

# Hierarchical System Synchronization and Signaling for High-Performance – Low-Latency Interconnects

Peter Müller, Urs Bapst, Ronald Luijten  
IBM Research GmbH, IBM Zurich Research Laboratory  
8803 Rüschlikon, Switzerland  
E-mail: pmu@zurich.ibm.com

## Abstract

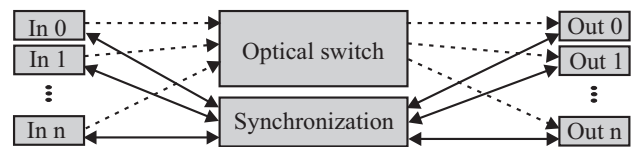
We address a hierarchical synchronization distribution architecture for high-performance and low-latency operations. Furthermore, the bandwidth overhead is minimized, and the accuracy can be adjusted to the application. A novel signaling channel with an open, user-extendable protocol is proposed. An approximation method to estimate system-wide clock jitter is introduced and applied to the Optical Shared MemOry Supercomputer Interconnects System (OSMOSIS). First measurement results, which reveal the challenges of future system synchronization requirements and the potential of the defined architecture, are presented.

## 1. Introduction

Synchronization is the art of distributing timing and signaling information over a set of connected processes within a specified area. Much work has already been done on various aspects of synchronization because almost all emerging technologies and any networked application require synchronization techniques: Hu and Servetto discuss the algorithmic aspects of synchronization in large-scale wireless sensor networks [1]. Sheu *et al.* have proposed an algorithm for dynamic clock synchronization in an ad-hoc multi-hop network [2]. Kelly and Manohar reported a simulation method for latency-critical applications in large networks [3]. Chen analyzed the effect of satellite networks on real-time applications [4]. Work based on the network time protocol (NTP) or coordinated universal time (UTC) has been reported by Mills [5], Butner and Vahey [6], Pásztor and Veitch [7], and Schossmaier and Weiss [8]. Liebing *et al.* [9] and Abali *et al.* [10] discuss the synchronization of core networks within the service server infrastructure using massively parallel systems. Furthermore, high-performance super-computer interconnect system synchronization is reported by Fehrer *et al.* [11] and Gambini *et al.* [12]. Efforts

targeting internal synchronization at the system and chip levels with high reliability are discussed by Yashiro *et al.* [13]. Mitra analyzes a hybrid master-slave and mutual synchronization architecture [14].

In our paper, we focus on the synchronization of a small packet-switched interconnection system based on electro-optical technologies for high-performance computers, named Optical Shared MemOry Supercomputer Interconnects System (OSMOSIS) [15]. As a full optical data path contains no buffers, the most challenging requirements of such an application are in terms of synchronization. Specifically, the input/output participants have to be phase- and frequency-synchronized over the entire space of the system. Figure 1 depicts a simplified view of OSMOSIS.



**Fig. 1.** The central unit is the electro-optical switch of the OSMOSIS. The upper box represents the optical data switch, whereas the lower box symbolizes the synchronization function. The input and output participants are shown on the left- and right-hand side, respectively. The straight arrows are synchronization channels, and the dashed arrows the optical data paths, which are shown only for the sake of clarity. Physically the two channels may be multiplexed together.

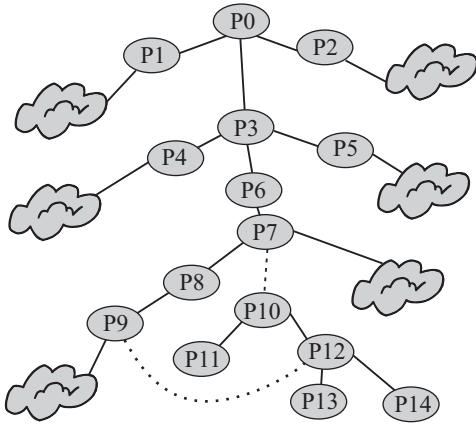
We address a generalized top-down strategy of an overall system-wide architecture for synchronization. The paper is organized as follows. Sections 2 to 5 define the structure, clock, signaling, and protocol required by the system. Section 6 introduces an experimental setup, Section 7 presents results on real-time signaling and reference clock transfer jitter, and Section 8 contains the conclusions.

