

Fast FPGA-based Serial Receiver Design

Ondřej Urban, Vjačeslav Georgiev, Jan Zich

Abstract — This paper describes a fast serial digital signal receiver for applications in nuclear instrumentation. The proposed design uses a Microsemi Polarfire FPGA embedded Ethernet transceiver for data oversampling (with frequency up to 12.7 GHz) and deserialization. The subsequent FPGA implemented digital signal processing chain then analyses the oversampled data array (at least 4 samples per data bit are required by the processing logic). This processing chain begins with a frame buffer, which ensures that the entire sampled data frame can be captured and a 5-bit majority parallel filter. Following start sequence detection logic uses a comparator array for valid data triggering and data offset evaluation. These information are then used by the sampling point selection logic for data restoration. Thanks to the single clock cycle operation of each of these logic blocks, the processing chain provides a constant propagation delay and no dead time is required between individual data frames. The device prototype based on this design is described and measurement results for a data bit rate of 400 MHz and a sampling rate of 3.2 GHz are presented.

Keywords — *Serial communication, FPGA, transceiver, Ethernet, data reconstruction, NIM*

I. INTRODUCTION

THE applications in the field of nuclear instrumentation often require high speed communication, especially tailored for a specific experiment. One of such application is for example data transmission between trigger module and a data acquisition system in experiments like AFP ToF [1], [2]. In this particular possible application, the specific requirements are for example a fixed 40 MHz transmission period, 400 MHz bit rate, specific frame format, or a NIM [3] compatible physical-level layer.

The traditional serial communication standards are not suitable for such applications, due to the relatively low bit rates, of up to tens of kbit/s for single ended transmission lines (e.g. 20 kbit/s for RS-232 [4]) or units of Mbit/s for differential lines such as RS-485. [5], or a CAN bus [6]. These standards also do not provide a NIM compliant physical layer.

Proposed solution uses signal restoration circuitry compliant with the NIM standard. Digital oversampling required to successfully receive single ended serial data is

This research was supported by the projects Engineering Applications of Microworld Physics (CZ.02.1.01/0.0/0.0/16_019/0000766), Research Infrastructure for CERN experiments (LM2015058) and Microworld Research in the frame of CERN infrastructure (LTT17018).

Corresponding author Ondřej Urban is with the Faculty of Electrical Engineering, University of West Bohemia, Univerzitni 22, 306 14 Pilsen, Czech Republic (e-mail: urbano@fel.zcu.cz). Co-authors ¹Vjačeslav Georgiev, and ²Jan Zich are with the Faculty of Electrical Engineering, University of West Bohemia, Pilsen (e-mail addresses: ¹georg@fel.zcu.cz, ²zichj@fel.zcu.cz).

achieved by the transceiver embedded within the Polarfire FPGA [7]. The subsequent processing chain (filtering, frame start trigger, and data restoration) has been implemented in the FPGA fabric using the vhdl language. The device realization requires minimum external circuitry and is thus easily adaptable for an exact application by modifying the FPGA design only.

II. TOPOLOGY OVERVIEW

The block diagram of the prototype device is shown in the figure 1. It can be seen that the only external components to the FPGA are the circuitry for the voltage level adaptation and the USB 2.0 interface for the received data readout.

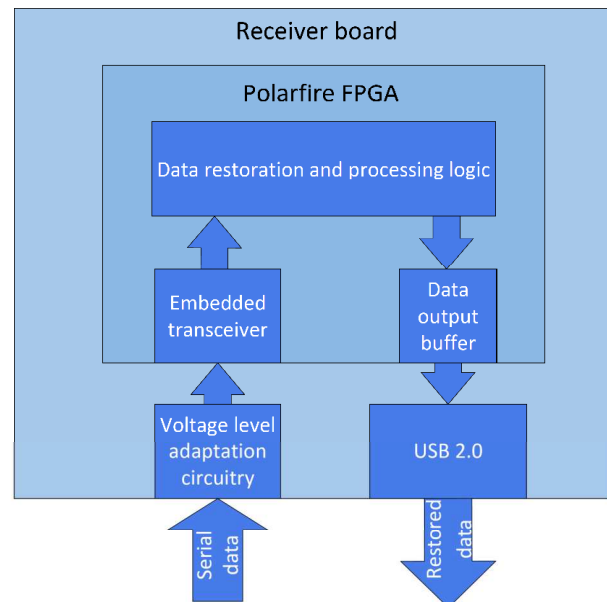


Fig. 1: Block diagram of the receiver

Each of the data processing chain blocks shown in the figure 1 is described in the following chapters.

A. Physical layer

Since the applications are targeted especially at the field of physical instrumentation, the receive circuitry has been designed in compliance with the NIM (Nuclear Instrumentation) standard [3]. The NIM standard uses a current based logic, where the logical “1” bit is represented by driving -16 mA to the 50 Ω load (i.e. -800 mV), and logical “0” by 0 V. In order to achieve the best performance for an exact application, the input stage of the receiver comprises a fast and an adjustable comparator. The output of the comparator provide suitable voltage levels for the FPGA transceiver and therefore can be

B. Frame start detection

Once the eventual errors are corrected, the frame is analyzed in order to get the frame start value. This is done by comparing the bit vector segments with a particular frame start pattern, e.g.: number of successive logical “0” followed by a logical “1”.

