# High Rate Wave-pipelined Asynchronous On-chip Bit-serial Data Link

Rostislav (Reuven) Dobkin, Yevgeny Perelman, Tuvia Liran, Ran Ginosar, Avinoam Kolodny
*VLSI Systems Research Center, Technion—Israel Institute of Technology, Haifa 32000, Israel*
*rostikd@tx.technion.ac.il*

## Abstract

*A high data rate asynchronous bit-serial link for long-range on-chip communication is presented. The data bit cycle time is equal to a single gate delay, enabling 67Gbps throughput in 65nm technology. The serial link incurs lower power and area costs relative to bit-parallel communications, and enables higher tolerance to PVT variations relative to synchronous links. The link uses differential dual-rail level encoding (LEDR) and current mode signaling over a low-crosstalk interconnect layout. Novel circuits used in the link are described, including a novel splitter shift register, a fast LEDR encoder, a high-speed toggle element, a channel driver with adaptive control and a differential channel receiver.*

## 1. Introduction

Global on-chip interconnect does not scale with technology, calling for new solutions that mitigate its growing latency, throughput and power requirements [1][2]. In addition, as Systems-on-Chip (SoC) integrate an ever growing number of modules, on-chip inter-modular communications become congested and modules may require serial interfaces, similar to the trend from parallel to serial inter-chip interconnects.

Typical long-range communication is mostly based on bit-parallel data links that provide high data rates at the cost of area, routing difficulty, noise and power. Such links are often utilized during only a small portion of the time, but dissipate leakage power at all times. Leakage is incurred at the line drivers and also at the repeaters, which are often necessary for long interconnects [3][4]. Parallel link performance is bounded by available clock rate, clock skew, delay uncertainty due to process variations, crosstalk noise, and layout geometries.

Bit-serial communication offers an alternative to bit-parallel interconnects, mitigating the issues of area, routability, and leakage power, since there are fewer wires, fewer line drivers, and fewer repeaters [5][6]. However, simple synchronous serial links are unable to match parallel link throughput. Consequently, novel serial links that can provide high throughput are of great interest.

High-speed serial circuits, having data cycle of a few gate delays (down to single gate-delay cycle), have been recently proposed [5]—[15]. These fast circuits exploit wave-pipelining, low-swing differential signaling, fast clock generators and asynchronous protocols.

Very fast circuits for high rate serial communications were previously explored. Wave-front train serializers presented in [15][16] are based on chains of MUXes. The single-ended link employs wave-pipelining and achieves data cycle of approximately $7 \cdot d_4$ (3Gbps@180nm), where $d_4$ is a single inverter FO4 delay. Low-voltage differential pairs for on-chip serial interconnect was discussed in [13][14] and a three level voltage swing was presented in [17], requiring non-standard amplifiers. In [12], circuits that had been originally designed for off-chip communications [18][19] were adopted for on-chip serial link. The link employs an output-multiplexed transmitter and a multiplexed receiver, requiring clock calibration at the receiver side (both transmitter and receiver use multi-phase DLL circuits). A fabricated circuit demonstrated $2 \cdot d_4$ data cycle and operation over 3mm with 8-bit words.

One gate delay data cycle serial structure was presented in [8][11] employing wave-pipeline operation inside the transmitter and receiver and over the channel. The transmitter and receiver employ fast shift-registers and a wave-pipeline control. The fast control limited the maximal word length due to signal degradation over long shift registers, incurring also excessive power.

In this paper we improve on the original work of [8] analyzing long-word and power issues. A new 'splitter' architecture is presented, which allows serialization of

long words at reduced power dissipation. We describe a novel layout of the channel wires for cross-talk noise reduction, as well as high-speed current-mode differential driver and receiver circuits that achieve channel signaling at the targeted speed of 67Gbps (at 65 nm process).

The rest of this paper is organized as follows. In Sect. 2 high rate serial link is briefly presented. Sect. 3 details the LEDR encoding and its possible decomposition. New serial link transmitter and receiver architectures are presented in Sect. 4. We analyze the wave-pipeline control performance in Sect. 5. New Splitter Shift-Register is presented in Sect. 6, while channel interconnect and channel driver and receiver circuits are detailed in Sect. 7. In Sect. 8 we show the simulation results.

## 2. High-rate serial communication

High-rate serial links require less interconnect area and incur less routing congestion relative to parallel links. For links longer than a certain length, the serial link outperforms the parallel link in terms of active area, leakage and dynamic power [6]. The relative improvement grows with technology scaling [6], as shown in Figure 1 and Figure 2 for a single gate delay serial link. The figures show the link length at which single gate delay cycle serial link becomes superior to a parallel link in terms of leakage power (Figure 1) or dynamic power (Figure 2), assuming word width $N$=8, equal bit-rate and fully-shielded parallel link with $8 \cdot d_4$ clock cycle. The serial link should employ the higher metal layers, which offer lower resistance.

This paper addresses design issues for the architecture of a single gate delay cycle serial link shown in Figure 3 [8]. The serial link (Figure 3) employs low-latency synchronizers at the source and sink [20], two-phase non return to zero (NRZ) Level Encoded Dual Rail (LEDR) asynchronous protocol (allowing non-uniform delay intervals between successive bits) [21][22], serializer and de-serializer using fast asynchronous shift-registers [8], line drivers and receivers, and differential channel encoding (P and S are differential pairs). Acknowledgment is returned

only once per word, rather than bit by bit, enabling multiple bits in a wave-pipelined manner over the serial channel. The wires are designed as wave-guides, enabling multiple simultaneously traveling signals.

Special high speed asynchronous circuits are required to operate at a single gate-delay data cycle. In this paper we discuss design issues of these circuits, focusing on the critical parts – wave-pipeline control and the interconnect. We first discuss LEDR encoding, the transmitter and the receiver, and then turn to circuit details.
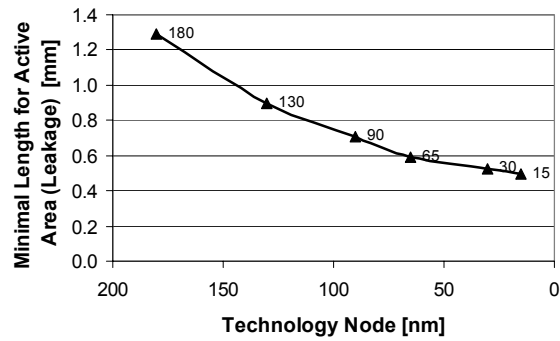


**Figure 1. Minimal length for serial link employment: At longer lengths, the serial link incurs lower active area / leakage power than a parallel link [6]**
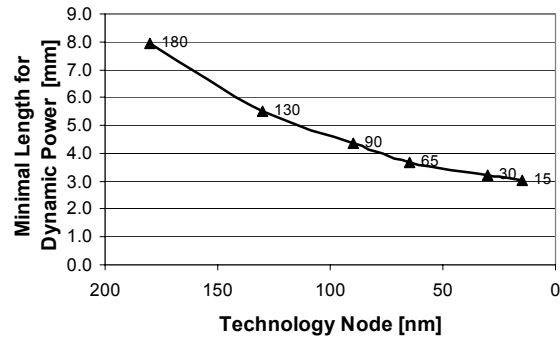


