LVDS interface for AER links with burst mode operation capability

Carlos Zamarreño-Ramos, Rafael Serrano-Gotarredona, Teresa Serrano-Gotarredona, Bernabé Linares-Barranco

Instituto de Microelectrónica de Sevilla IMSE-CNM, CSIC and Universidad de Sevilla

41012 Sevilla, E-mail: bernabe@imse.cnm.es

Abstract— This paper presents the design and simulation of a serial AER LVDS communication link. It converts data from classical AER parallel bus with a 4-phase handshaking protocol into a bit stream which is transmitted serially into a single LVDS wire. At the receiver side data from the LVDS cable are transformed back to a parallel AER bus and handshaking signals are also properly managed. The link has been designed in a 90 nms technology. Extensive simulations have been performed demonstrating that the link can operate at a speed of 1 Gbps for all the technology corners, exhibiting a power consumption of 27.8 mW for the transmitter and 12.3 mW for the receiver. In the simulation the transmission channel was modelled as a 50 cm cat5E UTP cable, connected to the AER chip through 5 cm PCB traces modelled as a coupled microstrip transmission line. The design has been completed up to the layout level and has been submitted for fabrication. The transmitter and the receiver take up an area of 311x148 μm^2 and 300x148 μm^2 respectively.

I. INTRODUCTION

Computer capabilities for processing data have evolved at such high speed that the actual state of the art was unbelievable a few years ago. However, the sequential inspiration of the classic approach is not appropiate for certain applications (i.e. real-time vision processing). A possible solution to improve the processing efficiency is to emulate the human brain structure: a huge amount of computational units (neurons) with low processing capability, but massively interconnected between them. However, the physical interconnection of neurons in 2-D silicon systems is practically limited to a few neighbours.

AER (Address Event Representation) [1] is a neuromorphic protocol which allows the communication of a large array of neurons to another array using a reduced number of physical connections. The protocol operation is shown in Fig. 1. In the sender side, the neurons activity is encoded in their output spike flow frequency; an arbitrer+encoder circuit translates the neuron spikes into digital addresses which are sent through the digital bus. In the receiver side, a decoder circuit sends a spike to the neuron which is identified with the received address. Receiver neurons just have to integrate the input spikes to reconstruct the activation state of the sender neurons. Communication between chips is performed using a classic four phase hand-shaking protocol. The AER allows a real-time virtual connectivity between two sets of n neurons placed in different chips using $log_2(n) + 2$ physical wires.

However, one of the main drawbacks of present AER systems is poor scalability. The size of AER parallel connectors and buses makes present state of art AER multichip systems
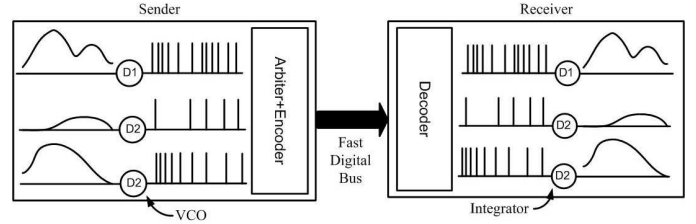

Fig. 1. AER link with parallel bus

bulky and difficult to scale up [2]. Recently, a word serial protocol has been proposed [3], but it is not easily scalable to build large multichip AER systems. In this paper we propose to move from the parallel AER bus to a bit-serial LVDS AER link. The bits of each address are serialized and transmitted using just a single physical LVDS wire. LVDS standard combines the serial transmission with low power consumption and high speed data rates [4]. The high speed capability of the LVDS link allows us to transmit the bits of each AER address serialized in a single wire while preserving the transmission throughput.

As we are using a single wire to interconnect the AER chips, AER systems with a large number of AER chips can be assembled much more easily. Furthermore, the proposed bit-serial AER-LVDS link also allows easily scalability of the arrays size, as the number of bits of the AER addresses can be arbitrarily increased without need of replacing the connectors nor the boards or the physical wires.

II. THE BIT-SERIAL AER LVDS LINK

Fig. 2 shows the block diagram of the proposed bit-serial AER LVDS link. This link interfaces with classical parallel AER buses using the 4-phase handshaking protocol at the transmitter and receiver chips.

The LVDS standard defines a serial differential link which uses low voltage differential excursions. This low voltage excursions reduce power consumption and at the same time allow to transmit data at rates in the order of Gbps. If we consider a large AER chip using 16 bit addresses (which correspond to a 256 x 256 array of neurons), a LVDS link of 1 Gbps and 2 bits for heading can achieve a transmission rate of 55 Mevents/second. This figure is above the data transmission rates of current state of the art parallel AER bus solutions [5].

