A 100 Gigabit Measurement System with State of the Art FPGA Technology for Characterization of High Speed ADCs and DACs

Damir Ferenci, Manfred Berroth
Institute of Electrical and Optical Communications Engineering
University of Stuttgart
Stuttgart, Germany
damir.ferenci@int.uni-stuttgart.de

***Abstract*—Speeding up the data rate in today's fiber optical networks requires a sophisticated higher order modulation. For this purpose high speed ADCs and DACs are necessary with sampling rates above 25 GS/s and a resolution of 6 bit. This drives the combined data rate to 150 Gbit/s. To realize a cost-efficient and fully scalable measurement system for the characterization of the high speed ADCs and DACs, a FPGA board is used which is equipped with gigabit interfaces. These interfaces provide data rates between 6.5 Gbit/s and 11 Gbit/s. In this paper we describe the hardware which is necessary to build up the measurement system and we describe the architecture of the FPGA logic which is implemented by means of VHDL.**

I. Introduction

The commonly used on-off keying is reaching its physical limits in today's fibre optical networks. To increase the data rate a higher spectral efficiency is needed. Higher order modulation schemes like quadrature phase-shift keying (QPSK) or quadrature amplitude modulation (QAM) would solve this problem and increase the data rate per wavelength. The goal for the next generation of optical data communication systems is a data rate of at least 100 Gbit/s. For this application high speed components like analog-to-digital converters (ADCs) or digital-to-analog converters (DACs) are necessary which operate at data rates between 25 GS/s and 50 GS/s and have a resolution of 6 bit. This requires a digital bandwidth of 150-300 Gbit/s at the interface of a digital signal processor and the ADC or DAC [1], [2].

To handle the data rate at the interfaces 12 to 48 channels with data rates between 6.25 Gbit/s and 12.5 Gbit/s are necessary. These data rates can be handled by field-programmable gate arrays (FPGAs) which are equipped with gigabit interfaces. For example, the Xilinx Virtex4 series offers a data rate up to 6.5 Gbit/s at up to 24 differential channels. This paper describes the architecture of a measurement system used to characterize a fourfold parallel 20 GS/s 3 bit CMOS ADC with 12 differential 5 Gigabit interfaces [3]. Although in this application the maximum data rate is limited to 60 Gbit/s, the presented measurement system is fully scalable for future designs and FPGAs to handle data rates beyond 100 Gbit/s. In chapter II the hardware platform is described. The digital system, which is designed in the very high speed hardware description language (VHDL), is explained in chapter III and in chapter IV data synchronization methods are introduced.

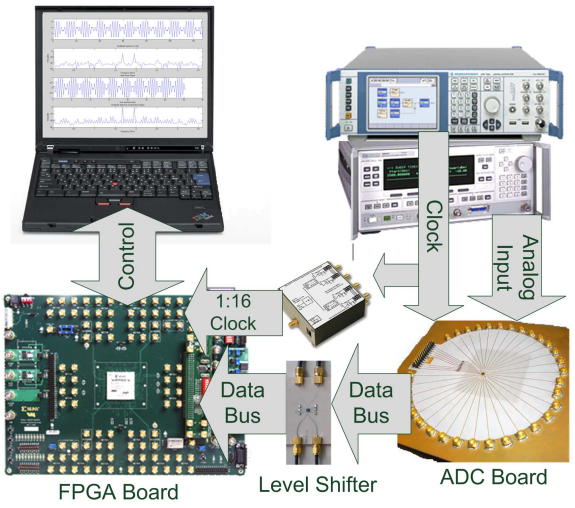

Fig. 1. The hardware platform.

II. Hardware platform

The hardware platform is depicted in Figure 1. The main part of the measurement system is a Xilinx FPGA board, the so-called RocketIO Characterization Platform, which is equipped with the Virtex4 FPGA XC4VFX100 [4].

A 1-43 GHz sinusoidal signal generator (Rohde & Schwarz, SMF 100A) is generating the clock signals for the device under test (DUT) and the 1:16 clock divider (Pulse Research Lab, PRL-258-8). The clock divider is supplying the clock signals for the FPGA board. As the clock signals are generated from the same clock source, the FPGA and the DUT operate clock synchronous.