**Figure 2. Minimal length for serial link employment: At longer lengths, the serial link dissipates less dynamic power than a parallel link[6]**
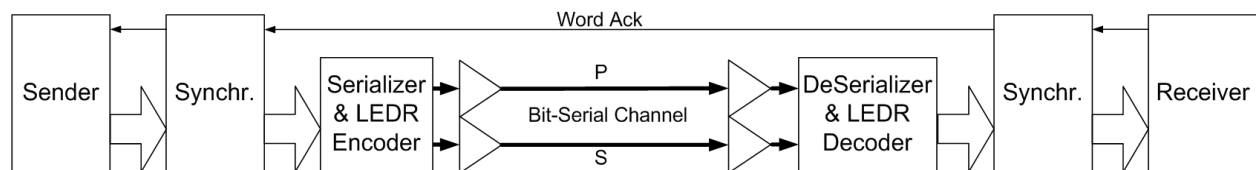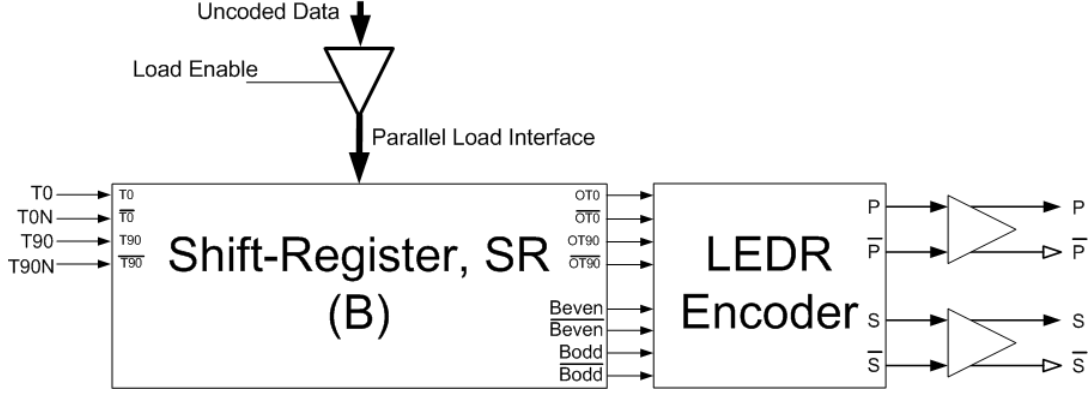


**Figure 3. Serial communication scheme**

**Figure 4. Serializer and LEDR encoder**

## 3. LEDR encoder

The LEDR code [21][22] is defined as follows. A bit sequence B(i) is encoded into bit sequences S(i), P(i) of State and Phase bits, respectively. S(i)=B(i) for all i. If B(i)=B(i-1) then P(i) is the inverse of P(i-1), otherwise P(i)=P(i-1) (Eq. (1)).

$$P(i) = \begin{cases} \overline{P(i-1)}, & B(i) = B(i-1) \\ P(i-1), & B(i) = \overline{B(i-1)} \end{cases} \quad (1)$$

For use in the high-rate serial link, a low latency implementation of P(i) is required. The boolean expression for P(i) is:

$$P(i) = \overline{B(i) \otimes P(i-1) \otimes B(i-1)} \quad (2)$$

Assuming any given starting state, e.g., P(0)=0, B(0)=0, we obtain a closed form solution of the recurrence expression:

$$P(i) = \begin{cases} \overline{B(i) \otimes B(i-1) \otimes B(i-1)}, & i \quad odd \\ \overline{B(i) \otimes \overline{B(i-1)} \otimes B(i-1)}, & i \quad even \end{cases} =$$

$$= \begin{cases} \overline{B(i)}, & i \quad odd \\ B(i), & i \quad even \end{cases} \quad (3)$$

$$S(i) = B(i) \quad \forall i \quad (4)$$

Thus, the LEDR encoder is a very simple circuit consisting of N/2 inverters and having latency of a single inverter. It is thus appropriate for use in a single gate delay high-rate asynchronous serial link. In the following sections we show how this simple implementation is simplified even further, enabling a smaller area and fewer gates implementation relative to the previously published shift register (SR) [8].

## 4. Transmitter and receiver

While the previously published serial link SR [8] encoded the parallel data before serializing it, in the present design the order is reversed and the data is first serialized and then encoded (Figure 4). This results in reducing the number of shift registers in the transmitter from two to one. The *N*-bit splitter shift-register (further discussed in Section 6 below) comprises two *N*/2-bit internal shift-registers, one for the even bits and the other for the odd bits of the original word. The bits are shifted 90 degrees in phase (data is shifted on both the rising and falling edges of T0, T90). The control signals (T0/T0N, T90/T90N) are generated by a multi-phase transition signal generator [18].

The transmitter operates as follows. A parallel *N*-bit word, preceded by a constant '1' start bit, is loaded into the shift-register. Subsequently, for each incoming transition on T0 or T90, SR shifts one bit out on either $B_{EVEN}$ or $B_{ODD}$ outputs, respectively. Each output bit B(i) is encoded by the LEDR Encoder according to Eq. (3) and Eq. (4). The odd and even bit streams are combined into a single stream inside the LEDR Encoder. The data cycle at the output of the encoder is single gate-delay, while the data cycle in each of the internal shift-registers is twice longer.

The encoded data (P/PN, S/SN) drive the differential channel drivers. The internal structure of the shift-register and LEDR are discussed in Section 6 and the drivers are described in Section 7.

The de-serializer (Figure 5) consists of a dual-rail XOR gate for transition detection [23], a shift-register for data sampling and storage and a parallel-load output register. The XOR gate [23] is based on pass transistors for low and symmetric delay.

