Advances in Radio Science (2004) 2: 187–193
© Copernicus GmbH 2004

Advances in
Radio Science

Different approaches of high speed data transmission standards

M. Ehlert

Micronas GmbH, Freiburg, Germany

Abstract. A number of standards addresses the problem of high-speed data transmission on serial or serial-parallel data lines. Serial-parallel data transmission means the transmitted information is distributed on parallel data lines. Even though several standards exist, there are only a few basic techniques used in most of these standards. This paper is giving an overview of these different basic techniques used in the physical layer of today's data transmission standards, for example DVI/HDMI, USB2.0, Infiniband, SFI5, etc. [1–9]. The main focus lies on the approaches used for physical signaling, line coding and information synchronization in serial and serial-parallel systems. In addition, currently discussed techniques to improve data transmission in the future will be presented.

1 Introduction

High-speed data communication standards originally have addressed fiber optics systems for telecommunication and datacom. Nowadays, the increasing operation speed of modern processors, computer busses as well as low cost data interfaces e.g. USB2.0 have reached a high-speed data communication of several hundreds of megabits data rate per second and gigabits per second which are about to come to mass markets soon.

Therefore, in addition to fiber optics systems, backplane and bus architectures are addressed by high-speed data transmission standards. Since these applications usually do not need to cover long distances, copper cables are used as interconnection media for speeds up to several gigabit per second to reduce system cost which is the key to the mass market. Today, distances of 20–30 m can be implemented with such cables which are sufficient for wide areas of applications.

Despite the different applications like fiber optics, copper cable, or backplane data transfer, the various synchronization techniques used in the different standards are still quite close to each other. Differences mostly originate from the various coding schemes used to encode the data and to transmit special characters used in the synchronization process. This paper will address the most often-used design techniques in today's standards. The interfaces are described starting with the electrical signal form as the lowest physical level and going up to line coding and frame processing as the highest level of data transmission.

First, an example of an electrical data path and the electrical signals used for data transmission is explained in Sect. 2. Furthermore, timing budget issues are discussed here. In Sect. 3, phase detection techniques used for bit synchronization are explained. Then, in Sect. 4, data coding schemes are discussed using the examples of 8B10B and TMDS signaling. Word synchronization and speed matching techniques are presented in Sect. 5. A summary and conclusions are given in Sect. 6.

2 Physical data transmission

Electrical data paths suffer from damping and dispersion and, in the case of parallel data transmission, from skew between different signal lines. Moreover, connectors add considerable distortion to the signals due to their usually bad high frequency characteristics which leads to reflections. An example of a data path for backplane data transmission is given in Fig. 1.

For backplane transmission, two ICs placed on different PCBs are connected via two connectors to a backplane or main board. With FR4 PCB material, skew and damping are about 250 fs/mm and 12 mdB/mm, respectively. For example, in the Infiniband standard [6], 20" of transmission length is allowed. This leads to a damping of about 6.1 dB and a skew of 130 ps, respectively, only for the signal lines. Since Infiniband allows an eye closure of 2/3, at 3.125 Gbit/s there is only 106 ps of eye opening left which is smaller than the total skew. In addition, there is the influence of the connectors which usually is even worse than that of the signal

Correspondence to: M. Ehlert
(martin.ehlert@ieee.org)

Table 1. Electrical signal specification ranges.

Parameter	Encoder	Decoder
Eye Opening	60–80%	30–50%
Amplitude	250–1000 mV	150–400 mV

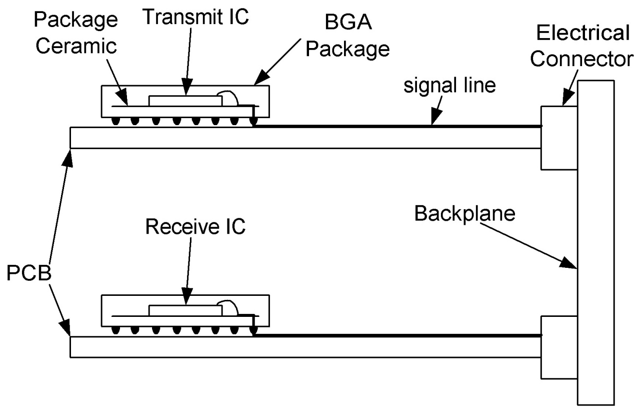


Fig. 1. Datapath for backplane data transmission.

lines. Therefore, retiming of several parallel data lines with only one common clock phase is not possible. Techniques to achieve retiming under such conditions will be discussed in Sect. 5.