The input and output logic levels of the FPGA are between 1 V and 1.5 V. As most of the high speed interfaces use logic levels between 0 V and -0.3 V, level shifters must be used between the DUT and the FPGA. The level shifter NBSG16M from On Semiconductor is used to shift the voltage level up and NBSG16VS is used to shift the voltage level down. The

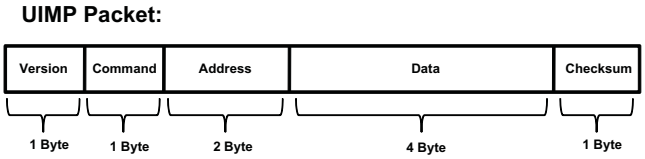

Fig. 2. The UIMP protocol.

digital bandwidth of the connected DUT, here a 3 bit 20 GS/s ADC, is 60 Gbit/s. The 3 bit output is fourfold demultiplexed on-chip, resulting in 12 interfaces each with a data rate of 5 Gbit/s. The FPGA is configured to store a sequence of up to 6 Mbit. This sequence can be read out by a PC, where a discrete Fourier transform (DFT) is performed to calculate the effective resolution of the ADC at a given input signal frequency. The necessary input signal for the ADC is generated by another sinusoidal generator which is synchronized to the clock generator.

The communication with the FPGA is initiated and controlled by the PC. Each request from the PC will be acknowledged by the FPGA, making the PC the master and the FPGA the slave. The protocol architecture defines two address types, a hardware address and a data address. A hardware address can address a hardware block within the FPGA and a data address can address data within a previously set hardware address.

The protocol between the PC and the FPGA is the Universal INT Measurement Protocol (UIMP). It has a length of nine bytes (cf. Figure 2). The first byte contains the protocol version, the second byte contains control flags. The third and the fourth byte contain the address information. The next four bytes contain the data that is written or read from the data address. The last byte contains a simple checksum.

III. FPGA Design

The block diagram of the FPGA logic design is depicted in Figure 3. The serial data from the PC is decoded in the serial interface and passed to the protocol decoder. Here the protocol is decoded and data is transferred via an internal 32 bit data bus to or from the protocol decoder to one of the connected units. The transmit units contain RAM blocks with the data that is intended for transmission over the gigabit interfaces. The receive units store the input data from the high speed interfaces. The receive and the transmit units are connected to the gigabit transceivers which perform the serialization of the data. The transmission of the data is performed cyclically over the data in the transmit RAM. The reception starts with an external command from the PC and ends when the RAM in the receive unit is fully written.

There are two main clock domains within the design. A 50 MHz clock is used for clocking of the internal data bus which connects the protocol decoder with all main units. The second clock domain is clocked from the gigabit transceiver block. The gigabit transceivers serialize 32 bit data words to a single output stream, thus the internal data rate is 1:32 of the high speed data transmission rate.

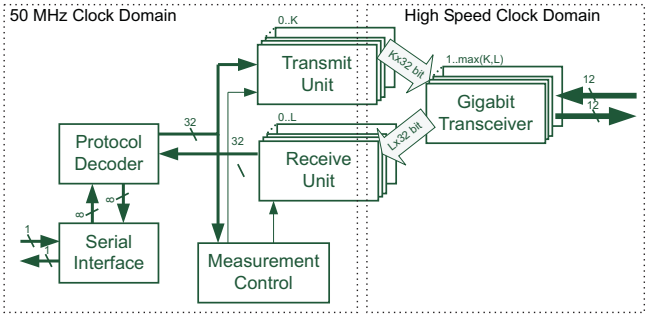

Fig. 3. Block diagram of the VHDL design.

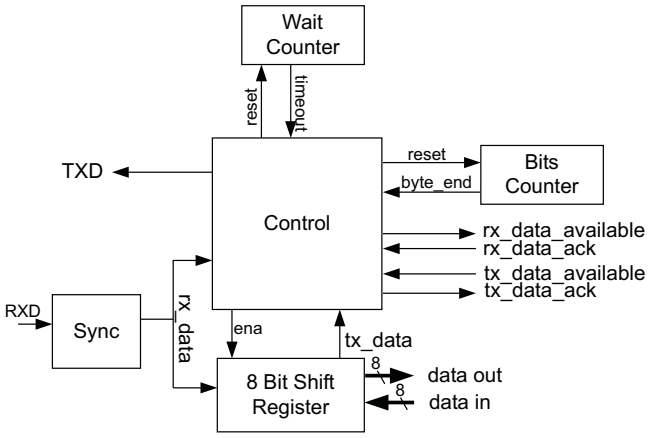

Fig. 4. Block diagram of the serial interface.