In LEDR encoding only one of the differential pairs at the input (S/SN, P/PN) makes a transition per bit. Each transition on either of the two differential pairs is translated into a transition on the C/CN control lines.
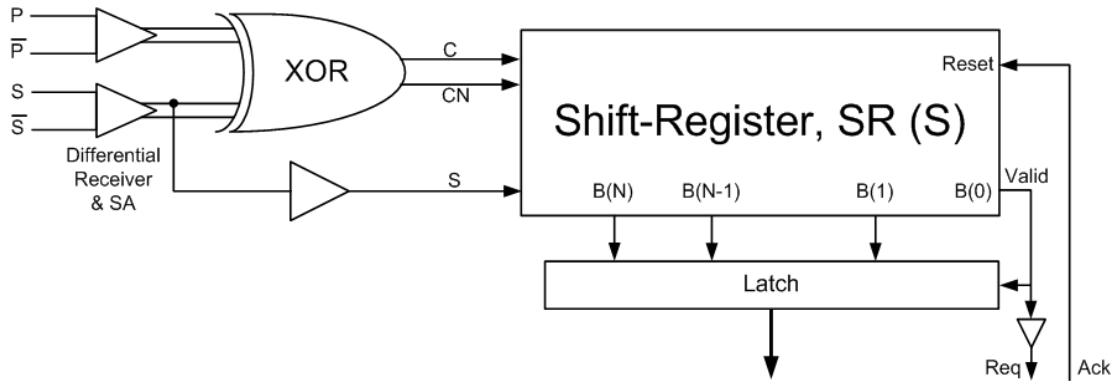
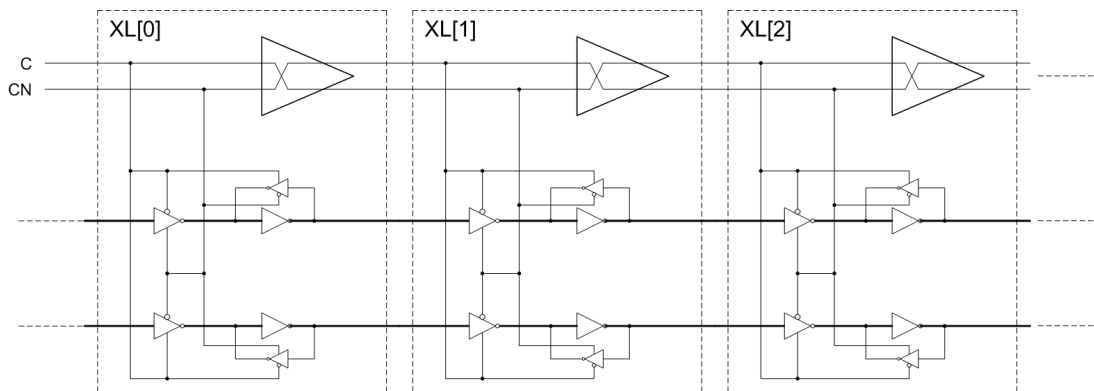**Figure 5. De-serializer & decoder module**



**Figure 6. One gate-delay shift-register, basic architecture [8]**

Each transition causes a single shift of the entire shift-register. Thus, the dual rail XOR retimes the input data. The start bit alleviates the need for a completion detection SR [8].

LEDR decoding (S(i)=B(i) ∀i; Eq. (4)) is combined with de-serialization. Once a transmission is completed ("Valid" switches to high), the SR contains a decoded word that is sampled into the output register by the "Valid" signal. The Valid signal is produced when the '1' start bit passes through the SR [15].

The de-serializer asynchronous SR is based on a novel 'splitter' architecture designed to provide very high throughput, storing the data and accommodating varying delays between successive bits. Section 6 describes the splitter architecture and its operation in detail.

## 5. Wave-pipelined control characteristics

The previously published shift-register [8] comprises transition latches XL (Figure 6). Each XL comprises a differential control signal C/CN, a dual-rail inverting buffer for the controls, and two separate data paths including tri-state buffers and internal latches. Each latch consists of an inverter and a (weak) keeper. The differential control lines C/CN are connected to the tri-state buffers such that when one buffer is active, the other one in the same XL is in high-Z state, and the situation is reversed in the next XL. The data is divided between two data paths, where the bottom path holds the even bits and the upper one holds the odd bits. The control transitions on C/CN propagate without stopping through the control wave-pipeline shifting the data in the pipe. Note that the data is shifted on both the rising and falling edges of C/CN.

A simple implementation of the wave-pipelined control dual-rail inverting buffer consists of two inverters. The large-signal behavior of a single inverter as function of frequency is depicted in Figure 7, where the different lines represent different unbiased input swings. Biased input swing causes stronger signal degradation (Figure 8). The behaviors were obtained by SPICE simulations, measuring amplitude ratio between the device output and input. The wave-pipelined control can operate at the highest speed of

the single gate-delay cycle, namely around the pole of the Bode diagram (~33GHz in Figure 7). This operating point results in signal degradation along the inverter chain, as shown in Figure 9. That figure was drawn by repeating the Bode analysis on inverter chains of different lengths. The figure shows the inverter chain -3dB cut-off frequency as a function of the inverter chain length. For operating at 33GHz or higher, the inverter chain is limited to about 17 stages or less. In the following section we propose a novel splitter architecture that allows working with any word size and is more resilient to in-die variation.

## 6. Splitter shift-register architecture

The splitter architecture (Figure 10) is designed to lower the data rate of the shift-register wave-pipelined control, while maintaining one gate-delay data cycle at the output of the transmitter (Figure 10a) and input of the receiver (Figure 10b). This is similar to wagging FIFOs [9][10]. This goal is achieved by splitting the original $N$-bit shift-register (Figure 6) into two $N$/2-bit shift-registers. One of the two sub-shift-registers stores odd bits and the other stores the even bits. Note that the basic SR, as shown in Figure 6, already contains two parallel data paths; the splitter architecture employs two such SRs. Each of the $N$/2-bit SRs operates at half speed of the original $N$-bit SR, eliminating the problem described in Section 5. Next we describe the construction of the transmitter and the receiver SR circuits employing the splitter scheme.

### 6.1. Transmitter SR

In the transmitter, the two parallel N/2-bit SRs are driven by transition signals C0/C0N and C90/C90N having minimum time separation (between two successive transitions) of two gate delays and shifted by one gate delay relative to each other. The transition signals are generated (as in the basic architecture [8]) by means of a multi-phase transition signal generator circuit [18]. The data coming out of the two parallel SRs are first merged at the output of each N/2-bit SR by means of Merge stage (Figure 11), producing two data streams at a cycle time of two gate delays. Thanks to the relative shift of C0 and C90, the streams are shifted relative to each other by one gate delay. The two streams are subsequently combined by the encoder into a single coded stream with single gate-delay minimum data cycle. Thus, the highest rate operation is localized inside the encoder, while the SRs work at twice slower rate.
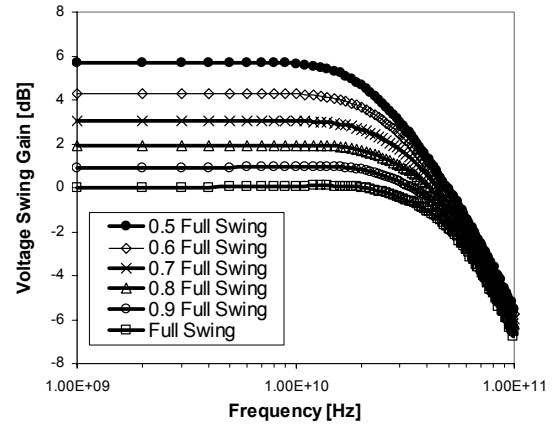


**Figure 7. Inverter amplification – Bode diagram for different unbiased input swings**
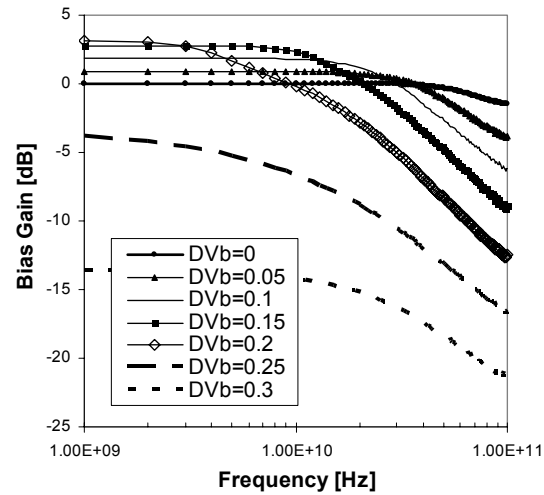


