

Original research paper

UDC 004.78:621.38]:621.313.333.064

DOI 10.7251/IJECC2102059S

# On- and Off-chip Signaling and Synchronization Methods in Electrical Interconnects

Mile Stojcev, Bojan Dimitrijevic

Univerzity of Niš, Faculty of Electronic Engineering, Niš, Serbia

E-mail address: mile.stojcev@elfak.ni.ac.rs, bojan.dimitrijevic@elfak.ni.ac.rs

**Abstract**—Advances in integrated circuit (IC) fabrication technology, coupled with aggressive circuit design, have led to an exponential growth in speed and integration levels. However, to improve overall system performance, the communication speed between *on*-chip subsystems and between ICs in printed circuit board must increase accordingly. Currently, communication bus links in various applications approach Gbps (gigabits per second) data rates. These applications include high-speed network switching, local area network, memory buses, and multiprocessor interconnection networks. In this article, we analyze the most common (popular) CMOS implementations of high speed *on*- and *off*-chip links (electrical interconnects) and show that the links' performance should continue to scale with technology. In addition, we point to the fact that global electrical interconnects are widely acknowledged as a limiting factor in future *on*-chip and *off*-chip designs. Novel electrical interconnect driving techniques like multi-level multi-wire signaling, and GALS synchronization method primarily intended to improve performance of *on*-chip electrical interconnects, has been shortly analyzed. In the future, in order to handle the electrical interconnects' finite bandwidth (data rate higher than 100 Gbps), however, more sophisticated signaling and synchronization methods will be required.

**Keywords:** Electrical Interconnects, *on*- and *off*-chip links, GALS, VLSI design

## I. INTRODUCTION

The rapid advances in VLSI CMOS ICs fabrication technology, combined with aggressive circuit design, have bring to an exponential growth in operational speed and integration levels. Generally, the digital system performance consists of two parts: computation performance and communication performance [1], [2], [3]. Thanks to the development in the CMOS technology scaling (especially in the field of the multicore architecture), the computation performance of a chip has been increased dramatically [4]. As computation performance goes up, the required communication bandwidth (or throughput) needs also increase with the same rate. However, the number of input/output (I/O) pins and the total I/O bandwidth of a single chip have been scaling much more slowly [5], [6]. Therefore, the communication between multiple *on*- or *off*-chip semiconductor intellectual property (IP) blocks is becoming a dominant cost, performance, and power factor in modern digital systems. In essence, the I/O bandwidth increases by approximately 2× every two years or 10× every five years, while the I/O pin number grows at a much smaller rate, 1.7× every five years, because of process and mechanical constraints. This results in a widening gap between I/O bandwidth requirements and capabilities, notoriously named the interconnect gap, which has been a major and long-standing challenge for more than a decade [7], [8]. With aim to improve the overall system performance, the

communication speed between systems and ICs must increase adequately.

In this article, we briefly discuss various physical *on*- and (short-range) *off*-chip electrical interconnect methods which specifies the physical organization of the signaling technologies and synchronization techniques which control communication data transfer speed among IC building blocks at circuit level. The most critical performance metrics of the electrical interface are the following: *a) the bandwidth density* – defined as gigabits per second per square micrometer; *b) energy efficiency* - defined as pico-joules per bit; *c) bit error rate* (BER) - it is established by the signal-to-noise ratio (SNR), component and channel dispersion, pre/post-processing complexity and capabilities, and so forth; *d) latency* – it is affected by the component response and processing time and the time of flight through channels; and *e) cost* - it is governed by initial manufacturing expenses, yields, reliability, product life spans, and related factors.

The above mentioned design challenges have inspired research from both the electrical interconnect and communications to find an efficient electrical interconnect which will provide the possibility to achieve high-speed data communication (at rates of order up to several gigahertz) among different functional building blocks according to the system specification (requirement) [1], [4], [9].

In practice, electrical interconnects use one of the following two main signaling schemes (interfaces): single-

ended and differential signaling. In single ended signaling each link has one dedicated wire and all the links use a shared ground signal for return current. Single-ended interfaces allow for relatively high frequencies (up to 70 MHz) when applied in close proximity to a system controller. In differential signaling each link consists of two dedicated wires connecting the transmitter to the receiver. In essence, a differential pair is realized with two transmission lines that have equal and opposite polarity signals propagating on them, so that the positive path and the negative path (of a differential pair) are tightly timed. Differential interfaces possess significant higher noise immunity and drastically reduced EMI and can, therefore, transmit data per wire at frequencies of up to 500 MHz and above. The need for high input-output bandwidth has led to widespread use of differential signaling [10]. Concerning signaling techniques, the main investigations in the field of coding theory has resulted in numerous capacity-approaching codes, the search for low-complexity coding schemes for practical electrical interconnect implementations [11], [12]. In chip-to-chip and block-to-block communication applications, the main challenge is to come up with low-complexity coding schemes that can be implemented at high speed. In principle, the use of coding in inter-chip and inter-block applications can be categorized into two subsections: two-level signaling and multilevel signaling [8], [10], [13]. In addition, in this paper we introduce the concept of a multi-conductor interconnect design solution (MIPI interface) able to transmit more bits of data per physical conductor in a given time interval, thus making it somewhat specific (unique) compared to traditional digital buses. The proposed design uses signaling methods that rely on a hybrid single-ended/differential-signaling scheme and multilevel signal modulation. In other words, it introduces the concept of multiphase encoding to help eliminate the need for transmitting a forwarded clock, thus also saving on wire count [14], [15].

In principle there are three global methods to use for synchronization of a system. The most commonly used today is the synchronous system where a global clock is distributed over the system with low skew. This clock is then used to time all the events and transactions in the system [16], [17]. The second method that is popular in research and extreme low power products is to run the system completely asynchronous. Completely asynchronous systems need to use handshaking or special timing circuitry for both computations and communications in order to keep synchronization within the system [18], [19]. The third method is to use blocks that are synchronous but communicate asynchronously, better known as the globally asynchronous, locally synchronous (GALS) methodology [20], [21], [22]. With the current increase in the number of different clock domains used on a single chip this is a very promising overall technique to use for IP block integration.

The remainder of this paper is organized as follows. Section 2 introduces several suitable and commonly used signaling schemes for *on-* and *off-*chip high-speed data transfer applications. Section 3 explains basic principle of clock synchronization approaches. In Section 4 we give the conclusion of this paper and point out to possible feature research.

## II. SIGNALING APPROACHES

Fig. 1. shows the components of a signaling system: transmitter (Tx), channel, and receiver (Rx). The transmitter converts digital information to a signal (waveform) on the transmission medium, or communication channel. This channel is commonly a board trace, coaxial cable, twisted-pair wire, or *on-chip* VLSI electrical interconnect. The receiver on the other end of the channel restores the signal, by sampling and quantizing it, to the original digital information. Clock generation and timing recovery are tightly coupled to signal transmission and reception. The timing recovery, often embedded in the receiving side, adjusts the phase of the clock that strobes the receiver. The receiver samples the signal waveform at the optimal position.