## 2. Synchronization Distribution Structure

As shown in Fig. 2, a high-accuracy system-wide synchronization leads to a hierarchical, tree-like domain architecture, with a master clock source at its top (process P0). This, however, does not preclude that participants maintain additional synchronization channels. As there is always a channel having the highest quality of service (QoS), which is shown in Fig. 2. If a synchronization channel breaks down, as indicated in Fig. 2 by the dashed line between P7 and P10, P10 will change to act as a master clock source for the processes P10 to P14. By definition, the sub-domain of processes {P10 ... P14} will sooner or later lose synchronization to the P0 reference unless a redundant synchronization channel is found. A more comprehensive description of the redundant synchronization channels is given in Section 5.

How to synchronize or re-synchronize a single process or a sub-domain onto a higher QoS domain without affecting ongoing user tasks is one of the most challenging topics in synchronization theory. Two basic functions are required to initialize and maintain reliable synchronous processes:

1. A phase-stable system-wide reference clock.
2. A signaling channel protocol.



**Fig. 2.** Structured master-slave synchronization tree. P0 is the system master because it has the highest quality of service. The dashed line between P7 and P10 shows a broken synchronization channel. The sub-domain {P10 ... P14} is being mastered by P10, but runs plesinchronously to the domain mastered by P0. The dotted line between P9 and P12 indicates a redundant synchronization channel.

## 3. Reference Clock

The stability of the reference clock can be characterized either by the phase noise, which is a frequency-domain view of the noise spectrum around a reference frequency;

or by the jitter, which is an equivalent measure of the accuracy of the clock period in the time domain.

From an engineering point of view, a synchronization channel consists of a number of different components, such as oscillators, phase-locked loops (PLL) and transmission lines, all of which cause noise. Accordingly, a large body of literature exists: The  $1/f$  noise and clock jitter in digital electronics systems were described by Zhang *et al.* [16]. An analysis of jitter in high-speed serial links was given by Hanumolu *et al.* [17], and theoretical insights on white noise, flicker noise, and modulation in oscillators and PLLs were presented by Mehrotra [18] and Herzel *et al.* [19]. Behavioral models and simulations were discussed in the papers of Hinz *et al.* [20], Manganaro *et al.* [21], and Lau and Perrott [22].

As the total system synchronization is a tree-like architecture, the effects of chaining synchronization channels should be considered. An analysis of jitter peaking using cascaded surface acoustic wave (SAW) filters for clock recovery is given by Fishman *et al.* [23], and the theory of jitter accumulation in synchronous networks is discussed by Yim and Hartmann [24].

Consider the system-wide distributed clock and its jitter by the means of a model. If  $t_n$  refers to the  $n$ -th zero node at the full wavelength of the reference signal, then the clock period is defined as  $T_n = t_{n+1} - t_n$ . Any deviation from this ideal reference signal ( $T_{\text{Ref}}$  is the mean period) is called jitter and noted as  $\Delta T_n = T_n - T_{\text{Ref}}$ . In [25], Herzel and Razavi presented definitions for three kinds of jitter, which are widely applied. The first one is absolute or long-term jitter:

$$\Delta T_{\text{abs}}(N) = \sum_{n=1}^N (\Delta T_n) , \quad (1)$$

which accumulates the total error signal. The second type is cycle jitter:

$$\Delta T_c = \lim_{N \rightarrow \infty} \sqrt{\frac{1}{N} \sum_{n=1}^N (\Delta T_n^2)} , \quad (2)$$

which represents the magnitude of fluctuations as the RMS value of  $\Delta T_n$ . The third type is cycle-to-cycle jitter:

$$\Delta T_{cc} = \lim_{N \rightarrow \infty} \sqrt{\frac{1}{N} \sum_{n=1}^N (T_{n+1} - T_n)^2} , \quad (3)$$

which defines the RMS difference between two consecutive periods.

Fakhfakh *et al.* [26] present a solution for the conversion of the jitter to phase noise in the case of independent white noise sources.  $\sigma_{\Delta T_{\text{abs}}}$  is the standard deviation from the “absolute jitter”,

$$\frac{\sigma \Delta T_{\text{abs}}}{\sqrt{\Delta t}} = \frac{\Delta T_c}{\sqrt{T_{\text{Ref}}}} = \frac{\Delta T_{cc}}{\sqrt{2} \sqrt{T_{\text{Ref}}}}, \quad (4)$$

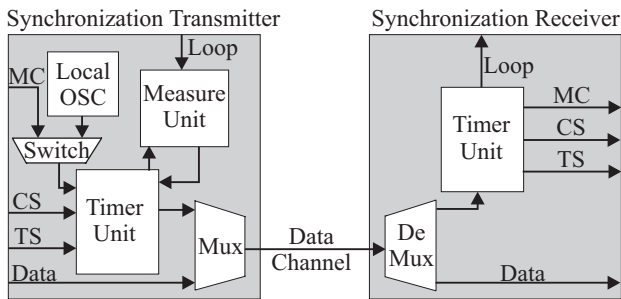
where  $\Delta t$  is the measurement time interval.