**Figure 8. Inverter bias amplification – Bode diagram for different input bias values DVb is the difference between bias and Vcc/2**
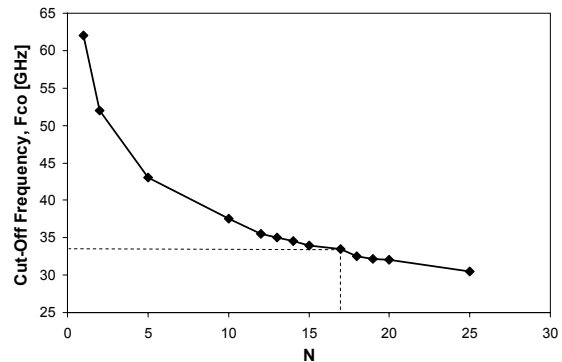
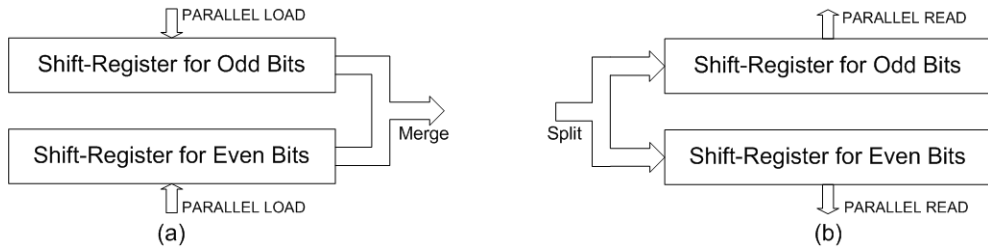

**Figure 9. Inverter chain cut-off frequency**

**Figure 10. Splitter architecture. The basic shift-register is partitioned into multiple shift-registers**
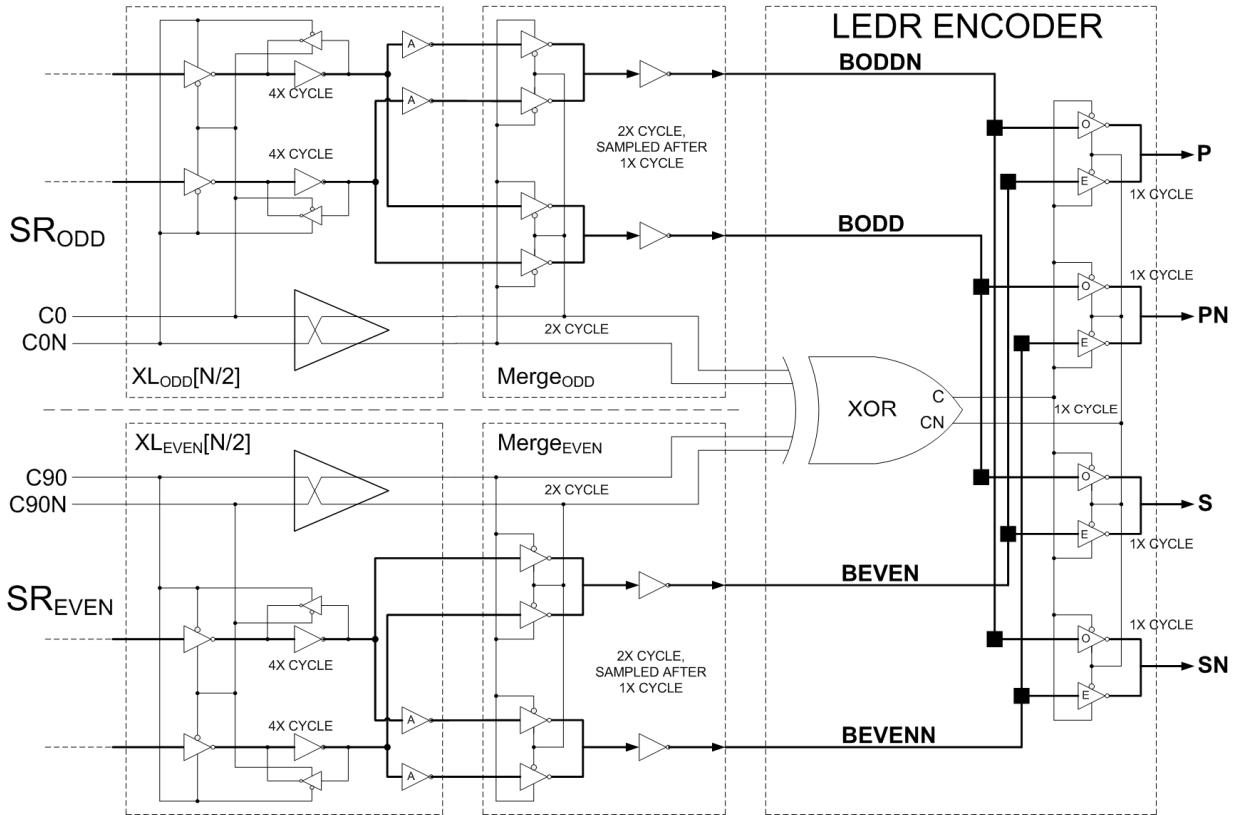


**Figure 11. Splitter architecture for the transmitter**

The separate storage of the odd and even bits allows simple and fast LEDR encoding as noted in Section 2 (Figure 4). Additional inverters "A" are introduced in the last stage of each SR ($XL_{ODD}[N/2]$ and $XL_{EVEN}[N/2]$) to produce inverted values of the uncoded bits. Thanks to the reduced speed of the XL stages there is enough time to produce both values. The two pairs of bits (BODD/BODDN and BEVEN/BEVENN) are used to produce the coded symbol (P/PN, S/SN) according to Eq. (3) and Eq. (4), and the interconnects inside the encoder follow those equations. The XOR gate produces control signals C/CN that switch the output between the odd and even bit streams.