In order to achieve a single-clock evaluation of the pattern, the comparators are designed as a parallel structure for each offset of the vector. The position of the start frame pattern within the vector is then given by a positive output of a corresponding comparator. This then enables the following logic to offset the sampled bit vector and select appropriate sampling points.

C. Data reconstruction logic

The data reconstruction logic unit shifts the sampled bit vector according to the information from the “Frame start detection” unit, so that only the region of the expected data bits is used for the further evaluation.

The reconstruction of the data from the shifted sampled vector is complicated by the fact, that no receive clock reference is available. Therefore the rough ratio between the signal bit rate and sampling frequency has to be known. The design uses this ratio to estimate the distances between the expected data bits. Due to the asynchronism of the sampling clock and the bit clock and possible signal distortion, this bit duration estimation is never exact. The design therefore comprises an edge detection logic, which is used to select only the sampling points which are not close to the sampled signal transition.

The implemented edge avoidance logic leads to the fact that certain minimal amount of samples (in this case 3 samples) of each bit is required for the frame acceptance.

D. Implementation results

The described design has been implemented in an FPGA with the transceiver sampling frequency set to 3.2 GHz and the deserialization ratio 1:64 bits. This results in a required processing frequency of 50 MHz (i.e. the frequency at which SERDES produces a 64 bit sampled data). The implementation uses following resources.

TABLE 1: RESOURCES USED BY THE PROCESSING CHAIN COMPONENTS.

Component	LUTs
Majority filter	182
Frame start trigger	659
Data reconstruction	215

These relatively large numbers of used LUTs are caused by a parallel implementation of the processing, which is required for fast data reception without any dead time between frames. The amount of required LUTs also depends on an oversampling ratio (greater oversampling results in larger bit vectors required for processing).

The design is however small enough to be implemented in most low-cost FPGAs with the capability of bypassing the PMA part of the transceiver (in order to bypass the comma detection and decoding).

IV. MEASUREMENT

For the measurement, a specific serial frame format has been chosen. Similar frame format is used in one of the possible applications of the proposed device. The frame is shown in the figure 4 below. It is 10 bit long, and consists of 5 bits of bus idle, 1 start bit and 4 data bits.



Fig. 4: Frame format of the tested serial pattern.

In order to test the proposed design, two series of measurements were carried out. The first measurement targeted at static data consistency evaluation, while the other measurement targeted at dynamic data evaluation.

The measurements were performed for a bit rate of 400 MHz, with the data frame repeat frequency of 12.5 MHz (static test) and 3.2 MHz (dynamic test). The tested design used a sampling frequency of 3.2 GHz, providing approximately 8 sampling points per bit.

The sampling frequency of 3.2 GHz has been chosen as a compromise between sufficiently large amount of samples taken per one signal bit duration, and design complexity (e.g. for 10 GHz sampling frequency 250 bit long buffer is needed to fit the entire frame, and due to the parallel processing structure the FPGA resources usage increase considerably).

A. The static data test

The first test was performed using a digital generator, driving the input electronics with a fixed pattern every period. I.e. the same pattern has been transmitted with the period of 12.5 MHz and the bit rate of 400 MHz.

The test design used a sampling frequency of 3.2 GHz and the output data were buffered in a large 32 kB FIFO. The buffered data were then outputted via the USB 2.0 interface and stored in a PC. In order to prevent the FIFO overflow the device stopped the data sampling once the buffer has been filled.

The measurement has been carried out repeatedly for over a 10^5 data frames in total, and for all possible pattern combinations (in this case 16 combinations).

The measurement has been evaluated using a script in order to find any values other than expected. The measurement has not shown any errors values, suggesting that the design operates as expected.

B. The dynamic data test

In the next phase of testing, attention was given to the capability to successfully receive each data frame. For this test an up-counter has been implemented using another FPGA device and connected to the receiver device. The data bit rate has been set the same as in the previous test, i.e. 400 MHz. The repeat frequency has been set to approximately 3.2 MHz.

In order to test the capabilities of the receiver more

thoroughly, the data frame generator was designed to be able to distort the data bits. This has been achieved by increasing or reducing the duration of each bit, while preserving the total frame duration. This corresponds with the expected signal distortion in real applications. The test setup diagram is presented in the picture below.

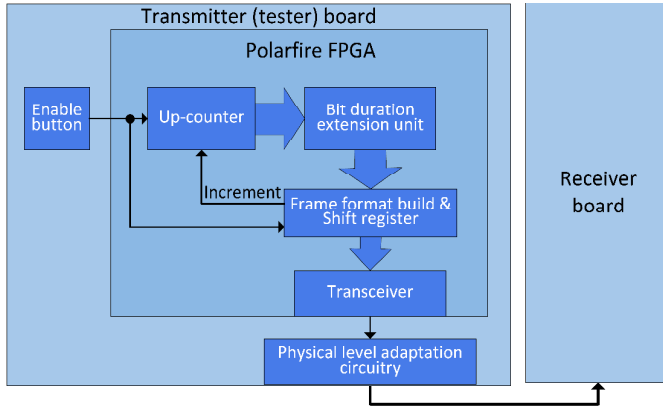


Fig. 5: Block diagram of the transmitter logic for the receiver test.

