PROCEEDINGS OF SCIENCE

An Ultra-Fast 10Gb/s 64b66b Data Serialiser Back-end in 65nm CMOS Technology

Thomas Gardiner[1]
Science and Technology Facilities Council
Rutherford Appleton Laboratory, Harwell Oxford, Didcot OX11 0QX, UK
E-mail: Thomas.Gardiner@stfc.ac.uk

With future pixel ASICs trending towards mega-frame rate readout, the development of ultra-high-speed readout systems is increasingly important. Here we present an ultra-fast readout system developed to operate at 10Gbps, and intended to surpass a more conventional highly-parallel LVDS bus approach. The system generates a 5GHz clock (LC Oscillator), scrambles and serialises the parallel input data in accordance with the Aurora 64b66b protocol, and transmits the data off-chip through a Current Mode Logic (CML) line-driver at 10Gbps. A prototype is under evaluation having been fabricated in early 2018 on a 65nm Multi-Project Wafer. Serialiser ASIC ran at 10.312Gbps under test for 60 hours without a bit-error event.

Topical Workshop on Electronics for Particle Physics (TWEPP2018)
17-21 September 2018
Antwerp, Belgium

PoS(TWEPP2018)088

1. Introduction

Future pixel ASICs are now trending towards readout speeds in the mega-framerate order. This has put considerable strain on traditional readout systems and has resulted in a significant increase in area, power and IO pads. These readout systems typically function on a simple interface scheme and LVDS drivers to transmit the data off chip. This work has developed a new sophisticated readout system with a significant increase in link speed and a large reduction in area, power and IO pads. Furthermore, an interface module is proposed and is in development to maintain a low-complexity interface despite the large internal complexity increase.

Figure 1 shows the top-level hierarchy of the 10Gbps serialiser. The serialiser interface in development is to be asynchronous; externally interfacing data is timed to an unconstrained clock generated elsewhere in the ASIC, while internally a 5GHz clock generated from a LC oscillator PLL is the governing clock domain for all internal data. Once the data is processed and transferred to the 5GHz clock domain, data is encoded and scrambled according the Aurora 64b66b protocol. Data is then split, mixed and serialized into a 10Gbps data stream to be transmitted off chip through a differential CML driver.

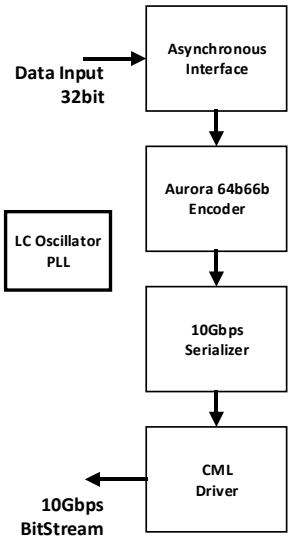

Figure 1: 10Gbps Serialiser Readout System.

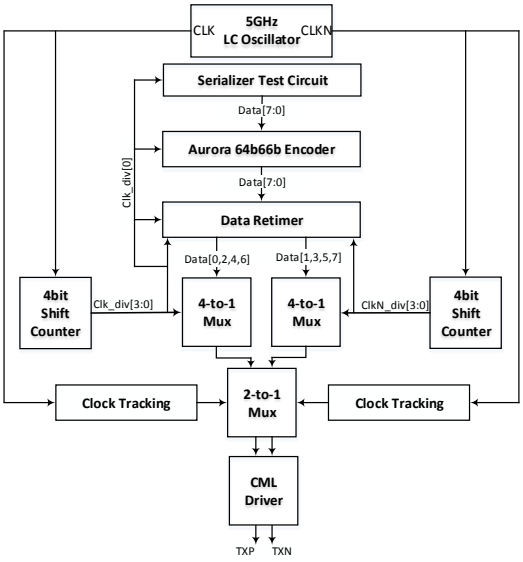

Figure 2: 10Gbps Serialiser Architecture.

2. Serialiser Architecture

The serialiser architecture is shown in Figure 2 and compromises of a 5GHz PLL module, Interface logic module, 10Gbps Serialiser and CML Driver. The final output of this module produces an Aurora 64b66b compliant 10Gbps bitstream and is transmitted off chip.

2.1 LC Oscillator PLL

An LC oscillator was chosen for its superior performance in jitter, duty cycle and stability. Additional switchable damping capacitors were added to provide flexibility to the target frequency. The LC oscillator was attached to an analogue charge pump PLL architecture to

control the frequency and stability of the clock. Dynamic TSPC flip-flops were implemented to divide the 5GHz clock down for comparison with the reference clock *(÷128)*. The TSPC flip-flops were required to allow state-based digital circuitry to operate at 5GHz, these flip-flops were also used in the serialiser circuit [1]. Figure 3 and Figure 4 show the schematic of the TSPC flip-flop and the LC oscillator respectively.

2.2 10 Gbps Serialiser

The serialiser has been designed to accept 8-bit words per clock cycle (which operates at 1.25GHz) at the input and output a differential full-swing10Gbps bitstream.

