Achieving Best Performance with VME-based Data Acquisition Systems and 2eSST

A. Aloisio[a,b], L. Capasso[a,b], R. Giordano[a,b], V. Izzo[a,b]

[a] INFN - Sezione di Napoli - Via Cintia, 80126, Napoli, Italy

[b] Università degli Studi di Napoli "Federico II" - Via Cintia, 80126, Napoli, Italy

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

Abstract

The double edge Source Synchronous Transfer (2eSST) is the fastest block transfer cycle offered by the VME64x standard. The maximum achievable data-rate foreseen by the protocol is 320 MByte/s. In this paper we present a reference design based on a FPGA, for the reader willing to implement 2eSST in his VME64x application. By using this template, we have designed a custom Bit Error Rate Tester, in order to probe the block transfer reliability within and beyond the data rate limit presently set by the standard. Our results show that 800 MByte/s data transfers can be achieved in a 21 slots crate with a BER smaller than 10^{-12}.

I. Introduction

THE ANSI/VITA 1.5-2003 standard [1] adds to the VME64x bus [2] a source-synchronous, double data rate block transfer cycle known as 2eSST. It allows the bus master to move across the backplane bursts of up to 256 words, each 64-bit wide, for a total of 2 Kbyte. In order to achieve the highest bandwidth, data is clocked by both positive and negative edges of a strobe, with no handshake. The protocol adopts an iso-synchronous scheme and the data producer is also in charge to drive the double rate strobe. According to the transfer direction, different lines are used to strobe the data. With this approach, the value of propagation delay does not affect by itself the timing performance and the maximum data transfer rate is set by the skew between the specific strobe and the data lines. Different loading on the bus lines and spread in the driver's behaviors shrink the timing budget for a correct data transaction. Even board layouts with a significant difference in the flight time of signals between bus connectors and on-board logic contribute to this effect. In a real system, the designer faces the challenge to optimize the system performance taking into account line transmission effects, asymmetric bus loading of legacy boards and even different behavior between read and write cycles. The latter is due to the privileged master location, which usually sits nearby the impedance termination network in the leftmost slot, at the far end of the line. Data driven in a read cycle is then received by the master in the cleanest environment. In a write cycle, on the contrary, a slave board located in the middle of a populated backplane might sample a signal distorted by multiple reflections. The protocol greatly improves the skew tolerance by placing the strobe at the center of the data window. The double data rate architecture invites the designer to work with a dual phase clock or at double the frequency and this makes it quite simple to set the strobe timing. However, in order to toggle the strobe as close as possible to the ideal central position, the transmitter logic needs a careful layout with balanced I/O timing between data and strobe. On the receiver side, the setup and hold requirements of the input logic should define a timing window centered on the strobe edge with the smallest width. The present version of the 2eSST protocol specifies three speed grades, with nominal strobe widths of 50, 30 and 25 ns, equivalent to a total bandwidth of 160, 267 and 320 MByte/s. For each speed, the standard analyzes the timing budget, stating the maximum allowed skews on the transmitter section, on the backplane, on the receiver and the worst-case setup and hold of the input registers. The burst speed is negotiated in the first steps of the transaction by transmitting a specific code in a bit field. It is interesting to note that the protocol reserves codes for even faster speed grades, which could be announced in the future. In this view we have tested the present physical layer beyond the maximum specifications. With a careful Printed Circuit Board (PCB) layout and keeping in mind the main sources of skew in the transmitter and receiver sections, we have designed a companion board-set capable to transfer data beyond the 320 Mbyte/s limit. The two boards are strictly VME64x-compliant and are designed around off-the-shelf components, including mature low cost Xilinx Spartan IIE Field Programmable Gate Arrays (FPGAs). The backplane as well is a commercial 21-slot, 6-layer VME64x implementation with active terminations. Transmitter and receiver suffer from the ground bounce and the simultaneous switching noise out of the 64-bit busses. While already important at standard speeds, these noise sources become a serious issue when pushing the data transfer rate; they challenge the signal integrity and add jitter and skew to the lines. The payload switching activity during a burst determines the noise level on the boards and on the transmission lines. At higher transfer rates this parameter affects significantly the achievable performance.