As a first-order approximation, the synchronization channel can be treated as an oscillator. This is based on the fact that the PLL configurations on a serialized synchronization channel only contribute in a minor way to the low-frequency system jitter (see Section 6).

## 4. Signaling Channel

The purpose of the signaling channel and its protocol are the distribution of information to initialize, optimize, and maintain reliable channel timing. A sketch of the signaling channel is given in Fig. 3, where only the path from master to slave is depicted. A real implementation scenario also consists of an additional backward channel.

The synchronization transmitter has a local oscillator (OSC). If the master clock (MC) is lost, the switch-over unit (Switch) activates the OSC, which obtains a master-ship. The timer unit contains a real-time register. If the transmitter is master, then timer synchronization (TS) and command synchronization (CS) signals are generated locally. All slave processes receive their TS and CS synchronously with the MC. Another task of the timer unit is to manage the time delay for lower-QoS processes, which is discussed in Section 5. Finally, the synchronization information is modulated onto the data channel (Mux). Much work on this topic has been published: For example the 8B/10B coding scheme introduced by Widmer and Franszek [27], or a variant thereof, modified by Chen *et al.* [28], is characterized by high transition rates for transferring the clock signal and includes some out-of-space symbols to carry the TS, CS, and possible user extensions.



**Fig. 3.** Architecture of a simplex signaling channel. The clock and signaling information are multiplexed onto a data channel by the means of coding and out-of-space symbols. The backward channel is symmetric and therefore not shown in the figure. See text for further details.

The synchronization receiver consists of a de-mux (DeMux) to extract the MC, TS, and CS from the data channel. These signals are fed to the transmitter for the backward channel and to all lower-QoS processes.

## 5. Signaling Protocol

The master synchronization transmitter uses MC or OSC as reference to synchronize the data channel. The slave synchronization receiver locks onto the data channel. The TS signal is a periodically distributed beacon from the higher-QoS processes (master) to all processes with lower QoS. It indicates the calibration state of the real-time register in the system. The maximum beacon frequency is determined by the maximum delay, which is given by the longest possible path in the entire distribution tree.

The TS frequency divided by the MC frequency yields the minimum real-time register length. The CS signal appears arbitrarily and is used for exchanging information between master and slave.

To achieve high performance, the channel delay under operation has to be analyzed and corrected. Therefore, the synchronizing transmitter starts the measure unit and sends a “measure-delay” CS to the synchronization receiver. The receiver feeds the CS via the loop signal to the measure unit of the backward channel, which sends the command back. Finally the returning “measure-delay” CS arrives at the synchronizing transmitter measure unit. The result is a number of MC cycles plus a phase delay, which is of very high accuracy. The synchronizing transmitter is now able to calculate the real synchronization channel delay by dividing the result by two. This is the synchronization channel delay compensation, relative to the system-wide master process, which can be applied to calibrate (pre-load) the real-time register on the slave side. This simple protocol allows maintaining very high accuracy without interrupting the data channel operation. The “measure-delay” process can be initiated independently of either the master or the slave side.

Other important QoS arguments are the minimization of the number of processes to the MC origin, and the indication of isolation from the MC origin, as shown in Fig. 2. Therefore, three cases have to be considered:

1. In Fig. 2, the synchronization distribution is in a static configuration with the dashed channel between P7 and P10 connected. P0 broadcasts all synchronization information down to the tree. The CS shows that P0 is the actual synchronizer (MC), and the distance, given as a number of hops, is incremented in each process and sent to all lower-QoS processes (slaves and redundancy channels).
2. If the channel between P7 and P10 breaks down, the sub-domain {P10...P14} continues to synchronize

on the independent base of P10. P10 keeps broadcasting all synchronization information to the sub-domain members, as new pseudo-master. The CS indicates by a tag that P10 is no longer synchronized by P0. The QoS is now interpreted as very low.

3. If a sub-domain is being reconfigured or runs independently as mentioned above, the redundant channels will be activated. In Fig. 2, as long as the channel between P7 and P10 exists, the dotted line between P9 and P12 indicates a redundant channel of low QoS (high number of hops). However, when the channel between P7 and P10 breaks, the channel between P9 and P12 now is of higher QoS than any other channel in the sub-domain. Therefore it becomes the new synchronizing channel to the P0 domain, and the entire sub-domain is reconfigured, with P12 connected to P9 as a master.