The particular minimum eye opening and amplitude of the various standards specify the electrical signals (see Fig. 2). The ranges for these values are given in Table 1. Typically, with these values, a bit error rate (BER) of 1e12 must be achieved. So far, only DVI/HDMI accept a higher BER of 1e9 since single bit errors are less important in video applications.

In addition to the specification of a simple eye diagram, some standards like OpenLDI [22] specify waveforms with pre-amphasis (Fig. 3) as an option to reduce the effects of band limitation of the data path. This, however, is not a must, yet.

Moreover, it is discussed to include reconstruction filters at the receiver to widen up the eye opening again for the retiming. These filtering options are interesting for three reasons. One is to upgrade older equipment for higher data transmission rates without a need to change the transmission cables or PCBs, second is to reduce system cost allowing low cost connectors and PCB material, and finally to enhance the transmission distance.

The most important difference in the specification of the electrical signal between the various standards is the minimum eye opening allowed at the receiver. The retiming clock at the receiver needs to retime the data within this residual eye opening. All internal error sources of the receiver must be subtracted from this opening. Error sources are e.g. clock jitter of the retiming clock, phase error of the clock alignment, setup and hold time of the retiming Flip-Flops as

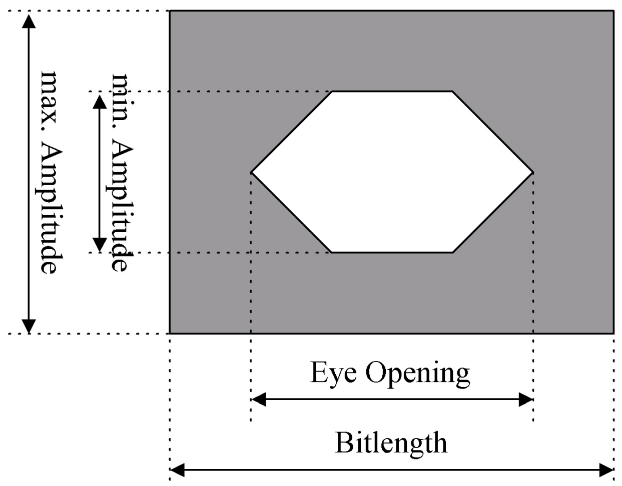


Fig. 2. Definition of eye diagram.

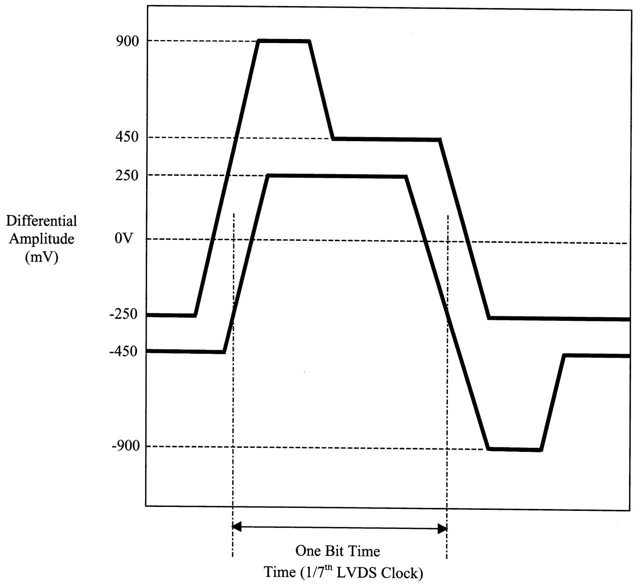


Fig. 3. OpenLDI specification for optional pre-amphasis.

well as influences from the receiver package. All these error sources must be accounted for in an overall timing budget, which yields the final real eye opening usable for the retiming operation.

3 Bit synchronization

The next level of data transmission following the physical signal is the recovering of single bits from the electrical signal. This process is called bit synchronization. The most common technique to establish bit synchronization is to use a phase locked loop (PLL) [10–15]. This can either be a clock-and-data recovery (CDR, see Fig. 4) or, in case a correct clock is available already, a delay-locked-loop (DLL).