II. The Bit Error Rate Tester

Our tests on VME64x bus aim at studying the performance beyond the current data rate limit set by 2eSST specifications. We have designed a custom Bit Error Rate Tester (BERT) that can work with data rates up to 1 Gbyte/s. The BERT is based on two VME boards: a transmitter (TX), which sends custom data pattern across a VME backplane, and a receiver (RX), which compares it with a local copy of the pattern, flagging an error if a difference occurs (Fig. 1).The TX includes a programmable circular pattern generator with a length of 2k words. The clock frequency of the TX is set using an external clock generator and is halved to provide the strobe burst.

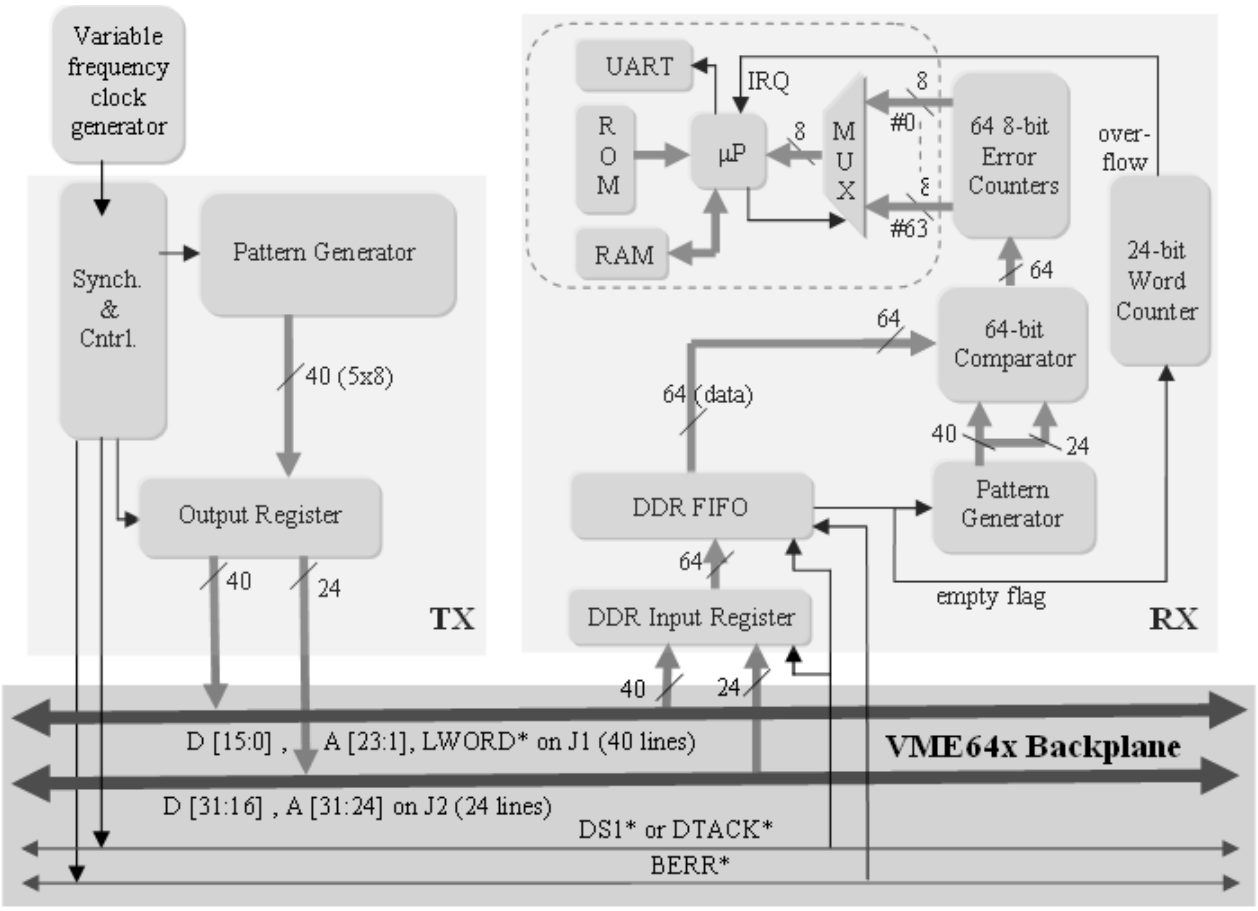

Figure 1: Block diagram of the Bit Error Rate Tester.

