The Read-Out Driver for the ATLAS MDT Muon Precision Chambers

H. Boterenbrood, P. Jansweijer, G. Kieft, A. König, J. Vermeulen, T. Wijnen

***Abstract*— The 1172 Monitored Drift Tube (MDT) precision chambers of the muon spectrometer of the ATLAS experiment at the LHC will be read out by 204 MDT Read-Out Drivers (MRODs). The MRODs receive event data via 1136 optical links (up to 6 per MROD), build event fragments at a maximum rate of 100 kHz, output these to the ATLAS data-acquisition system and take care of monitoring and error checking, handling and flagging. The evolution of the design of the MROD (a 9U VME64 module in which this functionality is implemented using FPGAs and ADSP-21160 Digital Signal Processors programmed in C++) is reviewed and critical issues are discussed.**

***Index Terms*— ATLAS, LHC, Detector read-out, Field Programmable Gate Arrays, Digital Signal Processors, Real time systems.**

I. Introduction

At the European Laboratory for Particle Physics CERN in Geneva a new accelerator, the Large Hadron Collider (LHC) is under construction. It is a circular colliding beam accelerator, which is scheduled to come into operation in the year 2007. To record the results of the collisions in LHC, four detectors are being constructed, one of which is the ATLAS detector [1]. It is conceived as a general-purpose detector for high-energy colliding beam experiments. The detector consists of subsystems each with their own dedicated read-out electronics.

The ATLAS muon subsystem consists of 1172 Monitored Drift Tube (MDT) chambers [2] with a total of about 300,000 individual drift tubes. The largest MDT chambers contain 432 drift tubes. The measurements with the MDT chambers require time recordings with an accuracy of 1 ns. A schematic overview of the MDT read-out chain is depicted in Fig. 1. On-chamber Time to Digital Converters (TDCs) generate time stamps from the wire signals. The data from the TDCs mounted on one, or for some small chambers, on two chambers, are sent via a Chamber Service Module (CSM) to an MROD (MDT Read-Out Driver). This contribution focuses on the MROD [3].

II. MDT Read-Out Chain

Directly on the MDT chambers, front-end cards equipped with ASD (Amplifier-Shaper-Discriminator) chips [4] and 24-channel TDC-chips (ATLAS Muon TDC or AMT [5]) process the wire signals. Each front-end card thus serves a maximum of 24 drift tubes and the largest MDT chambers have 18 front-end cards with as many TDC-chips. The TDCs are entirely data driven and record time stamps for any wire signal above threshold, which are stored in the internal buffer memory. Upon receipt of a level-1 trigger signal, each TDC-chip selects from its internal buffer any time stamps pertaining to this particular trigger and sends them over a serial point-to-point connection to the Chamber Service Module (CSM) [6]. The CSM deserializes the 32-bit TDC data words, multiplexes them, and outputs the resulting data stream after serialization via an optical link (GOL) [7] to the MDT Read Out Driver (MROD). The CSM therefore acts as a time division multiplexer. A fixed time slot is associated with each TDC, in which a single 32-bit word is transmitted. If a TDC happens to have no data supplied, a word with all bits set to 0 is inserted by the CSM in the data stream as a placeholder (words output by the TDCs never have all bits set to 0). This word will be ignored by the MROD. Note that the optical link provides a strictly one-way connection from the CSM to the MROD. The MRODs are physically located in a different underground area (USA15) than the detector itself, shielded from the radiation in the detector cavern. The design of the MROD is therefore not constrained by requirements on radiation hardness. The length of the optical fibers connecting CSMs and MRODs is of the order of 100 m.

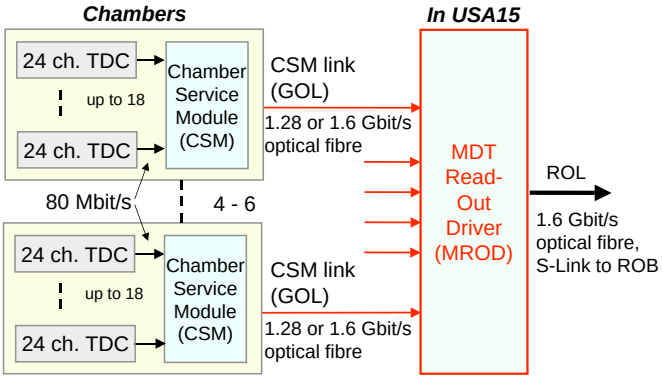

Fig.1. Overview of the MDT read-out chain.

Data overruns in the CSM are impossible as long as the TDCs do not send more than a single word per CSM read-out cycle. This requires the speed of the CSM output link to be high enough with respect to the speed of the serial TDC-to-CSM links. The initial design choice of 40 Mbit/s for these

Manuscript received June 16, 2005, revised March 8, 2006.

H. Boterenbrood, P. Jansweijer, G. Kieft, and J. Vermeulen (corresponding author, phone: (31) 020 5925108, e-mail: i73@nikhef.nl) are with NIKHEF, PO Box 41882, 1009DB Amsterdam, The Netherlands.