## 6.2. Receiver SR

In the receiver, the input stream from the channel is split by the *O* and *E* tri-state buffers into two slower streams of length *N*/2 bits each that enter the two SRs (Figure 12). These are the odd and even bit streams of the transmitter. The incoming control signal C/CN is immediately divided by the toggle element into two control signals C0 and C90, which are phase-shifted by a single gate delay (more precisely, thanks to asynchronous operation, C0 and C90 are time-separated by the actual time between the two most recent successive input transitions). The toggle element structure and operation are detailed in Section 6.3.
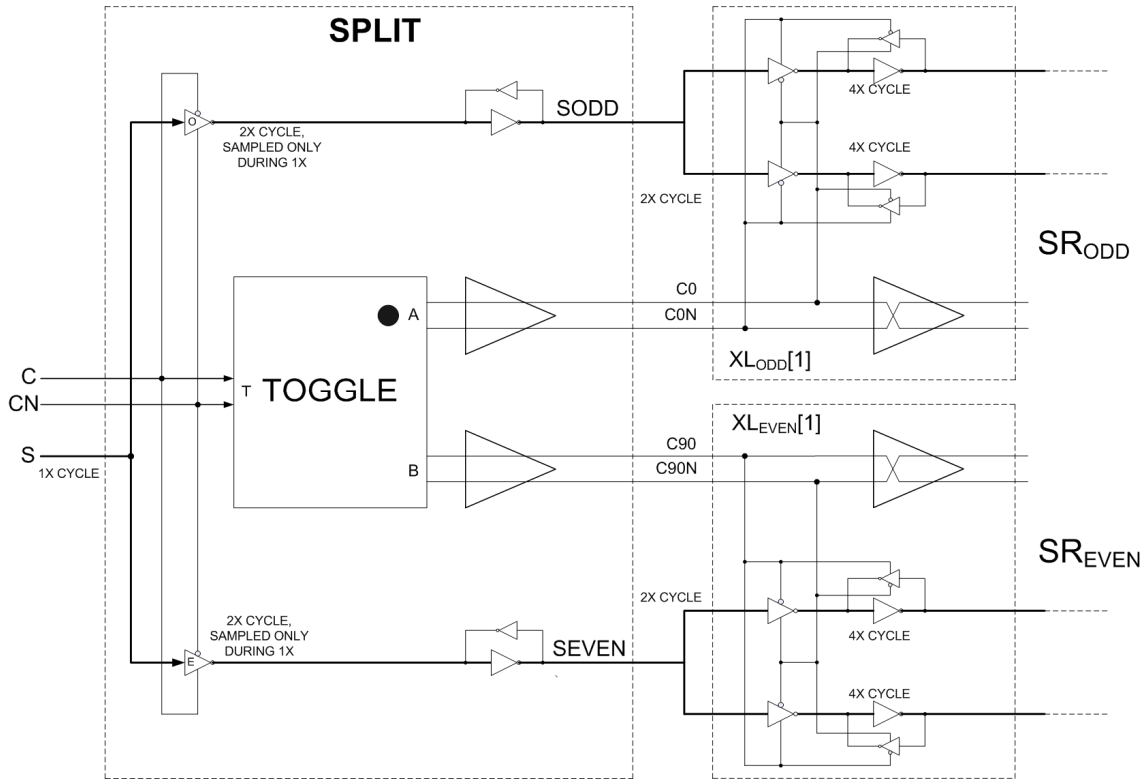
**Figure 12. Splitter architecture for the receiver**

In this architecture the SR needs to operate only twice slower than when a single SR is employed without a split (as in [8]).

Since *S* bits are sampled and *B=S* (Eq. (4)), at the end of word transmission (when $XL_{EVEN}(N/2)$ holds logic one) the decoded word is present inside the receiver SR and can be read out in parallel (Figure 5).

### 6.3. High-speed toggle element

The Toggle element (Figure 13a) is specified as follows. For each incoming transition on input T/TN the toggle produces a transition that alternates on output A/AN and B/BN. The signal transition graph (STG) of Figure 13b defines this operation. The first output transition is generated on the output marked with a dot.

The straightforward implementation is a fundamental asynchronous state machine. Several implementation options for such a machine were considered, and the fastest and smallest one is presented in Figure 13c (this implementation is also suggested by Petrify [24] for the given STG when complex gate synthesis is requested). That circuit shows correct operation only down to a data cycle of

21 ps (47GHz), which is approximately $1.4 \cdot d_4$. Therefore, although being very fast, that design does not fulfill the single gate delay requirement. In order to overcome this problem an alternative toggle element circuit is proposed as follows (Figure 14).

The new circuit takes advantage of the observation that each output (A or B) toggles after every other transition of the input T. We divide the time between two successive odd (even) input transitions into two phases: The first phase happens between the first odd
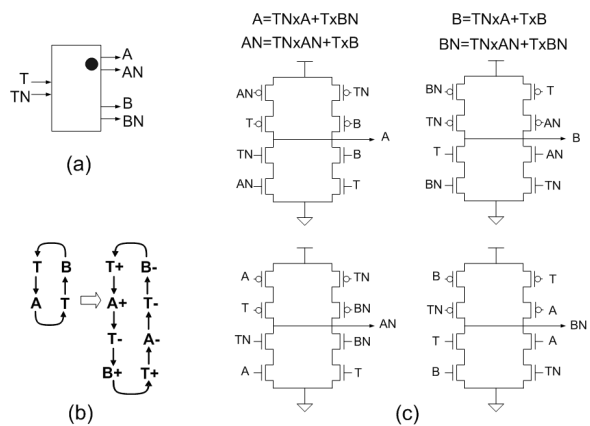


**Figure 13. Toggle element specification and fundamental machine implementation**
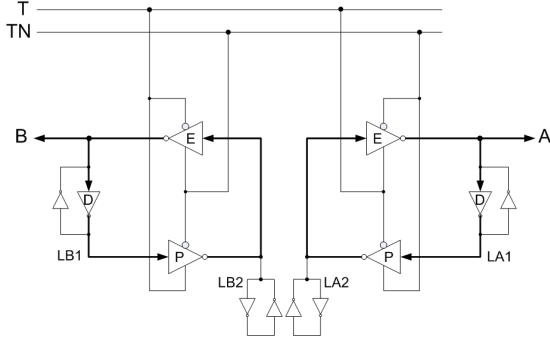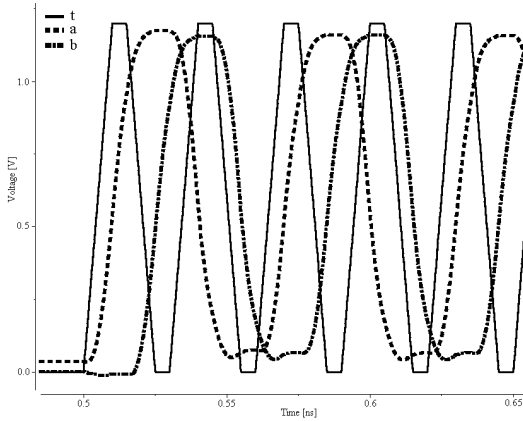
**Figure 14. High-speed toggle element**



**Figure 15. Toggle element simulation results, 15 ps data cycle**

(even) transition and the next even (odd) transition. The second phase happens between the even (odd) transition and the next odd (even) transition. Let's designate the first phase as "precharge" (P) and the second one as "evaluation" (E).

The circuit of Figure 14 consists, for each output, of two tri-state buffers that are controlled by T so that while one of them is open the other one is closed (e.g., when E is open P is closed). On each side of the buffers there is a keeper latch implemented by two inverters.

The toggle circuit operates as follows. In the beginning, the latches are reset (reset is not shown in Figure 14). Let's assume that the initial state is A=0, B=0 and T=0, and focus on generating the output A. In the initial stage, the right-hand circuit (which generates A) is in the "precharge" phase: E is closed, P is open (T=0), LA2 is reset to logic zero and LA1 is preset to logic high. Once the next transition comes in (T=1), the evaluation phase begins. Buffer E opens, changing A to logic high. The new value of A propagates through the delay inverter D that also serves as part of output latch LA1. Depending on operating conditions,

LA1 either fully or almost fully resets when the next transition (T=0) arrives. That transition triggers the precharge phase, opening P and closing the E buffers. During the precharge phase the output A is driven by the output latch LA1. LA2 is changed to a new value (LA2=1), creating the next value at the input of buffer E. In our example, during the next evaluation phase, the next A output value will be A=LA'=0.