The output is transmitted in VME block transfer cycles in 256 word bursts, according to 2eSST VME specifications. The structure of a burst is shown in Fig. 2. A transition on a strobe line (DS1* or DTACK*), either positive or negative, validates data on A31-A1, LWORD, D31-D0 lines. Every burst begins with a negative transition on the strobe line and ends with a positive transition. The time interval between data bursts is programmable. Using an output register, the 40 bits output of the encoder are used to feed the J1 connector lines (D0-D15, LWORD,A1-A23), and the 24 least significant bits are also sent on J2 connector lines (D16-D31, A24-A31). The RX stores the data in a DDR input register clocked by the strobe line. Data is then transferred to a DDR FIFO, which has independent clocks domains for double rate writing and single rate reading. The VME strobe line is used as write clock and a 80 MHz internal clock as read clock. The DDR FIFO is used to decouple the VME clock domain from the receiver's internal clock domain. Also, using an input FIFO reflects what a designer would do in a real application. The BERR* VME bus line allows the TX to provide an external reset to the FIFO and to the input register. This signal makes only one transition at power up, thus doesn't affect the signal integrity of the adjacent data lines. Data coming out from the FIFO is compared in parallel with the output of a local pattern generator identical to the one in the TX. Each of the 64 comparators drives a clock enable of a 8-bit error counter. If a received bit is different from its local copy, the corresponding counter is incremented. An embedded microprocessor handles those counters and takes care of error displaying via a serial link. A 24 bit counter counts the number of VME words received and, when it overflows, issues an interrupt request (IRQ) to the microprocessor. The interrupt service routine reads the error counters and resets them. The CPU accumulates the value read from each error counter in a different 16-bit variable, thus extending the countable errors from $2^8 - 1$ to $2^{16} - 1$. If more than 255 errors occur on the same bit between two interrupts, the corresponding counter saturates. We did not take measurements in data rate ranges that make any counter saturate. However this has not been a limiting condition for our measurements which aimed at very low BER levels. The microprocessor periodically prints out the content of the variables and the number of the received words on a RS-232 serial port in ASCII format. Receiver and transmitter units have been implemented using two identical boards hosting a Xilinx Spartan IIE XS300 FPGA. Boards are equipped with SN74VMEH22501 bus transceivers [3]. They offer the latest available driving technology for 2eSST cycles and are required by the ANSI standard. A detailed characterization of these devices in VME applications can be found in [4] and [5]. Our boards have test points for some VME bus lines either on input and on the output of each driver. Timing performance of the data transfer has been optimized both on the TX and RX FPGAs. The output register of the transmitter is implemented into FPGA IOBs, thus all the bits have almost the same clock to output delay (T_{co}). As Spartan IIE FPGAs don't provide Double Data Rate (DDR) primitives, each bit of the receiver's input

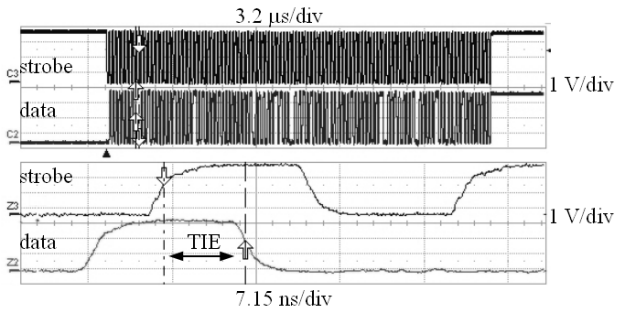

Figure 2: Top: example of VME64x 2eSST burst. Bottom: close-up on data and clock edges.