A. König, and T. Wijnen are with IMAPP, Dept. of Experimental High Energy Physics, Radboud University Nijmegen, PO Box 9010, 6500GL Nijmegen, The Netherlands.

links and of 0.8 Gbit/s (25 MHz of 32-bit word transfers) for the CSM output link speed meets this condition. Thus only simple data handling by the CSM was originally foreseen. This is desirable in view of the radiation levels in the detector cavern. To diminish the probability of internal TDC buffer overflows, it has recently been decided to increase the speed of the TDC-to-CSM links to 80 Mbit/s. To still exclude CSM data overruns the CSM output link (GOL) speed should also be doubled to allow for a 1.6 Gbit/s data rate. As of now, operation of the GOL at 40 MHz (1.28 Gbit/s) has been demonstrated. Running the GOL at 50 MHz (1.6 Gbit/s) may be possible and will be attempted. In the meantime a more complex data handling scheme in the CSM, allowing data selectively to be discarded if necessary, has been designed and implemented.

III. MROD Functionality

The main task of the MROD is to receive the data streams from up to six MDT chambers, which in most cases together form a "tower". The MROD builds event fragments from the incoming data and outputs these via an optical link. The output links of the RODs are S-Link [8] connections and are referred to as Read-Out Links (ROLs) in the ATLAS trigger and data-acquisition system [9]. The ROLs connect to Read-Out Buffers (ROBs), which are located on ROBIN cards [10], from where the data can be retrieved by the second-level trigger and by the Event Builder [9]. The MROD detects and reports errors and inconsistencies in the incoming data streams (and where possible initiates corrective action). Ideally it also collects statistics and it allows to "spy" on the data. Moreover, data reduction schemes are implemented in the MROD.

IV. Data format

The event format for the MDT data is imposed by the format of the data output by the TDCs and by the ATLAS ROD output format. The event format is described in detail in [11] and can best be visualized as consisting of three nested levels of "envelopes", see Fig. 2. The lowest of the three levels is the TDC level. The next higher level is formed by the CSM level, whereas the highest level corresponds to a group of up to 6 chambers (i.e. MROD). Each envelope has the same basic structure: one or more header words followed by a number of data words and terminated with a trailer containing the word count for the envelope as a whole.

For the TDC and CSM levels, the respective trailers include a 12-bit event number. Envelopes may be empty, i.e. contain no data words. At each level the envelopes of the directly lower level are integrally included as data words in the envelope of the current level. The TDC envelopes are generated by the TDCs in the form of header and trailer words. In the absence of hits, a TDC outputs an empty envelope for each level-1 trigger. Since the CSM is virtually transparent to the data, the CSM level envelope is generated by the MROD (MROD specific information is also added), as is the MROD level envelope.

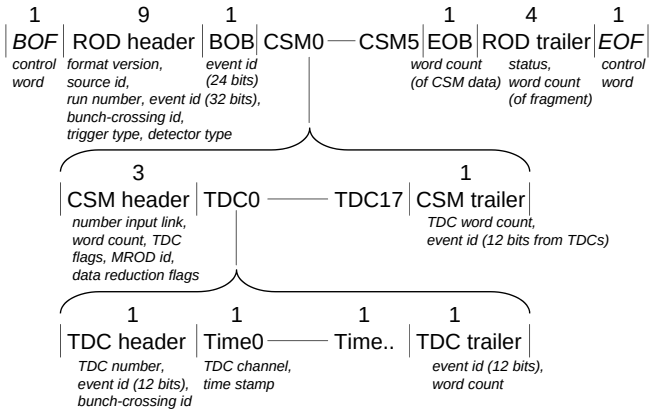

Fig. 2. Organization of the data output by the MROD. The numbers specify the number of 32-bit words. The first and last word (BOF and EOF) are control words, which separate fragments and are not stored by the receiving ROB. Redundant information (event id's, word counts) allows for checking for errors. Apart from the ROD header and trailer words, the 4 most significant bits define the type of each word output. In case of parity errors (parity is checked for the transmission from TDC to CSM and for the transmission from CSM to MROD), the TDC word is replaced by an error word with its own 4-bit identifier, and with the identifier of the TDC word placed in bits 24 to 27, while the 24 least significant bits are copied to the 24 least significant bits of the error word.

V. Data rates

The dominant contribution to the hit rate of the chambers is coming from hits due to conversions of neutron-induced photons. Based on the most recent calculations of the neutron background and assuming the full LHC design luminosity, the maximum output data rate for any single MROD has been estimated [12]. In view of uncertainties in the rates it has been agreed that the read-out chain of the ATLAS muon system should be able to handle a data rate five times larger than the nominal rate. The MROD can reduce the output data rate to an acceptable level by discarding empty TDC envelopes and / or by discarding TDC trailer words, see Fig. 3 for the estimated average number of words per event and per input if TDC trailer words are discarded. The figure is for the case that two 32-bit words, i.e. two time stamps, one for the leading edge and the other for the trailing edge of the input signal, are produced per wire hit and for a background rate five times higher than the nominal rate, in combination with the maximum level-1 trigger accept rate of 100 kHz. The second time stamp allows to determine the charge collected from the wire, as the amount of charge can be encoded in the length of the input signal of the TDC.

VI. Evolution of the design of the MROD

The final design of the MROD, the MROD-X, is based on the MROD-1 prototype, which was designed making use of initial experience gained with hardware originally developed as part of a study for hardware for the ROB function in the ATLAS DAQ system [13].

