# Implementation of a Serial Protocol for the Liquid Argon Calorimeters of ATLAS

F. Hubaut, B. Laforge, O. Le Dortz, D. Martin, Ph. Schwemling
*LPNHE-Paris, Universités de Paris 6&7, IN2P3-CNRS*

*Abstract*
This article presents the status of a serial protocol developed to control the front-end electronics of the ATLAS calorimetry (SPAC). After a technical description of the communication system architecture, we detail the features and performances of the designed components. A prototype of the slave ASIC in DMILL technology is expected to be delivered by mid-Nov 2000.

## 1 Introduction

The electronic modules of the front-end crates of the liquid argon calorimeters of ATLAS (front-end boards, calibration boards, tower builder and controller boards [1]) will be driven through a serial link known as SPAC (Serial Protocol for the ATLAS calorimeters). This link will be used to load, to update or to check the various registers and memories of the various front-end modules. This information is intended to be transfered before the data taking. Consequently, the serial link will be silent during data-taking in order to reduce electronic noise on the boards. A first version of the protocol, based on the $I^2C$ protocol [2], has been developed in programmable devices to drive the electronics of the module 0 front-end crate [3]. From this point, several modifications were applied to the original design in order to improve the reliability of the system, to make the protocol easily transmitted on an optical fibre, and to produce the serial slaves in ASIC.

## 2 Overview of the protocol

### 2.1 Main features

The general layout of the serial communication system is shown on figure 1 : each serial network consists in one master and multiple slaves. The SPAC master board will be situated in the readout crate, outside of the detector area, in a radiation free environment. The SPAC master is connected through four unidirectional optical links (the number of fibres is doubled to reinforce the reliability of the system) to the *controller board*, housed in the front-end crate, in the detector area. The optical signals are converted into electrical differential signals on the controller board and transfered on a *copper link* to the various front-end electronics boards, housing the serial slaves. While, in the ATLAS framework, the master boards, sitting in a radiation free environment, consist in programmable FPGA housed on VME interfaced boards, the serial slaves, located in the front end crates, will be radhard ASICs.

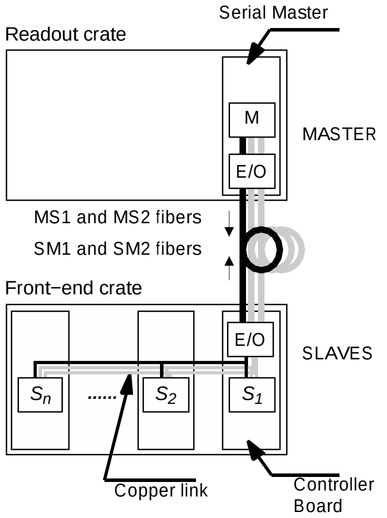

Figure 1: Architecture of the serial communication

The protocol enables point to point read-write data accesses or broadcast write data accesses to all or a part of the slaves of a network, at a raw data bit rate of 10 Megabits per second.

Each serial slave in the network is identified by a unique 7-bits address. One of these addresses is reserved for *global broadcast* accesses, and 15 others are reserved for *local broadcast* accesses. Each serial slave is also assigned a 4-bits local broadcast address that determines which local broadcast command it should consider.

The SPAC slave provides to the board it is mounted on a parallel interface to read and write, under request of the serial master, the on-board resources. These resources can be 8, 16 or 32 bits registers, or a block of memory bytes of arbitrary length, and are identified with a 7-bits *sub-address*. Several sub-addresses are reserved to access some internal resources of the SPAC slave.

The SPAC slave also provides to the board a I$^2$C master interface. I$^2$C interfaced components on the board can therefore be controlled through read or write accesses to some dedicated internal registers of the SPAC slave.

The protocol requires, at least, two unidirectional lines: one Master to Slaves line , and one Slaves to Master line. For each direction, both data and clock are encoded in Manchester type coding system. But for reliability purposes, the downstream and upstream serial lines are duplicated. At any time, only one of the two downstream lines carries the useful information, the other line remaining idle (the decision is done at the SPAC master level). Each serial slave will receive both lines and automatically detect which one is idle and which one is active. On the opposite, the upstream serial lines will carry both the same information at the same time. This scheme limits the decision logic on the remote slave side and therefore improves the reliability of the system.

