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

F. Hubaut, B. Laforge, O. Le Dortz, D. Martin, P. Schwemling

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

F. Hubaut, B. Laforge, O. Le Dortz, D. Martin, and P. Schwemling

***Abstract*—This paper presents the status of a serial protocol developed in the framework of ATLAS (A Toroidal LHC ApparatuS), a high energy physics experiment that will be installed at the Large Hadron Collider (LHC) based at CERN. This link will be used to control the front-end electronics of the liquid argon calorimeters, located in an irradiated environment. After a technical description of the communication system architecture, we detail the features and performances of the designed components. This protocol could be applied in other contexts (nuclear physics, industry) where a fast, reliable and radhard serial communication is needed.**

***Index Terms*—ATLAS, control, radiation tolerance, serial communication.**

I. Introduction

THE ATLAS (A Toroidal LHC ApparatuS) experiment [1] is to be carried out at the Large Hadron Collider (LHC), a particle accelerator facility of the CERN Laboratory in Geneva, Switzerland. The detector, the overall layout of which is shown in Fig. 1, is due to begin operation in the year 2005. 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 [2]) will be driven through a serial link known as SPAC (Serial Protocol for the ATLAS Calorimeters).

The SPAC will be used to load, to update or to check the various registers and memories of the various front-end modules. These modules are located in an irradiated environment (20 Gray/year and 10^{12} neutrons/cm^2/year) [3]. The information is intended to be transferred before the data taking. Consequently, the serial link will be silent during data-taking in order to reduce electronic noise on the boards.

II. Overview of the Protocol

A. Main Features

The general layout of the serial communication system is shown in Fig. 2. Each serial network consists of one master and multiple slaves. There will be about 115 independent networks for the whole experiment, with one master and 15 slaves each.

The SPAC master board will be situated in the readout crate, outside of the detector area, in a radiation-free environment. It is connected through four unidirectional optical links (the number of fibers is doubled to reinforce the reliability of the system)

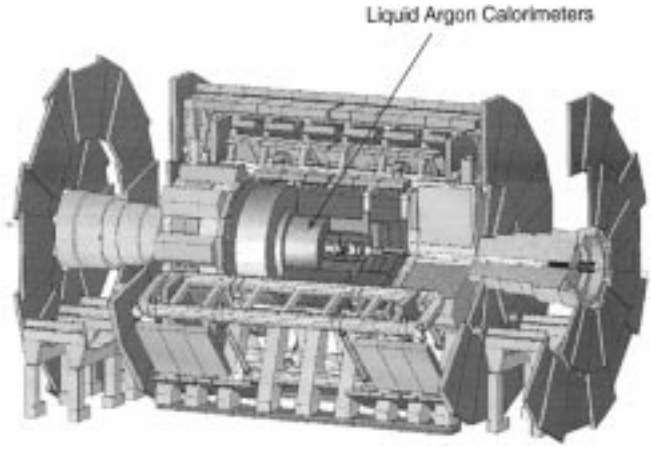

Fig. 1. Overall layout of the ATLAS detector. The outer diameter is roughly 20 m.

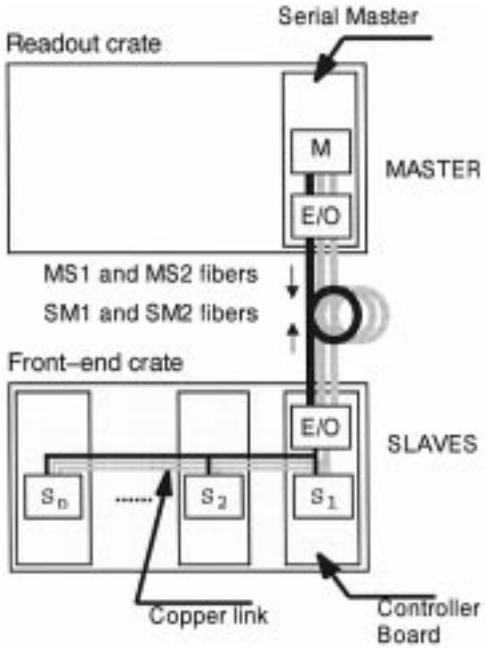

Fig. 2. Architecture of the serial communication system.