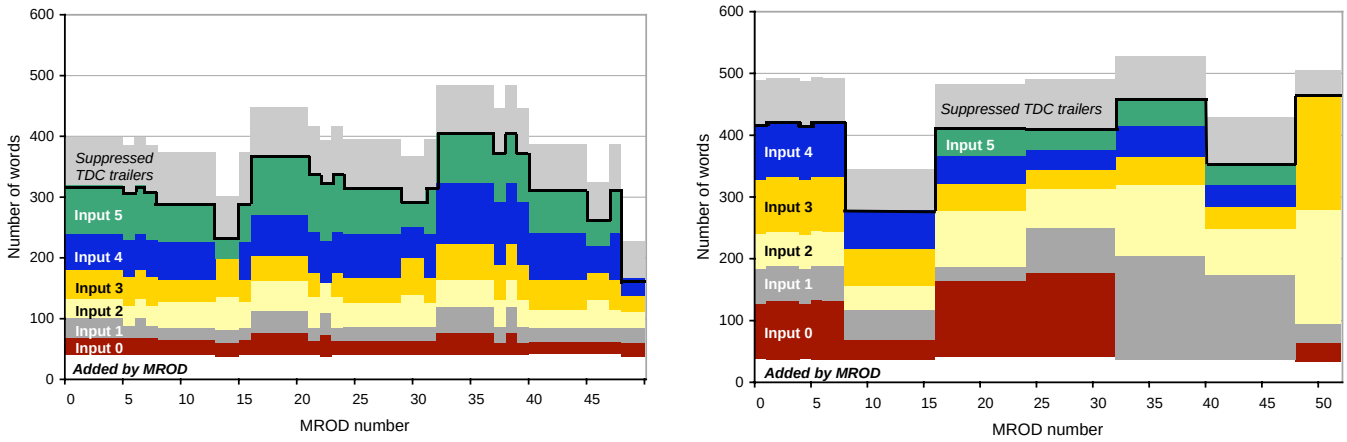

Fig. 3. Estimates of the number of words per event and per input, for two words per hit, a background rate five times larger than the nominal rate and if all TDC trailer words are discarded. The plots are for the MRODs receiving data from the barrel chambers at one side of the interaction point (left) and from the chambers of one endcap (right). The number of words added by the MROD is indicated, as well as the sum of the number of TDC trailer words discarded. The thick lines indicate the average size of the fragments output by the MRODs. At the maximum first level trigger accept rate of 100 kHz the highest data rate is about 180 MByte/s or 1.5 Gbit/s. Discarding also the TDC headers of empty TDC envelopes results in a further reduction of the fragment size of up to about 35 words.

The MROD-1 and MROD-X make use of FPGAs and of Analog Devices ADSP-21160 "SHARC" Digital Signal Processors (DSPs) [14], which have an internal fast memory of 512 kByte and six half duplex data links. The links are used for fast point-to-point interconnections between the DSPs. For the MROD-1 the clock frequency is 80 MHz, for the MROD-X 100 MHz. The maximum bandwidth per link is specified to be 80 or 100 MByte/s for an 80 or 100 MHz clock respectively. However, for the MROD design it was found that reliable data transfers are only possible for link speed settings for which the bandwidth is a factor of two or more lower, i.e. the maximum bandwidth is 40 or 50 MByte/s. DMA controllers are associated with the links. Data transfers between the DSPs are automatically synchronized due to hardware handshaking across the links. The DSPs also have an external parallel bus with associated DMA controllers. All DMA controllers support "chained transfers", i.e. the execution of a sequence of transfers of data blocks without any intervention of the CPU of the DSP.

The MROD has a modular design with separate input and output sections called MRODin and MRODout respectively. A schematic overview is given in Fig. 4 for the MROD-1 prototype, which is physically built as a two slot wide 9U VME64x module with three dual-input daughter boards (the MRODins) mounted on a motherboard, the MRODout. Each MRODin serves two inputs, the MRODout connects to the VME64x bus and to an S-Link interface. The interfaces for the six CSM links and for the S-link are also implemented as daughter boards (mounted on both sides of the MRODout).

For the MROD-1 Altera® APEX™ 20K200 Field Programmable Gate Arrays (FPGAs) [15] have been used for the de-multiplexing and associated management of the data received from the CSMs. Two FPGAs connect to one DSP. The DSP builds event fragments from the fragments input from the two FPGAs and forwards these to another DSP in the MRODout via a link.

The MROD-1 prototype has been used on a routine basis in the 2003 and 2004 runs of the H8 test beam at CERN, Geneva. In total three MROD-1 modules were used to read out up to 15 MDT chambers. In addition, MROD-1 modules have been used at NIKHEF, Amsterdam, in the quality assessment procedure, using cosmic rays, for the MDT chambers constructed at NIKHEF, and are used for the commissioning of the MDT chambers in the ATLAS experiment.

