16	15...12	11...00
1001	ESPC	EFLG	WFLG	FELink

FELink: 12 bits FELink number
WFLG: 4 bits warning flags
EFLG: 8 bits error flags
ESPC: 4 bits Error type specification:

27...24	Function
0001	error in FELink
0010	reserved
0100	reserved
1000	etc. ...

ROD Checksum word:

Type	Description
31...28	27...00
1110	CSUM

CSUM: 28 bits Checksum

ROD EOE (End of Event) word:

Type	Description	
31...28	27...24	23...00
1111	EOEF	WCNT

WCNT: 24 bits word count for the total event block of this ROD (including this word)
EOEF: 4 bits flags:

27...24	Function
0001	reserved
0010	reserved
0100	reserved
1000	reserved

4.1.2 TDC header types

TDC BOG (Begin Of Group) word:

Type	Description		
31...28	27...24	23...12	11...00
1010	TDC	ECNT	BCID

TDC: 4 bits TDC identification of the Master TDC
ECNT: 12 bits event counter (should match lower bits of 24 bit L1ID)
BCID: 12 bits bunch crossing identifier of the trigger signal (L1A)

TDC BOT (Begin Of TDC) word:

Type	Description		
31...28	27...24	23...12	11...00
1011	TDC	ECNT	BCID

TDC: 4 bits for the TDC identification
ECNT: 12 bits event counter (should match lower bits of 24 bit L1ID)
BCID: 12 bits bunch crossing identifier of the trigger signal (L1A)

TDC EOT (End Of TDC) word:

Type	Description		
31...28	27...24	23...12	11...00
1100	TDC	ECNT	WCNT

TDC: 4 bits for the TDC identification
ECNT: 12 bits event counter (should match lower bits of 24 bit L1ID)
WCNT: 12 bits number of words in this event block (including itself)

TDC EOG (End Of Group) word:

Type	Description		
31...28	27...24	23...12	11...00
1101	TDC	ECNT	WCNT

TDC: 4 bits TDC identification of the Master TDC
ECNT: 12 bits event counter (should match lower bits of 24 bit L1ID)
WCNT: 12 bits number of words in this group of event blocks (including itself)

4.2 Data types

The upper four bits of a 32 bit word are used as a type identifier. The data type is denoted by a zero in bit 31 and the eight possible data types are shared by the ROD and the TDC:

Type 31...28	Description
0 000	ROD Reserved words
0 001	ROD Leader information words
0 010	TDC Masked channels flag
0 011	TDC Single measurement
0 100	TDC Combined measurement, charge
0 101	TDC Combined measurement, trailing edge
0 110	TDC Error flags
0 111	TDC Debug and Reserved information words

4.2.1 ROD data types

ROD Reserved words:

Here are the special six words that are fixed at the beginning of an event and one word at the end as proposed in the document "ATLAS Read Out Link Data Format by R. McLaren and O. Boyle:

Description			Description
	31...28	27...24	23...00
BOB	1011	0001	Begin Of Block MDT data
RFVN	0100	0100	ROD Format Version Number
RMID	0001	1010	ROD Module Identifier
L1ID	0000	0000	Level1 Identifier
BCID	0000	0000	Bunch Crossing Identifier
TTID	xxxx	xxxx	Trigger Type Identifier
EOB	1110	0001	End Of Block MDT data (the very last word)

ROD Leader Information words:

Type	Description		
31...28	27...24	23...12	11...00
0001	DSPC	data :	
0001	0000	reserved	
0001	0001	...	
0001	1000	FFLG	FELink
0001	1010	...	
0001	1111	reserved	

DSPC: 4 bits to specify the type of data in bit 23...00
FELink: 12 bits FELink number
FFLG: 12 bits FELink flags

4.2.2 TDC data types

TDC Masked channels flag:

Type	Description	
31...28	27...24	23...00
0010	TDC	channels with a masked hit signal preceding the trigger

TDC: 4 bits TDC identification

Note: each bit specifies one of the 24 channels that had a hit preceding the trigger that has possibly masked a hit within the trigger window.

TDC Single measurement:

Type	Description					
31...28	27...24	23...19	18	17	16...05	04...00
0011	TDC	Channel	L/T	flag	Coarse	Fine

TDC: 4 bits TDC number
Channel: 5 bits TDC channel number
L/T: 1 bit specifies coarse/fine to be Leading or Trailing edge
flag: 1 bit flag (to be specified)
Coarse: 12 bits time stamp in BC units (25 ns)
Fine: 5 bits time stamp in BC/32 units (.78 ns)

TDC Combined measurement (leading edge + trailing edge):

Type	Description				
31...28	27...24	23...19	18...11	10...05	04...00
0100	TDC	Channel	Trail	Coarse	Fine

TDC and Channel: see single measurement

Trail: 8 bits trailing edge measurement in 6.25 ns units, relative to leading edge.
Coarse: 6 bits leading edge time stamp in 25 ns units, relative to BCID of trigger.
Fine: 5 bits leading edge time stamp in 0.78 ns units, relative to BCID of trigger.