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 transferred on a *copper link* to the various front-end electronics boards, housing the serial slaves. While the master boards, sitting in a radiation-free environment, consist of programmable logical devices housed on VME interfaced boards, the serial slaves, located in the front end crates, have to be radhard ASICs.

Manuscript received October 25, 2000; revised January 22, 2001.

The authors are with LPNHE-Paris, Universités de Paris 6&7, IN2P3-CNRS, 4 place Jussieu, Tour 33 RdC, F-75252 Paris Cedex 05, France (e-mail: hubaut@in2p3.fr; ledortz@in2p3.fr).

Publisher Item Identifier S 0018-9499(01)07032-0.

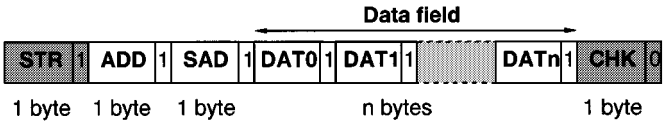

Fig. 3. Sketch of a typical SPAC frame, made of 9-bits data words.

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/s. 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^2C master interface [4]. I^2C 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 combined to form a single signal in the same way as in the Manchester coding system: a logical one (respectively, zero) is coded by a low to high (respectively, high to low) transition, whereas the idle state is coded by a low level with no transition. 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.

B. Frame Definition

Any SPAC frame circulating on the network is a sequence made of 9-bits data words, as sketched in Fig. 3.

- All the data bits are Manchester encoded.
- The 9-bits data word consists of:
  - — a byte of useful data, sent in *big endian* order (least significant bit first);
  - — a `Continue/Last` bit (logical zero for the last word).
- The start (STR) of a frame is indicated by a *preamble*, made of a pre-defined sequence of logical zeros and ones ($35 and a continue bit, see Fig. 4). The second aim of this start pattern is to enable the receivers on the network to recover and lock on the clock of the Manchester encoded incoming signal.

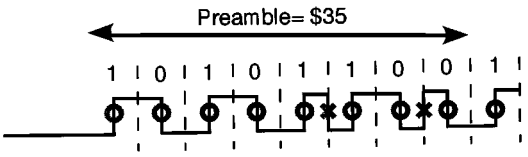

Fig. 4. Predefined sequence indicating the start of a frame and called preamble (noted STR in the other figures).

TABLE I
MEANING OF THE DIFFERENT PACKETS OF A TYPICAL SPAC 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]	

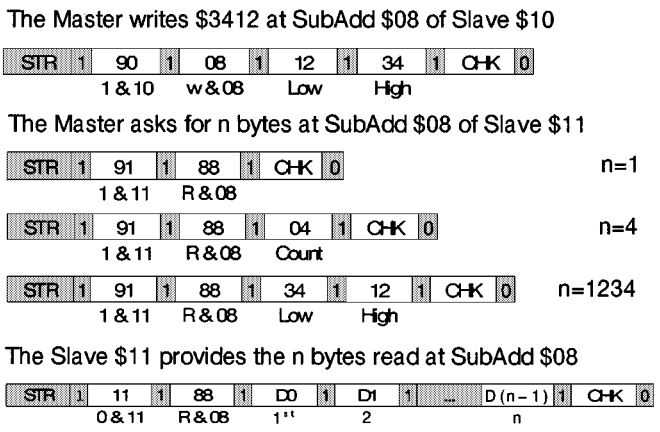

Fig. 5. Examples of frames circulating in the network in case of a read or write master request and a typical answer of a slave.

- The last word of the frame is a checksum (CHK), calculated on the whole data bytes of the frame, excluding the preamble.

Table I 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.