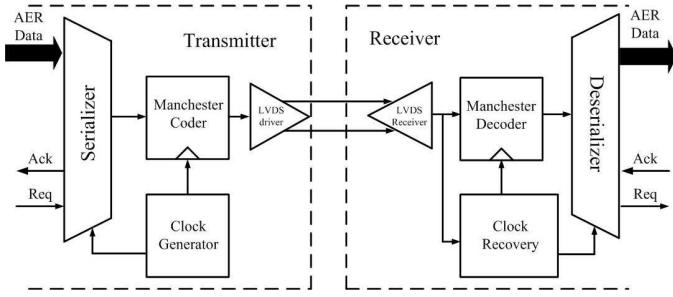One of the problems when designing a LVDS interface is the clock/data synchronization. The receiver needs a clock

Fig. 2. Block Diagram of the bit serial AER LVDS link

Fig. 3. Schematics of the serializer and Manchester coder

reference to recover the transmitted data and put them back in a parallel format. One possible solution is to transmit the clock in a separate LVDS link [6]. This approach limits seriously the transmission speed, as desynchronization between clock and data may easily occur. Another commonly used strategy to recover the clock it is the use of a PLL-based interface. However, it complicates greatly the receiver design and it needs a long synchronization preamble. In the present AER link we have chosen to use a Manchester codification which allows to recover the clock from the data in a simple and fast way. In Manchester code, a transition of the transmitted data always exists in the middle of the clock period. As we ensure that at least one transition exists in each clock period, the receiver can easily extract the clock from the transmitted data. The inconvenience of this approach is that the number of transitions is doubled so that the LVDS driver and receiver have to operate at twice the speed than the bit rate.

Another inconvenience to deal with is that in AER links data are transmitted in bursts. In non-saturated AER-links events are sparse in time and there can be silent periods, so the receiver must be kept synchronized despite there are no transitions in the data line. We need some kind of memory to save the receiver clock frequency during inactive periods.

Another question to take into account is the data alignment; the receiver has to know when the beginning of a new address occurs. To overcome this, a preamble of two bits set to '1' is included at the beginning of each new address.

III. THE TRANSMITTER CIRCUITRY

The left side of Fig. 2 shows the block diagram of the transmitter circuitry. A serializer circuit converts the AER parallel data into a serial bit stream. The serializer also interfaces the protocol signals of Acknowledge and Request of the parallel AER generator. The bit stream going out of the serializer enters into a Manchester coder. Finally, the Manchester coded bit stream is transformed to a differential low voltage signal (LVDS) by the LVDS driver.

Fig. 3 details the schematics of the serializer and the Manchester coder. Upon reception of the request signal, the two-bit counter controlling a 2:1 multiplexer is activated and generates a stream of two '1's', which is the preamble. The signal of count completion of the 2 bit counter activates the 4 bit counter which controls a 16:1 multiplexer that transforms the parallel AER data to a serial format. Manchester encoding
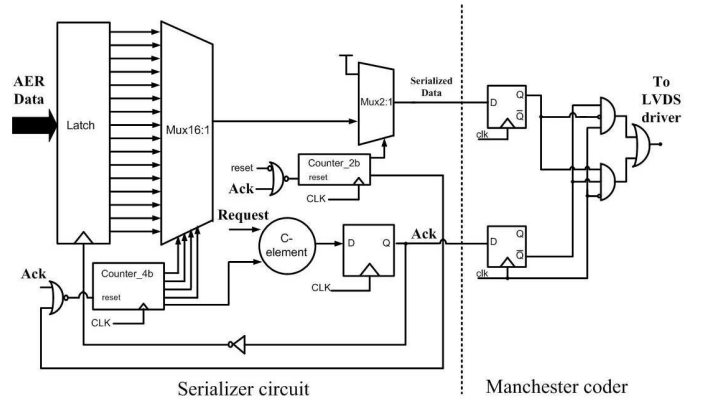
is just a simple XOR operation between the data and the clock used for the synchronization [7]. The Ack signal is generated from the C-element operation of the request and the signal indicating the end of count of the 4-bit counter. That way the Ack signals remains active during all the serialization process indicating the serial-AER generator is busy. The Ack signal is desactivated when the serialization process is completed and the Rqst has been also deactivated.

