

A clock-less ultra-low power bit-serial LVDS link for Address-Event multi-chip systems

Ning Qiao

Institute of Neuroinformatics
University of Zurich and ETH Zurich
Zurich, Switzerland
Email: qiaoning@ini.uzh.ch

Giacomo Indiveri

Institute of Neuroinformatics
University of Zurich and ETH Zurich
Zurich, Switzerland
Email: giacomo@ini.uzh.ch

Abstract—We present a power efficient clock-less fully asynchronous bit-serial Low Voltage Differential Signaling (LVDS) link with event-driven instant wake-up and self-sleep features, optimized for high speed inter-chip communication of asynchronous address-events between neuromorphic chips. The proposed LVDS link makes use of the Level-Encoded Dual-Rail (LEDR) representation and a token-ring architecture to encode and transmit data, avoiding the use of conventional large Clock-Data Recovery (CDR) modules with power-hungry DLL or PLL circuits. We implemented the LVDS circuits in a device fabricated with a standard $0.18\mu\text{m}$ CMOS process. The total silicon area used for such block is of 0.14mm^2 . We present experimental measurement results to demonstrate that, with a bit rate of 1.5 Gbps and an event width of 32-bit, the proposed LVDS link can achieve transmission event rates of 35.7 M Events/second with current consumption of 19.3 mA and 3.57 mA for receiver and transmitter blocks, respectively. Given the clock-less and instant on/off design choices made, the power consumption of the whole link depends linearly on the data transmission rate. We show that the current consumption can go down to sub- μA for low event rates (e.g., $<1\text{k Events/second}$), with a floor of 80 nA for transmitter and 42 nA for receiver, determined mainly by static off-leakage currents.

I. INTRODUCTION

The Address-Event Representation (AER) protocol has been widely used in neuromorphic computing systems to connect multiple cores and chips together [1]–[6], in single-chip devices for encoding sensory signals [7] or for implementing spike-based learning mechanisms [8], [9], and in multi-chip sensory-processing systems [10]–[12]. By exploiting the asynchronous principle, the AER protocol is extremely efficient for event-driven neural system in terms of power consumption and low latency. Bit-parallel AER is the most commonly used implementation, due to its ease of design and configuration. This strategy however is not scalable, as the width of the parallel bus and the power required to transmit these parallel event bits scales with the size of the network. This can become a critical issue for large scale neuromorphic systems, which typically employ multiple copies of AER buses for routing events to multiple destinations and receiving events from multiple sources [1], [3], [4]: these systems are normally arranged and tiled in 2D arrays with North-South, East-West, and possibly diagonal Input/Output (I/O) links between them. This requires a very large pin-count and can lead to significant leakage and dynamic power consumption. Rather than using the full parallel

AER protocol, some approaches have resorted to employing a “word-serial” protocol, which groups multiple row addresses for a given column address to reduce pin count [5], [13]. However, it has been argued that one of the most efficient solutions for transmitting AER data in terms of both speed and power consumption, is to use a bit-serial Low Voltage Differential Signaling (LVDS) scheme [14].

Event rates in neuromorphic systems tend to be sparse, but to have high peak values [13]. As the time information is typically important, low latency is an essential requirement. Traditional LVDS schemes are designed for continuous data transmission with power consumption that depends on clock frequency, and independent of the input data rate. In some LVDS schemes it is possible to send idle comma characters and signal a pause in the data transmission. However, these idle states may cause loss of synchronization between transmitter and receiver, and many (e.g., in the order of hundreds) clock cycles are typically required for these lock recoveries. Therefore, as traditional LVDS implementations are likely to cause significant latency for event transmission, they are not suitable for AER neuromorphic systems. Previous approaches have proposed to optimize the Clock-Data Recovery (CDR) scheme so that the phase lock of transmitter and receiver can be recovered on the fly [14], but they required additional clock generation and synchronization circuits, such as DLL or PLL circuits, for the CDR which are very expensive in terms of power and area requirements. The event-based nature of AER data transmission in neuromorphic systems calls for the development of a new fully asynchronous clock-less event-based switchable bit-serial AER LVDS link, that does not need clock recovery circuits. In this paper we propose a new clock-less LVDS scheme optimized for neuromorphic systems, and demonstrate its implementation in a prototype chip, fabricated using a standard $0.18\mu\text{m}$ CMOS process. We show that the chip designed successfully implements the following features:

- 1) Pure asynchronous design without PLL/DLL for CDR.
- 2) Instant on ($<0.5\text{ ns}$) wake-up for new event data and instant off ($<0.5\text{ ns}$) self-sleep in absence of data, maintaining low latency and low power consumption.
- 3) Sub-nW (220 nW) static power consumption and event-rate based dynamic power consumption.
- 4) Compact layout as a building block for multi-core and

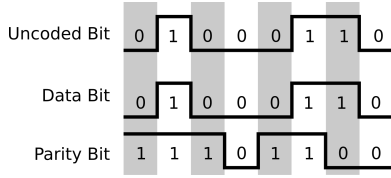


Fig. 1: Encoding example of LEDR. Shaded regions represent the even phase and the non-shaded regions represent the odd phase.

multi-chip neuromorphic systems.

The paper is organized as follows: Section II presents the data transmission scheme and link architecture; Section III describes the circuits implementation of the proposed bit-serial LVDS link; Section IV presents the measurements made with the prototype chip and describes the experimental results; Section V shortly concludes the work.

II. ENCODING SCHEME AND ARCHITECTURE

A. Data encoding

It is possible to implement a clock-less fully asynchronous event-driven bit-serial LVDS link by choosing a proper data encoding scheme that eliminates the need of traditional CDR, which is expensive for asynchronous systems. One scheme that is optimally suited for AER data is the LEDR signaling scheme [15]. In LEDR signaling data bits are encoded using two rails: given a sequence of bits, a data rail is used to represent the bit value and a parity rail is used to represent the parity relative to the encoding phase and data rail. The encoding alternates between an even and an odd phase. In the even phase the parity rail takes the inverted bit value, and in the odd phase the parity rail takes the same bit value of the data rail. Formally the data rail value $D[i]$ and the parity rail value $P[i]$ are:

$$\begin{cases} D[i] = B[i]; & P[i] = \overline{B[i]} & \text{for odd phase} \\ D[i] = B[i]; & P[i] = B[i] & \text{for even phase} \end{cases}$$

where $B[i]$ represents the encoded bit value of the sequence. Figure 1 shows an encoding example for an 8-bit data sequence.