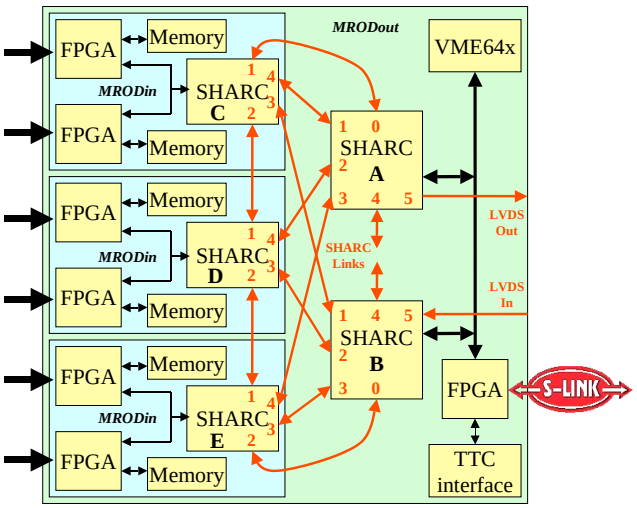

Fig. 4. Schematic overview of the design of the MROD-1.

In the final production version (the MROD-X) the Altera FPGAs have been replaced by Xilinx Virtex™ II-Pro FPGAs [16], just becoming available at the time the design of the MROD-X started. Each device has data links, 8 RocketIO™ links, which each can transfer up to 3 Gbit/s. In view of the time constraints dictated by the installation and commissioning phases of the ATLAS experiment the MROD-X retained the architecture of the MROD-1, with the addition of direct RocketIO link interconnections for direct transfer of event data between FPGAs in a later stage of the project.

The MROD-X (Fig. 5) could be made one slot wide by implementing the interfaces for the CSM links and the MRODins on the MRODout board. Originally it was planned to read out with each MROD a complete "tower" of the spectrometer. For most of the towers this can be done with MRODs having 6 inputs, but for some 8 inputs are required. For this reason the MROD-X has up to four MRODins (for the MROD-1 the use of a separate MRODin board mounted on a dummy motherboard and connected via a dedicated link to the MRODout section of the MROD-1 was foreseen). Four of the six MROD-X pre-production modules have been manufactured with four MRODins, the other two modules have three MRODins. In the autumn of 2005 it has been decided to use exclusively modules with three MRODins to make the logistics of production and of maintenance simpler. Collecting all the data from one tower in one MROD therefore in some cases is not possible, but on the positive side the data volume output by the MRODs can be distributed more evenly over the Read-Out Links. In total 204 modules will be needed to read out the spectrometer. The complete system will be housed in 16 9U VME crates, with 12 or 13 modules per crate. Each crate will have its own crate controller running Linux as operating system and will have a dedicated interface to the central Timing, Trigger and Control (TTC) system [17] of ATLAS and to the Busy logic in the form of a TTC Interface Module (TIM) [18]. The TTC signals received by the TIM are distributed to the MROD modules via a custom backplane, which also routes the busy signals from the MRODs to the TIM module. The MROD asserts its busy signal if internal congestion is detected, caused e.g. by assertion of the XOFF signal of the ROL by the ROB downstream of the MROD.

The six MROD-X pre-production modules have been extensively tested with simulated data. A few minor issues were corrected. Tests at environmental temperatures of 0 and 70 °C have also been carried out, these did not reveal problems. The modules have also been used for data taking with MDT chambers, using cosmic rays.

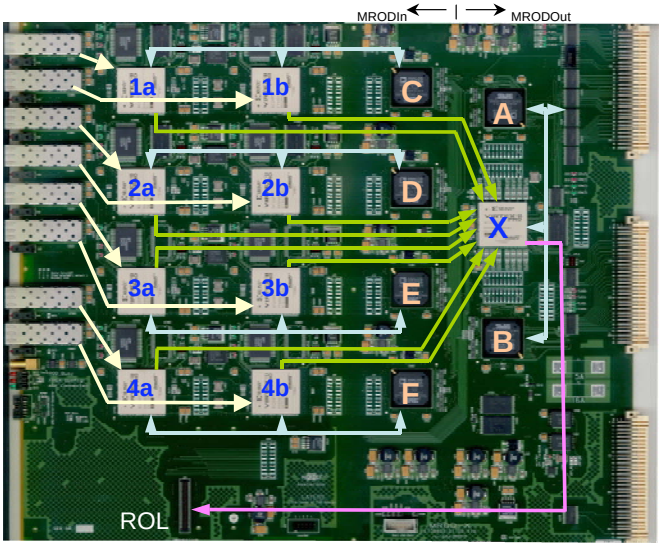

Fig. 5. The MROD-X board. CSM links connect to the left, the devices labeled with 1a, 1b, etc. are the Xilinx XC2VP7 FPGAs of the 4 MRODins, A and B are the MRODout DSPs, C, D , E and F are the MRODin DSPs, X is the Xilinx XC2VP20 FPGA of the MRODOut. The bus connections between FPGAs and DSPs are indicated as well as the RocketIO links connecting the MRODin FPGAs to the MRODout FPGA. SHARC links are not shown. The S-link interface is a daughter board mounted at the lower left corner. In the production version of the board MRODin FPGA 4a and 4b and DSP F will not be present, as at maximum 6 inputs will be used in ATLAS. Also DSP B will not be present.