IV. THE RECEIVER CIRCUITRY

The block diagram of the receiver is shown in the right part of Fig. 2. A LVDS receiver restores the LVDS transmitted signal into a rail-to-rail signal. A clock data recovery circuit (CDR) extracts the original clock from the Manchester coded data. This time reference is used to control a data deserializer that transforms the serial input bit stream into a parallel format. When the deserialization has been completed, the deserializer activates a Rqst signal to indicate the parallel AER decoder that there are new data available to be read.

Fig. 4 [7] shows a more detailed schematic of the CDR and the deserializer circuits, which also decode the data from Manchester format. As we will explain later, the delay inverting units in Fig. 4 are tuned to have a delay of 1/8 the bit time, thus every time a new edge is received in the input, the XOR gate generates a pulse of length $2T_{delay} = T_b/4$. This XOR output signal activates the flip-flop FFC. A delay path of 5 delay inverting units is introduced between FFC output and input. The result is a toggled flip-flop tuned in such way that only those clock edges which are more distant than $5T_{delay}$ will cause the output to toggle its state. Thus, the flip-flop state only changes with the edges located at the middle of the bit period, filtering the possible edges at the beginning of the Manchester bits. As a result, the recovered clock has a period twice the bit time and, therefore, input data must be captured in its falling and rising edges.

The deserializer is composed of two chains of 8-bit shift registers, one controlled by the recovered clock and the other by a inverted version of it. Two more flip-flops are added at registers output to generate the Rqst signal when the preamble reaches them. In this situation, two 1's are stored in the flip-
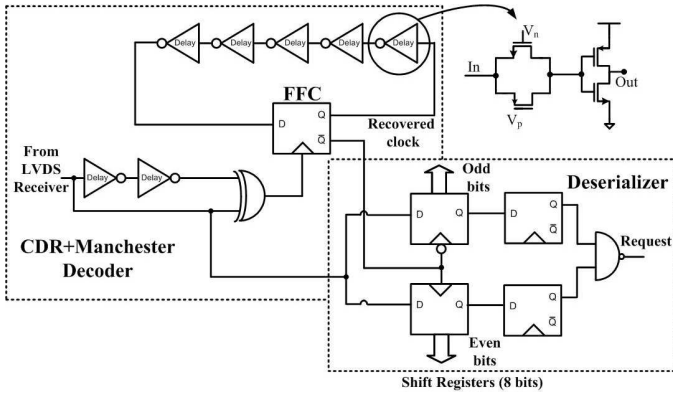
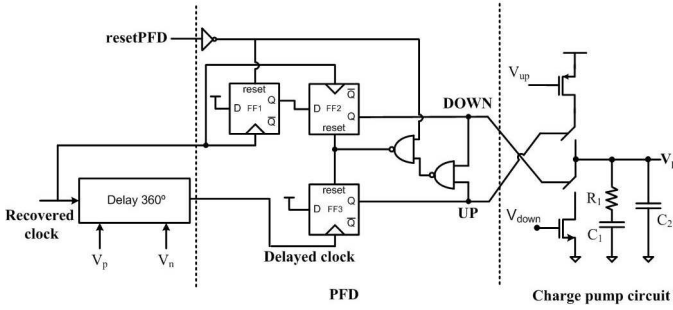Fig. 4. Schematics of CDR+Manchester decoder and Deserializers

Fig. 5. DLL scheme

Fig. 6. DLL modification for burst mode operation

flops, so the NAND gate produces a pulse in Rqst signal, indicating that new data are ready at the parallel output.

The nominal value for the delay elements is around 1/8 of the bit period, but this requeriment is not very strict. 5 delays must introduce a delay longer than half bit period and shorter than one bit period. The delay unit in Fig. 4 is composed by an inverter and a transmission gate. The control voltages V_n and V_p modify the input resistance, thus changing the unit delay.

To fulfill the delay requirements over process and environment variations, the DLL shown in Fig. 5 has been implemented. It generates the output voltage V_p that controls all delay units. The stage named *Delay* 360° is formed by 16 delay-units. Hence, when the delay units are properly tuned to $T_{delay} = T_b/8$, this block is generating a 360°-delayed version of its input. This delayed clock is compared by the Phase and Frequency Detector (PFD) versus the non-delayed clock. The PFD generates signals UP and DOWN. Edges in the recovered clock force a high level in DOWN and edges in the delayed version do the same in UP. When both signals are at high level the logic resets the flip-flops, returning their input to low level. FF1 is only used to avoid the initial comparison because the signals are shifted 360°.