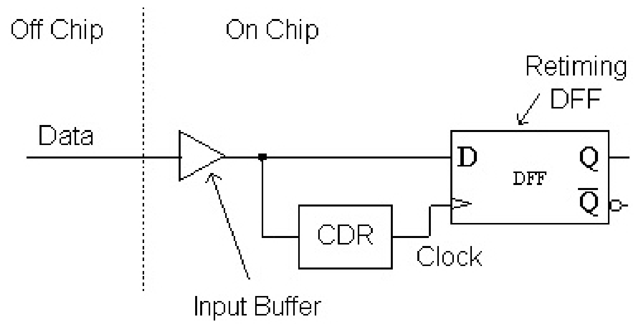


Fig. 4. Bit synchronization using clock-and-data recovery.

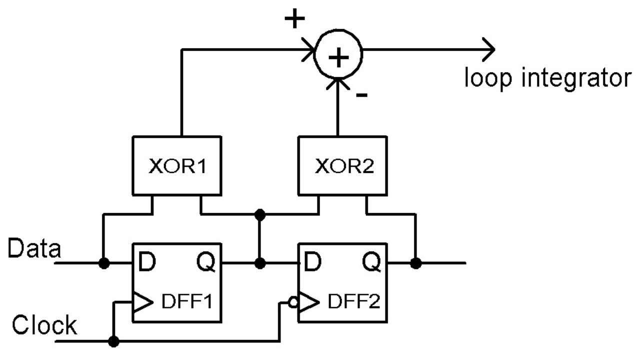


Fig. 5. Principle of phase detector proposed by Hooge [16].

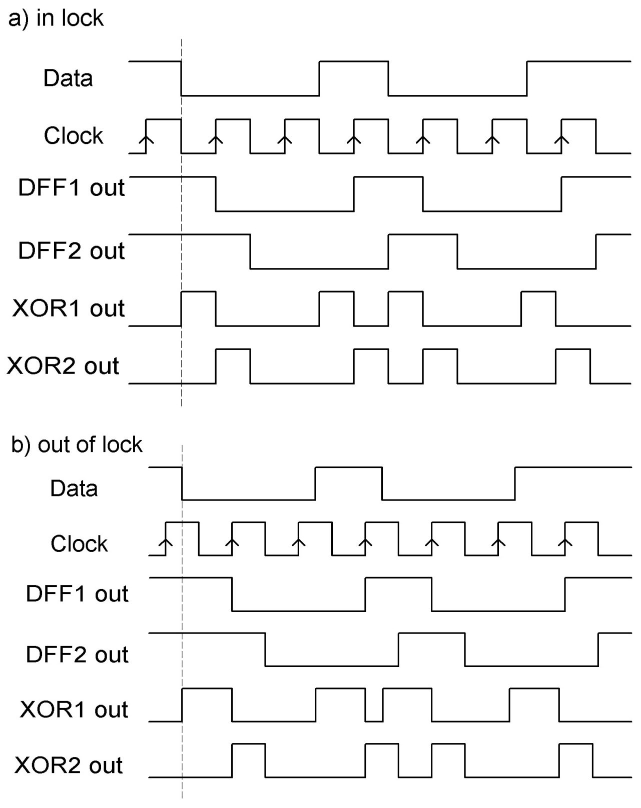
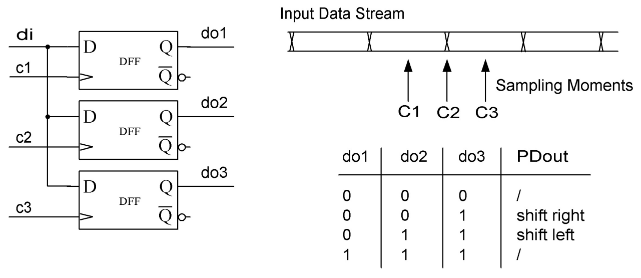


Fig. 6. Waveforms of Hooge phase detector **(a)** in lock **(b)** out of lock.



do1	do2	do3	PDout
0	0	0	/
0	0	1	shift right
0	1	1	shift left
1	1	1	/

Fig. 7. Example of bang-bang phase detector with timing and truth table.

Since this technique allows high phase accuracy, it is used for small eye openings at the receiver. In contrast, a majority vote is used if lower phase accuracy and/or if a very short lock in time is needed (as e.g. in USB2.0 with 50% eye opening and phase lock within 12 bit).

The evaluation of the phase error in a CDR can be done either analog or digital.

3.1 Analog phase detection

Analog phase detection is the traditional way of phase detection [16–18]. In analog phase detectors, an output voltage is generated which is proportional to the phase error. This is mostly done with a structure based on the principle proposed by Hooge (Fig. 5, [16]).