Figure 1. Signaling system components: transmitter, channel, and receiver.

Signaling approaches in high-speed input-output (I/O) links can be realized in one of the following four ways:

- Single-ended signaling,
- Differential signaling,
- Multi-level signaling, and
- Common-mode signaling

In the sequel we will analyze in brief the properties of all four signaling methods.

### A. Single-ended Signaling

Single-ended data transmission uses only one signal line, for which its voltage potential is referred to ground. While the signal line provides the forward path for signal currents, the ground line provides the return current path. Fig. 2. shows the basic schematic of a single-ended transmission path.

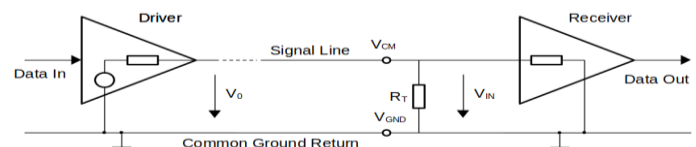


Figure 2. Single-ended transmission path

Single-ended interfaces benefit from their simplicity and their low implementation cost, but have three main drawbacks [1], [4], [8], [23], [24], [25]:

1) They are highly sensitive to noise pick-up, because noise induced into the signal or ground paths adds directly to the receiver input, thus causing false receiver triggering.

2) Another concern is crosstalk, which is the capacitive and inductive coupling between adjacent signal and control lines, particularly at higher frequencies.

3) Finally, due to the physical differences between the signal trace and the ground plane, the transversal electromagnetic waves (TEM) generated in single-ended systems can radiate into the circuit environment, thus representing a significant source of electromagnetic interference (EMI) to adjacent circuits.

B. Differential signaling

Differential signaling uses signal pair consisting of two conductors: one for the forward, the other for the return current to flow (see Fig. 3.).

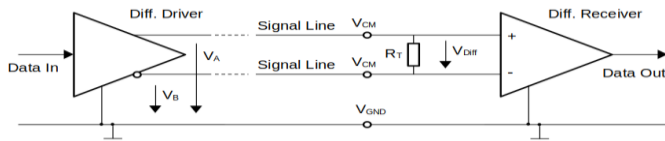


Figure 3. Differential transmission path

When the conductors of a differential pair are close to each other, electrically coupled external noise induced into both conductors equally appears as common-mode noise at the receiver input. Receivers with differential inputs are sensitive to signal differences only, but immune to common-mode signals. The receiver, therefore, rejects common-mode noise and signal integrity is maintained. Close electric coupling provides another benefit. The currents in the two conductors, being of equal amplitude but opposite polarity, create magnetic fields that cancel each other. In general, differential signaling is designed to transmit logic signals between two boxes or units that have logic grounds offset from each other by an amount too large for single ended logic signals to function correctly.

*Advantages:* The usage of differential signaling has several advantages and disadvantages [1], [4], [8], [10], [26]:

(a) Timing is much more precisely defined, because it is easier to control the crossover point on a signal pair than it is to control an absolute voltage relative to some other reference. This is one of the reasons for exactly equal length traces. Any timing control we have at the source could be compromised if the signals arrive at different times at the other end. Furthermore, if signals at the far end of the pair are not exactly equal and opposite, common-mode noise might result which might then cause signal timing and electro-magnetic interference (EMI) problems. In complex electrical systems that use differential signaling these two lines must be equal in length to within the timing tolerances of the logic IC family being used. In virtually all cases, the timing accuracy will allow length differences up to 500 mils or 12.5 millimeters.

(b) Since they reference no other signals than themselves, and since the timing of signal crossover can be more tightly controlled, differential circuits can normally operate at higher speeds than comparable single-ended circuits.

(c) Since differential circuits react to the difference between the signals on two traces (whose signals are equal and opposite) the resulting net signal is twice as large, compared to ambient noise, as is either of the single-ended signals. Therefore, differential signals, all other things equal, have greater signal/noise ratios and performance.

*Disadvantages:* The primary disadvantage of differential circuitry is the increased number of traces. So, if

none of the advantages are particularly significant in your application, differential signals and the associated routing considerations are not worth the cost in increased area. But if the advantages make a significant difference in the performance of your circuit, then increased routing area is the price we pay.

However, when complex electrical systems are realized, this cost may be far less than would be incurred by trying to build a power and ground structure that has enough copper in it to maintain the voltage drops in power and ground within limits imposed by single ended logic signaling.

Fig. 4. shows a single-ended signaling system and a differential signaling system with 4 data wires.

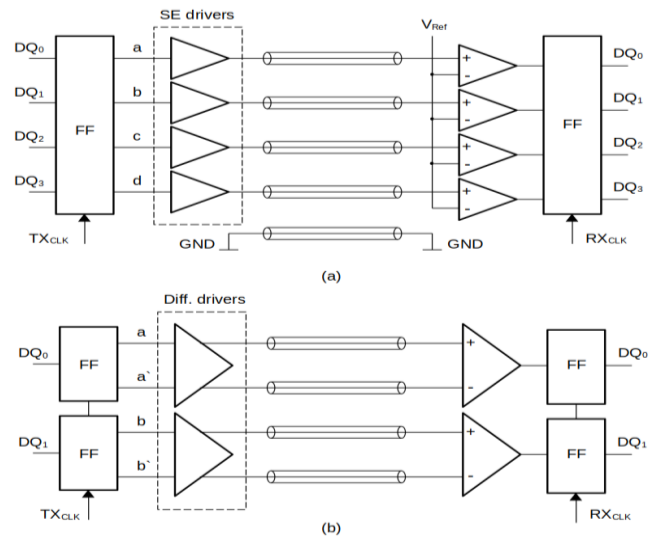


Figure 4. (a) A single-ended signaling system (b) A differential signaling system.

C. LVDS circuits: typical components for differential signaling

During the last two decades, the pronounced demands for high speed data transmission over chip-to-chip, board-to-board, and even longer distances have resulted in the development of the *Low Voltage Differential Signaling* (LVDS) I/O standard. Based on CMOS logic, LVDS features high speed with low noise generation, EMI resistance, and low power requirements. LVDS is used in high bandwidth data transfer applications, in particular backplane transceivers or clock distribution applications. A common reason for choosing LVDS is its low signal swing voltage of 350 mV, much lower than TTL, ECL and CMOS logic.

This lower swing voltage presents added benefits over other alternatives [27]:

- LVDS is a power efficient standard. AC power is low because the signal switch-over voltage is small, leading to low power dissipation per signal transition. DC power is also low because although each channel requires 3.5 mA, it is likely a single channel will be replacing a number of existing parallel channels.
- LVDS generates reduced levels of EMI. Device-generated EMI is dependent on frequency, output voltage swing and slew rate. Due to the low-voltage swing of the LVDS standard, the effects of EMI are much less than with CMOS, TTL, or other I/O standards.

SERDES technologies, based on LVDS circuits, have become increasingly popular as a method to meet these challenges for chip-to-chip, board-to-board, and backplane applications.