VII. Design Choices

Initially the SHARC DSP was chosen as data movement engine and for fragment building, because of its data transport facilities and the flexibility allowed by software. It was known at the start of the project that it would not be easy to satisfy the requirements, but the SHARC interconnection technology appeared more attractive than other options (like implementing busses for connecting FPGAs). Unfortunately, as already mentioned, the SHARC link speed was lower than foreseen. Therefore in some cases (i.e. for CSMs with high hit rates at rates five times higher than nominal, see also section V) it is not possible to transfer all event data output by a MRODin via a single SHARC link. Also the time-critical software needed for managing data transports and fragment building turned out to be quite complex. Nonetheless, handling of event rates of 100 kHz (for three data streams, two input and one output, in the MRODin and four data streams, three input and one output, in the MRODout, each with 100 kHz event fragment rate) has been proven to be possible.

A different and relatively simple approach with respect to fragment building is made possible by the Xilinx devices used in the MROD-X. The speed of the RocketIO links can be set equal to or higher than the speed of the MROD output link, so that fragment building in the MRODout FPGA basically consists of generating the correct envelope data words and of routing the output streams of the MRODin FPGAs one after the other to the MROD output link. This mechanism has been implemented for the first pre-production versions of the MROD-X and functions as expected. Fragment building by the DSPs and event data transfers across their links are therefore no longer needed, although still possible in the same way as in the MROD-1. The DSPs still take care of initialization, control and error handling and can also be used for monitoring, as a programmable fraction of the event data still can be transferred to the DSPs. In the production version of the MROD-X the MRODout will have only a single DSP instead of two DSPs, as it is now clear that the DSPs will not be used in the same way as in the MROD-1 (the two DSPs would allow to increase the throughput and maximum event rate at the cost of additional synchronization, however, in the studies performed with the MROD-1 the event data was handled by only one of the two DSPs).

VIII. The MRODin FPGA

The FPGAs of the MRODins process the incoming data without any intervention of the MRODin DSP in the absence of errors. The FPGA demultiplexes the CSM data and removes words only containing zeros, and thus reconstructs the data streams of the individual TDCs. These streams are stored in the memory associated with the FPGA, each in its own 32 kByte partition. On the fly the FPGA recognizes the TDC trailer words and reads the event number. The trailer word flags that the TDC has completed the transmission of data for the event indicated by the event number. The individual TDCs produce their data strictly time ordered event per event. However, the data streams of the different TDCs connected to a CSM are not necessarily in phase, while the event fragments to be output by the FPGA need to contain all data associated with the same event number from all TDCs. To signal that all these data have arrived and are stored in the memory, a bit is set in a two-dimensional bit array upon arrival of a TDC trailer word. The TDC number determines the column of the bit. The 4 least significant bits of the event number, extracted from the TDC trailer word, define its row. Once all or a programmable subset of the bits in the row that is associated with the "expected event number" are set, the event data of one entire chamber are complete. The (expected) event number and the bits in the row corresponding to it are sent to the output controller, which collects the data from the different memory partitions and outputs the data. Note that for this mechanism to work properly, it is crucial that each TDC sends at least the trailer word, even in cases when it has no data. Error conditions, which may result from corrupted trailer words or erratically behaving TDCs, are detected on the basis of the bit states.

Empty TDC envelopes may optionally be skipped by the output controller, i.e. data reduction may be applied. In the MROD-X also TDC trailer words may be skipped (see also section V). At this point also the CSM level envelope is generated by the FPGA and the TDC data are enclosed to form the event fragment. The data can be output to the internal memory of the MRODin DSP and / or, for the MROD-X, to the MRODout FPGA via a RocketIO link. In the latter case a programmable fraction of the event fragments can be transferred to the MRODin DSP for monitoring purposes. Transfers to the DSP proceed by way of DMA transfers, with handshaking between the FPGA and DSP. For each event fragment the FPGA passes one word, containing the length and the 12-bit event number of the event fragment, via a separate FIFO to the DSP. The availability of this "length word" indicates that all event data have been read from the buffer memory associated with the FPGA. Due to pipelined data handling, these data may not all have arrived yet in the DSP memory when the information in the FIFO becomes available. Transfers to the RocketIO link are processed likewise. The event fragment, as well as one word containing the event fragment length and 12-bit event number, are stored in two separate FIFOs. The content of these FIFOs is sent over the RocketIO link where it should be noted that the FIFO for the event fragment length and event number is given priority. The FIFO for the event fragment is chosen big enough to accommodate a complete event fragment. This scheme assures that whenever there is a fragment length word present in the FIFO at the receiving side of the RocketIO link, a complete event fragment is waiting to be read out from its corresponding FIFO.

The MRODin FPGA checks for a number of error and exception conditions:

- parity errors on the TDC to CSM link (this parity is checked and errors are encoded in the data by the CSM),
- link and/or parity errors on the CSM to MROD link,
- memory partition overflow,
- incorrect or too long event fragments from a TDC,
- absence of expected trailer words or corruption of trailer words, as mentioned above.

In case an error is detected, the FPGA may interrupt the DSP, which then intervenes appropriately. For very serious conditions (too large event fragment, memory partition overflow, absence of data) the FPGA will independently decide to ignore (shut off) an individual TDC channel. This is flagged in one of the envelope words. Also in this case the DSP will be notified by means of an interrupt.

