ACES 2011

Fixed-Latency, Multi-Gigabit Serial Links with FPGAs

www.unina.it www.infn.it

Alberto Aloisio[1], Raffaele Giordano[1] and Vincenzo Izzo[2]

[1] Università di Napoli "Federico II", Physics Dept. and INFN – Napoli, I-80126 Napoli, Italy

[2] INFN – Napoli, I-80126 Napoli, Italy

Abstract

Most of the off-the-shelf high-speed Serializer-Deserializer (SerDes) chips do not keep the same latency through the data-path after a reset, a loss of lock or a power cycle.
This implementation choice is often made because fixed-latency operations require dedicated circuitry and they are usually not needed for most telecom and data-com applications. However timing synchronization applications and triggers systems of the high energy physics experiments would benefit from fixed-latency links.
In this work, we present a link architecture based on the high-speed SerDeses embedded in Xilinx Virtex 5 and Spartan 6 Field Programmable Gate Arrays (FPGAs). We discuss the latency performance of our architecture and we show how we made it constant and predictable. We also present test results showing the fixed latency of the link.

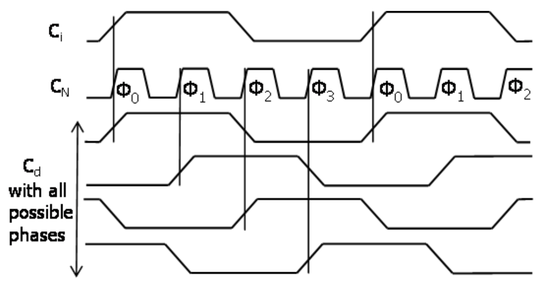

Fig. 2. Clock multiplication by 4 and subsequent division.

Latency Variations in Serial Links

Latency variations may come from both the serial and parallel sections of a SerDes device.
In a serializer, the parallel transmit clock (i.e. c_i in Fig. 2) is multiplied to provide the high-speed clock to the serial side of the PISO. In the deserializer, the high-speed recovered clock from the CDR is divided to provide the low-speed recovered clock (i.e. c_d) to the parallel side of the SIPO. If the divided clock phase is chosen randomly, we have a potential phase variation of the parallel recovered clock with respect to the transmit clock in terms of integer numbers of UIs. A phase variation of the recovered clock implies a latency variation of the data transferred on the link.
As far as it concerns the parallel section, latency variations may be induced by the presence of elastic buffers. A dedicated mechanism is needed to ensure that always the same number of words has been written in the buffer before they start being read (the receiver elastic buffer of the GTP implements this feature).

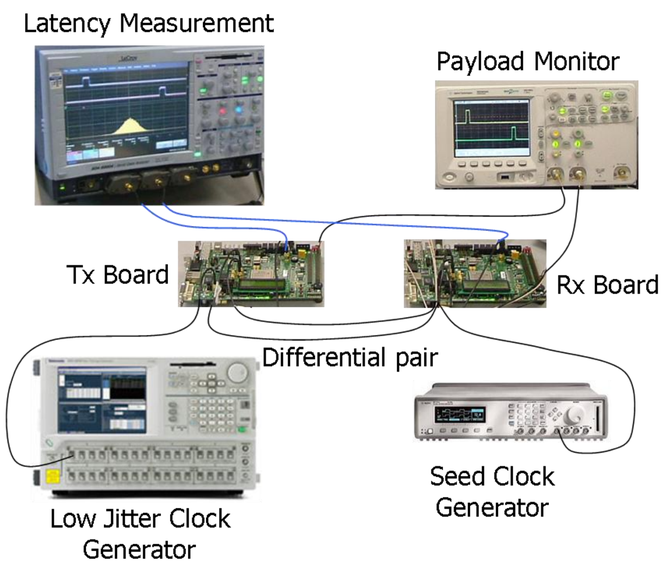

Fig. 6. Experimental setup for latency tests.

Test Results

In order to test our architecture, we used two off-the-shelf boards (Xilinx ML-505) to implement an 8b10b-coded 2.5Gbps link. We checked that the transmission latency and the phase of the recovered clock remained always the same during transfers and between subsequent power-ups of the system. We performed a 24-hour test, resetting the transmitter and the receiver every 3 seconds (i.e. simulating a power-cycle) and holding all the acquired waveforms on the scope screen, in order to record latency variations. The histogram in Fig. 7 shows the distribution of the link latency during subsequent resets. The standard deviation of the latency is ~ 40 ps and it includes the contributions from all the subsequent power-ups of the system. We note that the latency in terms of integer number of parallel clock cycles and UIs is fixed at each power up. The distribution we observe is only due to the jitter of the received data edge with respect to the transmitted one.