The LEDR is a Delay Insensitive (DI) protocol: sequential bits can easily be distinguished by checking whether $D[i]=P[i]$ or not. So, by encoding address event data strings using LEDR, it is possible to build fully asynchronous bit-serial LVDS links without using a clock generation block or a clock synchronization block for CDR.

According to this LEDR encoding scheme, it is possible to implement both asynchronous encoder and decoder. On the encoder side, the data rail should always take the original sequence bit value while the parity rail should take the inverted sequence bit value for the odd phase, and the original sequence bit value for the even phase. On the decoder side, it is sufficient to check if $D[i] = P[i]$ or $P[i] = \overline{D[i]}$ to determine the bit phase, and then to read incoming bits one by one. This scheme leads to a very compact design in terms of hardware resources. Because LEDR encoding follows a two-phase handshaking (or

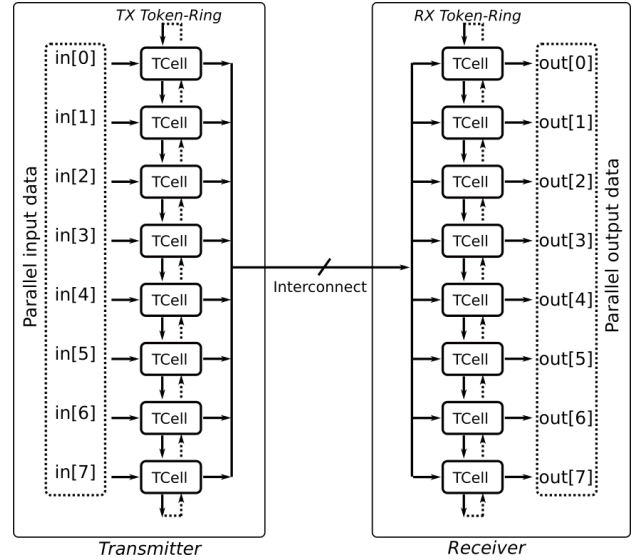


Fig. 2: A typical 8-bit transceiver based on token-ring architecture. Token-cells are labeled as “TCell”.

Non-Return-to-Zero (NRZ)) protocol, it allows a full bit rate and provides a significant bandwidth advantage comparing to alternative schemes based on Phase Encoding (PE) or Dual-Rail Return-to-Zero (DR-RZ) methods.

B. LVDS with token-rings

Token-ring schemes have already been proposed for asynchronous sequential data transmission [16]. A token-ring comprises a number of mutually exclusive token-cells to transmit their data content one by one. Figure 2 shows a typical 8-bit transceiver based on token-ring architecture [16]. Token-cells in the “Transmitter block” are activated sequentially to take one bit at a time from a parallel data bus and to write it on a shared interconnection link. Token-cells in the “Receiver block” take bit values sequentially from the shared bus to reconstruct the parallel data.

A token-ring based serializer can be built following the LEDR scheme to sequentially encode both data and corresponding parity bits from a parallel bus to a shared serial one. Accordingly, a token-ring based de-serializer can be built to de-serialize and decode the data by taking data bit-by-bit from shared data/parity wires. The block diagram of the asynchronous bit-serial LVDS link we propose based on these concepts is shown in Fig. 3. It comprises the following blocks: “Input Buffer”, “TX Token-Ring”, “RX Token-Ring”, “LVDS Drivers”, “LVDS Receivers”, “Output Buffer” and “Control Queue”.

The “TX Token-Ring” block is implemented to serialize and encode event parallel bits into data and parity rails following the LEDR scheme. The “RX Token-Ring” is implemented to de-serialize and reconstruct parallel bits from data and parity rails. The “LVDS Drivers” convert data and parity rails into low-voltage differential signals for low-power consumption and high-speed inter-chip data transmission. Similarly, the “LVDS

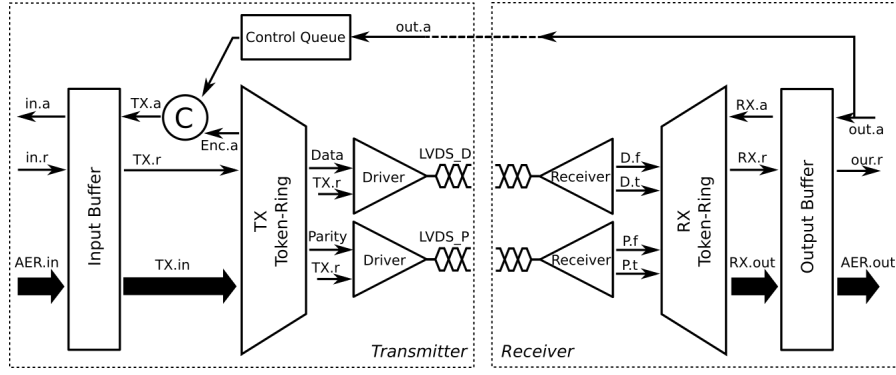


Fig. 3: Architecture of the proposed bit-serial LVDS link.

Receivers” convert LVDS signals back to normal digital signals. In order to minimize power consumption and make it depend only on the event-rate, we propose a novel instant on/off scheme for LVDS Drivers and Receivers, described in the following section. Finally, the data transmission is done in a “burst mode”, such that the acknowledge signal is returned once per address event word, rather than bit-by-bit. Address event input and output buffers are included to pipeline the transmission cycle and increase data depth on both sides. A small “Control Queue” block with the same depth of the output buffer is employed to pre-store multiple acknowledges, so that the transmitter can keep on sending events without waiting for their corresponding acknowledge signals to arrive, in order to minimize latency.