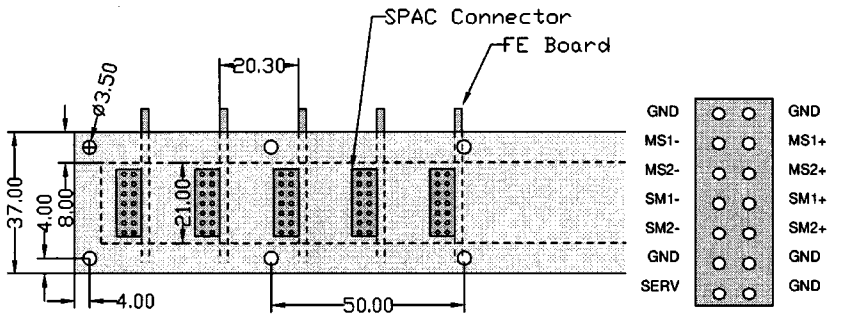

Fig. 6. Schematics of the SPAC copper bus.

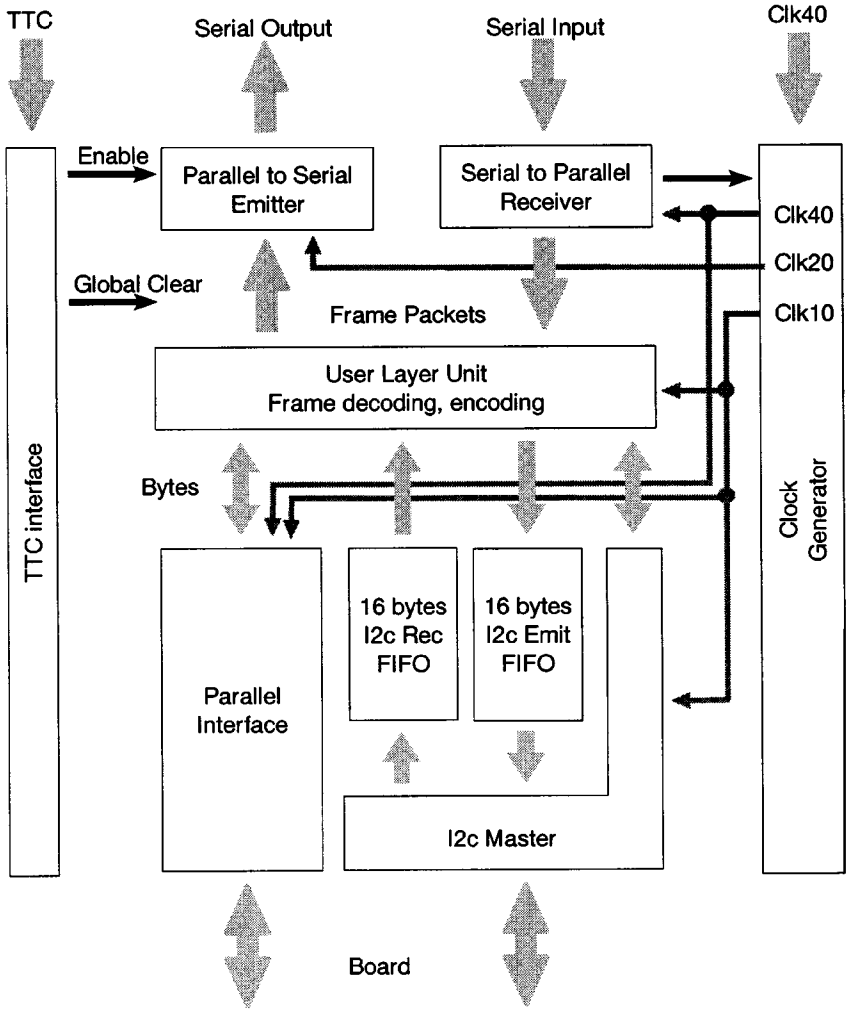

Fig. 7. Functional diagram of the SPAC slave.

Some examples of frames that can be sent in case of a read or write request produced by the master and the corresponding answer of the slave are shown in Fig. 5.

III. Implementation

A. Serial Master and Copper Link

The serial master of the SPAC network consists of a VME module, made of two programmable logical devices (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.