D. What is SERDES?

SERDES (serializers/deserializers) are devices that can take wide bit-width, single-ended signal buses and compress them to a few, typically one, differential signal that switches at a much higher frequency rate than the wide single-ended data bus. SERDES enable the movement of a large amount of data point-to-point while reducing the complexity, cost, power, and board space usage associated with having to implement wide parallel data buses. SERDES usage becomes especially beneficial as the frequency rate of parallel data buses moves beyond 500 MHz (1000 Mbps). Today's SERDES ICs are highly integrated devices optimized for the specific application niche they target. One typical application of the SERDES circuit known as *Embedded Clock (Start-Stop) Bits* SERDES is presented in Fig. 5.) [28].

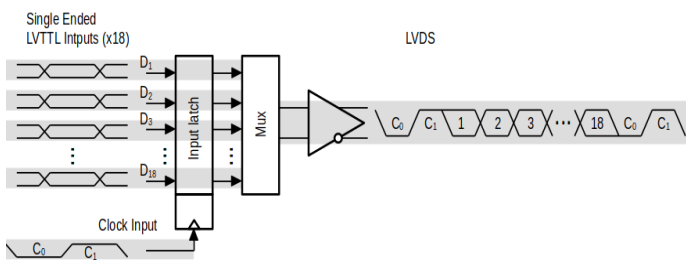


Figure 5. 18-bit embedded clock bits serializer.

The embedded clock bits' architecture transmitter serializes the data bus and the clock onto one serial signal pair. Two clock bits, one low and one high, are embedded into the serial stream every cycle, framing the start and end of each serialized word (hence the alternative name "start-stop bit" SerDes) and creating a periodic rising edge in the serial stream. Data payload word widths are not constrained to byte multiples; 10- and 18- bit widths are popular bus widths.

E. Multi-level-signaling

Multilevel signaling is often used as a means of compressing the bandwidth required to transmit data at a given bit rate, in other words it specifically attempts to reduce the number of interconnect wires [13], [29]. In a simple binary scheme, two single symbols, usually two voltage levels, are used to represent a 1 and a 0. The symbol rate is therefore equal to the bit rate. The principle of multilevel signaling is to use a larger alphabet of  $m$  symbols to represent data, so that each symbol can represent more than one bit of data. As a result, the number of symbols that needs to be transmitted is less than the number of bits (that is, the symbol rate is less than the bit rate), and hence the bandwidth is compressed. The alphabet of symbols may be constructed from a number of different voltage levels. PAM4 is the signal with 4 different levels, where each level corresponds to one symbol representing 2 bits. It means that using PAM4 gives the same bitrate with the half frequency. Symbolically, the levels can be 0, 1, 2, 3. One of the most common methods for converting binary signal to PAM4 signal is gray-coding (Fig. 6.).

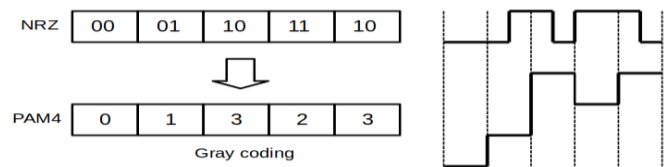


Figure 6. Gray-coding of binary to quaternary signal Notice: The term NRZ relates to Non Return to Zero signal coding

In the four-level scheme, groups of two data bits are mapped to one of four symbols. Only one symbol need be transmitted for each pair of data bits, so the symbol rate is half the bit rate. The drawback of the multilevel scheme is that symbols are separated by a smaller voltage than in the binary scheme. This means that when noise is added to the data signal (cross talk or impulse), the probability of the noise changing one symbol to another is increased. The symbol separation could be increased to that of the binary scheme by increasing the peak-to-peak transmitted voltage by a factor of  $(m - 1)$  for an  $m$ -level scheme, but this is generally not possible given fixed power supply voltages, and in any case it increases the power required for a transmitter.

Multilevel Pulse Amplitude Modulation (PAM) signaling is done by having each symbol containing  $k$  bits of binary information transmitted in a single clock cycle by one of the  $2^k$  signal levels. Information is then being transmitted at a rate of

$R_{PAM} = k/T_b$  bit/sec. Where  $T_b$  is the bit interval. The new signaling frequency as a result of using multilevel signaling is:  $f_{NEW} = f_{OLD}/k$ . Where the  $f_{OLD}$  is the old signaling frequency.

For a given interval  $T_b$ , the bit rate is  $R_{PAM} = kR_B$ , i.e. it is  $k$  times faster than the original transmission using binary pulses. For PAM-4 signals, each symbol contains 2 bits of binary information. In PAM-4 signaling, the same amount of data can be transmitted using half the signaling frequency. In other words, the ternary signaling ( $m = 3$ ) or quaternary signaling ( $m = 4$ ) can for example reduces the symbol-time by a factor 1.58 and a factor 2 respectively, while keeping the same data rate. Related to multi-level signaling technique we can conclude the following. There has been a great deal of interest in the use of multiple levels to increase signaling bandwidth without increasing clock frequency. Signaling with multiple voltage levels uses lower fundamental frequencies than binary signaling at the same data rate, offering the potential of higher performance in systems which have limited bandwidth. In order to achieve this performance, however, circuit improvements are needed or the reduced signal swing directly impacts the system signal-to-noise ratio (SNR).

F. Common-mode signaling using "phantom" line

The main disadvantage of differential signaling as opposed to single-ended signaling is that two wires are required to carry symbols from transmitter to receiver, rather than one. In a simple encoding scheme, where the two possible symbols encode a binary digit (bit), the efficiency of a differential signaling system is exactly 0.5 bit/wire. This efficiency may be improved if a signaling system consists of several parallel channels. One such method is outlined in Fig. 7. [30].



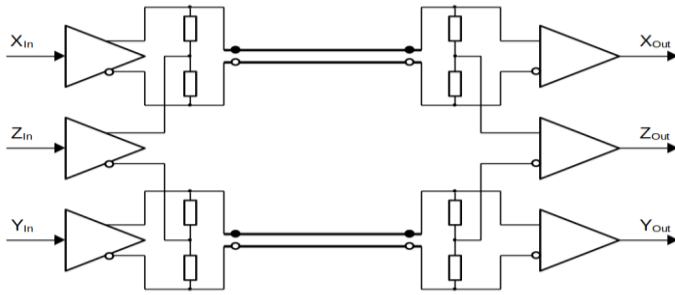


Figure 7. Improving signaling efficiency with a “phantom/ghost/wraith” line.

Channels “X” and “Y” are conventional differential signaling channels. Each is equipped with center-tapped terminators at both transmitter and receiver. The center taps in no way affect the operation of the “X” and “Y” channels, and in the absence of the “Z” circuitry, the center taps of all terminations resistors would be at a fixed voltage (assuming perfectly balanced voltages/currents out of the transmitters), namely the common-mode voltage,  $V_{cm}$ , of the signaling system. The “Z” channels transmitter establishes a voltage difference between  $V_{cm}$  on the “X” channel and  $V_{cm}$  on the “Y” channel. This voltage difference propagates down the two transmission lines of the “X” and “Y” channels and creates a voltage difference between the center taps of the receiver terminators on the “X” and “Y” receivers. This difference is detected by the “Z” receiver which recovers the “Z” bit stream. In this improvement, the efficiency of the signaling system is increased for 0.5 to 0.75, since 3 bits can be transmitted per symbol on 4 wires.