C. Instant On/Off driver and receiver

Instant on/off LVDS drivers and receivers that implement event-driven wake-up and sleep-mode mechanisms are crucial for minimizing consumption in neuromorphic systems that operate with sparse activity and low average event rates. Since the main digital blocks communicate with each other following a four-phase handshaking protocol, no dynamic power is dissipated in idle states. The “LVDS Drivers” can be easily turned on or off by a digital signal, such as $TX.r$ of Fig. 3, as new event data appears on the “Input Buffer” block. In order to turn on/off the “LVDS Drivers” instantly, we exploited the common voltage of LVDS pairs. As shown in Fig. 4, during the idle state, when no data is being transmitted, the two pairs of LVDS signals are both pulled down to Gnd , resulting in a 0V common mode voltage. In this way the “LVDS Receivers”, designed with NMOS input transistors, will be fully tuned off and power consumption will be due only to off-leakage level static power dissipation. As soon as a new event arrives, the common-mode feedback circuit will drive both data pair and parity pair voltage lines back to a $Vref$ common mode voltage, which is set to about 1V in this design. Simultaneously, the differential voltages of data pair and parity pair will recover back to their previous bits value with $D = P$. In this way the “LVDS Receivers” with NMOS input transistors will be turned on and will start to convert the LVDS signals to standard digital ones. Because the first odd token-cell in the decoder will only take data when $P[i] = D[i]$, the receiver will ignore

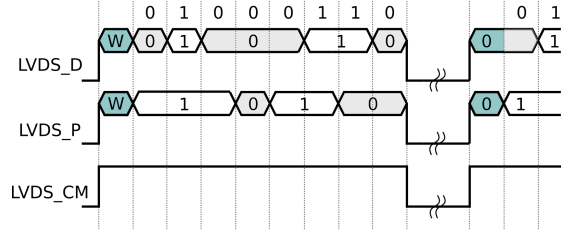


Fig. 4: Proposed signaling scheme with LVDS for data transmission and common-mode voltage for instant on/off receiver.

potential spurious repeated LSB bits until a new MSB bit arrives. After transmitting the full word, the common-mode voltage of the LVDS pairs will be pulled down to Gnd again, turning the receiver off. The recovery speed of common-mode voltage is controlled by a common-mode feedback circuit in the LVDS driver. In our measurement, the recovery latency of common-mode voltage is less than 0.5 ns, which is much shorter than previously reported values (e.g., 6.6 ns in [17]).

D. Transmission Scheme

Figure 5 describes the timing diagram for the transmission of one event in the proposed bit-serial LVDS link. A four-phase handshaking protocol is implemented between “Input Buffer” and “TX Token-Ring”. Once the event data $D_1 < n-1 : 0 >$ appears on the “TX Token-Ring” input bus $TX.in$, the signal $TX.r$ will be set to high by the “Input Buffer”, thus requesting a new data transmission which will trigger the first stage token-cell of “TX Token-Ring” to take the first bit. Meanwhile, this request signal will turn on the LVDS drivers to be ready for sending new data. After a tunable delay t_{wd} , the first odd token-cell of “Token-Ring” will push new bit value $D = D_1 < n-1 >$ and parity $P = \overline{D_1} < n-1 >$ to shared data and parity wires *Data* and *Parity*. It will then enable the following stage, i.e., the first even token-cell to take new bit value. After a cycle delay t_d , the enabled stage will disable previous stage and push new data/parity $D = D_1 < n-2 >$ and $P = D$ on the shared wires, while enabling its following stage. Mutual exclusion is implemented stage by stage till the end the token-ring. After pushing data/parity of the last bit value to shared wires, the

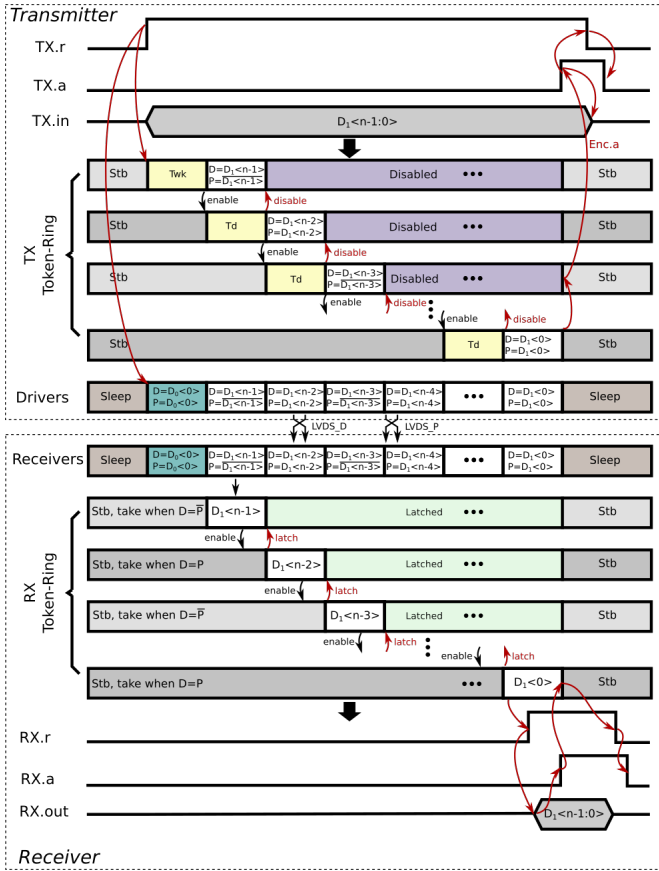


Fig. 5: Event transmission timing diagram of the bit-serial LVDS link. The “Stb” represents the stand-by state.

“TX Token-Ring” block will acknowledge the “Input Buffer” by asserting *Enc.a* to high. Subsequently, *TX.r* will be reset to low for the successful removal of data $D_1 < n - 1 : 0 >$. Finally, *TX.a* will be reset to low to complete the four-phase handshaking cycle.

It should be noted that during the wake-up stage the “LVDS Receivers” will need to be turned on for recovering the common-mode voltage of LVDS pairs with data and parity value $P = D$. The first token-cell of “RX Toke-Ring” will only take data and parity with $P = \overline{D}$. For an event data with even bit width, a safe approach is to fully recover both common-mode and differential values of previous bit by repeating the LSB of the previous event data with data $D = D_0 < 0 >$ and party $P = D_0 < 0 >$.