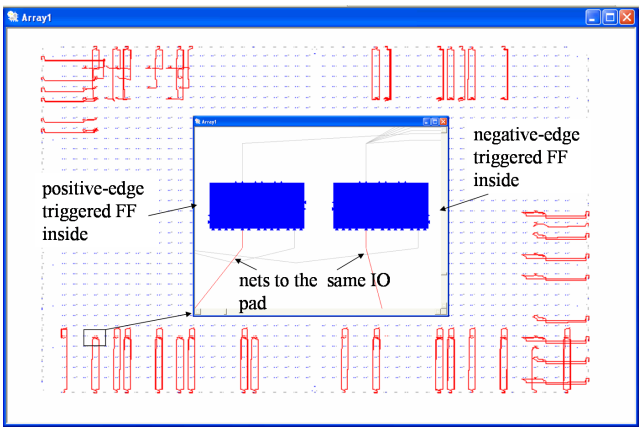

Figure 3: Layout of the DDR input register on receiver's FPGA.

register has been implemented with a pair of flip flops, each in a different slice of the same CLB, with inverted clocks (Fig. 3). The layout has been manually optimized to have balanced setup and hold times around the clock edge and to make them as similar as possible for all the flip flops (Fig. 4). The setup (hold) time has a mean of 0.20 ns (0.27 ns) and a standard deviation of 0.13 ns (0.13 ns).

Newer devices (e.g. Xilinx Virtex-II Pro) do offer DDR capable primitives. By means of those, we would not have to manually optimize the layout to equalize setup and hold times from different flip-flops, which would have been all the same. The DDR structure we designed using one IOB and two single rate flip-flops from two different slices could have been conveniently obtained by using just one double data rate IOB (Fig. 5).

Data patterns in the transmitter and receiver FPGAs are stored in ten 1k x 8 bit RAM banks. Their content is embedded in the bit-stream and they are initialized during the FPGA configuration. We developed a simple software utility to generate user defined data patterns and to load it in the right spots of the bit-stream. In this way we don't need to regenerate FPGA configuration files on a pattern change. Our setup makes it possible to measure the bit error rate of every VME data line as a function of the data rate and pattern. It also allows us to investigate the effect of the bus invert coding on the BER and signal integrity.

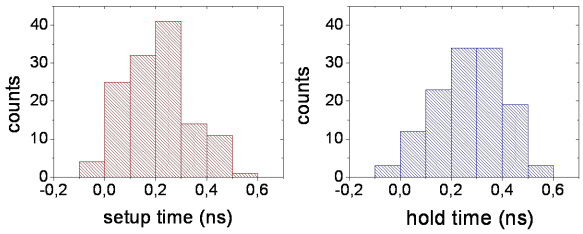

Figure 4: Setup and hold times histograms for the DDR input register on receiver's FPGA.

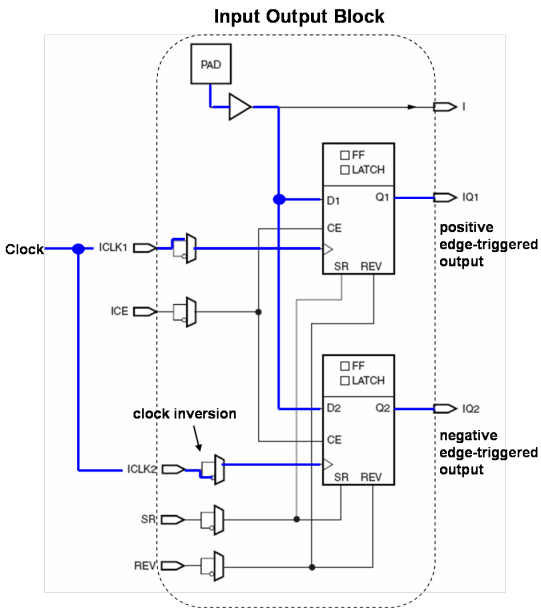

Figure 5: Virtex-II Pro Input Output Block configured as a DDR input.