IX. The MRODout FPGA

The FPGA of the MRODout interfaces to the VME bus, to the S-link interface and to the external bus interface(s) of the MRODout SHARC DSP(s). In the MROD-1 (or in the MROD-1 compatible mode of the MROD-X) event fragments are passed to the FPGA from one of the SHARC DSPs, and are then output via the S-link. For normal operation of the MROD-X the event data are transferred to the FPGA via RocketIO links, as described in the previous section, and are passed on to the output S-link. The FPGA also takes care of sending the ROD header and trailer and checks for error conditions, which again are handled by the DSP. A programmable fraction of the fragments output to the S-link can also be transferred to the DSP for monitoring purposes, with the same mechanism as implemented in the MRODin FPGA.

X. MROD-1 software for data transport management and fragment building

The software for the DSPs of the MROD-1 used in the 2003 test beam run was written in C. For the run of 2004 most of the software has been rewritten in C++ to improve its maintainability, to add functionality and also to optimize the performance.

In the MROD-1 software all event fragment transfers take place for each DSP under control of a subset of the 14 DMA controllers of the DSP. Chained transfers are used, in which a DMA controller of the DSP after completion of a transfer fetches a new so-called Transfer Control Block from memory. This contains the information needed to start the next transfer and also a pointer to the location of the next Transfer Control Block. Synchronization between senders and receivers of data is automatically achieved by hardware handshaking, which is implemented between the FPGAs and DSP of the MRODin

(see section VIII), and which is inherent to the SHARC-links. For output to the S-link the situation is different. A 1024 word deep FIFO buffer absorbs data output by the MRODout DSP(s), unless it is full. The FIFO has a half-full flag. The DSP bus cycle will stall if the FIFO is full and only finish after space for a new word has become available and the word is transferred. This presents a problem, because an access from the VME bus during the cycle stall will cause a VME bus error. A DMA transfer cannot be paused once it is started. Each DMA transfer is therefore limited to at maximum 512 words and can be started only if the half-full flag is not set. This makes additional checks necessary, while single data blocks (event fragment output by a single MRODin) longer than 512 words have to be transferred as a number of smaller blocks with a size of at maximum 512 words. The additional overhead reduces the maximum event rate that can be handled, see also section XIII.

Starting of data transfers and waiting for them to finish is decoupled as much as possible from setting up the necessary control information for the transfers and from buffer management. This makes it possible to determine the actions to be taken on the basis of polling, hence avoiding the overhead associated with the use of interrupt driven operation, while the throughput is maximized as data transfers start autonomously if possible and as waiting of processing tasks for data transfers to finish is minimized

Minimization of the time required for the necessary processing is accomplished by causing the compiler to replace method invocations as much as possible by “inline” code (to avoid performance loss due to the overhead associated with method invocation). Furthermore no use is made of dynamic memory allocation and Transfer Control Blocks are set up as much as possible during initialization of the DSP program.

XI. Software environment

The DSPs are booted via the VME bus, the MRODin DSPs via their SHARC-links (in Fig. 4 the links connecting A to link 4 of C, to link 4 of D and to link 4 of E). This is possible because the SHARC-link interfaces of the MRODout DSPs can be accessed directly from the VME bus. The MRODout DSPs are booted via their internal memories, which are also accessible from the VME bus. Per DSP a combined loader and server program can be run on the crate processor, which runs Linux. A run-time library linked to the DSP program facilitates Unix-like handling of command line arguments, and terminal and file I/O by the DSP program. These facilities have proven to be essential for checking and debugging the software. For data taking the MRODs in a crate will be under control of and communicating with the ROD Crate DAQ (RCD) software [19], which will include the loader and server functions.

Communication between the RCD and MROD DSPs is necessary during running for control and acquiring error and monitoring information. For this purpose a communication framework has been developed and implemented which makes it possible to send messages to individually addressed DSPs, to broadcast messages to the DSPs as well as to send messages from any DSP to the RCD. The framework also supports a special type of messages for terminal output. The messages are transferred between the DSPs via SHARC-links not used for event data transfers and not used for booting. Using these links a ring of interconnected DSPs is formed, via which data are always transmitted in one direction. The MRODout DSP, or, if there are two MRODout DSPs, one of the two DSPs, communicates via its internal memory with the crate processor. All messages between the DSPs again are transferred under DMA control. The framework is operated on the basis of polling and consumes a minor fraction of the DSP processor time, as it needs to be invoked only relatively infrequently with respect to the polling loop associated with event data processing. The framework has been tested with the help of the loader and server program mentioned earlier. The framework has a negligible effect on the maximum event rate that can be handled if the DSPs take care of management of event data transfers and building of event fragments.

XII. Software development

To ease code development and debugging an emulation of the complete MROD has been implemented as a C++ program, which can be compiled for any environment with a suitable compiler and a command line interface. In the program each DSP is represented by an object, from which the code for the real MROD is invoked. Only a very small amount of code needs to be changed for the emulation, this is achieved with the help of conditional compilation. All DMA transfers are emulated and are started in the same way as on the real hardware. The exact timing of the various transfers is not emulated, but it is possible to control the relative order of the transfers. This has helped to reproduce errors and to find and remove the bugs responsible for these errors. Furthermore it has been found to be possible to speed up the development and debugging process considerably by making use of the interactive symbolic debugging and code browsing facilities of a modern integrated development environment.