Concerning common-mode signaling, as conclusion we can say the following. In computer communications systems, the usage of phantom/ghost/wraith line cannot be carried far in practice because the absolute value of the signals at a given receiver may be driven outside the *common-mode range*, the allowable range of voltage on input terminals of the receiver. In practical receivers, this range is almost always restricted to within the power supply voltages, and often much less.

### G. Multi-wire interconnects

In general, we can have an interface consisting of  $m$  wires over which we transmit information differentially using  $n$  levels. The values of  $n$  and  $m$  determine the Information transmission rate of this interface. There are three possibilities for the relative values of  $n$  and  $m$  ( $n > m$ ,  $n = m$ , and  $n < m$ ) [31] and we will discuss their implications in the sequel.

$n > m$ : operation with wasted levels: In the case when there are more signal levels than wires, for any one transmission there will be signal levels that are not assigned to any of the wires. Since the information is stored only in the sign of difference signals, the same information can be coded in such way that  $n-m$  signal levels are never used. Using this method,  $n-m$  signal levels can be removed from any code with  $n > m$ , therefore these additional levels are wasted: they do not contribute to the information transmission rate of the interface.

$n < m$ : operation with wasted wires: An interface using more wires than signal levels will have to assign one signal level to at least two wires. Therefore, it is not possible to use the difference between these wires to transfer additional information, and the information transmission rate of these

wires is partially wasted. An implementation of such an interface is much more complicated.

$n = m$ : the optimum solution: The most efficient implementation is achieved with the number of wires equal to the number of signal levels, provided that the number of signal levels is not creating implementation problems. For such an interface, the information can be coded by choosing a signal level for the first wire. For the each following wire, any signal level that has not yet been used by the previous wires can be chosen. For the last wire, only one choice remains. Coding the information this way, all wires have different levels. Also, all levels can be derived from the signs of the differences of each pair of wires in the interface by sorting the wires according to the sign of their differences. This will result in a sorted sequence of wires, and the level of each wire corresponds to their position in the sorted sequence. The number of different symbols that can be transmitted by this interface is  $n!$  since the signal for the first wire can be chosen from  $n$  levels, the signal for the second wire can be chosen from  $n-1$  levels, and so on until there is only one choice for the last wire. The 4 wire differential which uses 4PAM signaling method can therefore transmit 24 different symbols, or more than 4 bits of information, per transmission. An additional advantage of choosing  $n$  equal to  $m$  is that at all times, all possible levels will be in use. This will cause the interface to not only be insensitive to external electromagnetic noise sources, but will also virtually prevent the interface from radiating any electromagnetic signals.

### Information transmission rate of $n$ -wire $n$ -level interfaces:

Although the number of different symbols that can be transmitted over an  $n$ -wire  $n$ -level differential interface is equal to  $n!$ , the information transmission rate of such an interface does not increase as fast as this increasing number of symbols would imply because the maximum transmission rate decreases with the number of signal levels. With more signal levels, the signal has to settle for a longer time before the signal level can be determined. How much this affects the information transmission rate depends on the channel and decoders.

As we have already mentioned, one possible solution for reduction of high wiring complexity, without affecting performance of a chip, is injecting more than two levels of signal into a single wire. The logic that can implement this signaling method is known as multiple-valued logic (MVL). It performs its operation by using more than two discrete signal levels. In voltage-mode circuits a number of signal levels is limited by the power-supply voltage and signal to noise ratio. In current-mode circuits a number of signal levels is limited by resolution of current comparators and signal to noise ratio for given technology.

### H. Extending Differential Signaling to Multi-Wire Multi-Level Systems – “Trifferential Signaling”: MIPI Interfaces

During the last two decades, the processing speed of mobile personal computers, tablets, network transmission components (intelligent hubs and routers), and many personal devices such as mobile phones equipped with 60 Mpix high resolution cameras and increasing frame rates ( $\geq 75$  frames/s) is pushing the off-chip data rate into the gigabits-per-second Gb/s range. As a consequence, this continuous technological progress has a significant impact for the needed system bandwidth. However, in spite of a drastic increase of on-chip frequency, chip-to-

board signaling gains little benefit in terms of on-chip operating frequency from the increased CMOS VLSI silicon integration. During the past time period, high data rates were mainly achieved thanks to implementation of parallel interconnects, with the drawbacks of increased complexity, power consumption, cost for the IC package and the printed circuit board. In order to bypass this problem efficiently MIPI Alliance (MIPI) develops interface specifications for mobile and mobile-influenced industries [32], [33]. In essence, MIPI specifications address only the interface technology, such as signaling characteristics and protocols that support *M-PHY*, *D-PHY* and/or *C-PHY* interfaces.

*M-PHY* (v3.1, June 2014) is an embedded clock serial interface technology with ultra-high bandwidth capabilities, specifically developed for the extreme performance and low power requirements of mobile applications. It's designed for next generation point-to-point interfaces and high speed component networks using dual simplex architectures.

*D-PHY* (v1.2, September 2014) is a serial interface technology using differential signaling for bandlimited channels with scalable data lanes and a source synchronous clock to support power efficient interfaces for streaming applications such as displays and cameras. It offers half-duplex behavior for applications that benefit from bidirectional communication at transmission rates up to 2.5 Gigabit per lane.

*C-PHY* (v1.0, October 2014) requires few conductors, does not require a separate clock lane, and provides flexibility to assign individual lanes in any combination to any port on the application processor via software control. Due to similarities in basic electrical specifications, *C-PHY* and *D-PHY* can be implemented on the same device pins. 3-phase symbol encoding technology delivers approximately 2.28 bits per symbol over a three wire group of conductors per lane. This enables higher data rates at a lower toggling frequency, further reducing power.

The *C-PHY* (in respect to *D-PHY*) is more complex *PHY* because it operates on three signals (called trio) whereas the clock signal is embedded into data causing a separate clock lane sufficient (i.e. unnecessary). In addition, the *C-PHY* interfaces uses encoded data (with aim) to pack  $16/7 \approx 2.28$  bits/symbol, while *D-PHY* does not use any kind of encoding. Compared to *D-PHY*, for the same symbol rate, the *C-PHY* can achieve higher data rate. The *C-PHY* employs multi-level signaling, but its receiver does not need to detect the difference between the multi-level signal. A practical *C-PHY* configuration consists of one or more three-wire lanes.

As sketched in Fig. 8. (b) the *C-PHY* lane consists of a trio, A, B, and C. The *C-PHY*'s receiver is composed of three differential receivers (RX's), each one looking at the difference between two of the three signals, (A-B), (B-C), and (C-A), respectively.