## 2.2 Frame definition

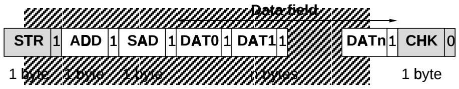

Figure 2: SPAC frame

Any SPAC frame (figure 2) circulating on the network is a sequence made of 9-bits data words.

- All the data bits are Manchester-type coded. A logical one (respectively zero) is coded by a low to high (respectively high to low) transition. The idle state is coded by a low level with no transition.
- The 9-bits data word consists in:
  - a byte of useful data, sent in *big endian* order (least significant bit first).
  - and then, a Continue/Last bit (logical zero for the last word).
- The start of a frame is indicated by a *preamble*, made of pre-defined sequence of logical zeros and ones (35 and a continue bit, see figure 3). The second aim of this start pattern is to enable the receivers on the network to recover the clock of the Manchester encoded incoming signal.
- The last word of the frame is a checksum, calculated on the whole data bytes of the frame, excluding the preamble.

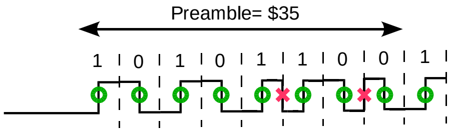

Figure 3: The preamble that indicates the start of a frame

| *Bit* | *8* | *7* | *6:0* |
|---|---|---|---|
| 1st word | 1 | Start Pattern[7:0] | |
| 2nd word | 1 | Direction | Slave Address[6:0] |
| 3rd word | 1 | R/W | Sub−Address[6:0] |
| (4th word) | 1 | Data[7:0] lowest order byte | |
| (5th word) | 1 | Data[7:0] higher order byte | |
| ... | | | |
| Last word | 0 | Checksum[7:0] | |

Table 1: SPAC frame

The table 1 summarizes the different packets of a SPAC frame. The frames produced by the master (resp. a slave) are such that the direction bit of their second word is 1 (resp. 0). The words between the third (sub-address word) and the last one (checksum word) are the data field. Its function varies according to the context:

- In a write frame produced by a master, it contains the data bytes to write into the slave boards.

- In a read request frame produced by the master, it indicates the number of data bytes to read from the target slave board (or *byte count*). In this configuration, the data field cannot contain more than two bytes, and if it is empty, the byte count is assumed to be 1.
- In an answer frame produced by a slave, in consequence of a read request produced by the master, the data field contains the data read from the board.

# 3 Implementation

## 3.1 Serial master and copper link

The serial master of the SPAC network consists in a VME module, made of two programmable FPGAs (one for the VME interface, and the second handles the function of SPAC master itself). For the time being, this module can drive one SPAC network, either in PECL or in BTL format, but will, in the near future, provide an optical link. It is also foreseen to gather in a single VME master module the functions of two independent SPAC masters.

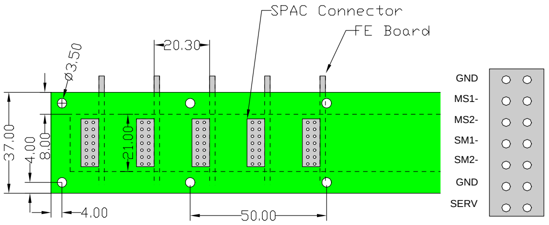

Figure 4: SPAC copper bus

The copper link that will connect the various electronic boards of the front-end crate to the controller board will consist in a solid PCB plane attached on the front part of the front-end crate (see figure 4). The connection between the PCB and the electronics boards will be achieved by inserting a traversing pin comb through the PCB and a connector on the receiving board. This design, similar to the one of the front-end crate power bus [4] enables to remove one single electronic board from the crate without needing to unwire all the connections. The signals existing on the bus are the two downstream MS1 and MS2 and the two upstream SM1 and SM2 serial lines (in differential electrical format). The bus will also carry an analogue general purpose line (SERV), intended for monitoring purposes: the users needing to monitor a given parameter on their board (temperature, voltage...) may put the analogue voltage corresponding to this parameter on this line, that can be digitized by the controller board.

## 3.2 Slave ASIC

Each front-end electronics board will house a slave ASIC to communicate with the SPAC master through the serial network. A functional diagram of the ASIC is shown on figure 5. The fundamental 40 MHz clock of the slave is provided by the TTCrx chip [5] of the board. It receives the incoming frames from the serial master on the serial inputs MS, synchronises its internal 10 MHz and 20MHz clocks to the incoming manchester-coded frame, decodes the different fields of the frames and, if required, communicates with the host board either by its parallel interface module or by its I$^2$C master interface. Finally, it encodes its reply frame and sends it to the master via the serial outputs MS. If an incoming frame is corrupted (e.g. if the ASIC has detected a wrong checksum or could not detect the preamble pattern), the ASIC generates an interrupt frame (high level without transitions during about one microsecond). This feature can be disabled or enabled through a dedicated configuration pin of the chip.