The mutually exclusive token-cells of “RX Token-Ring” will take data from LVDS receivers bit-by-bit. Each bit cycle is distinguished by either $P = \overline{D}$ or $P = D$. The output of each token-cell is latched, once the current token-cell is disabled by its successor. As soon as the last token-cell gets its bit, it will request “Output Buffer” to take the whole data packet from all token-cells and reset the “RX Token-Ring”. In this design the “RX Token-Ring” is required to have the highest throughput. The tunable delay T_d is added in “TX Token-Ring” to enforce the timing assumptions that “RX Token-Ring” has a higher

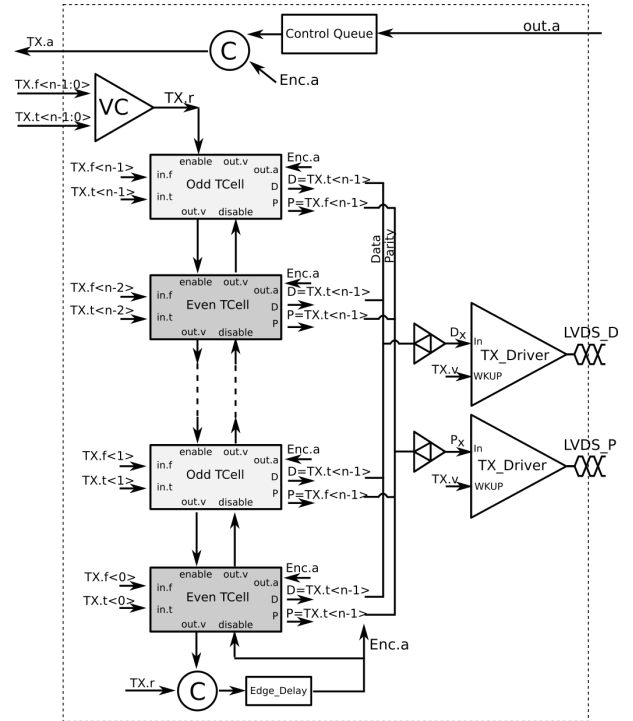


Fig. 6: Transmitter Token-Ring for encoding data into data/phase scheme in proposed bit-serial LVDS link.

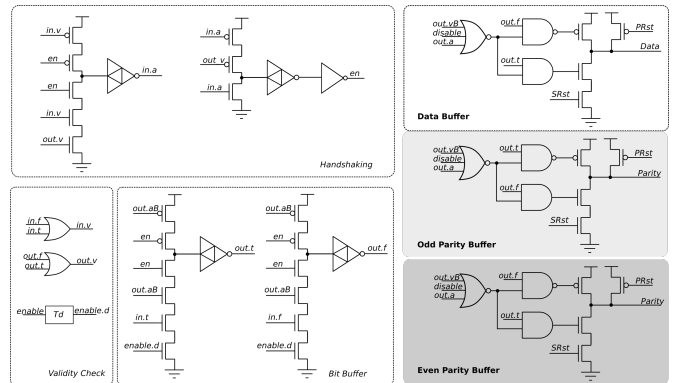


Fig. 7: Circuit implementation of the TX token-cell based on a bit-buffer. Each token cell comprises “Handshaking”, “Validity Check”, “Bit Buffer”, “Data Buffer” and “Odd/Even Parity Buffer” blocks.

throughput than “TX Token-Ring”, to get sequence bit within one TX bit cycle.

III. CIRCUITS IMPLEMENTATION

A. Transmitter Token-Ring

The block diagram of “TX Token-Ring” is shown in Fig. 6. A dual-rail asynchronous protocol and four-phase handshaking are used for processing input data. The “TX Token-Ring” comprises an input “Validity Check” block, “Token-Ring” with odd and even token-cells, “LVDS Drivers” and a “Control Queue” block. The “Validity Check” block first checks and indicates a valid

input event data by $TX.r$. The “LVDS Drivers” can then be turned on by $TX.r$ for a valid input event. Meanwhile, the first token-cell starts to take the first bit value $TX.f < n-1 > / TX.t < n-1 >$ and push relative data and parity outputs to shared wires. For odd bits, data and parity outputs are $D = B$ and $P = \overline{D}$, respectively, while for even bits, they are $D = B$ and $P = D$, respectively. So the first odd token-cell will push $D = TX.t < n-1 >$ to shared data wire and $P = TX.f < n-1 >$ to shared parity wire. After a tunable delay t_{wk} when the first token-cell successfully pushes data and parity value of MSB of input event to shared data and parity wires, the first token-cell will send the *enable* signal to enable its successor for the next bit value. As a response, its successor will send back the *disable* signal as soon as it successfully takes a bit value. After a set “bit cycle” time t_d when the last token-cell pushes its output to shared data and parity wires, *Enc.a* will be asserted to high to reset the whole “TX Token-Ring” and acknowledge “Input Buffer” to erase old data, and will be reset to low to acknowledge that old data has returned to zero ($TX.f < n-1 : 0 > = 0, TX.t < n-1 : 0 > = 0$). At this point the “TX Token-Ring” is free to take new data.

Figure 7 shows the circuit implementation of the proposed token-cell, based on an asynchronous buffer following a dual-rail protocol and four-phase handshaking. The token-cell comprises a “Handshaking” block, a “Validity Check” block, “Bit Buffer”, “Data Buffer” and “Odd/Even Parity Buffer” blocks. The “Validity Check” block checks the validity of input bit value and indicates the state by signal *in.v*. The “Handshaking” block generates the acknowledge signal *in.a* to acknowledge a valid bit input and control signal *en* to enable Bit Buffer block for buffering current input bit value ($en = 1$) or reset the “Bit Buffer” block for the next cycle ($en = 0$). The “Data Buffer” and “Odd/Even Parity Buffer” blocks will convert buffered bit value *out.t* and *out.f* to data and parity value according to LEDR protocol and push them to shared *Data* and *Parity* wires. Once the current token-ring generates a valid output which is indicated by *out.v* = 1, it will enable its successor and disable its predecessor for mutual exclusion.

B. LVDS Drivers

Current mode LVDS Drivers, shown in Fig. 8, are used to convert data and parity value on shared wires to LVDS signals. The “LVDS Driver” is implemented such that it can convert input value D_{in} into LVDS signals for a valid input ($WKUP = 1$) and fully turned off for a standby mode ($WKUP = 0$). When no data is transmitted ($WKUP = 0$), the “CM-FB” block is switched off. The signals *DN* and *D* are then both set to logic “1” to tune off their gating PMOS transistors and tune on their gating NMOS to pull both *LVDS_f* and *LVDS_t* down to *Gnd*, with a common-mode LVDS pair voltage $V_{CM} = 0$. This will switch off its linked “LVDS Receiver” block following the proposed instant on/off scheme. Once there is a valid input ($WKUP = 1$), the “CM-FB” block will supply property common-mode LVDS pair voltage $V_{CM} = V_{ref}$ to switch on the “LVDS Receiver” on the receiver side, and the “Driver” block will start to convert D_{in} into LVDS. The V_{B1} and V_{B2}

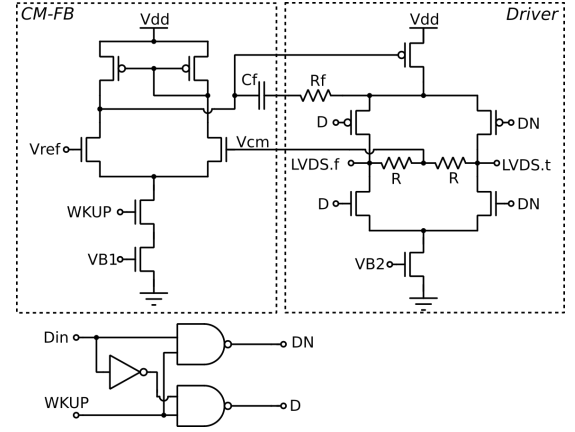


Fig. 8: Circuit implementation of the “TX LVDS Driver”.