A. Serial Interface

In Figure 4 the block diagram of the serial RS-232 interface is shown. The control block is responsible for a correct transmission or reception flow by correct setting of the control signals of all registers and counters. The data is received or transmitted by means of the shift register. Incoming serial data is synchronized by a dual rank synchronizer and then shifted through an 8 bit shift register. For transmission the data is loaded parallel into the shift register and finally shifted out. The wait counter defines the length of a bit period of the serial interface. By changing a value in the wait counter the transmission rate can be adjusted. The bit counter is used to count until all eight bits are received or transmitted.

B. Protocol Decoder

The block diagram of the protocol decoder unit is depicted in Figure 5. The control unit is responsible for correct encoding and decoding of the UIMP packet. The incoming data bytes are shifted through a nine byte register chain. The data bytes are counted by the word counter. A timeout counter is used to reset the flow if the PC sends less then nine bytes or a byte gets lost. If an internal error occurs, for example if the user tries to read from a non-existing hardware address, the timeout also resets the control unit. The correctness of the checksum and the version byte are also reported to the control unit. A new hardware address is stored in the hardware address

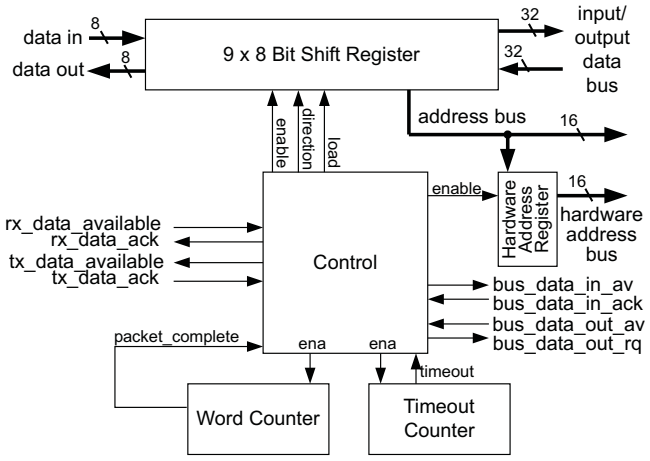

Fig. 5. Block diagram of the protocol decoder.

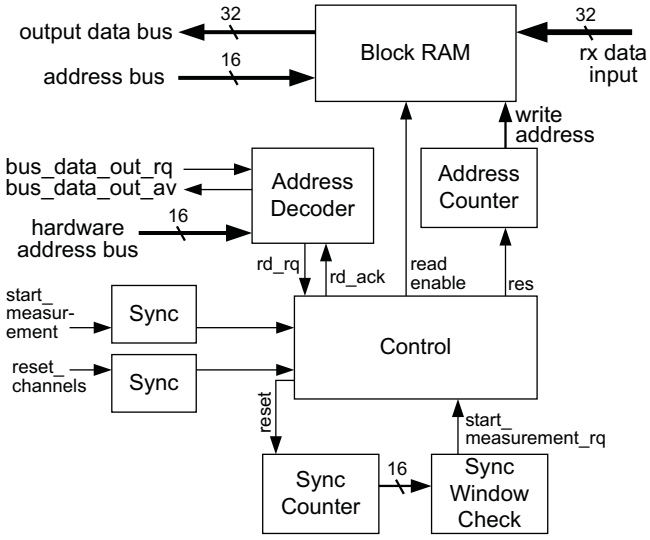

Fig. 6. Block diagram of the receive unit.

register. The data and the address busses are connected to the internal units. Measurement and control data is interchanged between the PC and the internal units by means of these busses.