The measurement has been carried out for various distortions of both logical 1 and 0 bits (in order to preserve the total frame duration, an increase in duration of one logical value leads to decrease in duration of the opposite logical bit value). Each measurement was done for approximately 10^5 data frames. The results are presented in the tables below.

TABLE 1: MEASUREMENT RESULTS FOR BIT "0" DISTORTION WITHIN THE SAMPLED SIGNAL

Bit "0" duration decrease:	25,0%	50,0%	62,5%
Errors percentage:	0,0%	0,0%	38,2%

TABLE 3: MEASUREMENT RESULTS FOR BIT "1" DISTORTION WITHIN THE SAMPLED SIGNAL

Bit "1" duration decrease:	25,0%	50,0%	62,5%
Errors percentage:	0,0%	0,0%	53,8%

It can be observed, that the receiver prototype is able to recognize the patterns successfully up to bit duration distortion around 50% of the original bit time. After that we see a sharp rise in error count. This is caused by the filter implementation, and the sampling point selection logic, which in this case rejects all bits sampled with less than 3 samples.

It can also be observed, that the reduction in the logical "0" bit duration affects the signal restoration capability of

the device much less than in the case of the logical "1" bit.

The further analysis of the errors has also shown that only around 13% of the detected errors are caused by "missed" data frame (where the frame is not accepted at all). The rest of the errors were caused by wrong data restoration, caused likely by inaccurate frame start offset evaluation.

V. CONCLUSION

The proposed serial receiver design intended for nuclear instrumentation applications uses an FPGA embedded transceiver to oversample data at frequencies up to 12.7 MHz, and a following processing logic, tailored for any user-defined frame format, restores the received data. The processing chain blocks use a pipelined architecture, which allows data reception without any necessary dead time between frames.

The design has been successfully implemented in a prototype board with the sampling frequency set to 3.2 GHz and tested by a user defined serial communication at the 400 MHz bit rate. The results presented in this paper prove that the proposed solution provides good reliability even in case of considerable signal distortion caused by e.g. long transmission lines.

REFERENCES

- [1] S. Grinstein on behalf of AFP Collaboration, "The ATLAS Forward Proton Detector (AFP)," in *Nuclear and Particle Physics Proceedings*, Science Direct, vol. 273-275, April-June 2016, pp. 1180–1184.
- [2] J. Zich, V. Georgiev, M. Holik, V. Pavlicek, and O. Vavroch, "Multichannel coincidence circuit with settable threshold level for TOF AFP Detector," *2019 27th Telecommunications Forum (TELFOR)*, 2019.
- [3] Costrell, Louis, Lenkszus, Frank R., Rudnick, Stanley J., Davey, Eric, Gould, John, Rankowitz, Seymour, Sims, William P., Whitney, R. Roy, Dobinson, Robert W., Verweij, Henk, Latner, Norman, Negro, Vincent C., Barsotti, Edward J., Droegge, Thomas E., Kerns, Cordon, Turner, Kathleen J., Downing, Robert W., Kirsten, Frederick A., Larsh, A. E., Loken, Stewart C., Mack, Dick A., Wagner, Lee J., Lucena, Robert C., O'Brien, Dennis W., Gjovig, Allan, Naivar, Frank, Nelson, Ronald O., White, D. Hywell, Akerlof, Carl, Stilwell, Donald E., Trainor, James H., Gobbi, Bruno, Biggerstaff, John A., Hill, Nat W., Schulze, Gerald K., Gustavson, David B., Horelick, Dale, Kunz, Paul F., Paffrath, Leo, Walz, Helmut V., Dawson, W. Kenneth, Cresswell, John, Dhawan, Satish, and Gingell, Charles E. L. Standard NIM Instrumentation System. United States: N. p., 1990. Web. doi:10.2172/7120327.
- [4] "Fundamentals of RS-232 Serial Communication", 29-Mar-2001. [Online]. Available: <https://www.maximintegrated.com/en/design/technical-documents/tutorials/8/83.html>. [Accessed: 10-Oct-2021].
- [5] T. Kugelstadt, "The RS-485 design guide," *The RS-485 Design Guide*, 2021. [Online]. Available: <https://www.ti.com/lit/an/slla272d/slla272d.pdf>. [Accessed: 10-Oct-2021].
- [6] International Organization for Standardization. ISO 11898-1:2015. Geneva : International Organization for Standardization, 2015.
- [7] "PolarFire® FPGA and PolarFire SOC FPGA transceiver", 2019. [Online]. Available: <https://onlinedocs.microchip.com/pr/GUID-AE985A83-E181-4941-B39D-055A6746BB07-en-US-1/index.html>. [Accessed: 10-Oct-2021].