A Simple FPGA/PLL Based Protocol Converter for Serial Data Transmission

Boris Radin[1], Pavle Savković[2], Dušan Majstorović[3], Branislav Atlagić[4], Saša Vukosavljev[5]

Abstract – **This paper describes a design of simple protocol converter based on FPGA and PLL circuitry used for serial data transmission between data acquisition unit and solid state data recorder. Such converters are often employed in telemetry and data acquisition systems when communication link must be established between components running different communication protocols. Main aspects of this design are simplicity and reliability.**

Keywords – **FPGA, PLL, synchronizer, protocol converter**

I. INTRODUCTION

Typical data acquisition or telemetry system contains a number of transducers used for measurement of various process parameters. Transducers are usually connected to an acquisition unit which outputs a stream of measured data properly formatted according to standard applied. Data can be processed/stored locally, or transmitted to a remote location for further processing. Sometimes there is a need for equipment replacement, for instance obsolete tape recorders with new solid state models. New equipment is rarely compatible with existing solutions, therefore some sort of interface equipment (operating as protocol converters) are required. The protocol converter described in this paper was designed in order to provide direct connection between data acquisition unit running subset of IRIG-106 protocol, and solid state data recorder compatible with ARINC 573/717, Fig. 1.

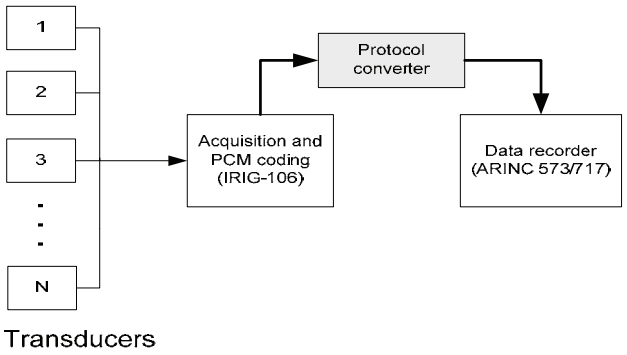

Fig. 1. The protocol converter in data acquisition system

[1, 2, 3, 4,5]All authors are with Faculty of technical sciences, Chair for Computer Engineering and Communications, Fruškogorska 11a, 21000 Novi Sad, Serbia
E-mail: Boris.Radin@krt.neobee.net
E-mail: Pavle.Savkovic@krt.neobee.net
E-mail: Dusan.Majstorovic@krt.neobee.net
E-mail: Branislav.Atlagic@krt.neobee.net
E-mail: Sasa.Vukosavljev@krt.neobee.net

II. SYSTEM STRUCTURE

Output from the acquisition unit is uniform synchronous data flow consisting of 13-bit words (12-bit words with additional parity bit). On the physical level this was realized with 3 digital signals having TTL levels – WS (Word Strobe), CLK (Clock) and input data, as depicted on Fig. 2. Words were packed in subframes and frames according to IRIG-106 standard, but this issue was not relevant since data recorder in question operated as plane bit-recorder. Also, the software used to retrieve recorded data took advantage of the frame structure.

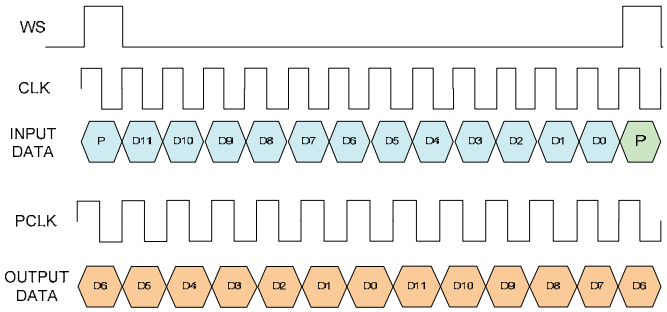

Fig. 2. Relation between input and output bit stream

The data recorder required 12-bit words with Harvard Bi-Phase coding and no parity information, Fig. 3. Both units could operate on various speeds (words per second, or WPS), but only three were found to be relevant for the application – 1024 WPS, 2048 WPS and 4096 WPS. Therefore, the main task for the converter was to accept 13-bit words via 3-wire PCM interface, remove parity bit and retransmit the data in Harvard Bi-Phase code maintaining constant word-rate.

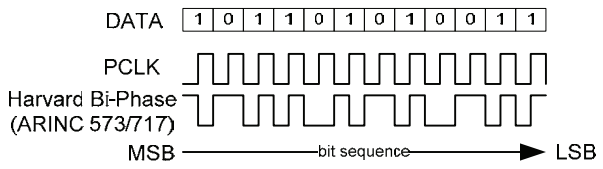

Fig. 3. Harvard Bi-Phase coding example

Protocol conversion based on words, not on the frames and subframes, brought in simplicity and generality. Besides, it was required to design the converter to be the most compact in order to fit the environment. Therefore the decision was made to base the design upon single FPGA with small number of additional integrated circuits (buffers and level translators).