Output B is produced similarly. Inverted values of A and B for differential signaling are obtained by duplicating the entire circuit of Figure 14 and applying an appropriate initial reset. Note that to achieve fastest operation, there is no time to invert A and B, hence the need for duplicating the circuit.

The circuit shows correct operation down to 14 ps data cycle (70GHz). This higher speed is achieved thanks to the fact that in the new toggle circuit each output is generated independently of the value of the other outputs, unlike the circuit of Figure 13. SPICE simulation example for the target speed is shown in Figure 15. Since the toggle element is the most complicated component in the Splitter architecture that is required to operate at the highest speed (others are XOR and inverters), it is essential to check its susceptibility to variations. We have performed Monte-Carlo simulations on the toggle element circuit using 14ps data cycle, and varying the threshold voltage $V_T$ and the channel length $L_{EFF}$. We observed output swing degradation of ~15% only at 5.75·σ. Even better results are expected for 15ps data cycle. In addition, the circuit was verified for 26 PVT corners. Note that the gate delay (FO4 inverter delay) changes as function of corner − in each corner the circuit was verified to support the data cycle of single gate-delay of the checked corner.

The 70GHz operation provides abilities that have not existed earlier. For example, the novel toggle element circuit can be used for on-chip clock or other fast signal probing by dividing the fast signals down to the frequencies of regular flip-flops.

### 6.4. Power reduction

Most of the power in the shift-registers is dissipated by the wave-pipelined control (WP-CTRL), due to its relatively big sizing and the fact that the entire control toggles for each shifted bit. Eq. (5) shows the dynamic power in the control of the non-split SR ($F_{SER}$ stands for the equivalent clock rate, which is one half the maximal bit-rate of the SR):

$$P_{WP-CTRL}^{NON-SPLIT} = N \cdot C_{BUF} \cdot V^2 \cdot F_{SER} \qquad (5)$$

The splitter architecture reduces power dissipation as follows. Instead of $N^2$ transitions per each $N$-bit word, the splitter SR employs $2 \cdot (N/2)^2$ transitions, resulting in a two-fold reduction in dynamic power of the wave-pipelined control:

$$P_{WP-CTRL}^{SPLITTER} = 2 \cdot \frac{N}{2} \cdot C_{BUF} \cdot V^2 \cdot \frac{F_{SER}}{2} =$$
$$= \frac{1}{2} \cdot P_{WP-CTRL}^{NON-SPLIT} \qquad (6)$$

A further spilt (e.g. 4-way, possibly useful for very long words) will result in even further power reduction. In addition to power reduction in the control section, the data moves twice slower than in the non-split SR, reducing power even more. A more detailed power analysis of the serial link can be found in [6].

## 7. High-Speed channel interconnect and circuit design

In this section we present the interconnect layout for LEDR codes, followed by the design of the current mode differential driver and receiver.

### 7.1. LEDR interconnect layout

When using LEDR codes, only one of the signals (either P or S) toggles per every transmitted bit. When LEDR symbols are signaled differentially, there are always two concurrent opposite transitions per every transmitted symbol. A special version of active shielding [25] is employed for these signals (Figure 16). This scheme minimizes cross talk and provides shielding as follows: Each toggling wire is surrounded by two quiet wires, and each quiet wire is surrounded by two wires that toggle in opposite directions.
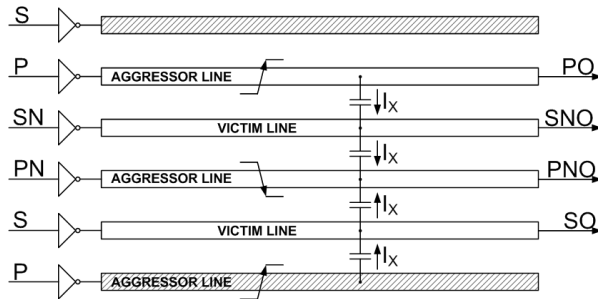


**Figure 16. LEDR shielding, P and S are LEDR encoding outputs and PN and SN are their inverted values. In this example, P/PN lines are aggressors and S/SN lines are victims. The cross talk is minimized**

Note that this structure not only mitigates the Miller effect, but it also reduces the proximity effect [26], since wires carrying opposing currents (the differential pair) are separated by thrice longer distance than if they were adjacent. If the wire width and separation are 1μm, the separation of the differential pair for LEDR interconnect is 3μm (instead of 1μm), resulting in effective increase in resistance of 3%, instead of 15%, namely five times smaller.

### 7.2. Channel differential driver and receiver

The differential channel circuit is shown in Figure 17. Novel adaptive current mode driver and low swing voltage mode receiver are used. The driver sends a differential signal comprising currents in opposite directions. The current results in a voltage drop on the terminating resistors, which is amplified by the voltage sense-amplifier.
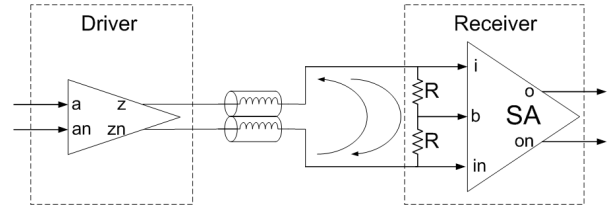


**Figure 17. Current mode driver, differential channel, and voltage mode receiver**

The driver circuit is shown in Figure 18a. The current flows through either M1 and M4 or M2 and M3. In addition, the driver contains an adaptive drive control circuit, designed to compensate for changes in the effective channel impedance. The channel characteristic impedance and effective resistance depend on signal frequency (Figure 19, Eq. (7), where δ is the skin effect depth, R is the interconnect resistance per unit length, L is the capacitance per unit length, C is the capacitance per unit length, and σ is shut inductance [26][27]). Note the 50% change in characteristic impedance value in Figure 19.

$$Z_0 = \sqrt{\frac{R + j\omega L}{\sigma + j\omega C}} \qquad R = R_{DC} + \delta \cdot (1+j) \cdot \sqrt{\omega} \quad (7)$$