C. Receive Unit

A block diagram of the receive unit is shown in Figure 6. The RAM is used to store the measurement data. The address counter is incrementing while the data is stored. This procedure is fully controlled by the control unit. The control unit is also responsible for a correct read out flow of the data and for a synchronous start of the data storing process, which is ensured by the following mechanism. The measurement control unit reports a measurement request to the control unit of all receive units. The control unit stores the measurement request until the synchronization counter is zero. Then a synchronous measurement starts at each channel. This procedure ensures equal bit shifts at all data channels for different measurements.

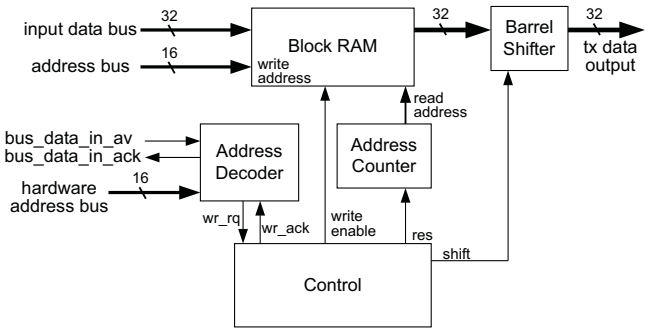

Fig. 7. Block diagram of the transmit unit.

D. Transmit Unit

The measurement platform contains a transmit unit for data transmission. The block diagram is depicted in Figure 7. A RAM is used to store the data that is transmitted cyclically to the high speed interface. New data can be written to a specific channel by setting the hardware address of the transmit unit that is associated to the channel. Subsequently new data can be written on an arbitrary data address within the RAM. For synchronization of the channels a manual synchronization is implemented where the output data can be shifted either word by word or bit by bit, depending on the difference between the channels. A barrel shifter is used to shift the data by a single bit. Shifting the data word by word is realized either by stopping the address counter for one clock cycle or by incrementing the address counter by two steps instead of one step.

E. High Speed Interface

The high speed interface (cf. Figure 8) contains the so-called RocketIO interface and finite state machines which are responsible for the initial reset of the RocketIO interface. The RocketIO interface is the actual gigabit transceiver, it is a hard-wired component in the FPGA which is designed by Xilinx. It contains serializers, deserializers, encoders, decoders and PLLs.

The serializers and deserializers are performing the serialization and deserialization between the internal 32 bit data bus and the serial link. The encoders and decoders are intended for the use of standardized communication protocols. In this design no communication protocol is used, thus the encoders and decoders are bypassed. The internal PLLs generate internal clock signals from the external reference clock signal. The external reference clock rate is 1:16 of the data transmission rate and is provided by the external clock divider. As the internal clock signals are individually generated, each RocketIO operates in an independent clock domain. This is considered in the design by passing the clock from the RocketIO interface to the appropriate receive and transmit unit.

For the receive channel an integrated clock and data recovery circuit (CDR) can be used to optimize the sampling point. The CDR can compensate for phase shifts between the reference clock and the incoming data. In the practical

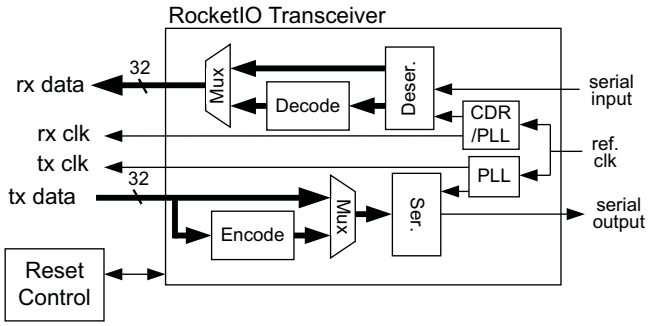

Fig. 8. Block diagram of the high speed interface.