In the Hooge phase detector, two flip-flops are sampling the incoming data. The clock for both retiming operations is shifted by 180°. An XOR gate evaluates the input and output of each flip-flop. In lock, the output pulse trains of both XOR gates are identical but shifted in time (Fig. 6a).

If the data signal is not sampled in the bit middle than the output of the first XOR of the Hooge phase detector changes its duty cycle depending on whether the clock is late or early. The second XOR gate stays unaffected thus working as a reference. Therefore, the phase error can be evaluated by the difference between the output pulses of the two XOR.

The problem of analog phase detectors is that the pulses generated for low phase errors are very narrow. This requires fast evaluation circuits to process the small pulses. Otherwise, the loop dynamics can be degraded due to dead zones. Therefore, it has become more common lately to use digital or bang-bang phase detectors for high speed applications.

3.2 Digital phase detection

In contrast to analog phase detectors, digital phase detectors only make the decision if the clock phase is early or late compared to the data (e.g. [19, 20]). The output is independent of the amount of phase error. In Fig. 7, an example of a bang-bang phase detector is given, together with a timing diagram and a truth table.

In this example, the incoming data is sampled at three times, two succeeding bits (phases C1 and C3) and the

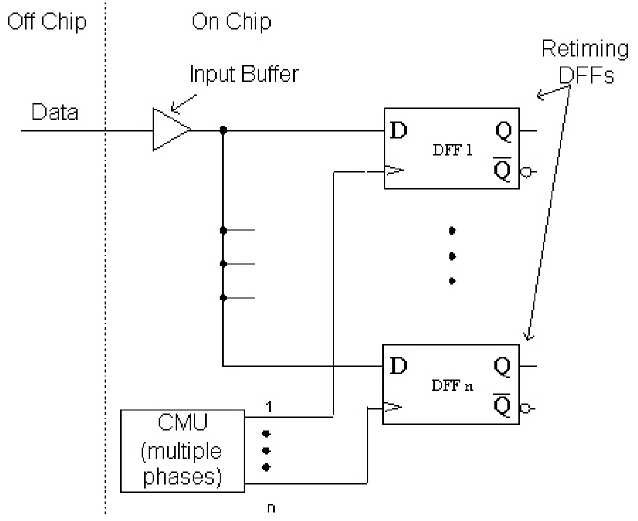


Fig. 8. Example for majority vote system, incoming data is retimed with n phases.

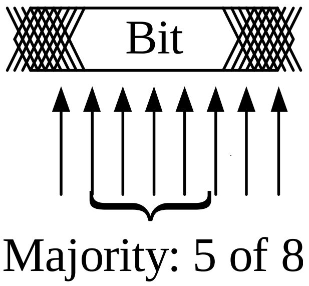


Fig. 9. Majority vote: more than half the sampling phases inside the eye opening.

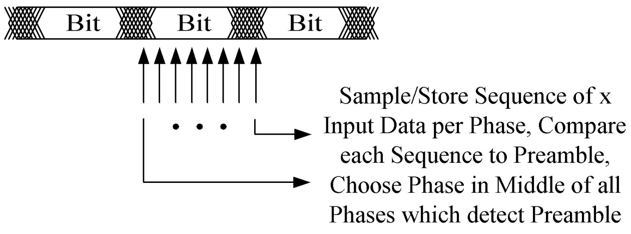

Bit
Bit
Bit
Sample/Store Sequence of x Input Data per Phase, Compare each Sequence to Preamble, Choose Phase in Middle of all Phases which detect Preamble

Fig. 10. Majority vote with preamble or training sequence.

transmission between those bits (phase C2). If C2 is identical to C1, the clock has been too fast. If C2 is identical to C3 the clock has been too slow. If all phases yield the same output, no transition has occurred and no decision could be made. In lock, C2 theoretically samples directly in the bit middle and therefore its output is not defined.

Other bang-bang structures sample the internal clock with the data signal which also yields the needed phase information. The advantage of bang-bang detectors, which are evaluating the data, is the perfect matching of phase evaluation and retiming operation since both actions are done at the same time and with the same flip-flops.

The general drawbacks of these kinds of phase detectors are an increased noise at the control node of the VCO and a more difficult handling. This is because a PLL using a digital phase detector is not a linear system anymore.