Note: all time stamps are relative to the BCID of the L1A trigger

TDC Combined measurement (leading edge + charge):

Type	Description				
31...28	27...24	23...19	18...11	10...05	04...00
0101	TDC	Channel	Charge	Coarse	Fine

TDC and Channel: see single measurement

Charge: 8 bits charge measurement in 0.75 ns units, relative to leading edge.
Coarse: 6 bits leading edge time stamp in 25 ns units, relative to BCID of trigger.
Fine: 5 bits leading edge time stamp in 0.78 ns units, relative to BCID of trigger.

Note: all time stamps are relative to the BCID of the L1A trigger

TDC Error flags:

Type	Description			
31...28	27...24	23...19	18...12	11...00
0110	TDC	Channel	WFLG	EFLG

TDC and Channel: see single measurement (present if appropriate)

WFLG: 6 bits warning flags
EFLG: 12 bits error flags

Note: The error flags are generated by the individual TDCs, not per block. The contents is to be defined by the TDC specification.

TDC Debug information:

Type	Description		
31...28	27...24	18...12	11...00
0111	TDC		

TDC: 4 bits TDC identification

Note: This word is reserved for TDC debug information and its contents will be specified by J. Christiansen when the design of the TDC chip is finalized!

4.3 Example of an event

In the example below an event is shown for which the NIMROD passed all the data from the FELinks to the ROB. (The words in gray are by default stripped off to reduce the data volume.)

Description	31...28	27...24	23 ... 00					
BOB	B	1	0	0	X	X	R	R
ROD FVN	4	4	ROD Format Version Number					
ROD begin	ROD MODULE ID							
ROD begin	0000	0000	L1ID					
ROD begin	0000	0000	BCID					
ROD begin	ROD Trigger type							
ROD status	0001	0000	reserved					
ROD status	0001	0001	reserved					

NIMROD

BOE	1000	flags	L1ID				
ROD data	0001	1000	FFLG			FELink	
BOG	1010	TDC1	ECNT			BCID	
BOT	1011	TDC1	ECNT			BCID	
TDC data	0010	TDC1	channels with a masked hit				
TDC data	0011	TDC1	Channel	L/T	flag	Coarse	Fine
TDC data	0011	TDC1	Channel	L/T	flag	Coarse	Fine
EOT	1100	TDC1	ECNT			WCNT	

BOT	1011	TDC2	ECNT			BCID	
BOT	1011	TDC3	ECNT			BCID	
TDC data	0010	TDC3	channels with a masked hit				
TDC data	0100	TDC3	Channel	Trail	Coarse	Fine	
TDC data	0100	TDC3	Channel	Trail	Coarse	Fine	
EOT	1100	TDC3	ECNT			WCNT	
EOG	1101	TDC3	ECNT			WCNT	

GROUP 1

more groups	••••			
ROD data	0001	0100	FFLG	FELink
BOG	1010	TDC1	ECNT	BCID
more data	...			
more data	...			
EOG	1101	TDC9	ECNT	WCNT

GROUP n

ROD ERF	1001	0000	EFLG/WFLG	FELink
EOE	1111	flags	Total WCNT	

NIMROD

ROD end	Number of Status words							
ROD end	Number of Data words							
EOB	E	1	0	0	X	X	R	R

5. Abbreviations

Abbreviations and other less well known definitions used in this (and other) documents.

ASD	Amplifier, Shaper, Discriminator (Boston).
BCID	Bunch Crossing IDentifier. Reset by BCR, counts 3564 clock cycles for one LHC orbit (88.924 μs).
BCR	Bunch Counter Reset, issued once per LHC orbit to synchronize Front-End electronics.
CTP	Central Trigger Processor. First level trigger, generates L1A signal.
ECR	Event Counter Reset.
FE_BCID	Front-End Bunch Crossing IDentifier.
FEBoard	Front-End Board. Contains ASDs, TDCs and phase calibration logic.
FELink	Front-End Link. Controls the FEBoards, transports the TDC data.
FEReset	Front-End Reset. From CTP, app. 1 Hz.
JTAG	(Joint Test Action Group) IEEE 1149.1. Functional test and programming.
L1A	Level 1 Accept, timed 2.5 μs after the actual event.
LVDS	Low Voltage Differential Signaling. ±400 mV. Compatible with 3V and 5V supply voltages.
NIMROD	NIKHEF MDT Read Out Driver. Gathers data from FELinks, sends complete events to ROB.
PCS	Phase Calibration System. Eliminates clock phase differences in the Front-End electronics.
ROB	Read Out Buffer. Memory module, input for the 2nd level trigger and data acquisition system.
ROD	Read Out Driver. Drives the ROL
ROI	Region Of Interest. Defined solid angle in Φ and Θ.
ROL	Read Out Link. 1 GHz link to ROB.
TTC	Trigger Timing and Control. Provides LHC clock and fast controls.
TTCrx	Receiver ASIC for TTC timing and control.
multidrop	Electrical adaptation to enable broadcasts etc..
multilayer	Set of 3 or 4 layers of MDTs.