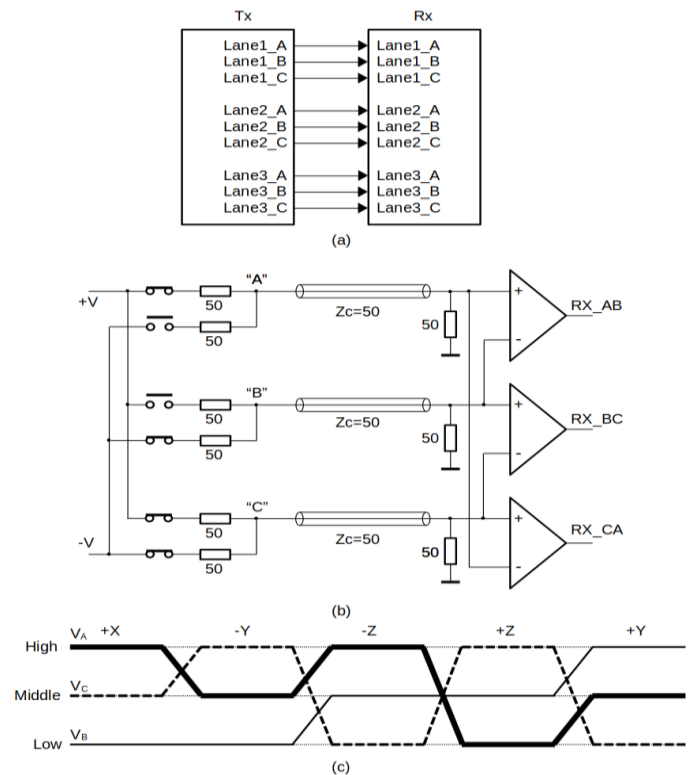


Figure 8. *C-PHY* (a) TX & RX connection, (b) different functions in *C-PHY* subsystem, (c) *C-PHY* signaling levels at TX and RX outputs

The *C-PHY* encoder warrants fulfillment of the following three design requirements: (R1) at least one edge transition per symbol exists; (R2) the differential input at all three receivers (RXs) is non-zero; and (R3) the common mode voltage of all three encoded signals is constant. The above mentioned requirements (R2) and (R3) are accomplished by limiting the combination of the transmitter signals during any single Unit Interval (UI) to high-level, mid-level, and low-level, and by preserving the voltage level on each of the three encoded signals different. The combination of the three transmitter signals levels (low-, mid- and high-level) that act in accordance with requirement (R1) results in generation of six signal level combinations (i.e. wire states). The number of wire states corresponds the permutation of three transmitter signal levels (i.e., factorial of three (3!)). To warrant that it exists at least one edge per symbol, imposed by requirement (R1), the *C-PHY* must change between different wire states as it moves away from one symbol to the next and cannot remain at the same wire-state during two successive symbols. Bearing this in mind, five different unique transitions between the six wire states exist. In other words, the encoded data has five possible states, what results the *C-PHY* to be a base-5 system. This is the reason why *C-PHY* mapper is needed to be installed in the hardware structure. Accordingly, the maximum theoretical number of bits/symbol is  $\log_2(5) = 2.3219$ . In practice the ratio  $16/7 \approx 2.28$  was chosen.

### III. SYNCHRONIZATION

#### A. Synchronous interconnect

A synchronous signal is one that has the exact same frequency, and a known fixed phase offset with respect to the local clock. In such a timing methodology, the signal is

synchronized with the clock, and the data can be sampled directly without any uncertainty. In digital logic design, synchronous systems are the most effortless type of interconnect. Two typical high-speed clocking schemes that use synchronous clocking are sketched in Fig. 9.

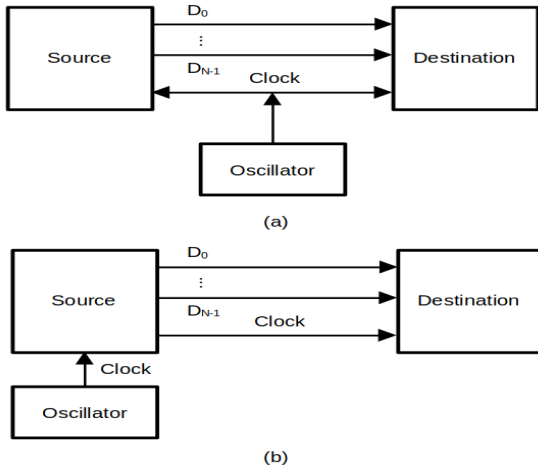


Figure 9. Synchronous clocking: (a) One clock drive for all devices in a system; (b) Clock signal transmitted with data (source synchronous clocking)

One typical source-synchronous point-to-point parallel link interface:

Conventional parallel links are generally source-synchronous, with a clock sent along with the data signals for receiver timing recovery. One typical source-synchronous unidirectional and differential point-to-point parallel link interface architecture is presented in Fig. 10. [34], [35].

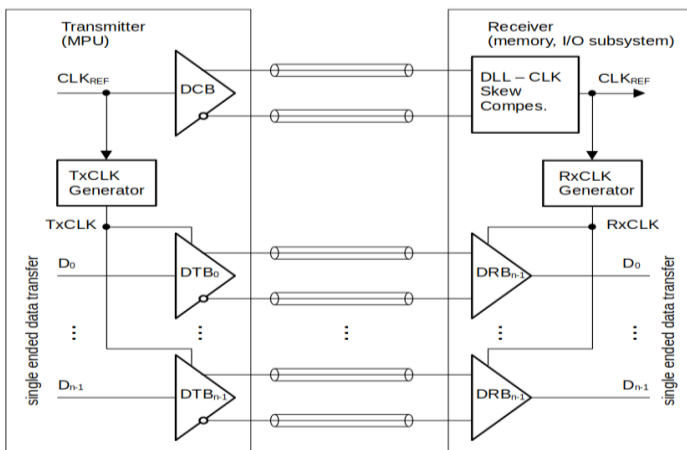


Figure 10. Source synchronous simultaneous unidirectional and differential point-to-point parallel link interface architecture  
Notice:  $CLK_{ref}$  stands for referent clock signal;  $TxCLK$  ( $RxCLK$ ) – global transmitter (receiver) clock signal;  $TxCLKGen$  ( $RxCLKGen$ ) – transmitter (receiver) clock generator;  $DCB$ - differential clock buffer;  $DTB_0, \dots, DTB_{n-1}$  ( $DRB_0, \dots, DRB_{n-1}$ )- differential transmitter (receiver) data buffer;  $DLL-CLK-Skew\_Comp$  – delay locked loop skew compensator;  $D_0, \dots, D_{n-1}$  – data signals

All data signals ( $D_0, \dots, D_{n-1}$ ) and a referent clock signal  $CLK_{ref}$  are transmitted synchronously. Data rate of signals  $D_0, \dots, D_{n-1}$  is determined by  $TxCLK$  ( $RxCLK$ ). At the receiver end, a delay locked loop skew compensator ( $DLL-CLK-$

$Skew\_Comp$ ) generates referent clock signal  $CLK_{ref}$ , while the receiver clock generator  $RxCLKGen$  generates a global receiver clock  $RxCLK$ . The  $RxCLK$  is used to sample all incoming data signals  $D_0, \dots, D_{n-1}$ . Correct sampling is achieved when  $TxCLK = RxCLK$ .