3.3 Majority vote phase detection (oversampling)

In a majority vote or oversampling phase detection system, the incoming data is retimed several times with different equally spaced clock phases (see Fig. 8).

The correct phase is then found either by a majority decision or by comparison with a test data sequence during an initial training interval. For a majority decision, the eye opening must be wide enough to have more than half of the sampling phases inside the data eye (Fig. 9). Therefore, this is only possible for eye openings wider than 50%.

More common for the usage of higher frequencies is the comparison of the sampled data to a preamble or training sequence during startup or certain intermediate training intervals (Fig. 10). For example, USB2.0 has an initial sequence of a minimum of 12 bit of 01 pattern followed by a single 1 which can be searched for in the different sampled data streams of the phases.

3.4 Chip border crossing

A specialty of data transmission is the communication of two chips sitting adjacent to each other on the same board or even in the same package. This is the case for serializer-deserializer (serdes) applications where it is not yet possible to implement the whole serdes function within one IC and InP or GaAs ICs are needed. This special problem is not directly addressed by any standard yet but has been discussed for parallel data transmission with data rates higher than 10 Gbit/s/channel. At those speeds, the eye opening is so small that even the skew introduced by the packages can be too high to retime the data on the receiving IC without deskewing.

Since the amount of circuitry on the upstream side IC with InP and GaAs is limited due to technology reasons, the idea is to implement a DLL across the chip borders to do the deskewing. To achieve this, on the upstream side only the phase detector must be implemented. Then, during a training period delay elements are trained on the downstream side to equalize the skew between the different data lines. As a result, the amount off circuitry on the upstream side is greatly reduced (see Fig. 11).

Common to all the bit synchronization techniques is the need for transitions in the input signal. If there are no bit changes than no phase information can be derived. Therefore, a minimum number of transitions must be specified in order to guarantee correct operation of the synchronization. Since this usually cannot be guaranteed by the data itself, line coding is used to establish a minimum number of data signal transitions.

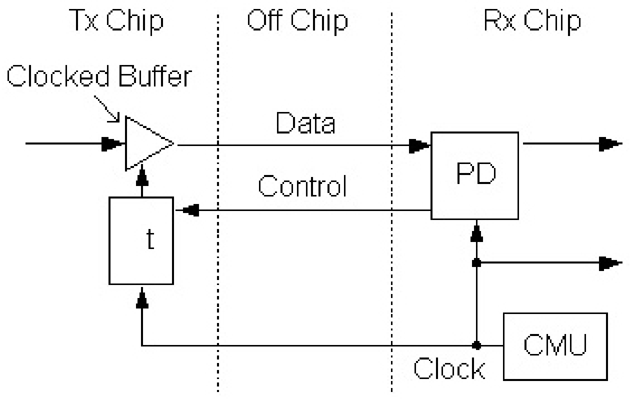


Fig. 11. Block diagram for deskewing of data lines across chip borders, only a phase detector (PD) is placed on the receiver (RX), the deskewing is done on the transmitter (TX).

4 Line coding

The next level of data recovery following the bit synchronization is the extraction of the information from the bit stream. To achieve this, the word boundaries must be found from the data stream.

In package oriented transmission standards, this can simply be done by defining a known start pattern. For example, in USB2.0, a minimum sequence of 12 succeeding low-high pattern followed by a single low is used. After this preamble, the transmitted bits contain real data information. Therefore, the start of the word boundary is known after the initial synchronization of the link.

In standards where a constant data stream is applied the word boundary must be found without initial knowledge of the position of a special bit. For standards that allow the distribution of the transmitted information on parallel data lines (serial-parallel data transmission), also the problem of skew compensation is addressed. In that case, deskewing information must be added to the data stream. Only if the skew on the bit lines do not exceed one bit (this includes the bit synchronization!) [15] this is not necessary.

All these tasks are achieved by encoding of the transmitted data. The biggest differences between the various standards result from theses kinds of line coding that are used. There are several more reasons to do line coding in general. Some of these are:

1. Provide a 50% mark ratio for DC balance (equal numbers of “0” and “1” on average).
2. Allow frame border detection through synchronization patterns or known sequences.
3. Transmission error detection/correction.
4. Supply special control patterns.
5. Achieve a certain spectrum for the transmitted data.

Table 2. TMDS coding table for 20 bit alternating sequence, bits are transmitted LSB first.