III. Test Results

In our setup we have the two tester boards acting as master (TX) and target slave (RX). The master executes write cycles toward the slave. Five more boards, all VME64x compliant, have been used to load a 21-slot backplane and to take into account the change in the bus line impedance present in a real environment. The master has been placed in slot 0. According to the VME64x protocol, all the slaves participating to the 2eSST should be clustered in adjacent slots. By grouping them at the center of the backplane, we obtain a significant perturbation of the line impedance, giving us a typical scenario in data acquisition systems [6]. The target slave has been placed in slots from 9 to 14, with the other boards filling all the remaining slots in the range 9-14. We focused on this configuration, because earlier studies indicate this one as signal integrity worst case [7]. We measured the BER as a function of the data rate in all slots from 9 to 14. We spanned the range from 320 Mbyte/s to 800 Mbyte/s, equivalent to 40 Mbps to 100 Mbps rate on each data line. As data pattern, one would like to use the worst case switching activity pattern, which is obtained making all the data lines toggle at every cycle.

This choice would not generate a reasonably good pseudo-

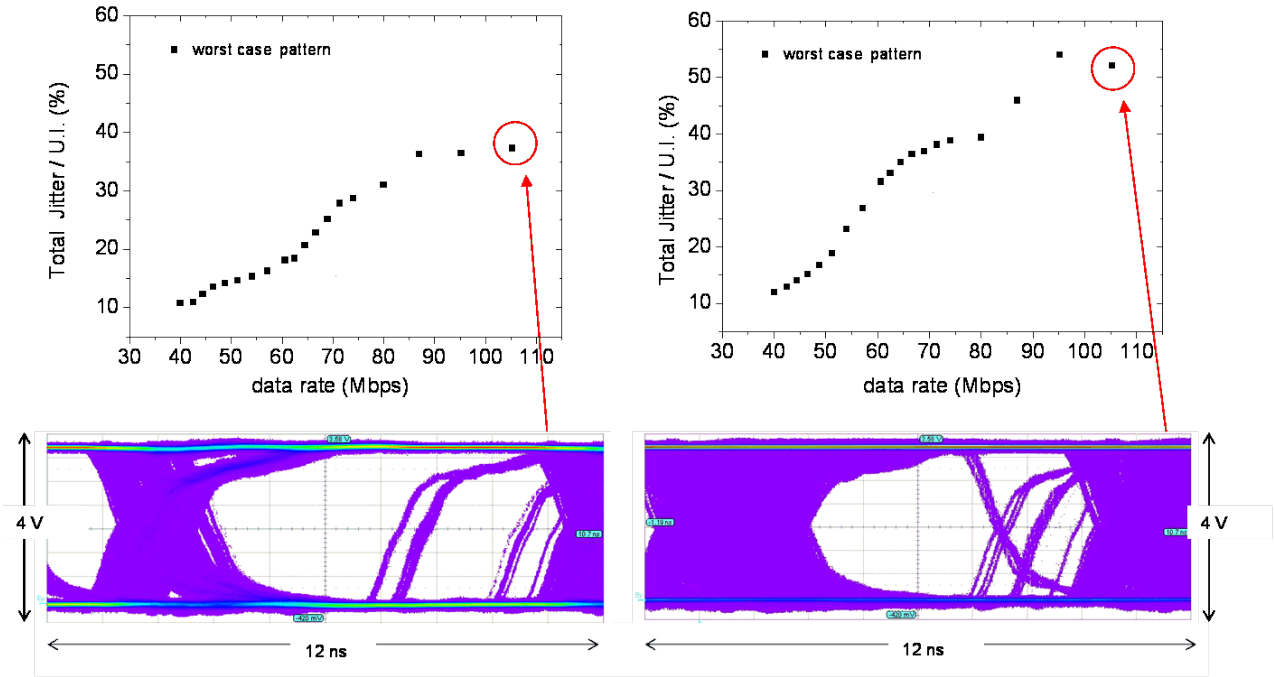

Figure 6: Top left (right): Tj/UI for slot 14 (9). Bottom left (right): eye diagram corresponding to the maximum data rate point for slot 14 (9).