In order to verify proper operation of data recorder, a special test mode of operation was introduced. In this mode no acquisition unit is required since protocol converter itself acts as a data source, Fig. 4. A predefined data pattern containing one full frame could be loaded into the internal FPGA memory, and repeatedly transmitted towards the data recorder at various word-rates. Also, optional pattern generator based upon counters could be used instead. Both methods allowed large amounts of known data to be recorded and subsequently retrieved for validation. Large data blocks up to 10^9 bits were written and read out with no errors detected. As a timing reference, an external crystal-controlled oscillator was used. Standard operating frequency of 6144kHz was chosen since it is integer multiplier of highest bit-rate in use (12*4096 Hz).

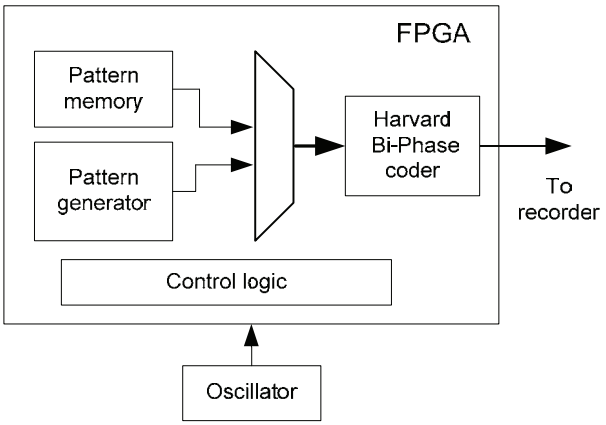

Fig. 4. The converter in test-mode

In full mode of operation the converter accepts serial data from acquisition unit in a synchronous manner, using incoming clock (receiver block on Fig. 5). Input data is sampled on falling edge of the clock signal, Fig. 2, which is positioned in the middle of every bit-interval thus providing very reliable operation. Individual bits are packed together and 12-bit words are clocked into resynchronization network at the word boundary (falling edge of WS signal, Fig. 2).

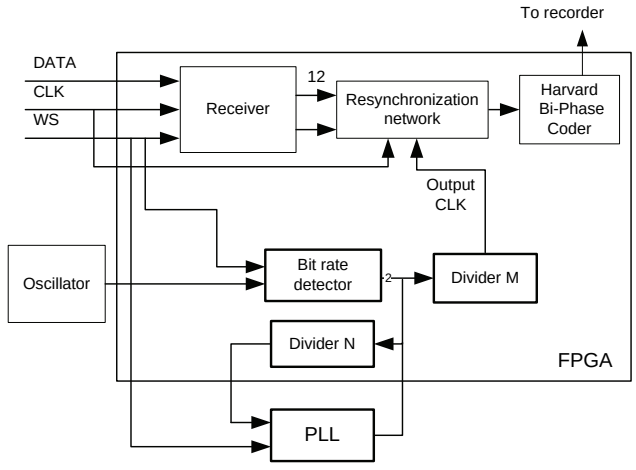

Fig. 5. The converter in full operation

The task of resynchronization network is to provide data integrity while crossing the boundary between two clock domains – input clock and output clock (CLK and PCLK on Fig. 2 respectively). It contains 12-bit register to hold the data word through whole word-period until it is acquired by 12-bit parallel-to-serial shift register and shifted out serially by PCLK. Besides, it contains a sequence of thirteen edge-triggered flip-flops and additional logic used for synchronization and verification. Any underrun or overrun error is detected and flagged-out to the user.

Output data stream is controlled by PCLK signal generated by a external PLL (Fig. 5). It multiplies the frequency of it's reference input (signal WS) by 12 thus providing effective 12/13 ratio compared to input CLK signal. In order to narrow the output frequency range of the PLL with changes of input WS frequency from 1024Hz to 4096Hz, a pair of digital dividers was used (dividers N and M in Fig. 5). Their values (12/24/48 for N and 1/2/4 for M) are chosen according to incoming bit-rate. Thus constant PCLK frequency of 638976 Hz is maintained for all three input frequencies, providing low-jitter clock source with correct frequency, locked to the input.

The method used to provide reliable boundary crossing between clock domains was delay. With each new word clocked into resynchronization network a single bit '1' is propagated through flip-flop chain by CLK. Output of the fourth flip-flop in chain is resynchronized to PCLK with simple double flip-flop synchronizer and used to transfer the word from internal register to output parallel/serial shift register, Fig. 2. This transfer takes place near the middle of the word period, thus making the transmission immune to jitter accumulated on PCLK signal. Synchronization sequence described above occurs only once, when first word enters resynchronization network. After that, every boundary crossing is periodic and controlled by dividers clocked by PCLK, making proper operation dependant upon PLL locking.

III. CONCLUSION

The fully assembled unit was subject of thorough examination. With consistent input data stream an error-free operation was obtained with data blocks exceeding 10^9 bits. Fully digital design and careful synchronization techniques allowed excellent performance with data buffer equal to single word. The only weak point in the design is PLL that must be properly shielded against external electromagnetic fields.

REFERENCES

[1] Mike Stein "Crossing the abyss: asynchronous signals in a synchronous world", *EDN designfeature*, July 2003

[2] Cadence technical paper, "Clock domain crossing - closing the loop on clock domain functional implementation problems", *Cadence Design Systems, Inc*., 2004

[3] Octavian Petre, Hans G. Kerkhoff, " Increment the Fault Coverage in Multiple Clock Domain Systems by Using On-Line Testing of Synchronizers", *Proceedings of the Seventh IEEE International On-Line Testing Workshop (IOLTW'01)*, 2001