The copper link that will connect the various electronic boards of the front-end crate to the controller board will consist of a solid PCB plane attached on the front part of the front-end crate (see Fig. 6). 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 enables to remove one single electronic board from

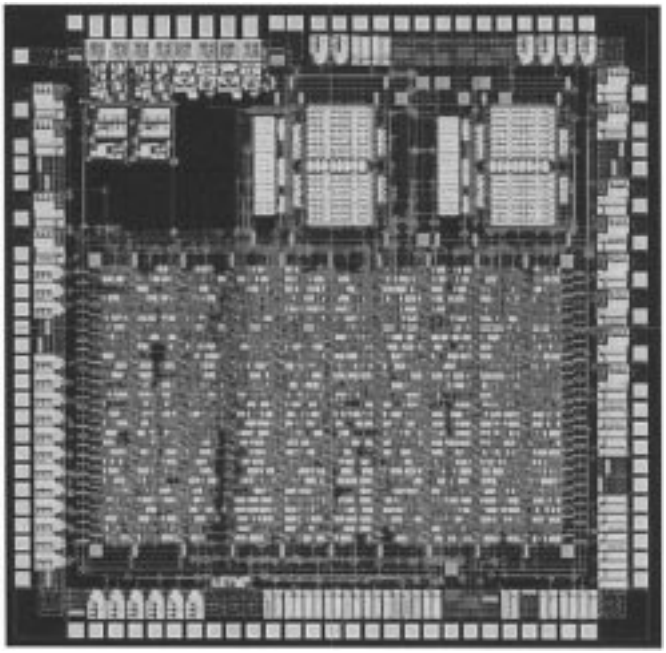

Fig. 8. Layout of the SPAC slave ASIC, DMILL process, with a size of $5.2 \times 5.2 = 27$ mm^2.

the crate without needing to unwire all the connections, as the boards are extracted from the top of the crate.

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 analog general purpose line (SERV), intended for monitoring purposes: the users needing to monitor a given parameter on their board (temperature, voltage, ...) may put the analog voltage corresponding to this parameter on this line, that can be digitized by the controller board.

B. 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 Fig. 7. The fundamental 40-MHz clock of the slave is provided by another chip of the board (called TTCrx [5]). The slave receives the incoming frames from the serial master on the serial inputs MS, synchronizes its internal 10- and 20-MHz clocks to the incoming Manchester encoded 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.

The slave ASIC also provides to the board an I^{2}C master interface, the behavior 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.

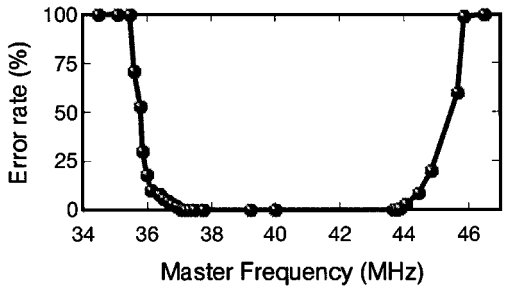

Fig. 9. Data transmission error rate as a function of the master clock frequency. The slaves use a constant 40-MHz clock.

The serial slaves, located in the front end crates, have to be radhard ASICs. A prototype of the slave ASIC was designed and submitted at the DMILL [6] multiproject run 471 (end of May 2000). The design, described in VHDL code, was targeted to the radiation hardened 0.8-μm DMILL process. A power-on-reset 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 transceiver cells were also implemented to enable (by external connections) to send and receive the serial lines (MS1, MS2, SM1 and SM1) in LVDS levels.

The chip layout is shown in Fig. 8. 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.

C. Performances of the Protocol

A test bench consisted of a VME master board and slaves implemented in programmable devices has been implemented and tested successfully. 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 frequencies. Fig. 9 presents the error rate observed in the data transmission when the master clock frequency is varying while the slaves ones remain at the nominal value of 40 MHz.

Fig. 10 presents the region without any error observed in the data transmission when the frequency and the duty cycle of the slave clock are varying while the master clock remains at the nominal value of 40 MHz.

These results show that the protocol is very robust and reliable as the transmission is performed correctly even when the frequencies of the master and the slaves vary by ten percents from each other or when the duty cycle is not perfect.