When signal toggling slows down or stops, the effective frequency is reduced, the effective resistance decreases towards its DC value and the characteristic impedance is increased. Conversely, when fast toggling resumes, the effective frequency is increased, the effective resistance grows due to the skin effect and the channel impedance decreases, calling for a stronger drive. The AND gates in the circuit are driven by inertial delays, and control a variable load on the driver output. When the input is stable (or switches slowly),
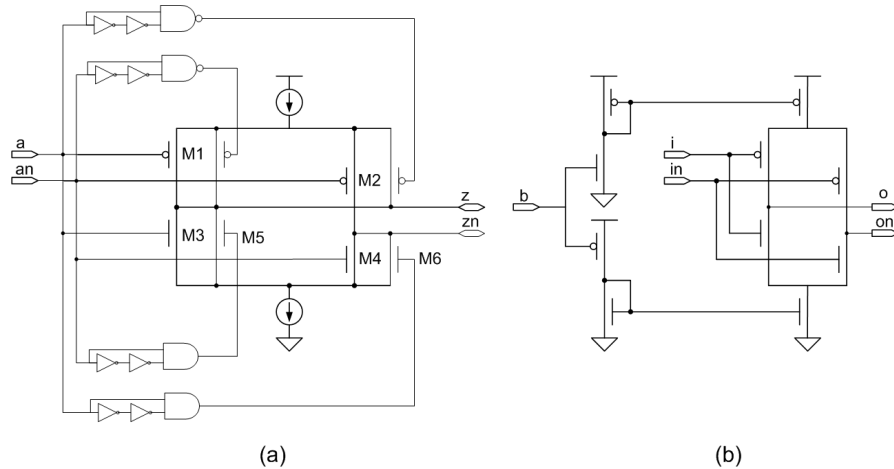
(a)                                                                (b)

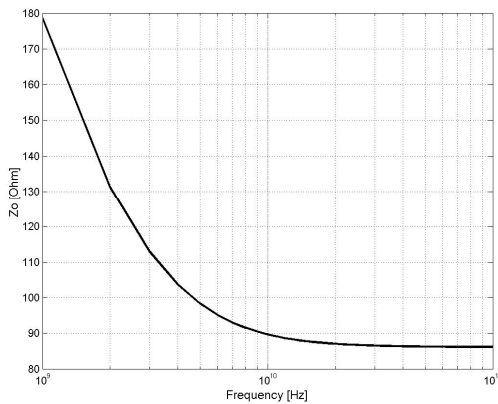**Figure 18. Transmitter differential driver and receiver differential sense amplifier**



**Figure 19. Characteristic channel impedance frequency dependency.
Based on data from BPTM [28] and Eq. (7).
Drawn for constant R, L, C**

the drive strength is reduced. When the input toggles fast, the AND gate is never turned on and the drive strength is increased.

The most important effect of the adaptive control is the ability to handle fast transients. During a long period of no toggling, the characteristic channel impedance is high. Hence, once a new transition arrives, the first transmitted toggle of the channel is distorted. The adaptive control mitigates this effect because it presents reduced impedance to the drive at the time of this first toggle. Shortly afterwards, the AND gate turns off and the extra load is removed. By that time, the channel impedance decreases, and normal transmission continues.

Another way of describing this operation is possible by observing voltage levels of the channel wires. Consider a simple driver without the added adaptive control circuit. During fast toggling, the voltage swing on the wires is very low and the average voltage is near Vdd/2. Once toggling stops, the wires would slowly converge to Vdd and Gnd. Then, upon the first toggle following a long period of no transmission, the wires will have to traverse a long voltage swing from the supply rails towards middle voltage, and that would take longer time than typical other toggles of lower swing. To prevent this distortion of the first toggle following a rest period, the adaptive circuit pulls the wire that would go up (down) to Vdd (Vss) down (up) towards middle voltage. The required swing when the first toggle arrives is thus reduced, resulting in faster first toggle and less distortion.

The receiver sense-amplifier (Figure 18b) comprises a balanced differential amplifier and a dynamic bias control for common mode rejection. The center tap of the terminating resistor (Figure 17) is connected to the dynamic bias control through input 'b.' The dynamic bias control manages to maintain a bias near the middle voltage at the output of the receiver. For instance, when the channel bias level increases, the output of the receiver tends to decrease in bias, and the control circuits pulls it back up.

The combination of the adaptive driver, the differential resistive-terminated channel and the dynamic bias-controlled receiver offers a number of advantages, including current signaling (dissipating lower power and enabling faster signaling) and good control of the current return path. The channel and circuit were SPICE simulated and operated correctly up to 4 mm at 67Gbps, using 1μm wires on 65nm technology. RLC parameters, obtained by Raphael-like three-dimensional field solver [29], were employed for channel modeling. When links longer than four millimeters are required, repeaters could be employed
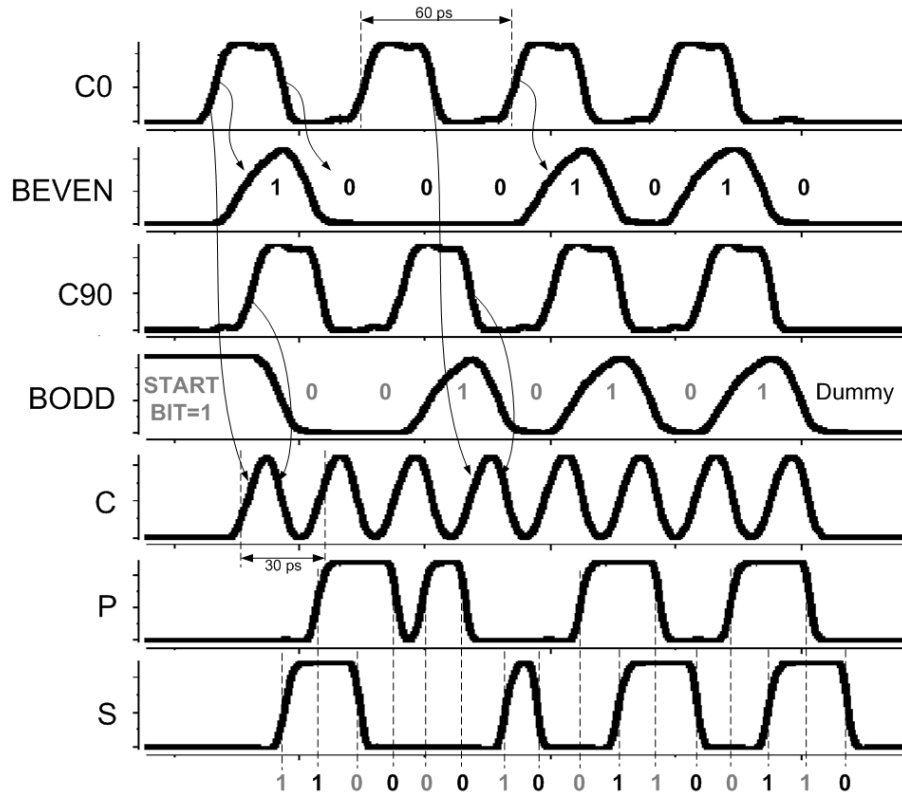
**Figure 20. Transmitter splitter shift-register simulation example**

along the channel, preserving the data rate. The repeaters can be constructed of a differential sense amplifier and a driver connected back to back.

## 8. Simulation results and discussion

A SPICE simulation example is shown in Figure 20. EVEN-SR is shifted first by the C0 control signal. The data is shifted out to BEVEN each $2 \cdot d_4$ (30ps). C90 is delayed by one gate delay (15ps) relative to C0. The XOR of C0 and C90 generates signal C. The encoded output bit-streams on P and S are the combination of the odd and even bit-streams. Note that either P or S toggle, but not both simultaneously.

67Gbps four millimeter serial link with 16-bit words, utilized 100% of the time, dissipates 150mW. The channel differential pair uses 18mW. The observed leakage power is 4mA, due to low $V_T$ transistors.