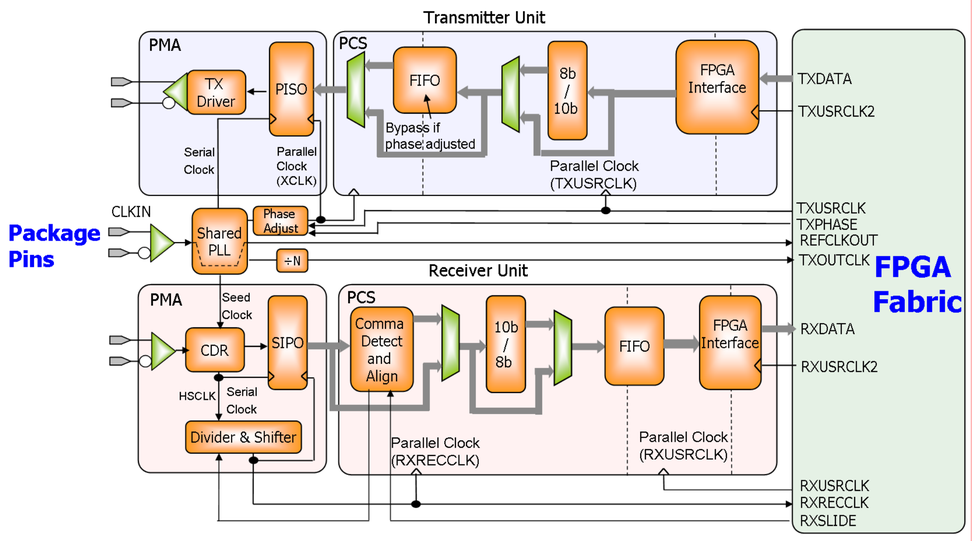

Fig. 1. Simplified block diagram of the GTP transceiver. Half a tile is shown.

The GTP Transceiver

Inside the FPGA, GTPs are available as configurable hard-macros, each including a pair of transceivers and some shared resources such as a PLL and the reset logic. Fig. 1 shows the architecture of the transmitter (Tx) and the receiver (Rx) included in each transceiver. The transmitter requires two input clocks TXUSRCLK and TXUSRCLK2, which, as far as it concerns this work, are always driven with the same signal. Analogously to the transmitter, the receiver requires two input clocks RXUSRCLK and RXUSRCLK2, which in our architecture are always driven with the same signal. The receiver outputs a clock recovered from the stream (RXRECCLK). Under appropriate conditions the FIFO in the transmitter can be by-passed, as well as the 8b10b encoding/decoding and "the comma detect and align" block in the receiver. The GTP does not work with fixed latency in configurations based on its internal resources. We have developed a configuration based on an external logic controlling the alignment, which forces the SerDes to have a deterministic latency through its data-path.

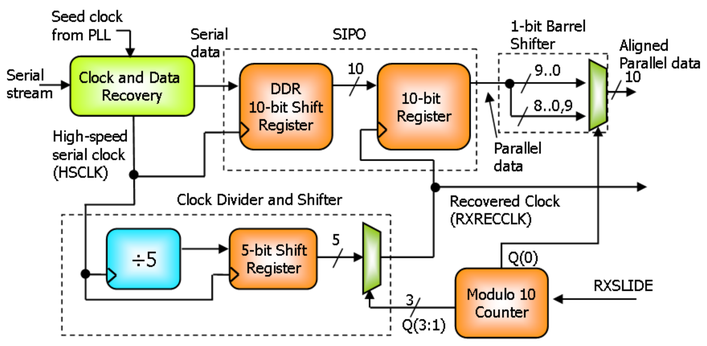

Fig. 3. Conceptual block diagram of the shift architecture used in PMA mode.

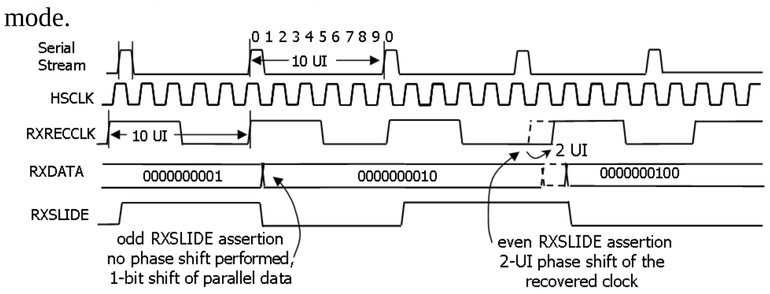

Fig. 4. Recovered clock phase adjustment by means of bit slips.

Word Alignment Mechanisms

The GTP allows the alignment to be driven by the logic in the FPGA fabric. There are two modes for the bit sliding to be achieved: the first one is realized in the PCS by shifting the parallel data ("PCS mode") and the second one in the PMA with a shifting of the recovered clock phase combined with the logical shifting of the data ("PMA mode").
We propose a model of how the phase shift is performed (Fig. 3) in PMA mode and we show a timing-accurate example of bit shifting in Fig. 4. In this mode, the same word alignment can be achieved with two possible phases of the recovered clock, differing of 1 Unit Interval (UI). The phase depends on the parity (odd/even) of the number of the performed bit-shifts.
By itself, no one of the supported alignment modes (PCS or PMA) guarantees the fixed latency operation of the SerDes. As a consequence, fixed latency is not achievable by only using the internal GTP circuitry.