XIII. Throughput Tests

Throughput tests of the MROD-1 have been performed using hardware data generators, which supplied emulated data to the inputs of the MROD-1. For the measurements reported here event fragments consisting of 18 pairs of TDC headers and trailers were used. The data generators were triggered at a constant rate. Although possible, no TDC words with time stamps (see Fig. 2) were added. Data reduction was not applied, in contrast to what would be done during actual data taking. The DSPs of the MROD then handle fragments with roughly the same size as used for the measurements, due to the presence of time stamps in the data in combination with the removal of “empty” header-trailer pairs and of trailer words. For the condition described for each input event fragments of 40 32-bit words (4 CSM “envelope” words and 18 pairs of header and trailer words) were flowing from MRODin via MRODout to the S-link output interface. A

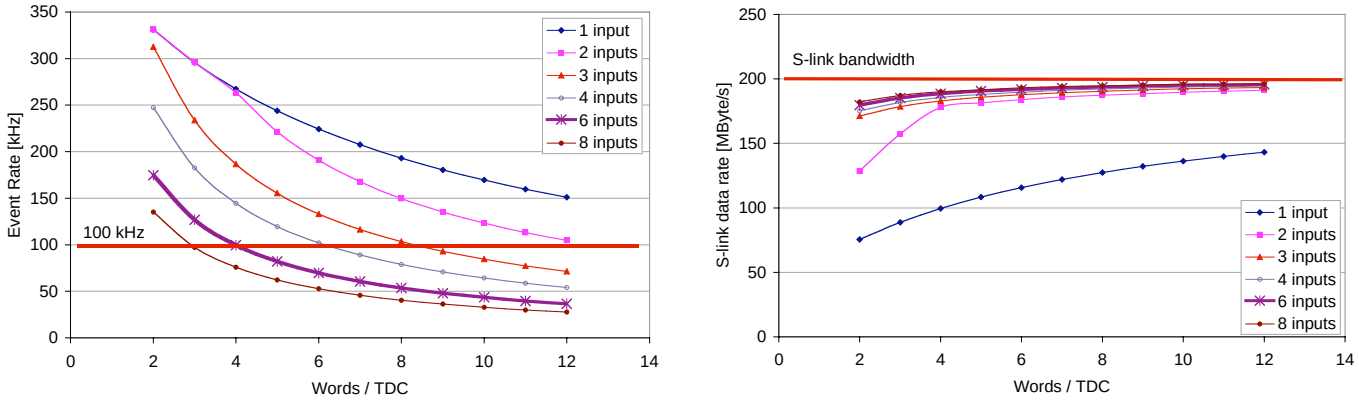

Fig. 6. Maximum event rate and associated output bandwidth for the MROD-X as function of the number of words per TDC, with each active input receiving simulated data from 18 TDCs and for each TDC the same number of words. For more than 2 inputs active the rate is limited by the S-link bandwidth (here: 200 MByte/s). For 2 or 3 words per TDC and 2 inputs active and for all measurement points for one input active, overheads associated with the MRODin FPGA limit the rate. However, in practice at least 4 inputs are used.

dummy S-link interface was connected to it and was acting as data sink. It was found that the maximum event rate is 80 kHz if data is handled from all 6 inputs. If only one MRODin is handling data from both its inputs the maximum event rate is 115 kHz. This maximum can be understood to come from the bandwidth required for transfer of the data from MRODin to MRODout. This is 37 MByte/s, close to the 40 MByte/s speed setting of the SHARC-link. With data reduction switched on per event only two fragments of 4 words need to be transferred from MRODin to MRODout for data without time stamps. A higher maximum event fragment rate of 125 kHz is found for this case: now the available processing power of the DSP of the MRODin is limiting the rate. With 6 input links active and 3 MRODins sending their data to one of the MRODout DSPs the processing in the latter DSP limits the maximum rate to 80 kHz. The part of the software needed to avoid overflow of the FIFO in the output S-link interface has been found to reduce the maximum achievable rate considerably, a maximum rate of 110 kHz has been measured without it. The performance of the DSPs and the bandwidths of the links scale linearly with the DSP clock frequency. Therefore, the maximum event rates can be expected to be a factor 1.25 higher for the MROD-X, as a 50 MHz clock frequency is used instead of 40 MHz, making it possible to sustain a fragment rate of 100 kHz in the MROD-1 compatible mode while using one DSP in the MRODout.

The throughput of the MROD-X with FPGA based fragment building has been determined with the help of test data generators built in the MRODin FPGAs. These test generators are only available in the MROD-X and can be triggered by a TIM. They transmit event data via the senders of the bi-directional optical interfaces for the CSM links and via loopback fibers to the receivers of these interfaces (the CSM link itself is uni-directional, the sender of the MROD optical interface is not used for normal operation). The trigger generator in the TIM can be throttled by means of the busy signal of the MROD.