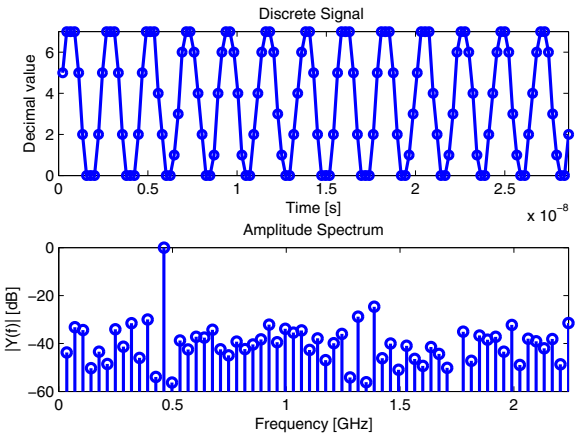

Fig. 9. Example measurement of a fourfold parallel 3 bit 20 GS/s ADC. The single channel sampling rate is 4.48 GS/s, the input signal frequency is 4.935 GHz. Due to undersampling the resulting frequency is 0.455 GHz.

application the CDR cannot lock to incoming data if the density of zero-one transitions is not high enough. For this reason the CDR is deactivated in this design and the internal receive PLL is locking to the external reference clock.

IV. Data Synchronization

A problem of using multiple RocketIO interfaces is that the channels are not synchronous to each other. There is a difference of multiple bits and also a phase shift that is less than a bit period. To compensate for this problem several possibilities are investigated.

To overcome the phase shift problem, the RocketIO receiver interfaces are operating at the double data rate (twofold oversampling). By this method digital phase shifting by 180 degrees is realized. Although the RocketIO interfaces are specified for a nominal data rate of 6.5 Gbit/s, an operation up to 11 Gbit/s is possible. With the oversampling method the appropriate sampling point is chosen and thus the phase shift problem is solved under the constraint of a limited data rate of about 5 Gbit/s per channel.

To synchronize the channels of an ADC two methods are introduced. The first method is to apply a digital sequence on the ADC analog input. By setting the ADC reference levels to zero the ADC will operate as a comparator and the sequence is synchronously output at all interfaces. After evaluation of the recorded data the bit shift can be calculated by reconstruction of the original sequence. For further measurements this shift will be constant as long as the input clock of the FPGA transceivers is stable. A second possibility relies on a DFT. As the input signal is known the signal-to-noise ratio (SNR) can be calculated by means of a DFT. By shifting all data channels until the SNR is maximized the delay between the data channels is compensated. Figure 9 shows a result of the synchronized data and the associated spectrum. All calculations are performed in Matlab. The advantage of this method is that the measurement setup needn't be modified to measure the delay between the channels.

The same problem as for the receive channels applies to the transmit channels. There is a phase shift and a difference of multiple bits between the transmit channels. To compensate for the bit shift the output stream of a selected channel can be shifted bit by bit by means of the barrel shifter in the transmit unit. To synchronize all channels an oscilloscope is necessary to observe the output stream and perform the shifting until all channels are synchronous. To compensate for the phase shift, either a phase shifter is needed between the DUT and the FPGA, or the length of the connection cable must be varied until the phase is well-adjusted.

Another possibility relies also on the double data rate method. By doubling the data transmission rate, as well as each transmitted bit, the data can be shifted by 180 degrees. As with the receivers the maximum effective data transmission rate is halved with this method.

An automatic synchronization circuit which compensates for all the described problems is currently under development. For this purpose an additional chip is placed between the FPGA and the DUT which provides a feedback channel to the FPGA. A circuit for correction of phase shifts is also included.

V. Conclusion

A cost-efficient multi gigabit measurement system for high speed ADCs and DACs is presented. It is fully scalable for future designs and FPGAs to handle data rates beyond 100 Gbit/s. The measurement system is based on the Xilinx RocketIO Characterization Platform equipped with a Xilinx XC4VFX100. Transmitted data is synchronized by manual bit shifting. For synchronization of incoming ADC data a software based synchronization method is introduced.

Acknowledgment

This work was funded by the German Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung, BMBF) with grant number 01BP0774.

References

[1] J. Cartledge, D. Krause, K. Roberts, C. Laperle, D. McGhan, H. Sun, K. Wu, M. OSullivan and Y. Jiang, "Electronic Signal Processing for Fiber-Optic Communications", IEEE Leos Newsletter, February 2009
[2] J. H. Sinsky and P. J. Winzer, "A Microwave Engineer's Perspective", IEEE Microwave Magazine, April 2009
[3] D. Ferenci, M. Grözing and M. Berroth, "Design of a 3 bit 20 GS/s ADC in 65 nm CMOS", Prime, July 2009
[4] Xilinx, RocketIO Characterization Platform, ML423 User Guide - UG087