*B. Mesochronous interconnect*

Mesochronous system clock (“meso” from Greek is middle) consists of communication partners that employ clock with the same frequency but with an arbitrary fixed phase shift. Mesochronous synchronizers need two mechanisms to safely interface the transmitter and receiver. First, they need a phase estimation mechanism with aim to determine the phase difference between the transmitter and receiver clock signals. Second, from the result of the phase estimation the synchronizer determines how to adjust a delay in the data path, or on the clock, or control lines. Usually adjustable delay line elements, or alternative data paths are used to adjust the arbitrary phase shift. One typical solution that uses mesochronous clocking is presented in Fig. 11. Mesochronous clocking can operate with or without full CDR. It is used in fast memories, internal system interfaces, MAC/PACKET interfaces, and other designs.

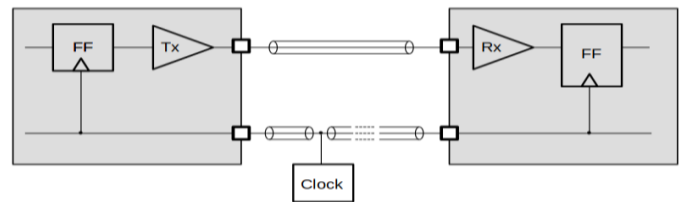


Figure 11. Mesochronous clocking

*C. Plesiochronous interconnect*

A plesiochronous signal is one that has nominally the same, but slightly different frequency as the local clock (“plesio” from Greek is near). In effect, the phase difference drifts in time. This scenario can easily arise when two interacting modules have independent clocks generated from separate crystal oscillators. This implies that the delay within the synchronizers needs to be adopted during operation in order to cope with changing relation in phase and to avoid duplicated or dropped data items. Typically, plesiochronous interconnect only occurs in distributed systems like long distance communications, since chip or even board level circuits typically utilize a common oscillator to derive local clocks. In Fig. 12. one common implementation of plesiochronous clocking is presented.

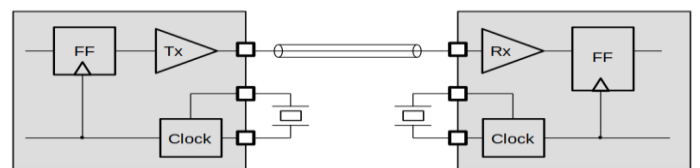


Figure 12. Plesiochronous clocking.

*D. Asynchronous interconnect*

Asynchronous signals can transition at any arbitrary time, and are not slaved to any local clock. As a result, it is not

straightforward to map these arbitrary transitions into a synchronized data stream. Although it is possible to synchronize asynchronous signals by detecting events and introducing latencies into a data stream synchronized to a local clock, a more natural way to handle asynchronous signals is to simply eliminate the use of local clocks and utilize a self-timed asynchronous design approach. In such an approach, communication between modules is controlled through a handshaking protocol to perform the proper ordering of commands.

Since there is no clock in asynchronous circuits, data has to be sent with extra control signals called *req*, for request and *ack*, for acknowledge (see Fig. 13.). Usually a four-phase protocol is used where *req* goes up, followed by *ack* and then *req* goes down, followed by *ack*. The data should be valid between *req* going to one and *ack* returning back to zero. Of course the signals *ack+* and *req-* doesn't matter and that is why a two-phase protocol using transition signaling might be preferred. Transition signaling differs from the "normal" signaling in that the level of the control signals has no meaning. Instead the only thing that matters is when the signal changes. This means that a rising edge is equivalent to a falling edge. These changes are called events. When transition signaling is used for the communication protocol it means that there is an event on *req* and then *ack* answers with an event. The data should be valid between the events.

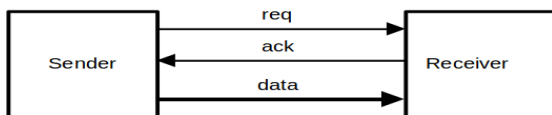


Figure 13. Asynchronous circuit sending data

TABLE I. SYNCHRONIZATION HIERARCHY SUMMARY

Type	Frequency	Phase	Remark
Syn-chronous	same	same	-Every chip gets same frequency AND phase -Used in low-speed busses
Meso-chronous	same	constant	-Same frequency, but unknown phase -Requires phase recovery circuitry -Can do with or without full CDR (Clock Data Recovery circuit) -Used in fast memories, internal system interfaces, MAC/Package interfaces
Plesio-chronous	small difference	slowly varying	-Almost the same frequency, resulting in slowly drifting phase -Requires CDR -Widely used in high-speed links
Asyn-chronous	N/A	arbitrary	-No clocks at all -Request/acknowledge handshake procedure -Used in embedded systems

Notice: Clock Data Recovery Clock data recovery (CDR) provides a technique of embedding the clock within the data to ensure data integrity. The transmission circuitry consists of a serializer and a synchronizer block. The synchronizer takes a clock source and uses this to serialize the data. This clock source is embedded into the data signal before transmission. The receiver consists of a clock recovery unit (CRU) and a deserializer. Data is fed via the receiver into the CRU, which takes the data stream and calculates the clock and phase from the transitions in the data. This clock can then be fed into the deserializer allowing for the data to be recovered in its original form.

E. Globally Asynchronous Locally Synchronous (GALS) clocking

Digital circuit design methods can be classified in two major categories: synchronous and asynchronous. Conventional synchronous digital circuits rely on a global clock signal to function. As advances in VLSI technology enable higher levels of integration in system-on-a-chip (SoC) designs, fully synchronous implementations are becoming less feasible. Globally asynchronous locally synchronous (GALS) clocking is a promising alternative [20], [21], [22]. Each IP core in a GALS system is a synchronous block (SB) of logic (usually with up to 50 000 gates per each IP core or clocked subsystem in Fig. 14.) whose locally generated clock has an independent frequency and phase, while communication between cores is asynchronous. By using GALS approach, it is possible to remove the global clock and replace it with an asynchronous communication scheme. Each core (unit) consists of an asynchronous wrapper and a synchronous module. The synchronous module handles all computations and the asynchronous wrapper handles all communication with other GALS units (see Fig. 15.). The idea of an asynchronous wrapper is that it is used as a camouflage to hide the fact that it is clocked on the inside. That means that a clocked circuit could be used inside the wrapper, but on the outside it acts like an asynchronous circuit [21]. The maybe most important thing needed to accomplish this is the stretchable clock. This clock acts like a normal clock if it is not required to stretch, but when the clocked circuit needs an input or it has to output some data the clock stretches the low part of the clock-period. This means that the clocked circuit sleeps when it is waiting for new data or for outputting data. Communication between wrappers has to be controlled by a handshake protocol since there is no global clock. GALS systems are popular both in software and hardware for specifying and producing embedded systems as well as complex electronic circuits such as SoC designs.