Thanks to the fact that most components in the splitter SR operate at a data cycle much longer than a single gate delay, this circuit is more resilient to process variation. Only the XOR, the toggle element, the channel driver, the channel receiver and the channel itself are required to operate at the highest speed. High variation in the latter components may lead to malfunction and they are carefully designed for variation mitigation (iterative design that identifies variation-sensitive transistors and applies legging and sizing to them). The circuit operates correctly as long as the minimal time between events in the fast components is preserved. Note that there is no restriction on maximal time between events thanks to asynchronous operation.

## 9. Conclusions

This paper described novel high rate asynchronous bit-serial on-chip communication architecture that outperforms bit-parallel links over long distances. The bit-serial link employs two-phase transition based LEDR encoding and differential signaling. The channel can handle multiple concurrent bits in a wave-pipeline manner. Wave-pipelining is also employed inside the transmitter and receiver shift-registers. The serial link achieves bit cycle time of a single gate delay, enabling 67 Gbps throughput in 65nm technology.

A novel 'splitter' architecture for the shift-register was presented, enabling the transmission of long data words and reducing power. The new architecture

results in reduced transmitter and receiver sizes. In addition, the highest speed operation is localized to a very small group of components: XOR in the transmitter and toggle element in the receiver. The architecture is highly resilient to PVT variations.

A novel low crosstalk interconnect layout, especially adapted to LEDR encoding, was introduced. Special circuits for channel driving and sensing were presented, employing an adaptive approach to the fast transient channel behavior.

The circuits were verified using SPICE simulations, including Monte-Carlo simulations for assessing the effect of in-die variation. RLC models were used for channel modeling. We plan to test the proposed serial link on real silicon in the future.

## 10. Acknowledgement

## References

[1] International Technology Roadmap for Semiconductors (ITRS), 2005, www.itrs.net.

[2] R. Ho, K.W. Mai, M.A. Horowitz, "The Future of Wires," Proc. of the IEEE, 89(4), pp. 490-504, 2001.

[3] A. Morgenshtein, I. Cidon, A. Kolodny, R. Ginosar, "Low-Leakage Repeaters for NoC Interconnects", Special Session "Repeater Insertion for Nanometer Technologies - Timing is NOT Everything", Proc. of ISCAS, pp. 600-603, 2005.

[4] R. Weerasekera, D. Pamunuwa, L. Zheng and H. Tenhunen, "Minimal-Power, Delay-Balanced Smart Repeaters for Interconnects in the Nanometer Regime," Proc. of SLIP, pp. 113-120, 2006.

[5] K. Lee, S.J. Lee, H.J. Yoo, "SILENT: serialized low energy transmission coding for on-chip interconnection networks," Proc. of ICCAD-2004, pp. 448-451, 2004.

[6] R. Dobkin, A. Morgenshtein, A. Kolodny, R. Ginosar, "Long Range On-Chip Communication: Parallel or Serial?" CCIT TR, Electrical Engineering Dept., Technion–Israel Institute of Technology, 2006.

[7] J. Teifel, R. Manohar, "A High-Speed Clockless Serial Link Transceiver," Proc. ASYNC'03, pp. 151-161, 2003.

[8] R. Dobkin, R. Ginosar, A. Kolodny, "Fast Asynchronous Shift Register for Bit-Serial Communication," Proc. of ASYNC'06, 2006.

[9] R.H. Mak, "Design and performance analysis of buffers: a constructive approach," Proc. of ASYNC, pp. 137-148, 2002.

[10] C.H. Kees van Berkel, M. Rem, R. Saeijs, VLSI programming, in: IEEE International Conference on Computer Design: VLSI in Computers and Processors, 1998, pp. 152–156.

[11] R. Dobkin, R. Ginosar, A. Kolodny, "High-Speed Serial Interconnect for NoC," NoC Workshop, DATE'06, 2006.

[12] A.P. Jose, G. Patounakis, K.L. Shepard, "Pulsed Current-Mode Signaling for Nearly Speed-of-Light Intrachip Communication," Journal of Solid-State Circuits, 41(4), pp. 772 – 780, 2006.

[13] I. Saastamoinen, T. Suutari, J. Isoaho and J. Nurmi, "Interconnect IP for Gigascale System-on-chip," Proc. of Euro. Conf. Circuit Theory and Design (ECCTD), pp. 281-284, 2001.

[14] T. Suutari, J. Isoaho and H. Tenhunen, "High-speed Serial Communication with Error Correction Using 0.25μm CMOS Technology," Proc. ISCAS, pp. IV-618-621, 2001.

[15] S.J. Lee, K. Kim, H. Kim, N. Cho, H.J. Yoo, "Adaptive Network-on-Chip with Wave-front Train Serialization Scheme," Proc. of VLSI Circuits, pp. 104-107, 2005.

[16] G. Lakshminarayanan, B. Venkataramani, "Optimization Techniques for FPGA-Based Wave-Pipelined DSP Blocks," IEEE Trans. VLSI, 13(7), pp. 783-793, 2005.

[17] C. Svensson and J. Yuan, "A 3-Level Asynchronous protocol for a Differential Two-Wire Communication Link," J. Solid State Circuits, 29(9), pp. 1129-1132, 1994.

[18] M.J.E. Lee, "An Efficient I/O and Clock Recovery for TERABIT Integrated Circuits Design," PhD Thesis, Stanford Univ., 2001.

[19] C.K.K. Yang, "Design of High-Speed Serial Links in CMOS", PhD Thesis, Stanford University, 1998.

[20] R. Dobkin, R. Ginosar and C. P. Sotiriou, "High Rate Data Synchronization in GALS SoCs," IEEE Transactions on VLSI, 14(10), pp. 1063-1074, 2006.

[21] M.T. Dean, T. Williams et al. "Efficient Self-Timing with Level-Encoded 2-Phase Dual-Rail (LEDR)," Proc. ARVLSI, pp. 55-70, 1991.

[22] D.H. Linder and J.C. Harden, "Phased Logic: Supporting the Synchronous Design Paradigm with Delay-Insensitive Circuitry," IEEE Trans. Computers 45(9), pp. 1031-1044, 1996.

[23] I.E. Sutherland, "Inverse XOR and XNOR Circuits," US Patent 5,861,762, 1999.

[24] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno and A. Yakovlev, "Petrify: a tool for manipulating concurrent specifications and synthesis of asynchronous controllers," IEICE Transactions on Information and Systems, E80-D(3), pp. 315– 325, 1997.

[25] H.Kaul, D.Sylvester and D.Blaauw, "Active Shields: A New Approach to Shielding Global Wires," Proc. Great Lakes Symposium on VLSI, pp. 112-117, April, 2002.

[26] H. Johnson, M. Graham, "High-Speed Digital Design. A handbook of Black Magic", Prentice Hall PTR, 1993.

[27] C. Svensson, P Caputa, "High bandwidth, low latency global interconnect", Proc. of SPIE conference, Canary Islands, pp. 126-134, 2003.

[28] Berkeley Predictive Technology Model (BPTM), http://www.eas.asu.edu/~ptm.

[29] Synopsys, Inc., Raphael. http://www.synopsys.com.