2.2.1 Interface Scheme

The serialiser module has been designed to operate asynchronously with the remainder of the ASIC hence an asynchronous FIFO transfer module is in development. 32bit words are written asynchronously and read out in 8bit words using standard clock domain crossing techniques and elastic buffers. Status flags such as FIFO_empty are used to automatically send idle packets when no written data is available.

2.2.2 Aurora 64b66b Encoding

Aurora 64b66b protocol was chosen due to being a robust and common protocol besides contains many high-level features besides scrambling the data to ensure a DC balanced bitstream for clean data-transmission. Features implemented in this design include: 1) Clock recovery packets - Allows the receiver to keep the clock locked to the data stream, 2) End frame packets – Allows the bit-stream to be packaged automatically at the receiver end and 3) Idle packets – Allows idle padded data to be automatically stripped out at the output of the receiver.

The encoder circuit encodes 8bits from the input and outputs the previously encoded 8bit word at the output. Since the protocol adds 2 header bits every 64 bits to form a 66bit data packet, the encoder circuit contains an elastic shift buffer to store the remaining bits. When 8 remainder bits have been accumulated, the encoder raises an output to indicate that the 8bit input word will not be accepted and instead proceeds to output the 8 remaining bits hence flushing the elastic buffer; this occurs every 4 encoded data packets. Note that this does not distort the bitstream order as the remaining bits are replaced every cycle with the 8bit input word. It's worth mentioning that the protocol requires all 64 bits to be streamed in consecutive cycles and only on the last streamed in 8bit word can the interface module decide whether to send an idle or end-frame packet or to continue sending data. In the test circuit, the interval of idle and end-frame packets is programmable.

2.2.3 8-to-1 bit Serialiser

The 8-to-1 bit serialiser circuit operates at 1.25GHz (generated from the 5GHz clock) and encodes and serialises 8 bits per cycle to form the 10Gbps bit stream. The 8bit words are encoded following the Aurora 64b66b protocol as described above before being serialised.

Firstly, the encoded 8bit word is retimed with each individual bit being consecutively shifted by 100ps to ensure a single valid bit is available at all times. Precise time shifting is achieved by the generation of eight 1.25GHz clocks using two 4bit shift counters with each counter being updated on opposite clock edges of the 5GHz clock. The eight different clocks

then retime the 8bit word by passing each bit safely through 1 or 2 different clock domains before finishing on the predestined clock domain. The retimed 8bit word is then split in two alternately and fed into two 4-to-1 serialisers to create two 5Gbps bitstreams. The two 5Gbps bitstreams are finally multiplexed into a single 10Gbps bitstream with a 2-to-1 multiplexer controlled by the phased-tracked 5GHz clock.

2.3 Current Mode Line-Driver

The Current Mode Line-Driver (CML) was implemented using a standard architecture as shown in Figure *5*. Charge balancing transistors (half width transistors) were added to correct the charge injected from the driver transistors due to a large gate to drain capacitance.

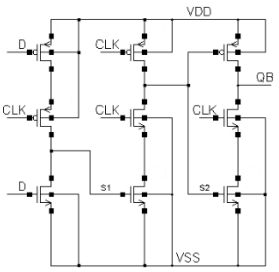

Figure 3: Schematic of TSPC Flip-flop.

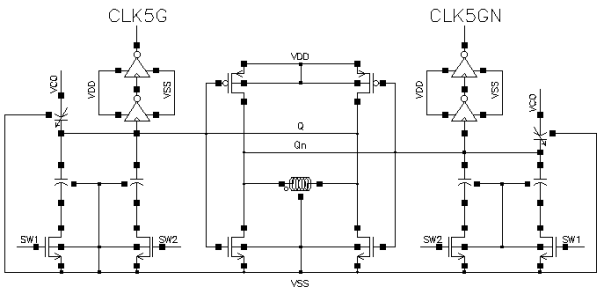

Figure 4: Schematic of LC Oscillator.

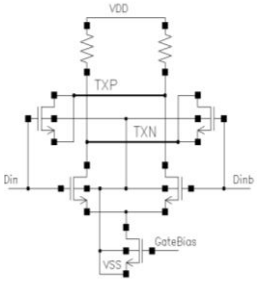

Figure 5: Schematic of CML Driver.

2.3.1 Link Modelling

To ensure the CML driver generated a clean eye off chip, the link was modelled in 3 parts: wire-bond, PCB and termination IC models and simulated in a spice simulation. Wire-bond model consisted of an RLC network and was derived from first principles due to being well understood. The PCB model was an extracted model of the link tracks taken from an early design of the test board with manufacturing properties included. The termination IC model contained loading capacitance and termination impedance.