As a demonstrative example, we present also a comparison between the timing performance of the serial link to read/write data from/to a memory component on the board using either the I^{2}C or the parallel interface of the SPAC slave. In this example, we used the I^{2}C with a clock rate of 2.5 MHz. Table II shows the

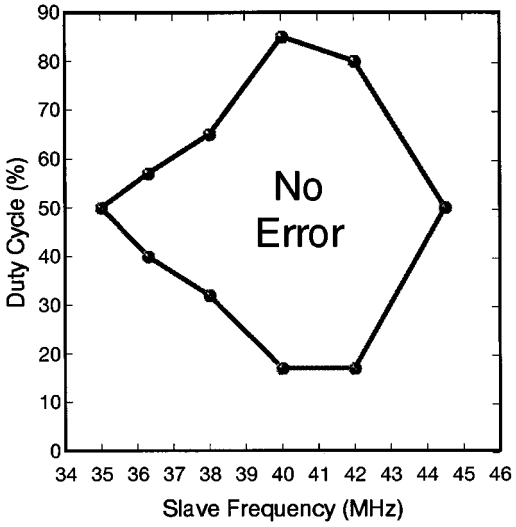

Fig. 10. Region without any transmission error when varying the frequency and the duty cycle of the slave clock. The master uses a constant 40-MHz clock.

TABLE II
COMPARISON BETWEEN THE I^2C AND THE PARALLEL INTERFACE TIMING PERFORMANCE

Data length	Parallel interface	I^2C at 2.5MHz	Ratio
Write Operation			
1 byte	4.5μs	18.0μs	4
2 bytes	5.4μs	22.5μs	4.2
4 bytes	7.2μs	31.5μs	4.4
15 bytes	17.1μs	81.0μs	4.7
Read Operation			
1 byte	9.0μs	27.0μs	3
2 bytes	10.8μs	33.3μs	3.1
4 bytes	12.6μs	44.1μs	3.5
15 bytes	22.5μs	103.5μs	4.6

different timing obtained to either read or write data blocks of different lengths. This shows that the parallel interface is much faster than the I^2C one.

IV. CONCLUSION

The SPAC protocol that will be used to control the front-end electronics of the liquid argon calorimeters of the ATLAS detector will provide a fast, reliable and radhard serial communication. It could be applied in other contexts (nuclear physics, industry) where such a communication is needed.

The DMILL slave ASIC expected by mid November 2000, will be extensively tested before going to the final production. The effect of the single event upsets caused by radiations will be especially studied. The optical fibers and transceivers also need to be validated in an irradiated environment. Some more tests on the reliability of the protocol and of the electrical bus are also underway.

ACKNOWLEDGMENT

The authors would like to thank S. Vilalte (LAPP) and F. Anghinolfi (CERN) for providing the power-on-reset and the LVDS transceiver cells, as well as E. Dupont-Nivet (CEA) for his collaboration.

REFERENCES

[1] The ATLAS Collaboration, "ATLAS: Technical proposal for a general-purpose pp experiment at the large hadron collider at CERN", Dec. 1994.
[2] ——, "ATLAS liquid argon calorimeter technical design report", Dec. 1996.
[3] P. Farthouat and H. Williams, "ATLAS policy on radiation tolerant electronics," ATLAS public note , ATL-ELEC-98-003. unpublished.
[4] Philips. (1998, Dec.) The I^2C-bus specification, version 2.0. Philips Semiconductors. [Online]. Available: http://www-us.semiconductors.com/i2c/facts/
[5] J. Christiansen, A. Marchioro, P. Moreira, and A. Sancho, "Receiver ASIC for timing, trigger and control distribution in LHC experiments," *IEEE Trans. Nucl. Sci.*, vol. 43, pp. 1773–1777, June 1996.
[6] M. Dentan, P. Abbon, P. Borgeaud, R. Delagnes, N. Fourches, and D. Lachartre, *et al.*, "DMILL, a mixed analog–digital radiation-hard BICMOS technology for high energy physics electronics," *IEEE Trans. Nucl. Sci.*, vol. 43, pp. 1763–1767, June 1996.