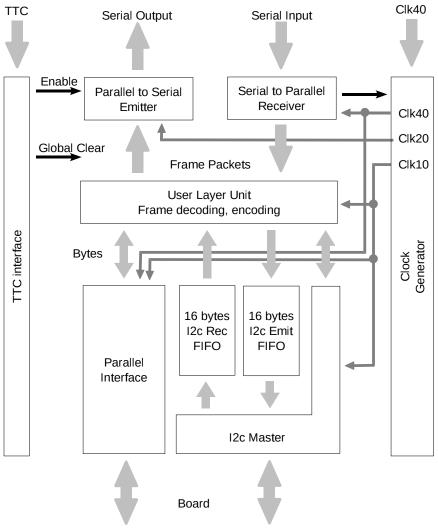

Figure 5: Functional diagram of the SPAC slave

An example of a write operation to an on-board resource through the parallel interface of the SPAC slave is shown on the chronogram (figure 6), describing the different outputs of the slave when a write frame is received from the master. More details and examples can be found in [6].

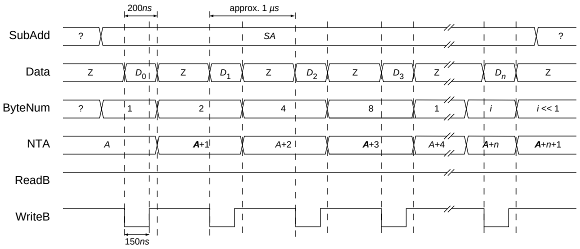

Figure 6: Chronogram of the parallel interface of the slave chip, when it receives a write frame from the master. The block of $n+1$ bytes $\{D_0, D_1, \cdots D_n\}$ is written into a memory of subaddress $SA$, at the addresses indicated by the $NTA$ value $\{A, A+1, \cdots A+n\}$.

The slave ASIC also provides to the board an I$^2$C master interface, the behaviour of which is controlled through the access to internal registers. The serial slave can then play the role of a master of two I$^2$C links on the board. Up to 15 data bytes can be read or written through SPAC, at a clock rate that is adjustable by the user from 2.5 MHz to approximately 150 kHz. The I$^2$C data bytes to be written on one of the I$^2$C output bus need first to be written through SPAC commands to an internal 16-bytes emission FIFO on the slave ASIC. Alternatively, the data bytes read from one of the I$^2$C buses are stored inside a 16-bytes reception FIFO in the slave ASIC and can then be read back though SPAC commands by the serial master.

One other feature of the SPAC slave is its ability to communicate with the TTCrx chip on the board. On one hand, the SPAC slave can configure, through I$^2$C the TTCrx. And on the other hand, the SPAC slave ASIC can be reset or inhibited by sending individual TTCrx commands to the TTCrx of the same board. Figure 7 shows an example of connections between the SPAC slave and the TTCrx chip of a board. To guarantee that the slave ASIC gets a valid 40 MHz clock from the TTCrx chip, the output TTCReady could be used as the global reset of the slave ASIC.

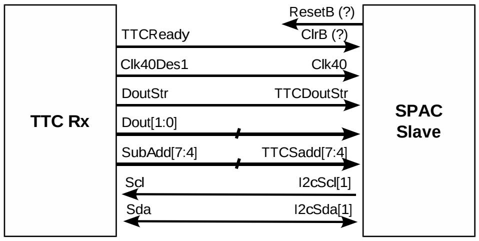

Figure 7: Example of connections between the SPAC slave and the TTCrx chip

## 3.3 Performances of the protocol

A test bench consisting of a VME master board and slaves implemented in programmable devices has been implemented and tested successfully in Paris. Care was especially taken to determine the robustness of the serial link when the master and the slaves do not have the same fundamental operation frequency. Figure 8 presents the error rate observed in the data transmission when the frequency of the master clock is varying while the slaves one remains at the nominal value of 40 MHz. This result ensures that the transmission is performed correctly when the frequencies of the master and the slaves vary respectively by less than ten percent.