8 bit data word (MSB...LSB)	10 bit transmit word (MSB...LSB)
0 0 0 0 0 1 0 1	1 1 1 1 1 1 1 1 0 0
0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 0 1 1	0 0 0 0 0 0 0 0 1 1
1 1 1 1 1 1 1 0	0 0 0 0 0 0 0 0 0 0

Table 3. TMDS coding table for maximum run length of 22 high bits, bits are transmitted LSB first.

8 bit data word (MSB...LSB)	10 bit transmit word (MSB...LSB)
0 0 0 0 0 0 0 0	0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 1	1 1 1 1 1 1 1 1 0 0
0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1 1 1
0 0 0 1 0 0 0 1	0 1 0 0 0 0 1 1 1 1

As an example, the coding schemes 8B10B and TMDS (Transmission Minimized Differential Signal) will be chosen for comparison.

Both codes transfer an 8 bit data word into a 10 bit transmit word. While for 8B10B the coding is done with a complex logic function [21], TMDS only needs a simple logic to generate the 10 bit transmit words [2, 9].

The 8B10B code provides a 50% mark ratio with a maximum run length of only 5 bits. This maximum run length is only achieved within the 12 special characters. Therefore, the special characters can easily be detected by this 5 bit sequence. They are used to recover the 10 bit word boundary and to transfer control information.

In the case of serial-parallel data transmission, special characters furthermore are used for deskewing. The information in each data line is framed by special characters. The receiver detects the special characters in each channel and compensates the skew by alignment of those characters. This is possible as long as the system skew does not reach half the length of the total data package including the framing.

The short run length of the 8B10B code helps in the design of the bit synchronization circuit at the receiver. This is because the jitter generated by those blocks depends on the number of transitions in the data stream.

For TMDS on the other side, there is no certain run length given in the specification. With some patience, one can find an input pattern that yields an alternating sequence of 20 highs and lows or a maximum run length of 22 bits (see Tables 2 and 3). Furthermore, fast bit changes after a long sequence (e.g. 21× high, followed by one low and one high) can be found. This makes the design of the bit synchronization circuit more difficult.

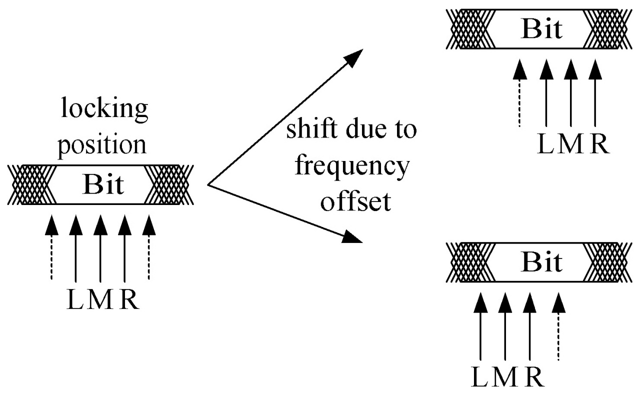


Fig. 12. Phase tracking for oversampling and majority vote synchronization systems.

As 8B10B, TMDS offers special transmit characters meant to transmit control information. The difficulty is that the four available patterns do not offer a unique sequence in the data stream, as does 8B10B. In contrast, control patterns must always be sent and detected in a sequence. This is because it is not possible to achieve a data pattern that matches two subsequent special characters from sending coded data patterns so the frame borders can be detected by at least two of these patterns in sequence.

One advantage of TMDS patterns is that electro-magnetic interference is reduced due to the reduced number of transitions.

Both codes have a 25% overhead for transmission data.

As can be seen from this evaluation, both codes have optimized different things. Despite looking close on the first side, they have some major implementation differences. While 8B10B encoding and decoding takes considerably more circuitry to implement, TMDS is more difficult to deal with for the bit synchronization at the receiver.

In that way, every other code, e.g. 64/66, scrambling etc. is optimizing different things but in general, the main tasks of line coding given above apply to all of them.

5 Matching transmission speeds

If transmitter and receiver are far apart as in most applications, they do not have the same physical clock source. Therefore, the clock speeds at the transmitter domain and at the receiver domain do not perfectly match. Usually, clock differences of up to 1000 ppm are allowed.