3. Testing

The sophisticated test system includes the following modules: 1) A retimer chip (*TI DS110DF410*) to sample the 10Gbps bitstream and buffer it off the PCB. 2) A Xilinx Kintex-7 FPGA with a 10Gbps Aurora 64b66b transceiver synthesized internally along with error detection and data capture code for evaluating the quality and robustness of the ASIC serialiser, 3) A programmable crystal (*Si570*) to generate a clean stable clock for the ASIC PLL, 4) A Clock multiplier (*Si5324*) to generate a compatible clock for the Kintex-7 transceiver at the defined 10.302Gbps bit-rate.

Once the system is initialized, an encoded bitstream containing a continuous PRBS-11 sequence using Aurora 64b66b protocol is sent from the ASIC to the FPGA where the PRBS-11 sequence is recovered. The synthesized error detection and data capture code checks the incoming bit sequence for long periods of time and records when bit errors occur. Figure 6 shows the difference between modelled and measured result. The degradation of the eye opening is likely to be caused by jitter noise in the PLL, non-idealities in the link model and

measurement probe distortion. Figure 7 shows a large eye opening for a very low Bit Error Rate (BER) when resampling at the retimer IC.

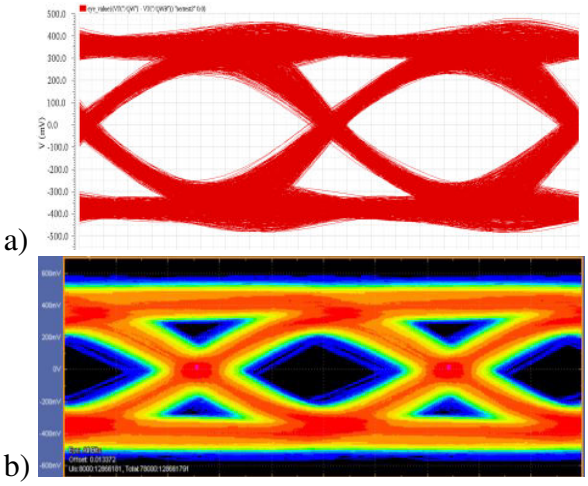

Figure 6: a) Simulated 10Gbps bitstream with an eye opening amplitude of 400mV and width of 80ps. b) Measured 10Gbps bitstream with an eye opening of 300mV and 45ps.

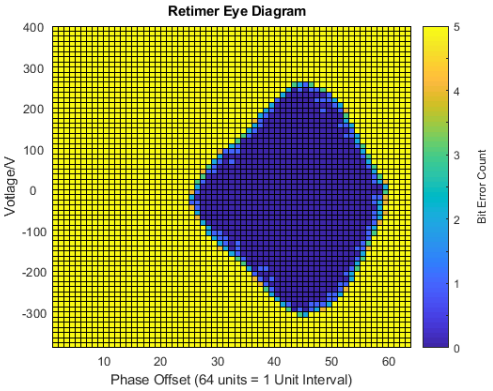

Figure 7: Detected 10Gbps bitstream eye diagram at the input of the Retimer IC. Recorded by the Retimer IC's debug circuitry.

4. Conclusion

In this paper, a 10.302Gbps serialiser readout system has been designed and tested successfully with a BER of 10^{-16}. Once the low-complexity interface finishes development, the readout system will be easily deployable onto ASIC designs due to its self-contained nature. Figure 8 shows the assembled system under test and Figure 9 shows the physical high-speed link.

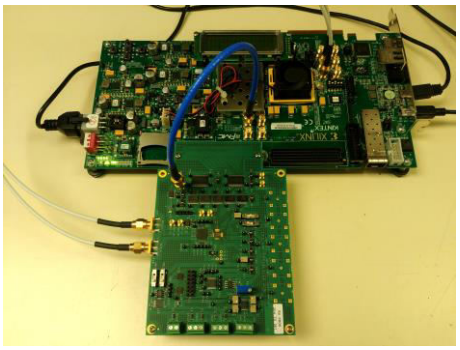

Figure 8: Readout system in operation with test board and FPGA board.

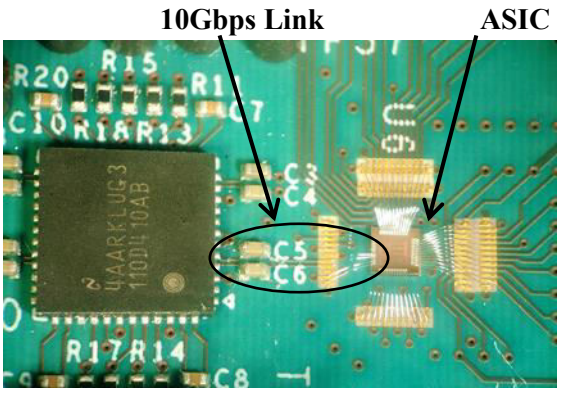

Figure 9: ASIC wire-bonded to test board with the high-speed link highlighted.

References

[1] D. Felici, S. Bertazzoni, S. Bonacini, A. Marchioro, P. Moreira and M. Ottavi, *A 20 mW, 4.8 Gbit/sec, SEU robust serializer in 65nm for read-out of data from LHC experiments*, 2014 JINST 9 C01004.