Contention in the process of reconfiguration or oscillation in a reconfiguration sequence can be excluded: The basic condition that reconfiguration only occurs if the QoS is higher than that of the existing synchronization channel, by definition, leads to stability in the system.

## 6. Experiment

For the experiment, the clock distribution has been applied to an optical shared-memory supercomputer interconnect system, see Fig. 1. The data channels run continuously at 40 Gbit/s, and the synchronization channels at 2.5 Gbit/s. A single switch node contains 64 data channels. The system-wide synchronization distribution can be represented as shown in Fig. 2.

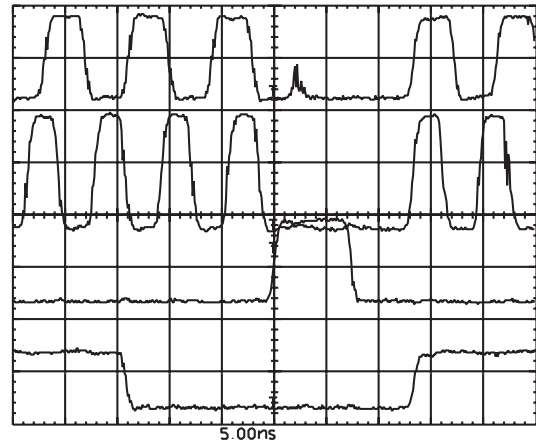
The experimental setup, which consists of the following components, is shown in Fig. 3:

On the synchronization master, which is the switch side, a 125-MHz MC is fed directly to the timer unit and the Mux (see Fig. 3). As the data channel is implemented as an optical transmission line, the Mux contains a PLL for clock multiplication up to 2.5 Gbit/s. This PLL is the main noise-inducing element on the transmission side, if we neglect thermal drift and channel skew of the hardware devices.

At the synchronization receiver side (slave side), the noise-inducing sources are the receiver photo diode and the run length of the 8B/10B-coded data channel together with the receiver PLL to re-synthesize the 125-MHz reference clock. The timer unit contains a second PLL, which is configured as a frequency synthesizer to generate the required system clocks, including the outgoing MC. The phase noise of the MC is improved by using a separate SAW-PLL on the backward channel path to the master and to all the connected slaves, including the redundant channels.

## 7. Results

Figure 4 shows the functionality of the timer unit at the synchronization receiver outputs. The waveforms represent the initial synchronization of the synthesizer PLL after power-up, all phases are synchronized simultaneously with the real-time register upon receipt of the MC and TS. Because the delay value is written to the real-time and phase-shift registers when the TS is received, all real-time registers in an entire domain are accurately synchronized.



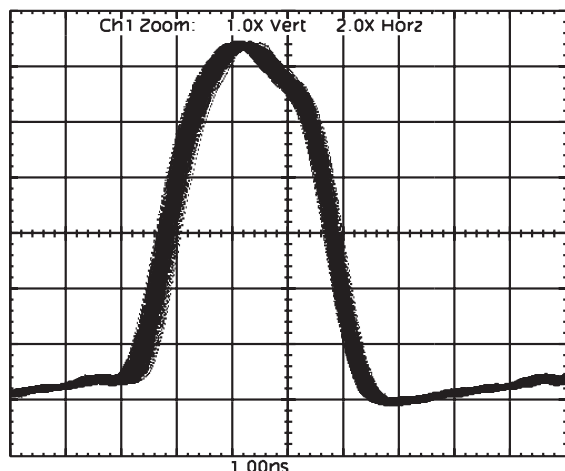
**Fig. 4.** Synthesizer PLL operations. From top to bottom, the topmost waveform depicts the output MC frequency. The next one represents an internally used 156.25 MHz frequency, which is phase-synchronized to the MC. This is initiated by the next waveform, which is the initial TS signal, i.e. prior to enabling the synchronization operation. The fourth waveform is the unit cell in which the phases of the first two waveforms match. The units per division for the X and Y axis are 5.00 ns and 1.500 V, respectively.