When *Delay* 360° is properly synchronized to a 360° delay, signals UP and DOWN will be at high level during exactly the same time. On the contrary, if the phase shift departs from its 360° target value, signals UP and DOWN will last different time at high level. These signals control a charge pump circuit to generate the control voltage. When UP is activated, the
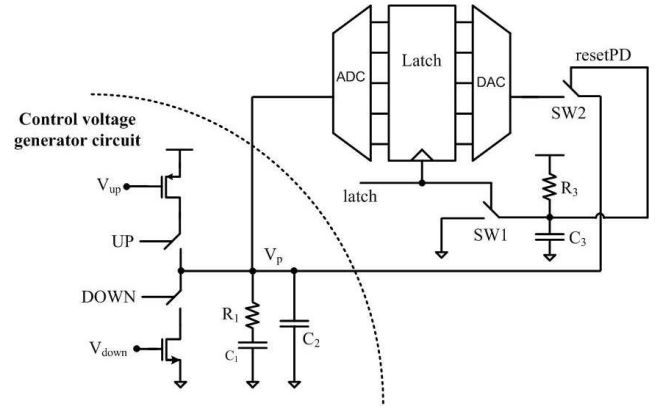
upper current source raises the control voltage V_p. DOWN makes voltage V_p decrease. In the equilibrium, the loop voltage V_p will be almost constant.

V. MODIFICATION FOR BURST MODE OPERATION

Enabling burst mode operation in the AER LVDS link was the main goal on this design. As the AER event flow is not uniform, the recovered clock will be non-uniform as well. The DLL described before needs an input clock with edges to perform its operation. In silent periods the control voltage will evolve uncontrolled and synchronization could be lost, so the receiver would need to begin a new synchronization process. Preserving the DLL state between consecutive transmissions, we can keep the receiver synchronized all the time with a constant value in the delay control voltage V_p.

Fig. 6 shows the modified DLL that memorizes the control voltage in silent periods. Signal latch is the output of the XOR gate in Fig. 4. Hence, when data are arriving this signal is set to a high level with every edge for a time $T_b/4$. In silent periods signal latch is always at low level. Consequently during active periods of the clock signal V_p is periodically sampled by the ADC/latch/DAC block.

During active clock periods signal resetPD is periodically set to ground by switch SW1, thus the output of the ADC/latch/DAC block is not connected to node V_p, as switch SW2 is always open. Node V_p is then driven by the DLL. However, when there is a long pause in the input data, node resetPD begins to be raised towards V_{dd} with a speed given by the time constant of node resetPD $\tau_3 = R_3 C_3$. When a certain threshold value is reached, the analog switch SW2 closes forcing the control voltage value to be the lastest stored value in the ADC/latch/DAC block. This stored value is not updated during silent periods.

The time constant $R_3 C_3$ needs to be designed carefully. It should be much larger than the time between consecutive pulses in signal *latch*, that is the bit period T_b. If we fulfill this requeriment, *resetPD* can never reach the analog switch threshold value during data reception. However, if this value is chosen too large, the circuit will need a long time to react after a pause and errors can occur due to an improper value

at node V_p. For example, for a 1 Gpbs design, a reasonable value for R_3C_3 could be 10 ns. The design requeriments on the ADC-DAC are not very restrictive. The analog signal to convert is almost constant, so the converters do not need to be too fast relaxing their design constraints. Furthermore, great accuracy is not required because the control voltage value is set dinamically by the DLL when new addresses arrive. From our circuit simulations we have concluded that 5 bits are enough for proper operation of the DLL.

VI. Simulation results

The proposed architecture for the LVDS-AER link has been implemented in a 90 nm STMicroelectronics technology and simulated to verify its performance. Simulations showed a power consumption of 27.8 mW for the transmitter (27.4 mW for LVDS driver and 0.4 mW for the serializer) and 12.3 mW for the receiver (11.2 mW for LVDS receiver and 1.1 mW for the deserializer). The LVDS driver and receiver and the basic digital cells have been taken from ST libraries. The transmission channel was composed by a 50 cm cat5E UTP cable connected using 5 cm microstrip PCB traces. To take into account all the high frequency effects, we also modelled the output LVDS pads, the ESD protection circuits and the connectors. The ADC/latch/DAC structure has been modelled with AHDL to change their characteristics and check their impact in the link performance. We have checked the correct operation of the circuit in all the technology corners, in a temperature range of $0 - 80^oC$ and with variations of 5% in the supply voltage.