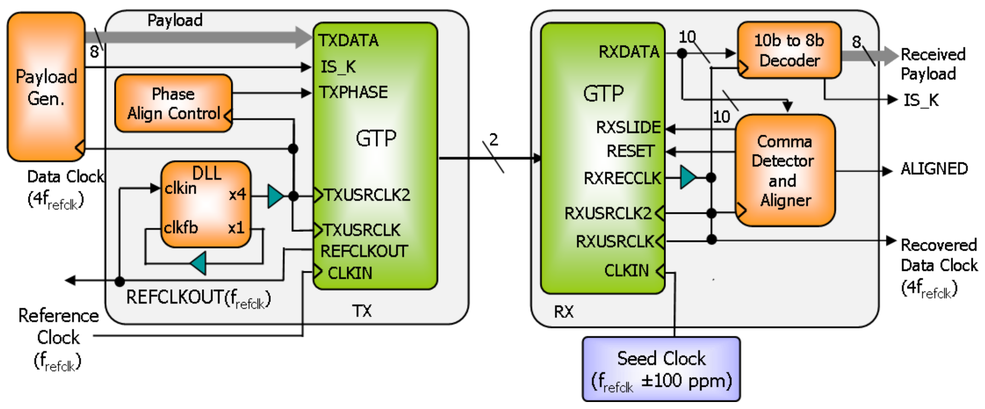

Fig. 5. Implementation of a serial link with fixed latency and fixed recovered clock phase, based on a GTP transceiver.

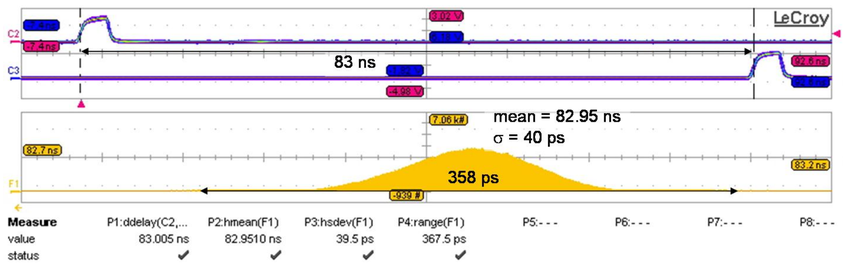

Fig. 7. Histogram of the latency of the link. Topmost trace: transmitted payload bit. second trace: corresponding received bit. Down: histogram of the latency.

Fixed Latency Transfers

On the transmitter, since the parallel clock for the PISO (XCLK) is generated by the PLL from the reference clock by multiplication and subsequent division, at each power-up its phase can be different. To work this issue around, we use the phase alignment circuit, which aligns the phase of XCLK to the one of TXUSRCLK. Under this operating condition the transmit FIFO is unnecessary and therefore we bypassed it.
For the receiver, we implemented a "Comma Detector and Aligner" (CDA) in the fabric. This block achieves the word alignment by monitoring the 8b10b encoded data from the GTP and controlling the bit sliding feature of the GTP in PMA mode. In order to establish a bi-unique relationship between the bit-shifts performed and the corresponding recovered clock phase, the CDA reject the CDR-locks leading to odd bit-shifts by resetting the GTP. In this approach, the latencies of the serializer and deserializer architecture are known and fixed (see Table 1).

Conclusions

High-speed SerDes chips are typically designed for variable latency transfers. However, protocols for timing synchronization and clock distribution applications, which sometime are present in HEP experiments, would benefit from fixed-latency serial links. By suitably configuring two GTP transceivers embedded in Xilinx FPGAs and adding to them some control logic in the FPGA fabric, we implemented fixed-latency operation. Our link transfers data with fixed latency and recovers the clock from the serial stream with a predictable phase, even after a reset or a power-cycle of the system. As an example of implementation, we designed a 2.5 Gbps serial link based on 8b10b encoding. Our architecture is independent of the data encoding and can be customized to support any.

This work is partly supported as Programma di Ricerca di rilevante Interesse Nazionale by the Italian Ministero dell'Istruzione, Università e Ricerca Scientifica.

	# of RXUSRCLK cycles	Block latency (ns)
Transmitter		
FPGA Interface	1	4
8b10b Encoder	1	4
FIFO (bypassed)	1	4
Serial Section	2	8
Total Transmitter Latency	5	20
Receiver		
Serial Section	1.5	6
Comma Detector (bypassed)	3	12
FIFO	5	20
FPGA Interface	2	8
10b8b Decoder	1	4
Total Receiver Latency	12.5	50
Total Link Latency	17.5	70

Table 1. Latency of the internal blocks of the transmitter and of the receiver.

aloisio@na.infn.it, rgiordano@na.infn.it, izzo@na.infn.it