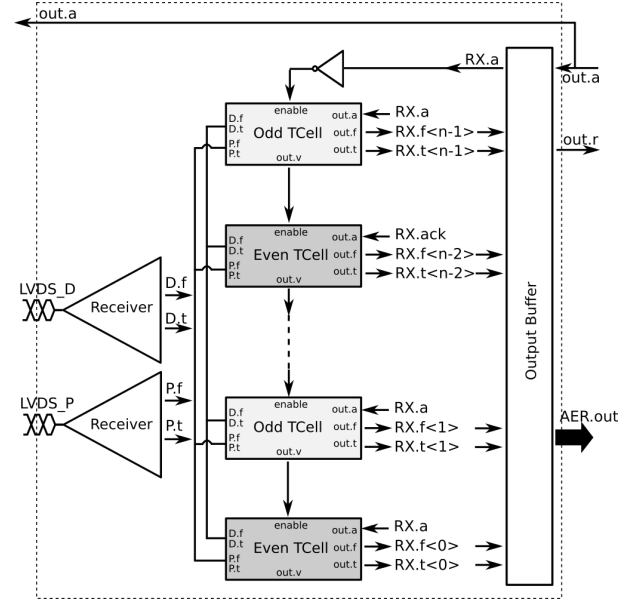


Fig. 9: Receiver Token-Ring for decoding data/phase to event data in proposed bit-serial LVDS link.

signals are biases to generate proper tail currents for the “CM-FB” and “Driver” blocks. Two resistors with value $R = 50\Omega$ (with another two resistors placed at the input terminals of the “LVDS Receiver”) are used to setup differential amplitude of LVDS pair.

C. Receiver Token-Ring

The architecture of the “RX Token-Ring” for processing and decoding LVDS pairs *LVDS_D* and *LVDS_P* is shown in Fig. 9. Following the dual-rail asynchronous protocol and four-phase handshaking, the “RX Token-Ring” comprises “LVDS Receivers”, “Token-Ring” with odd and even token-cells and an “Output Buffer” block. The “LVDS Receivers” first digitize the LVDS pairs *LVDS_D/P* to digital sequential bits *D.f/t* and *P.f/t*, respectively. The token-cells then take bits one-by-one till the end of this event transmission. Once all token-cells take and buffer bits value, the following the “Output buffer”

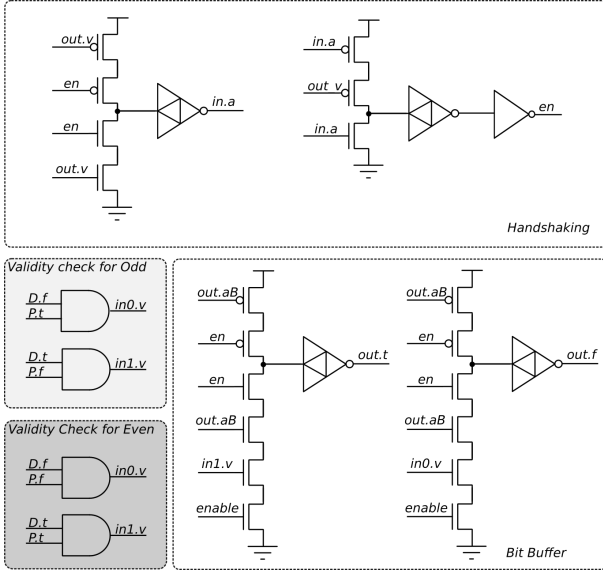


Fig. 10: Circuit implementation of RX token-cell based on 1-bit buffer. Each token-cell comprises Handshaking block, Validity Check block and Bit Buffer block.

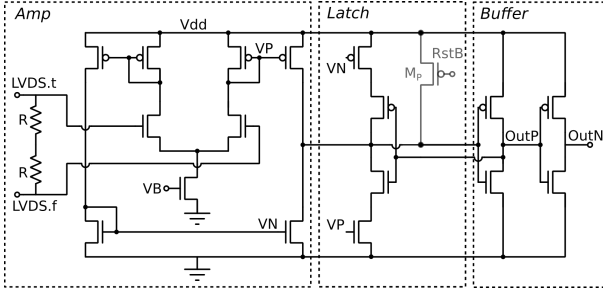


Fig. 11: LVDS Receiver for digitizing differential LVDS to digital signals.

will buffer received event data $RX.f < n - 1 : 0 >$ and $RX.t < n - 1 : 0 >$ to output bus $AER.out$ and reset the “RX Token-Ring” for new data.

The circuit implementation of the “RX token-cell” following the dual-rail protocol and four-phase handshaking is shown in Fig. 10. Each RX token-cell comprises a “Handshaking”, “Validity Check” and “Bit Buffer” block. The odd token-cell will only take and process bit value with $P = \bar{D}$ and even token-cell will only take and process bit value with $P = D$. When new bit value comes to the token-ring with proper data and parity value relationship, for example, $D.f = P.t$ and $D.t = P.f$ for $P = \bar{D}$, the current activated odd token-cell will take this bit value and buffer it with its “Bit Buffer” block. After generating a valid output bit value ($out.v = 1$), internal signal en will be set to logic “0” to latch the output bit value and block it to take new bit value. Meanwhile, this token-cell will enable its following token-cell for a new token.