This difference in the clock speeds leads to two problems. First, for oversampling and majority vote systems, which run at a local clock, the bit synchronization circuit at the receiver must be tuning all the time. This is necessary to keep the optimal sampling point for the incoming data for those systems and to prevent data loss. The tuning is done by constant evaluation of more than the optimal sampling phase found in the initial start up. If e.g. three adjacent phases are evaluated, no phase shift is necessary as long as all phases yield the same sampling result. If for example the evaluation of the left of three phases yields a different result than the other two phases, the former right phase will become the new middle phase, an additional phase is added at the right side of the sampling group, the former middle phase becomes the new left phase and the former left phase is not used for evaluation in the next time step anymore. This process is illustrated in Fig. 12.

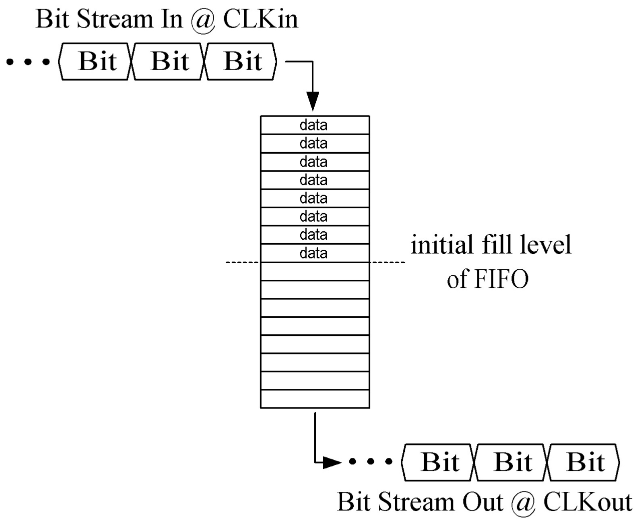


Fig. 13. Block diagram of simple elastic buffer.

For analog or digital phase detectors in a CDR, reference clock differences at transmitter and receiver are not a problem. Here, the CDR loop always runs at the clock speed of the transmitter due to the tuning of the local VCO.

The second problem related to the clock mismatches at transmitter and receiver is to loose or add data at the receiver. This is because the clock differences add up and eventually exceed one bit length.

In package-oriented standards like USB2.0, a simple elastic buffer can be used to compensate for these clock speed differences. An elastic buffer is a FIFO buffer where the write and the read operations are done at different clock speeds. In the case of USB2.0, not more than 12 bit mismatch can occur during the transmission of one package. Therefore, if an elastic buffer with a depth of 24 bits is filled up with 12 bits, no data loss or addition can happen. This is illustrated in Fig. 13.

For standards using constant data streams, the elastic buffer is more complicated. For example, in Infiniband, there are special areas defined in the data stream which do not contain real data. The elastic buffer detects these areas by the framing of special characters. Then, the elastic buffer adds or subtracts words from or to these areas respectively depending on the current fill level of the buffer. Due to this process, differences in the clock speeds are compensated. In Fig. 14, this is shown in principle. The shaded words are the fill pattern. In the Infiniband standard for example, with the maximum allowed clock offset of 200 ppm, a 10 bit word must be filled in or extracted from the data stream approximately every 16 us.

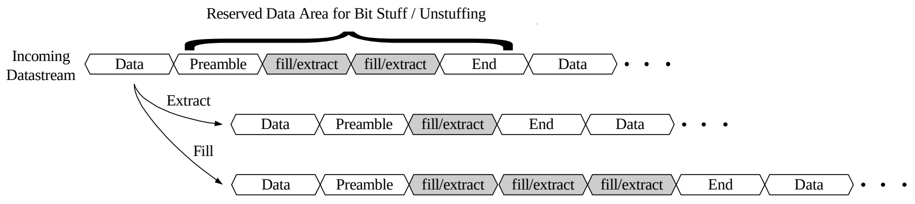


Fig. 14. Principle of clock speed matching process for constant data stream systems.

6 Summary and conclusions

It has been shown that for the physical layer of today's high-speed standards, there are not many different design techniques used to recover the data from the data stream. In general, two different techniques, oversampling and bang-bang based CDR loops, are used. In addition, techniques to enhance the transmission distance (or to reduce system cost) like pre-amphasis and reconstruction filters are about to be included in future implementations.

The main difference between standards today results from the various kinds of data codes used to encode/decode the data and to add synchronization information. These various coding techniques are the main source of differences in the implementation of transmitters and receivers.

Acknowledgement. The author would like to thank J. Höhn and A. Weggel for helpful discussions.