Fig. 7 shows the operation of the modified DLL when AER burst mode data arrive. We can see that the control voltage (V_p) remains constant while data are being received, but it evolves uncontrolled during a certain time (around 5 ns). This is the time that the receiver needs to force the control voltage value through the ADC/latch/DAC block in the pauses. Signal resetPD presents little variations while data are being received, but it raises to higher values when the input data flow stops. When it reaches the analog switch threshold value, V_p is fixed by the DAC until new AER data arrive.

Fig. 7 also shows the wavefom of the recovered clock. It can be checked that its period is $2T_b$ and it has an edge every time that the input Manchester coded data have a $T_b/2$ transition. That makes this signal suitable to recover the Manchester coded data. We can also observe how the delayed clock is shifted 360° with respect to the recovered clock and the signals UP and DOWN have the same duration at high level, indicating that the loop to control the delay has converged.

Simulations have shown that the initial time required to achieve the convergence is in the worst case around 60 ns. This is the time it takes V_p to converge to its equilibrium value from an initial value of 0V. During this time, V_p does not have an appropiate value and the CDR does not work properly.

VII. Conclusion

A novel serializer structure for AER links based on LVDS standard has been designed on a 90nm technology. The circuit
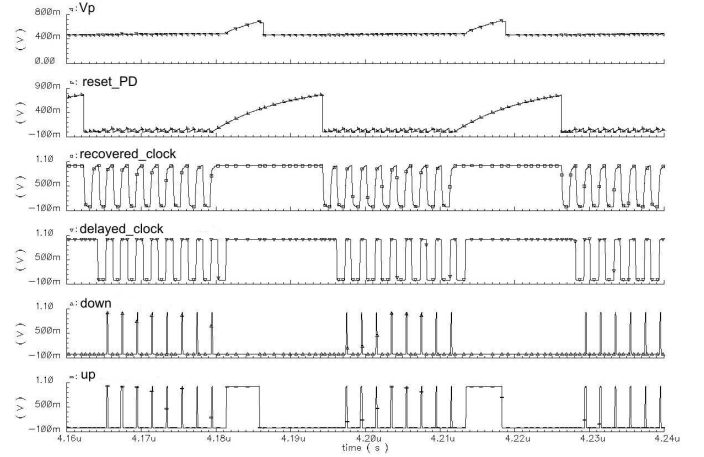

Fig. 7. Delay control loop operation

has been implemented up to the layout level. The serializer and the deserializer takes up an area of 112.35 x 18.14 μm^2 and 117.17 x 25.1μm^2, respectively. This interface can work with different data rates due to the dynamic control of the delay units, it is fully compatible with the LVDS standard and robust against process and environment variations. Moreover the architecture proposed can operate in burst mode keeping the receiver synchronized in the silent periods. This interface will allow the AER system designers to develop more complex and faster applications to make more sophisticated tasks.

References

[1] M.S. Silvilotti, *Wiring considerations in analog VLSI systems with applications to field programmable networks*. Ph. D. Thesis, California Institute of Technology, Pasadena, CA, 1991.

[2] R. Serrano-Gotarredona et al, AER Building Blocks for Multi-Layer Multi-Chips Neuromorphic Vision Systems. *Advances in Neural Information Processing Systems*, NIPS'06, vol.18, pp. 1217-1224.

[3] K. A. Boahen, A burst-mode word-serial address-event link-I, II and III. *IEEE Transactions on Circuits and Systems I*, vol. 51, num. 7, pp. 1269-1300, July, 2004.

[4] ANSI/TEIA/EIA-644-1995, *Electrical Characteristics of Low Voltage Differential Signalling (LVDS) Interface Circuits*. Telecommunications Industry Association, Nov. 15, 1995.

[5] Rafael Serrano-Gotarredona, *AER-Based Bio-Inspired Architecture for Real-Time Image Convolution*. Ph. D. Thesis, University of Seville, 2007.

[6] A. Boni, A. Pierazzi, D. Vecchi, LVDS I/O Interface for Gp/s-per-pin Operation in 0.35 μm CMOS. *IEEE Journal of Solid-State Circuits*, vol. 36, No. 4, pp. 706-711, April, 2001.

[7] P. Popescu, A. Solheim, M. Wight, Experimental Monolithic High Speed Transceiver for Manchester Encoded Data. *Proceedings of the 1995 Bipolar/CMOS Circuits and Technology Meeting*, pp.110-113, October, 1995.

[8] A.L. Coban, M.H. Korogolu, K.A. Ahmed, A 2.5-3.125 Gbps Quad Transceiver with Second-Order Analog DLL-Based CDRs. *IEEE Journal of Solid State Circuits*, vol. 40, No.9, pp. 1940-1947, September 2005.