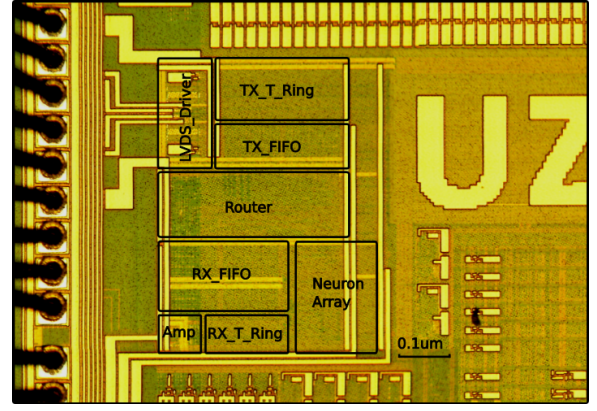


Fig. 12: Die photo of test chip with proposed event-driven bit-serial LVDS link in AMS 0.18 μm 1P6M process, in which TX block occupies an area of 0.08 mm^2 and RX block occupies an area of 0.06 mm^2 .

D. LVDS Receivers

In order to meet the requirement of instant on/off by means of the LVDS common-mode voltage, we implemented the amplifier-based LVDS receivers with NMOS inputs. The circuit implementation of the proposed “LVDS Receiver” is shown in Fig. 11. It comprises an “Amp” block, a “Latch” block and a “Buffer” block. The “Amp” block is responsible for digitizing the LVDS signals. In standby mode, the “Amp” stage will be fully tuned off with $LVDS.f/t = 0$. Once there is data from transmitter side that needs to be transmitted (i.e., when $V_{CM} = V_{ref}$), the “Amp” stage will be tuned on instantly. A latch stage with dynamic biases is implemented to latch the last bit value of previous event data once the “Amp” stage is switched to sleep mode so that the LVDS receiver will not wake up with a random output bit value. After a successful event transmission, V_{CM} of the LVDS pair will be switched from V_{ref} to Gnd , with VP and VN shifting to Vdd and Gnd respectively. This will strengthen the drive ability of the latch stage to store the current bit value when the “Amp” stage is turned off. As new event data arrives, the signals VP and VN will be shifted to near $Vdd/2$ to tune the latch stage weaker, so that it can be modified by the new data. An active-low reset signal $RstB$ is used to reset the circuit outputs to a proper initial condition ($P = D$) when powering up the chip.

IV. EXPERIMENTAL RESULTS

The proposed fully asynchronous event-driven bit-serial LVDS link was implemented using a standard $0.18 \mu\text{m}$ 1P6M CMOS process, occupying a silicon area of 0.14 mm^2 . Figure 12 shows the die photo of the fabricated test chip. The whole “Transmitter” block including the “TX_Buffer” occupies an area of 0.08 mm^2 , and the “Receiver” block including the “RX_Buffer” occupies an area of 0.06 mm^2 . Additionally, a small spiking neural array with tunable output event rate is implemented to provide events for testing. A 32-bit router is implemented for routing events from Receiver to Transmitter

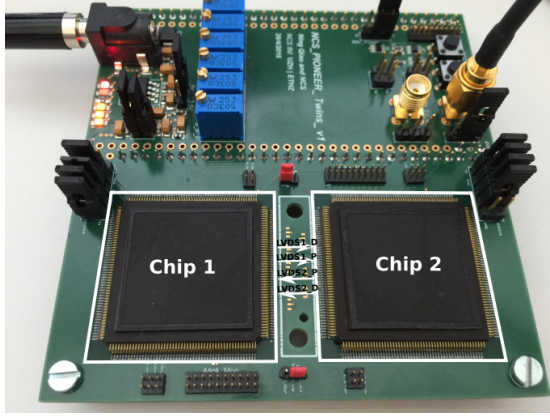


Fig. 13: The setup for testing LVDS links between two chips for bidirectional communication.

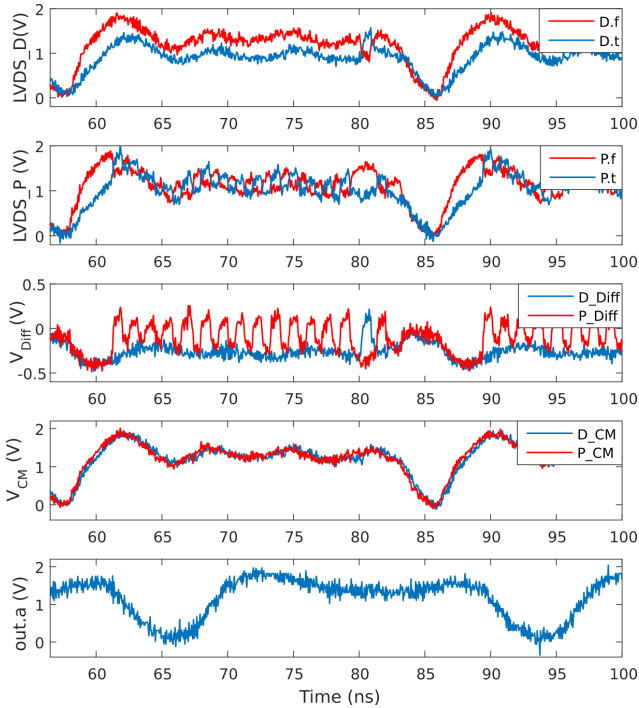


Fig. 14: Transient signals of LVDS pairs captured on the receiver's inputs: the traces $D.f$ and $D.t$ represent the differential signals for $LVDS_D$; the traces $P.f$ and $P.t$ represent the differential signals for $LVDS_P$; The D_Diff and P_Diff traces are differential voltages of two LVDS pairs; The D_CM and P_CM traces represent the common voltages of the two LVDS pairs; The last plot shows the RX_Ack signal, which is the acknowledge signal from the target chip to the source chip, representing a successful event transmission.

to realize a transmission loop between 2 chips to explore peak transmission throughput.