Figure 5 depicts an example jitter measurement, acquired with a Tektronix TDS 744, which is 3536 ns delayed from the trigger. Shown is a data accumulation over 72 h under ambient conditions. Equation (4) can be rewritten as

$$\Delta T_{cc} = \frac{\sigma_{\Delta T_{abs}} \sqrt{2} \sqrt{T_{Ref}}}{\sqrt{\Delta t}}$$

Substituting  $\sigma_{\Delta T_{abs}}$ ,  $T_{Ref}$ , and  $\Delta t$  with 300 ps, 8 ns, and 3536 ns, respectively, results in an average cycle-to-cycle jitter of only 20 ps. This number agrees with our expectations.

As Eq. (5) neglects any effects of the measurement setup, the value of 20 ps includes the trigger jitter of the oscilloscope. A systematic discussion of such measurement setups can be found in the paper of Zamek and Zamek [29].



**Fig. 5.** MC output signal of the synthesizer PLL, measured at the synchronization receiver side. The measurement shows the 442-th clock period after trigger position. The units per division for the X and Y axis are 1.00 ns and 500 mV, respectively.

## 8. Conclusion

A high-performance system-synchronization distribution architecture consisting of a highly accurate reference-clock distribution with real-time synchronization and a signaling channel with a novel flexible protocol has been defined and successfully implemented. The results obtained meet the requirements for accuracy, latency and flexibility, as required by the high-performance-computing systems research community.

The derived formula for  $\Delta T_{cc}$  can be used to determine the cycle-to-cycle jitter from a simple measurement. From this result, the RMS jitter can be calculated with Eq. (4).

The synchronization architecture contains a “measure-delay” feature together with a real-time register-calibration option. This is crucial to future research and engineering projects in precision real-time fields, where mixed configurations with static and mobile participants are highly likely. Theoretically, the proposed synchronization architecture is able to deliver the same accuracy to all participants, at no additional cost.

## Acknowledgments

This research is supported in part by the University of California under subcontract number B527064. The authors are grateful to the project sponsors and the technical teams at Corning Research, Corning, N.Y., and the IBM Zurich Research Laboratory.

## References

1. A. Hu and S.D. Servetto, “Asymptotically Optimal Time Synchronization in Dense Sensor Networks”, *Proc. Second ACM Workshop on Wireless Sensor Networks and Applications*, pp. 1-10 (2003).
2. J.-P. Sheu, C.-M. Chao, and C.W. Sun, “A Clock Synchronization Algorithm for Multi-Hop Wireless Ad Hoc Networks”, *IEEE Proc. Distributed Computing Systems*, pp. 574-581 (2004).
3. C. Kelly and R. Manohar, “An Event-Synchronization Protocol for Parallel Simulation of Large-Scale Wireless Networks”, *IEEE Proc. Distributed Simulation and Real-Time Applications*, pp. 110-119 (2003).
4. C. Chen, “A QoS-based Routing Algorithm in Multimedia Satellite Networks”, *IEEE Vehicular Technology Conference*, pp. 2703-2707 (2003).
5. D.L. Mills, “Precision Synchronization of Computer Network Clocks”, *ACM Comp. Commun. Rev.*, vol. 24, no. 2, pp. 28-43 (1994).
6. S. Butner and S. Vahey, “Nanosecond-scale Event Synchronization over Local-area Networks”, *IEEE Proc. Local Computer Networks*, pp. 261-269 (2002).
7. A. Pásztor and D. Veitch, “PC Based Precision Timing without GPS”, *ACM SIGMETRICS Performance Evaluation Review*, vol. 30, no. 1, pp. 1-10 (2002).
8. K. Schossmaier and B. Weiss, “An Algorithm for Fault-tolerant Clock State and Rate Synchronization”, *IEEE Proc. 18<sup>th</sup> Symp. on Reliable Distributed Systems*, pp. 36-47 (1999).
9. C. Liebig, M. Cilia, and A. Buchmann, “Event Composition in Time-dependent Distributed Systems”, *IEEE Proc. Cooperative Information Systems*, pp. 70-78 (1999).
10. B. Abali, C.B. Stunkel, and C. Benveniste, “Clock Synchronization on a Multicomputer”, *J. Parallel Distrib. Comput.*, vol. 40, pp. 118-130 (1997).
11. J. Feehrer, J. Sauer, and L. Ramfolt, “Design and Implementation of a Prototype Optical Deflection Network”, *ACM Proc. SIGCOMM*, pp. 191-200 (1995).
12. P. Gambini, M. Renaud, C. Guillemot, F. Callegati, I. Andonovic, B. Bostica, D. Chiaroni, G. Corazza, S.L. Danielsen, P. Gravey, P.B. Hansen, M. Henry, C. Janz, A. Kloch, R. Krähenbühl, C. Raffaelli, M. Schilling, A. Talneau, and L. Zucchelli, “Transparent Optical Packet Switching: Network Architecture and Demonstrators in the KEOPS Project”, *IEEE J. Sel. Areas Commun.*, vol. 16, no. 7, pp. 1245-1259 (1998).
13. H. Yashiro, Y. Takahashi, T. Fujiwara, “A High Assurance Timing Synchronization Technology for Space On-board Distributed Computer Systems”, *IEEE Proc. High Assurance Systems Engineering*, pp. 87-88 (2002).
14. D. Mitra, “Network Synchronization: Analysis of a Hybrid of Master-Slave and Mutual Synchronization”, *IEEE Trans. Commun.*, vol. 8, pp. 1245-1259 (1980).
15. R. Hemenway, R. Grzybowski, C. Minkenberg, and R. Luijten, “Optical-packet-switched Interconnect for Super-computer Applications”, *J. Opt. Netw.* 12, 900-913 (2004).
16. C.W. Zhang, X.Y. Wang, and L. Forbes, “Simulation Technique for Noise and Timing Jitter in Electronic Oscillators”, *IEE Proc. Circuits Devices and Systems*, vol. 151, no. 2, pp. 184-189 (2004).