random pattern, because the length of the pattern would be just of 2 words. To overcome this problem, we used a random pattern with a 7/8 switching activity per each byte, which has the next highest switching activity (for our bus partitioning). Our BERT recorded zero errors on all data lines with more than 10^{13} words transferred. This result gives a 10^{-12} BER with a confidence level bigger than 99% [8]. At 40 Mbps each BER run takes more than 120 hours, while the measurement lasts more than 48 hours at 100 Mbps. In order to reveal the BER trend versus the bit rate, one should increase the number of transferred bits. It is evident that this approach is not practical due to the too long run time needed. To gain insights into the available timing budget of the transaction, we instead performed alternative measurements based on data and clock edge relative timing. For this purpose we used a Serial Data Analyzer (SDA) oscilloscope with 20 Gsample/s maximum sample rate and 6 GHz bandwidth [9]. For each data edge the oscilloscope measures the timing interval between the data and clock edge, i.e. the Time Interval Error (TIE) (Fig. 2 bottom). The instrument collects the TIE measures in a histogram. This histogram is an estimate of the probability density function (p.d.f.) of the TIE, which is the probability to observe a data transition at a certain instant with respect to the clock edge. From the p.d.f. it is possible to calculate the BER as a function of the instant at which a data transition occurs. From this function one can obtain the timing budget (T_b) and thus the total jitter ($T_j = UI - T_b$, where UI is the unit interval) at a given BER. So, the TIE p.d.f. defines a biunique relationship between the total jitter of the data edge and the bit error rate.

The oscilloscope uses a specific software tool [10] to extrapolate the expected total jitter (T_j) at a given bit error rate (10^{-12} in our case) from the measured TIE histogram. In our tests, we measured T_j at 10^{-12} BER as a function of the data transfer

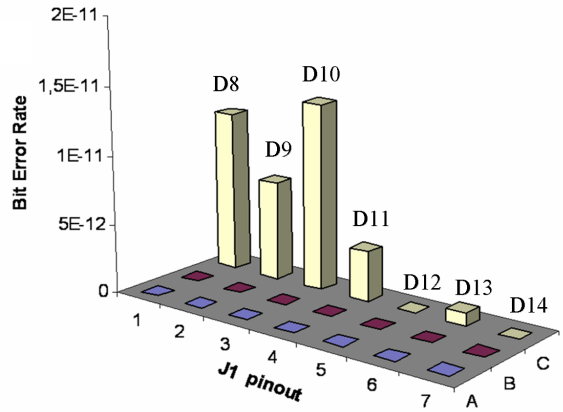

Figure 7: BER distribution on part of J1 connector (pins not shown have zero measured BER).

rate (from 40 Mbps to 100 Mbps) and of the slot occupied by the slave (from slot 9 to slot 14) . In the following discussion we will refer to T_j/UI instead of T_j. Our results show that T_j/UI grows moving the RX unit from the slot 14 toward 9, so in our set-up slot 9 (the nearest to the TX board) exhibits the worst behaviour. Fig. 6 illustrates the T_j/UI trend versus data rate measured at slot 9 and slot 14 for both the un-encoded and the coded case. T_j/UI is a strictly increasing function of the data rate. At slot 9 T_j/UI almost reaches 60% at a 100 Mbps data rate, while at slot 14 it does not exceed 40% . Note in Fig.6, how slot 9 shows a smaller open "eye" with respect to slot 14. The value of T_j increases as a function of the data rate (1/UI) because of the deeper impact of the transmission line effects on the signal integrity. Our setup is able to directly measure a non zero BER (with a run time of the order of one day) at data rate

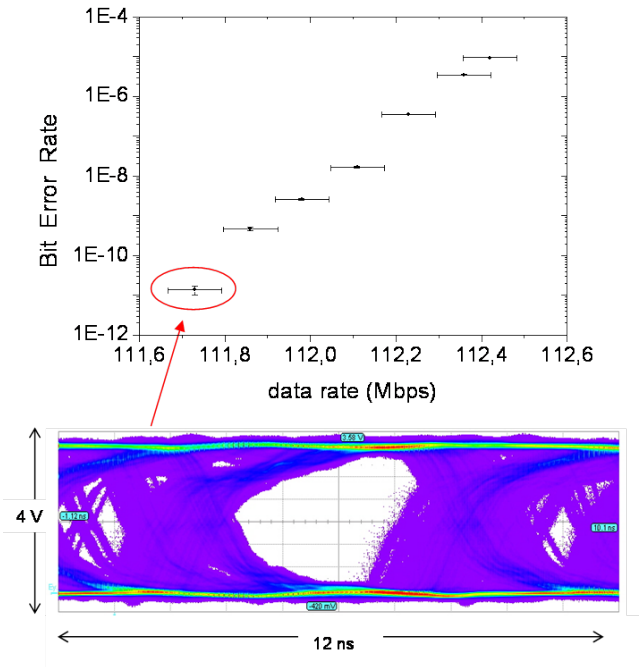

Figure 8: Top: D10 BER versus data rate for a fixed 7/8 switching activity data pattern. Bottom: eye diagram for the lowest data rate point.