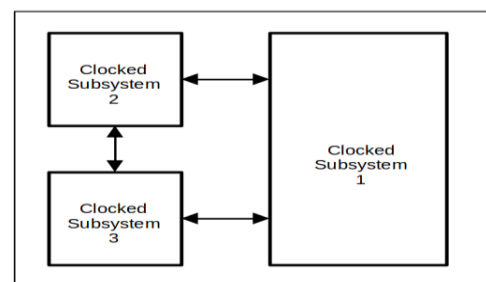


Figure 14. Example of an integrated circuit with several clocks

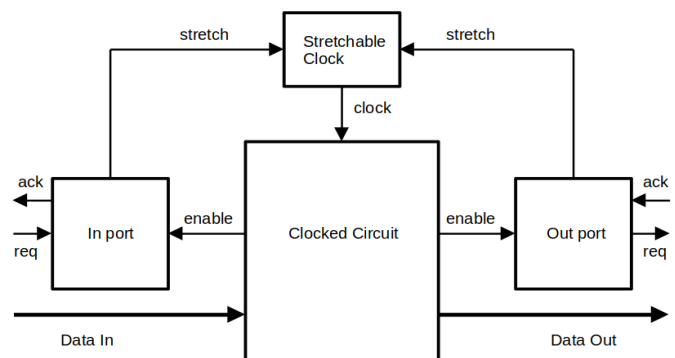


Figure 15. Principal design of an asynchronous wrapper. Inport is the control circuit for data input, Outport controls the data output.



Concluding remarks concerning to GALS: SoC architecture flexibility and scalability requires such an architecture that the performance does not decrease due to increased chip size, longer wires, and more complex clock tree when the number of the components increase. GALS architectures are solution to deal with multiple clock domains. GALS paradigm has been proposed as a compromise between fully synchronous and fully asynchronous architectures. In GALS, the IP blocks of the architecture are locally synchronous, but different blocks are asynchronous relative to each other. It provides different clock speeds for the IP blocks, which is beneficial for full re-usability, maximum performance, and power consumption minimization.

#### IV. CONCLUSION

Continued CMOS process scaling and system integration continues to increase the *on-chip* communication demands

beyond what conventional digital signaling and synchronization methods can efficiently provide. To this end, a short survey of various efficient electrical interconnects implementation techniques was presented in this article. As more promising design solution a multi-level multi-wire interface has been introduced which can transmit information at higher rates than would be possible through a binary differential interface with the same number of wires, while retaining the differential properties of the signal. Besides being rather insensitive to external electromagnetic noise sources, such an interface also generates very little electromagnetic radiation. The synchronizers discussed in this article have been grouped in four rough categories, *Synchronous*, *Mesosynchronous*, *Plesiosynchronous*, and *Asynchronous*. The basic concepts of synchronizers were identified. Furthermore, signal integrity parameter (like impedances, crosstalk, power loss, attenuation, reflections etc..) for high speed interconnects may also be considered for further improvement. The GALS concept as an innovative and effective technique (good compromise between fully synchronous and fully asynchronous architectures) for fast data transfer within the SoC design was also discussed.

At the end of our discussion let known that low-power IC consumption will remains one of the critical challenges for future VLSI systems [36], [37]. We will need innovation at all levels (data processing and data transfer) to continue performance scaling while maintaining power dissipation within acceptable levels. This is especially true since leakage currents have made it hard to continue to scale the supply voltage. The net result is that even high-end processors and fast interconnects are being forced to reduce clock rates and use parallel cores, multi-wire data transfer, and multi-level signaling to control the power dissipation. Driven by the aforementioned limitations, there has been a recent push in the interconnect design community toward a more efficient *on-* and *off-chip* high data transfer approaches such as optical (photonics)-, RF/wireless- and sub-THz/THz-interconnects. All new investigations in this field of research become increasingly important in light of the new trend how to achieve ultra-high-speed energy efficient interconnect [38], [39], [40].

Ideally future interconnect systems must encompass the following important features:

- ultra high data rates, usually > 100 Gbps,

- concurrent multi input-output service for simultaneous and bidirectional communications on a shared transmission medium,
- real-time re-configurability in connectivity and bandwidth for optimized channel efficiency and fault-tolerance, and
- the fabrication of interconnect systems must be compatible with the current SoC (System on Chip) and SiP (System in Package) technologies for low-cost system production, and

as battery-powered devices become popular, energy issues gain more importance and ultra-low-power consumption becomes an imperative and design challenge for long-life system operation.

#### REFERENCES

- [1] C. M. Yousuff, V. M. Y. Hasan, and M. R. K. Galib, *A Survey Addressing on High Performance On-Chip VLSI Interconnect*, International Journal of Electronics and Telecommunications, 2013, Vol. 59, No. 3, pp. 307–312
- [2] M. Stojčev, E. Milovanović, T. Nikolić, “*Multiprocessor Systems on Chip*”, University of Nish, Faculty of Electronic Engineering, 2012, (in Serbian)
- [3] J-P. Deschamps, E. Valderrama, L. Terès, “*Complex Digital Circuits*”, Springer Nature Switzerland AG, 2019
- [4] Moore, B., Sellathamby, C., Slupsky, S., and Iniewski, K. (2008). “*Chip to Chip Communications for Terabit Transmission Rates*”, APCCAS 2008 - 2008 IEEE Asia Pacific Conference on Circuits and Systems, pp. 1558-561
- [5] A. Ganguly, M. M. Ahmed, R. S. Narde, A. Vashist, Md. S. Shamim, N. Mansoor, T. Shinde, S. Subramaniam, S. Saxena, J. Venkataraman, and M. Indovina, “*The Advances, Challenges and Future Possibilities of Millimeter-Wave Chip-to-Chip Interconnections for Multi-Chip Systems*”, Journal of Low Power Electronics and Applications, Vol.8, No 5, 2018, pp. 2-36
- [6] R. Ho, K. W. Mai, and M. A. Horowitz, “*The future of wires*,” Proc. of the IEEE, vol. 89, no. 4, pp. 490–504, Apr. 2001.
- [7] J. D. Owens, W. J. Dally, R. Ho., D.N. Jayasimha, S. W. Keckler, L-S. Peh, “*Research Challenges for On-Chip Interconnection Networks*”, IEEE Micro Magazine, Sept./Oct. 2007, Vol. 27, pp. 96-108
- [8] M. Horowitz, C-K.K. Yang, and S. Sidiropoulos, (1998). High-Speed Electrical Signaling: Overview and Limitations. IEEE Micro, 18(1), pp.12–24.
- [9] E.Yeung, and M. A. Horowitz, “*A 2.4 Gb/s/pin Simultaneous Bidirectional Parallel Link with Per-Pin Skew Compensation*”, IEEE Journal of Solid-State Circuits, Vol. 35, No. 11, November 2000, pp. 1619-1628
- [10] L. W. Ritchey, *A Treatment of Differential Signaling and its Design Requirements*, Prepared by Speeding Edge, May 29, 2008, WWW.SPEEDINGEDGE.COM
- [11] K. Farzan, and D.A. Johns, “*Coding Schemes for Chip-to-Chip Interconnect Applications*”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 14, No. 4, April 2006, pp. 393-406
- [12] M. Šafak, “*Digital Communications*”, John Wiley & Sons Ltd, Chichester, UK, 2017
- [13] Sheikholeslami, A. (2009). *Multi-level Signaling for Chip-to-Chip and Backplane Communication (A Tutorial)*. 2009, 39th International Symposium on Multiple-Valued Logic, pp.203-207
- [14] P. Lancheres and M. Hafed, *The MIPI C-PHY Standard: A Generalized Multiconductor Signaling Scheme*, IEEE Solid-State Circuits Magazine, Spring 2019, pp.69-77
- [15] G. A. Wiley, “*Three phase and polarity encoded serial interface*,” U.S. Patent 8,064,535, 2011.