Figure 13 shows a setup with two chips placed side-by-side for the experiments. With this setup, we transmitted sequences of 32-bit AER events bi-directionally between two chips, through four LVDS pairs: The signals $LVDS1_D$ and

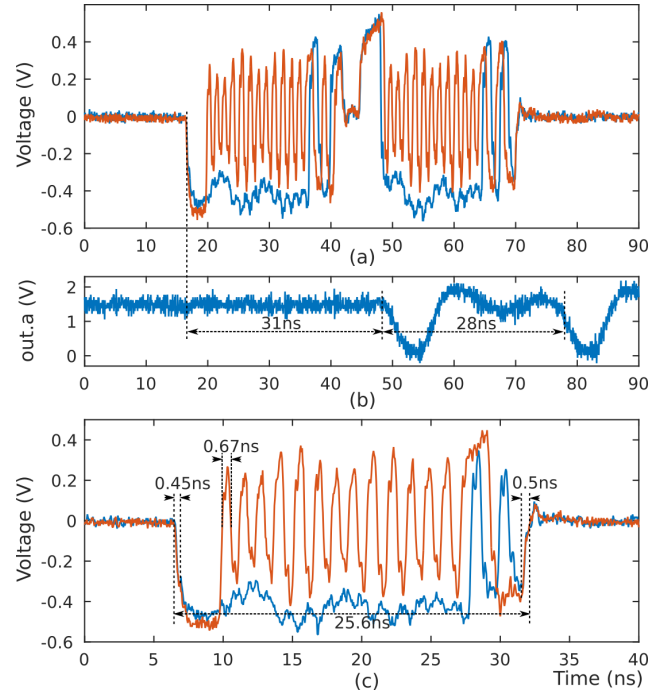


Fig. 15: Transient signals of LVDS pairs at receiver inputs: (a) differential mode of LVDS signals, (b) acknowledge signal from the receiver, (c) details of single event transmission signals.

$LVDS1_P$ were used to transmit events from Chip1 to Chip2, and $LVDS2_D$ and $LVDS2_P$ were used to transmit events from Chip2 to Chip1.

Transient Signals of LVDS pairs were observed and captured using a Tektronix DPO7000 Oscilloscope, from the input terminals of the “LVDS Receivers”. As shown in Fig. 14, the $LVDS_D$ plot shows data from the LVDS pair with differential signals $D.f$ and $D.t$. The $LVDS_P$ plot shows the parity LVDS pair with differential signals $P.f$ and $P.t$. The D_Diff and P_Diff traces in the V_{Diff} plot are the differential voltages of the data LVDS and parity LVDS pairs, respectively. The D_CM and P_CM traces in the V_{CM} plot are the common voltage of data LVDS and parity LVDS pairs, respectively. The $out.a$ plot shows the acknowledge signal from the target receiver chip for acknowledging a successfully event transmission. Sequence bits are presented bit-by-bit following the LEDR protocol, via the data and parity differential signals D_Diff and P_Diff . The common-voltages of the two pairs D_CM and P_CM are reset to Gnd at the end of a successful event transmission and are quickly recovered with new coming events. During the recovery of common-mode voltages of the LVDS pairs, the LSB of previous event with $P = D$ is repeated for sufficient long time to guarantee that the receiver is fully and successfully switched on.

Figure 15 shows the transient signals of the LVDS pairs at the input terminals of receiver chip, captured by an LVDS probe TektronixP6880. The bit cycle is set to be around 0.67 ns by tuning the delay cell t_d in the “TX Token-Ring” to achieve a bit-rate of 1.5 Gps. The observed switch on/off speed of the receiver

TABLE I: Performance comparison of LVDS transceiver VLSI implementations.

	[16]	[18]	[14]	This work
Technology	0.18 μm	90 nm	0.35 μm	0.18 μm
Power Supply	1.8 V	1 V	3.3 V	1.8 V
Area	0.016 mm^2	0.09 mm^2	0.352 mm^2	0.14 mm^2
Clocked CDR	No	Yes	Yes	No
Bit Rate	3 Gps	1 Gps	0.64 Gps	1.5 Gps
Event Rate	-	29.4 MEvent/s	13.7 MEvent/s	35.7 MEvent/s
P_{max}	77 mA	40.1 mA	15.9 mA	22.9 mA
P_{min}	-	40.1 mA	0.4 mA	0.122 μA
P_{max}/P_{min}	-	1	39	187.7k

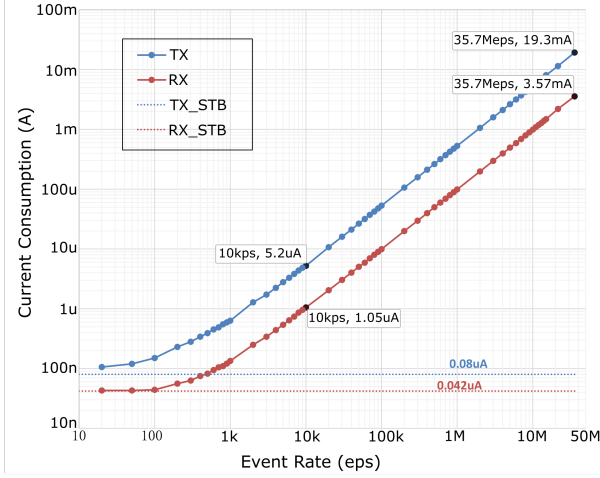


Fig. 16: Power consumption of asynchronous serial-bit LVDS link.

is approximately 0.45 ns and 0.5 ns, respectively, leading to a smaller latency for Address-Event (AE) transmission. As measured in Fig. 15(b), the latency needed for a successful transmission between chips (from switching on the Receiver to getting acknowledge signal *out.a* from receiver chip) for a 1.5 Gps bit-rate is 31 ns. The period of a successive events transmission is 28 ns. Since the transmitter has locally pre-stored the “out.a” signal in the “Control Queue” block (see Section II-B), it will keep on sending event data without waiting for acknowledge signals from the receiver chip until the “Control Queue” is fully empty, to further decrease latency. This is evidenced in Fig. 15(a) and (b), as the second event transmission happens before the arrival of first acknowledge signal *out.a*. In Fig. 15(c) we can observe that, for transmitting a 32-bit event data with a bit-rate of 1.5 Gps, the LVDS link will only be switched on for 25.6 ns, and will be switched off instantly on both transmitter and receiver sides, leading to a pure event-rate related power consumption.

In Fig. 16 we plot the measured power consumption for different event transmission rates. The peak event rate that can be achieved in our experimental setup is 35.7 MEvents/second (32-bit) with current consumption of 19.3 mA and 3.57 mA for transmitter and receiver part, respectively. The power consumption of both transmitter and receiver part scales linearly with the event transmission rate. At a 10k event rate, the power

consumption of the transmitter and receiver blocks are 5.2 μA and 1.05 μA , respectively. The power consumption can further go down to sub- μA for a lower event rates (<1k Events/second), with a floor of 80 nA for transmitter and 42 nA for receiver which is mainly dominated by leakage current of circuits.

Table I shows a performance comparison between different designs. However, area and power consumption of CDR circuits employed in the designs of [14], [18] are not reported. So it may be that significant additional silicon area and power consumption are required for those designs.