of 111 Mb/s. In the data rate range between 111 Mb/s and 112 Mb/s the BER gains 6 orders of magnitude. The bit with the highest BER is D10. The nearest bits on the connector's C row exhibit a similar behaviour. The histogram in Fig. 7 shows the BER distribution on part of the J1 connector, when D10's BER reaches 10^{-11}. Fig. 8 shows the bit error rate versus the data rate per line at slot 9.

IV. Conclusions

Our results show that VME 64x 2eSST block transfer can currently sustain data transfer up to 800 MByte/s with a BER of 10^{-12} (estimated with a 99% confidence level) for typical loading conditions found in a high performance DAQ system. By direct BER measurements and jitter analysis, we studied the timing budget versus the data rate as a function of the slot occupied by the slave. It should be noted that we did not study the BER and total jitter dependence on temperature, power supply voltage and different loading conditions (i.e., legacy boards, different number and location of the slaves). However, even at 800 Mbyte/s data rate, in the worst case conditions, we have found an available timing budget larger than 40% of the unit interval. Our results have been obtained with low cost FPGAs without DDR primitives, so we had to manually optimize the layout for the input logic. Newer devices offer DDR I/O resources and remove that need, making the design implementation easier. We conclude that the present technologies (VME64x backplanes, bus drivers, FPGAs) allow reliable 2eSST transfers with speed grades beyond the current limit.

References

[1] (2003). ANSI/VITA 1.5-2003 American National Standard for 2eSST. VITA

[2] (1998). ANSI/VITA 1.1-1997 American National Standard for VME64 Extension. VITA

[3] (2005). SN74VMEH22501 8-bit Universal Bus Transceiver. Texas Instruments. [Online]. Available: http://focus.ti.com/lit/ds/symlink/sn74vmeh22501.pdf

[4] (2002). VMEH22501 in 2eSST and Conventional VME Backplane Application. Texas Instruments. [Online]. Available: http://focus.ti.com/lit/an/scea028/scea028.pdf

[5] A. Aloisio, P. Branchini, F. Cevenini, V. Izzo, S. Loffredo, R. Lomoro, "Signal integrity and timing issues of VME64x double edge cycles", IEEE Trans. on Nucl. Sci., vol. 53, n. 2, pp. 520-525, Apr. 2006

[6] A. Aloisio, P. Branchini, F. Cevenini, "Timing analysis of asynchronous block transfer cycles on VME and VME64x physical layers", IEEE Trans. on Nucl. Sci., vol. 51, n. 3, pp. 401-406, Jun. 2004

[7] A. Aloisio, P. Branchini, F. Cevenini, R. Giordano, V. Izzo and S. Loffredo "Bus-Invert Coding For Low Noise, Low Power 2eSST VME64x Block Transfers", IEEE Trans. on Nucl. Sci., to be published, Jun. 2007

[8] (2000). Application Note 1095. HFTA-05.0: Statistical Confidence Levels for Estimating BER Probability , Dallas Semiconductor –Maxim. [Online]. Available: http://pdfserv.maxim-ic.com/en/an/AN1095.pdf

[9] (2006). Serial data analyzer operator's manual, LeCroy [Online]. Available: www.lecroy.com/tm/library/manuals/SDA/OperatorsManual/SDA-OM-E_Rev_F.pdf

[10] (2006). WHITE PAPER SDA, ASDA and SDM SDA Serial Data Analyzer and SDM Serial Data Mask Package –Theory of Operation , LeCroy [Online]. Available: www.lecroy.com/tm/Library/WhitePapers/PDF/SDA_Theory.pdf