References

[1] Electrical Characteristics of Low Voltage Differential Signaling (LVDS) Interface Circuits, Revision A, ANSI/TIA/EIA-644-A Electrical characteristics standard.
[2] DVI standard, www.ddwg.org.
[3] USB standard, www.usb.org.
[4] FireWire, www.apple.com/firewire.
[5] Optical Internetworking Forum (OIF), Physical Link Layer (PLL) Working Group, SFIX, www.oiforum.com.
[6] Infiniband standard, www.infinibandta.org/home.
[7] Intel Developer Network for PCI Express Architecture www.intel.com/technology/pciexpress/devnet.
[8] Gigabit Ethernet Alliance (XAUI), www.10gea.org/Tech-whitepapers.htm.
[9] HDMI standard, www.hdmi.org.
[10] Fukaishi, M., Nakamura, K., Sato, M., Tsutsui, Y., and Yotsuyanagi, M.: A 4.25-Gb/s CMOS Fiber Channel Transceiver with Asynchronous Tree-Type Demultiplexer and Frequency Conversion Architecture, IEEE Journal of Solid-State Circuits, 33, 12, 1998.
[11] Sakamoto, T., Tanaka, N., and Ando, Y.: Low-latency Skew-compensation Circuits for Parallel Optical Interconnections, ECTC 1999.
[12] Wadhwa, R., Aggarwal, A., Edwards, J., Ehlert, M., Höhn, J., Miao, G., Lakshmikumar, K., and Khoury, J.: A Low Power 0.13 μm CMOS OC-48 SONET and XAUI Compliant SERDES, CICC, 2003.
[13] Schmale, I., Heinemann, M., Drögemüller, K., Kuhl, D., Blank, J., Ehlert, M., Kraeker, T., Höhn, J., Klix, D., Plickert, V., Melchior, L., Hildebrandt, P., Leininger, L., Dröge, E., Kropp, J.-R., Wolf, H.-D., Wipiejewski, T., and Johnson, R.: High Speed 12×2.5 GBit/s parallel optical links (PAROLI) for increased transmission length, European Conference on Optical Communication (ECOC), 4–7 September, München, 2000.
[14] Drögemüller, K., Kuhl, D., Blank, J., Ehlert, M., Kraeker, T., Höhn, J., Klix, D., Plickert, V., Melchior, L., Schmale, I., Hildebrandt, P., Heinemann, M., Schiefelbein, F. P., Leininger, L., Wolf, H.-D., Wipiejewski, T., and Ebberg, A.: Current Progress of Advanced High Speed Parallel Optical Links for Computer Clusters and Switching Systems, 50th Electronic Components & Technology Conference (ECTC), 21–24 May Las Vegas, Nevada, USA, 2000.
[15] Kuhl, D., Drögemüller, K., Blank, J., Ehlert, M., Kraeker, T., Höhn, J., Klix, D., Plickert, V., Melchior, L., Hildebrandt, P., Heinemann, M., Beier, A., Leininger, L., Wolf, H.-D., Wipiejewski, T., and Engel, R.: PAROLI, A Parallel Optical Link with 15GBit/s Througput in a 12 Channel Wide Interconnection, 6th Parallel Interconnect (PI), 17–19 October, Anchorage, Alaska, USA, 1999.
[16] Hooge, Jr., C. R.: A Self Correcting Clock Recovery Circuit, Journal of Lightwave Technology, LT-3, 6, December, 1985.
[17] Lee, T. H. and Bulzacchelli, J. F.: A 155-MHz Clock Recovery Delay- and Phase-Locked Loop, IEEE Journal of Solid-State Circuits, 27, 12, December, 1992.
[18] Savoj, J. and Razavi, B.: A 10-Gb/s CMOS Clock and Data Recovery Circuit, 2000 Symposium on VLSI Circuits Digest of Technical Papers, 2000.
[19] Pottbäcker, A., Langmann, U., and Schreiber, H.-U.: A Si Bipolar Phase and Frequency Detector IC for Clock Extraction up to 8 Gb/s, IEEE Journal of Solid-State Circuits, 27, 12, December, 1992.
[20] Reinhold, M., Dorschky, C., Rose, E. et al.: A Fully Integrated 40-Gb/s Clock and Data Recovery IC With 1:4 DEMUX in SiGe Technology, IEEE Journal of Solid-State Circuits, 36, 12, December, 2001.
[21] Widmer, A. X. and Franaszek, P. A.: A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code, IBM J. Res. Develop., 27, 5, September, 1983.
[22] Open LDI standard, v0.95.