V. CONCLUSIONS

While neuromorphic electronic systems have the potential of solving the memory bottleneck problem [19], by construction they also face an important I/O bottleneck problem: large scale neuromorphic system implementations are typically composed of multiple cores and/or multiple chips tiled together, with grid-like communication networks. To transmit address-events across these cores and chips and to sustain the required bandwidth, current implementations use multiple parallel AER buses (e.g., for North-South, East-West, and possibly diagonal links). In this paper we argued that full parallel or even word-serial AER protocols are not scalable, as they require large number of pins/pads and large power consumption to quickly charge and discharge all these lines. To solve this problem, we proposed an ultra low-power fully asynchronous event-driven instant on/off bit-serial LVDS link, which is suitable for AER transmission in neuromorphic multi-chip systems. The proposed LVDS link uses LEDR encoding and a token-ring architecture to eliminate the need for clock-based CDR blocks with expensive on chip DLL/PLL circuits, leading to a very compact and low-power circuit implementation. A novel scheme is proposed to implement a low-latency event-driven transmission with sub-ns instant on/off feature. Experimental results demonstrate how the proposed bit-serial LVDS link can achieve an event rate of 35.7 MEvents/second with a bit-rate of 1.5 Gps. The power consumption of the proposed LVDS link is pure rate-dependent, with a sub- μA power consumption for low event rates (e.g., \approx 1k Events/second).

ACKNOWLEDGMENT

This work is supported by the EU ERC grant “NeuroP” (257219) and by the EU ICT grant “NeuRAM³” (687299).

REFERENCES

- [1] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs)," *Biomedical Circuits and Systems, IEEE Transactions on*, pp. 1–17, 2017.
- [2] J. Park, T. Yu, S. S. Joshi, C. Maier, and G. Cauwenberghs, "Hierarchical address event routing for reconfigurable large-scale neuromorphic systems," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2016.
- [3] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Risk, R. Manohar, and D. S. Modha, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, Aug 2014.
- [4] S. Furber, F. Galluppi, S. Temple, and L. Plana, "The SpiNNaker project," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 652–665, May 2014.
- [5] B. V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. R. Chandrasekaran, J. Bussat, R. Alvarez-Icaza, J. Arthur, P. Merolla, and K. Boahen, "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 699–716, 2014.
- [6] S.-C. Liu, T. Delbruck, G. Indiveri, A. Whatley, and R. Douglas, *Event-based neuromorphic systems*. Wiley, 2014.
- [7] S.-C. Liu and T. Delbruck, "Neuromorphic sensory systems," *Current Opinion in Neurobiology*, vol. 20, no. 3, pp. 288–295, 2010.
- [8] N. Qiao, H. Mostafa, F. Corradi, M. Osswald, F. Stefanini, D. Sumslawska, and G. Indiveri, "A re-configurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses," *Frontiers in Neuroscience*, vol. 9, no. 141, 2015.
- [9] M. Giulioni, P. Camilleri, M. Mattia, V. Dante, J. Braun, and P. D. Giudice, "Robust working memory in an asynchronously spiking neural network realized in neuromorphic VLSI," *Frontiers in Neuroscience*, vol. 5, no. 149, 2012.
- [10] E. Neftci, J. Binas, U. Rutishauser, E. Chicca, G. Indiveri, and R. Douglas, "Synthesizing cognition in neuromorphic electronic systems," *Proceedings of the National Academy of Sciences*, vol. 110, no. 37, pp. E3468–E3476, 2013.
- [11] R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gómez-Rodríguez, L. Camunas-Mesa, R. Berner, M. Rivas-Perez, T. Delbruck, S.-C. Liu, R. Douglas, P. Häfliger, G. Jimenez-Moreno, A. Civit-Ballcells, T. Serrano-Gotarredona, A. Acosta-Jiménez, and B. Linares-Barranco, "CAVIAR: A 45k neuron, 5M synapse, 12G connects/s aer hardware sensory-processing-learning-actuating system for high-speed visual object recognition and tracking," *IEEE Transactions on Neural Networks*, vol. 20, no. 9, pp. 1417–1438, September 2009.
- [12] E. Chicca, A. Whatley, P. Lichtsteiner, V. Dante, T. Delbruck, P. Del Giudice, R. Douglas, and G. Indiveri, "A multi-chip pulse-based neuromorphic infrastructure and its application to a model of orientation selectivity," *IEEE Transactions on Circuits and Systems I*, vol. 5, no. 54, pp. 981–993, 2007.
- [13] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240×180 130 dB 3 μ s latency global shutter spatiotemporal vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014.
- [14] C. Zamarreño-Ramos, R. Kulkarni, J. Silva-Martínez, T. Serrano-Gotarredona, and B. Linares-Barranco, "A 1.5 ns OFF/ON switching-time voltage-mode LVDS driver/receiver pair for asynchronous AER bit-serial chip grid links with up to 40 times event-rate dependent power savings," *Biomedical Circuits and Systems, IEEE Transactions on*, vol. 7, no. 5, pp. 722–731, 2013.
- [15] M. E. Dean, T. E. Williams, and D. L. Dill, "Efficient self-timing with level-encoded 2-phase dual-rail (LEDR)," in *Proceedings of the 1991 University of California/Santa Cruz conference on Advanced research in VLSI*. MIT Press, 1991, pp. 55–70.
- [16] J. Teifel and R. Manohar, "A high-speed clockless serial link transceiver," in *Asynchronous Circuits and Systems, 2003. Proceedings. Ninth International Symposium on*. IEEE, 2003, pp. 151–161.
- [17] C. Zamarreño-Ramos, T. Serrano-Gotarredona, and B. Linares-Barranco, "A 0.35 μ m sub-ns wake-up time ON-OFF switchable LVDS driver-receiver chip I/O pad pair for rate-dependent power saving in AER bit-serial links," *Biomedical Circuits and Systems, IEEE Transactions on*, vol. 6, no. 5, pp. 486–497, 2012.
- [18] C. Zamarreño-Ramos, R. Serrano-Gotarredona, T. Serrano-Gotarredona, and B. Linares-Barranco, "LVDS interface for aer links with burst mode operation capability," in *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*. IEEE, 2008, pp. 644–647.
- [19] G. Indiveri, and S.-C. Liu, "Memory and information processing in neuromorphic systems," in *Proceedings of IEEE*, vol. 103, no. 8, pp. 1379–1397, 2015.