The results of the measurements, presented in Fig. 6, show that handling of the event data in the MROD no longer limits the rate, apart from an overhead caused by the MRODout FPGA: for each event, data can not be transmitted via the output link during 18 + 2N clock cycles of 20 ns, where N is the number of active inputs. For 100 kHz event rate and N = 6 this results in a maximum output bandwidth of about 190 instead of 200 MByte/s under the conditions of the test. The rate is limited by the overhead associated with the MRODin FPGA if only a single input link is used, and also for 2 or 3 words per TDC for two input links. In ATLAS at least 4 links per MROD are used and in this case the throughput is entirely determined by the bandwidth of the MROD output link and the overhead of 18 + 2N clock cycles in the MRODout FPGA.

XIV. Concluding Remarks

The MROD-1 prototype has been evaluated on the basis of extensive general experience and in particular of the results of the throughput measurements. The performance tests showed that the SHARC-links between MRODin and MRODout form a bottleneck and moreover that most of or all the processing power of the DSPs may be needed for controlling data transfers, fragment building and adding of headers and trailers to the data at 100 kHz trigger rate. The fast serial FPGA interconnection technology available now offers possibilities not available at the time of the design of the MROD-1. The new MROD-X design utilizes therefore Xilinx Virtex-II Pro FPGAs [16]. The RocketIO connections of these FPGAs are superimposed on the MROD-1 design, yielding direct connections between the MRODin and MRODout FPGAs. The DSPs are left in place and are now relieved from the burden of taking care of the MROD-internal event data transfers and of fragment building. They still are necessary for control of the MROD, for error reporting and also for error handling, in so far as not implemented in the FPGAs. Their processing power may be fully exploited to check and monitor the data stream and to collect statistics.

The entire MROD-X has been implemented on a single board as a single-slot wide 9U VME64x module. The testing of the pre-production series of 6 modules is complete. The performance tests have shown that the throughput is determined almost completely by the bandwidth of the output link. The production of the MROD-X modules takes place in the first half of 2006, testing, installation and commissioning of 204 MRODs are foreseen for the second half of 2006.

ACKNOWLEDGMENT

The authors thank M. Barisonzi and R. van der Eijk for their contributions to the development of the MROD.

REFERENCES

[1] ATLAS Detector and physics performance technical design report. CERN-LHCC 99-13, The ATLAS Collaboration (1999), http://atlasinfo.cern.ch/Atlas/GROUPS/PHYSICS/TDR/access.html.
[2] The ATLAS Muon spectrometer design report. CERN-LHCC 97-022, The ATLAS Collaboration (1997), http://atlas.web.cern.ch/Atlas/GROUPS/MUON/TDR/Web/TDR.html.
[3] http://www.nikhef.nl/pub/experiments/atlas/daq/mrod/.
[4] C. Posch et al., "MDT-ASD, CMOS front-end for ATLAS MDT", ATL-MUON-2002-003, http://bmc.bu.edu/bmc/asd/asd_chip.html.
[5] Y. Arai, "AMT-3 (ATLAS Muon TDC version 3) User's Manual". Rev. 0.32. 18-05-2005, http://atlas.kek.jp/tdc/amt3/index.html.
[6] Y. Arai et al., "On-chamber readout system for the ATLAS MDT Muon Spectrometer", IEEE Trans. Nucl. Sci., vol. 51, 2004, pp. 2196-2200.
[7] P. Moreira et al., "G-Link and gigabit Ethernet compliant Serializer for LHC Data Transmission", in Proc. Nuclear Science Symp. (NSS) Lyon, France, 2000, pp. 9.6 – 9.9, and http://proj-gol.web.cern.ch/proj-gol/.
[8] CERN S-LINK, http://hsi.web.cern.ch/HSI/s-link/.
[9] ATLAS High-Level Trigger, Data-Acquisition and Controls Technical Design Report, ATLAS TDR 016, CERN/LHCC/2003-022, June 2003, http://atlas-proj-hltdaqdcs-tdr.web.cern.ch/atlas-proj-hltdaqdcs-tdr/.
[10] ROBIN, https://edms.cern.ch/document/555100/1.
[11] T.A.M. Wijnen, "The MROD data format and the tower partitioning of the MDT chambers". ATL-DAQ-2003-023, updated January 2006, http://doc.cern.ch/archive/electronic/cern/others/atlnot/Note/daq/daq-2003-023.pdf.
[12] C. Timmermans, "The effect of radiation background on the MDT data acquisition", Sept 2003, http://www.hef.ru.nl/atlas/Pub/rates2.pdf.
[13] R. Cranfield et al., "prototyping hardware for the ATLAS Readout Buffers", Proc. of the fourth workshop on Electronics for LHC Experiments, CERN/LHCC/98-36, 1998, pp. 397-401.
[14] ADSP-21160 SHARC DSPs, http://www.analog.com/en/prod/0%2C%2CADSP%252D21160N%2C00.html.
[15] APEX™ 20K FPGAs, http://www.altera.com/products/devices/apex/.
[16] Virtex™-II Pro FPGAs, http://www.xilinx.com/products/silicon_solutions/fpgas/virtex/virtex_ii_pro_fpgas/index.htm.
[17] TTC system, http://ttc.web.cern.ch/TTC/.
[18] TIM module, http://www.hep.ucl.ac.uk/atlas/sct/tim/.
[19] S. Gameiro et al., "The ROD Crate DAQ of the ATLAS Data Acquisition System", these proceedings.