- [16] D. Messerschmitt, "Synchronization in digital system design," IEEE Journal of Selected Areas in Communications, Vol. 8, No. 8, Oct. 1990, pp. 1404-1419.
- [17] D. J. Kinniment, "Synchronization and Arbitration in Digital Systems", John Wiley & Sons, Chichester, UK, 2007
- [18] J. Sparso, S. Furber, "Principles of Asynchronous Circuit Design – A System Perspective", Kluwer Academic Publishers, 2002.
- [19] R. Kutschera, "Efficient Interfacing Between Timing Domains" Ms.C. Thesis, Technische Universität Wien, 2014
- [20] M. Krstić, "Request-driven GALS for Datapath Architectures", Ph.D. Thesis, University of Cottbus, Germany, 2006
- [21] D.M. Chapiro. "Globally-Asynchronous Locally-Synchronous Systems", PhD thesis, Stanford University, 1984.
- [22] J. Mutersbach, "Globally-Asynchronous Locally-Synchronous Architectures for VLSI Systems" Ph.D. Thesis, ETH, Zurich, 2001.
- [23] C. R. Paul, Transmission Lines in Digital and Analog Electronic Systems: Signal Integrity and Crosstalk, John Wiley & Sons, Inc., Hoboken, New Jersey, 2010
- [24] C. Duan, B. J. LaMeres, S. P. Khatri, "On and Off-Chip Crosstalk Avoidance in VLSI Design" Springer New York, 2011
- [25] B. K. Kaushik, V. R. Kumar, A. Patnaik, "Crosstalk in Modern On-Chip Interconnects: A FDTD Approach", Springer, New York, 2016
- [26] C. Duan, B. J. LaMeres, S.P. Khatri, "On and Off-Chip Crosstalk Avoidance in VLSI Design", Springer, NewYork, 2010
- [27] S. Kempainen, "National Semiconductor, Application note 1382 – 6, Low Voltage Differential Signaling LVDS by Agilent", 2010, pp. 1-20
- [28] D. Lewis, "SerDes Architectures and Applications" DesignCon 2004, National Semiconductor Corporation, USA, 2004, pp.1-24
- [29] Dikhaminjia, N., He, J., Tsiklauri, M., Drowniak, J., Fan, J., Chada, A., Achkir, B. (2016). "PAM4 Signaling Considerations for High-Speed Serial Links", 2016 IEEE International Symposium on Electromagnetic Compatibility (EMC), pp. 906-910
- [30] J. W. Poulton, S. Tell, and R. Palmer, "Multiwire Differential Signaling", UNC-CH Department of Computer Science, 2003, pp.1-20, available 20.10.2021, at <https://www.semanticscholar.org/paper/Multiwire-Differential-Signaling-Poulton-Tell/c7a23e0b20c335898e2b966b565fefe5b9b3ec64>
- [31] P. Baltus, P. van der Meulen, R. Morley, "An Efficient Multi-Level Multi-Wire Differential Interface", Proceedings of the Twentieth International Symposium on Multiple-Valued Logic, 23-25 May, Charlotte, NC, USA, 1990, pp.181-188
- [32] J. Puskala, "High-Speed Camera Serial Interface Verification", MsC. Thesis, Tampere University of Technology, 2014
- [33] "MIPI Alliance Specification for C-PHY" MIPI Alliance, 2016. [Online]. Available: 22.10.2021 at: <https://www.mipi.org/specifications/c-phy>
- [34] G. Jovanović, M. Stojčev, T. Nikolić and G. Nikolić, Delay Locked Loop Clock Skew Compensator for Differential Interface Circuit, 53rd International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST 2018), Sozopol, Bulgaria, June 28-30, 2018, pp.115-118
- [35] D. Mitić, G. Jovanović, M. Stojčev, D. Antić, "Phase Control Loops in Analog and Digital Circuits", Monography, University of Nish, Serbia, 2020
- [36] M. Horowitz, "Computing's Energy Problem (and what we can do about it)": Proceedings of the 2014 IEEE International Solid-State Circuits Conference, Digest of Technical Papers (ISSCC), 2014, pp.10-14
- [37] M. E. Lee, W. J. Dally and P. Chiang, "Low-power area efficient high-speed I/O circuit techniques," IEEE J. Solid-State Circuits, vol. 35, pp. 1591–1599, Nov. 2000.
- [38] Q. J. Gu, "Sub-THz/THz Interconnect, Complement to Electrical and Optical Interconnects", IEEE Solid-State Circuits Magazine, Vol. 12, No. 4, Fall 2020, pp. 20-32
- [39] T. SaiWang, C. M-C. Frank, "RF/Wireless-Interconnect: The Next Wave of Connectivity", Science China, Information Sciences, Published by Science China Press and Springer-Verlag Berlin Heidelberg 2011, May 2011 Vol. 54 No. 5: pp:1026–1038
- [40] M-C. Frank Chang V. P. Roychowdhury, Z. H. Shin, and Y. Qian, "RF/Wireless Interconnect for Inter- and Intra-Chip Communications", Proceedings of the IEEE, Vol. 89, No. 4, April 2001, pp: 456-466



**Mile Stojcev** (1946) received BsC, MsC, and PhD degrees from faculty of Electronic Engineering, University of Nish, R. Serbia, in 1970, 1978, and 1984, respectively. He worked at RTV Skopje, R. North Macedonia, from 1970 up to 1978, and with faculty of Electronic Engineering, University of Nish, from 1978. His current research interests include embedded systems design, System-on-Chip (SoC) design, and wireless communications. He has published over 300 scientific papers, and 15 books.



**Bojan Dinitrijevic** (1972) received BsC, MsC and PhD degrees from faculty of Electronic Engineering, University of Nish, R. Serbia, in 1998, 2002 and 2006, respectively. From 2003 he worked with the faculty of Electronic Engineering, University of Nish, as researcher. His main areas of research are in the field of signal processing in telecommunication systems, modeling of the electromagnetic environment, and embedded system programming. He has published more than 120 scientific papers and 2 books.