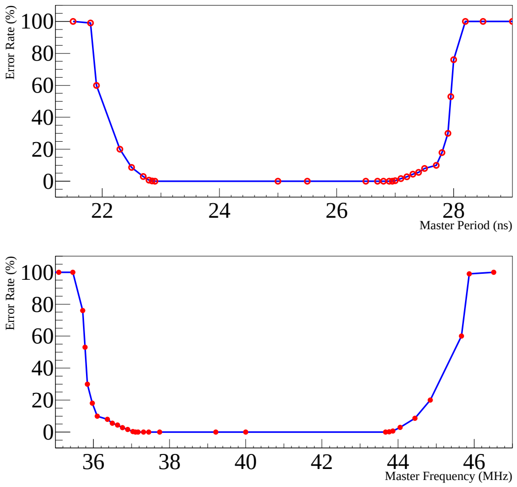

Figure 8: Data transmission error rate as a function the master clock period/frequency. The slaves used a constant 40 MHz clock.

## 3.4 DMILL prototype of the slave ASIC

A prototype of the slave ASIC was designed and submitted at the DMILL multi-project run 471 (end of May 2000).

The design, described in VHDL code, was targeted to the $0.8\mu$ DMILL process. A power-on-reset, provided by S.Vilalte (LAPP, Annecy, France) was implemented, and can be used to reset the ASIC by connecting externally its output to the global reset input of the chip. On the other hand, LVDS emitter and receiver cells, provided by F.Anghinolfi (CERN), were also implemented to enable (by external connections) to receive and emit the serial lines (MS1, MS2, SM1 and SM1) in LVDS levels.

The chip layout is shown figure 9. The size is $5.2 \times 5.2 = 27\ mm^2$, for about 30.000 transistors. The package is a CQFP-120 (0.8 pitch), for a total number of I/O pads of 96.

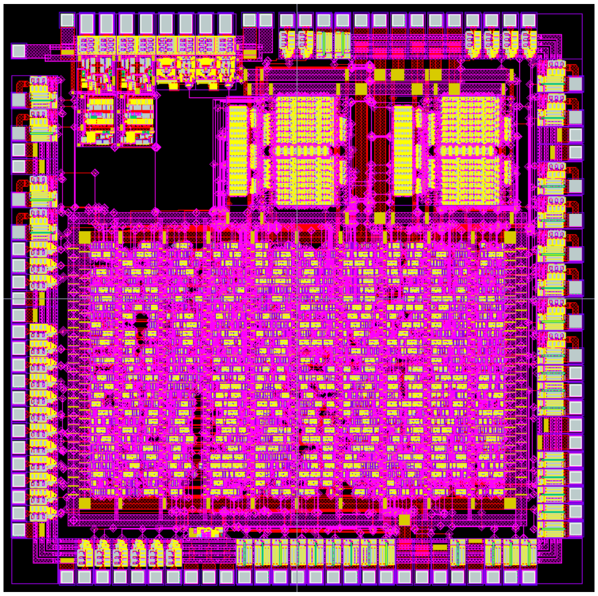

Figure 9: Layout of the SPAC slave ASIC, DMILL process, $5.2 \times 5.2 = 27\ mm^2$

# 4 Conclusion

The DMILL slave ASIC expected by mid-november 2000, will be extensively tested before going to the final production, especially the effect of the single event upsets caused by radiations. The optical fibres and transceivers also need to be validated in a radiated environment. Some more tests on the reliability of the protocol and of the electrical bus is also underway.

# References

[1] Technical Design Report on Calorimeter Performance, ATLAS Collaboration, 1997

[2] The $I^2$C-Bus Specification, version 2.0, Philips Semiconductors, December 1998 http://www-us.semiconductors.philips.com/acrobat/various/I2C_BUS_SPECIFICATION_2.pdf

[3] ATL-LARG-98-093, SPAC : Serial Protocol for the Atlas Calorimeter, R. Bernier, D. Breton, Ph. Cros, A. Gara, 24 March 1998

[4] ATL-AL-EN-0004, Liquid Argon Front-End Crate, H. Takai et al.

[5] TTCrx Reference Manual, version 3.0, J.Christiansen, A.Marchioro, P.Moreira and T.Toifl, Oct 99

[6] ATL-AL-ES-005, Specifications of Serial Protocol for the ATLAS Calorimeters, 0. Ledortz, D. Martin, Nov. 99