17. P.K. Hanumolu, B. Casper, R. Mooney, G.-Y. Wei, and U.-K. Moon, "Analysis of PLL Clock Jitter in High-Speed Serial Links", *IEEE Trans. Circuits Syst. – II: Analog and Digital Signal Proc.*, vol. 50, pp. 879-886 (2003).
18. A. Mehrotra, "Noise Analysis of Phase-locked Loops", *IEEE Trans. Circuits Syst. – I: Fundamental Theory and Applications*, vol. 49, no. 9, pp. 1309-1316 (2002).
19. F. Herzel, W. Winkler, and J. Borngreber, "Jitter and Phase Noise in Oscillators and Phase-locked Loops", *Proc. SPIE*, vol. 5473, pp. 16-26 (2004).
20. M. Hinz, I. Könenkamp, and E.H. Horneber, "Modeling and Simulation of Phase-locked Loops in the Time and Frequency Domain", *Proc. SPIE*, vol. 4228, pp. 330-341 (2000).
21. G. Manganaro, S.U. Kwak, S.H. Cho, and A. Pulincherry, "A Behavioral Modeling Approach to the Design of a Low Jitter Clock Source", *IEEE Trans. Circuits Syst. II: Analog and Digital Signal Proc.*, vol. 50, no. 11, pp. 804-814 (2003).
22. C.Y. Lau and M.H. Perrott, "Fractal-N Frequency Synthesizer Design at the Transfer Function Level Using a Direct Closed Loop Realization Algorithm", *ACM Proc. Autom. Design Conf.*, presentation 31.2, pp. 526-531 (2003).
23. D.A. Fishman, R.L. Rosenberg, and C. Chamzas, "Analysis of Jitter Peaking Effects in Digital Long-haul Transmission Systems Using SAW-filter Retiming", *IEEE Trans. Commun.* vol. 33, no. 7, pp.654-664 (1985).
24. C.H. Yim and H.L. Hartmann, "Jitter Accumulation in Time Synchronous Communication Networks", *ntzArchiv*, vol. 4, no. 12, pp. 371-375 (1982).
25. F. Herzel and B. Razavi, "A Study of Oscillator Jitter due to Supply and Substrate Noise", *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 1, pp 56-62 (1999).
26. A. Fakhfakh, N. Milet-Lewis, Y. Deval, and H. Lévi, "Study and Behavioural Simulation of Phase Noise and Jitter in Oscillators", *IEEE Proc. Symp. on Circuits and Syst.*, vol. 5, pp. 323-326 (2001).
27. A.X. Widmer and P.A. Franaszek, "A DC-balanced, Partitioned-Block, 8B/10B Transition Code", *IBM J. Res. Develop.*, vol. 27, no. 5, pp 440-451 (1983).
28. H.-Y. Chen, C.-H. Lin, and S.-J. Jou, "Low-jitter Transmission Code for 4-PAM Signaling in Serial Links", *IEEE Proc. Asia-Pacific Conf. on Advanced System Integration Circuits*, Presentation 15-6, pp. 334-337 (2004).
29. I. Zamek and S. Zamek, "Crystal Oscillator Jitter Measurements and its Estimation of Phase Noise", *Proc. IEEE Int. Frequency Control Symp.*, pp. 547-555 (2003).