DesignCon 2008

The Future of Multi-Clock Systems

Ransom Stephens, Ph.D., Ransom's Notes
Ransom@RansomsNotes.com

Roman Boroditsky, NEL Frequency Controls, Inc.

Jorge Gomez, NEL Frequency Controls, Inc.

Abstract

The clock requirements of high performance systems vary from the chip to the circuit to the system level, but in every case the ideal is a fully synchronous system. Problems with traditional synchronized architecture – reliability and redundancy, scalability and cost, noise amplification and propagation – have led designers to develop clever asynchronous designs at both the system and circuit level. We introduce a new approach to synchronized clock architecture that has the performance of full synchronization with no central clock, no fan-out, and the surprising effect that system jitter is reduced as more clock groups are added.

Authors Biographies

Ransom Stephens' company, Ransom's Notes, produces and presents content at every level of technical sophistication to help engineers advance to technology's cutting edge. He is the author of more than 200 articles in the electronics industry, science journals, and magazines. Dr. Stephens has introduced new measurement techniques for electrical and optical systems, invented methods for extracting signals from noise, led an engineering commando team, and served on high data-rate standards committees. Contact him at www.RansomsNotes.com.

Roman Boroditsky, Technology Director of NEL Frequency Controls, Inc., has nearly thirty years of experience in designing and developing frequency control products. Boroditsky earned his MSEE degree from the Minsk Radio-Engineering Institute in Minsk, USSR. He holds two patents and has published numerous articles related to oscillator design and applications.

Jorge Gomez is a Senior Design Engineer at NEL Frequency Controls. He has over 25 years of experience in design and development of electronic components, equipment, and systems. Last ten years he has worked in the field of frequency control products and has published several papers in the field. He holds BS in Automatic Controls Engineering from Las Villas Central University in Cuba.

The Future of Multi-Clock Systems

Clock signals are fundamentally the most important control signals in digital systems. Regardless of the modulation scheme, the timing of logic transitions is dictated by the clock. The performance of a system at any level, whether on a chip, on a board, or across boards, is predicated by the coordination of clock signals among system components.

It’s as simple as this: if a receiver samples a signal during a transition, the Bit Error Ratio (BER) will be high, if it samples in the middle of the bit, the BER will be low. The time position of sampling is affected by jitter [1] and noise on the clock signal, frequency and phase differences between the clock used in transmission and the clock used in reception, and skew between the data signal and clock signal.

There are three types of clocking schemes – synchronous, asynchronous, and plesiochronous. Synchronous systems provide clock signals that are frequency locked at every transmission/reception exchange and require a design with zero skew to set the phase relationship between signal and clock. Synchronous systems permit seamless data exchange at the highest possible rates and have the most elegant architecture.

Asynchronous systems have more autonomous components; they are not frequency or phase locked and, between components, delay and skew are not an issue. There are several design tradeoffs that make each approach attractive in different conditions but it is safe to assume that, in an ideal world with no noise, jitter, or skew, we’d all prefer synchronous systems.

Plesiochronous systems share traits with both synchronous and asynchronous systems but have few benefits of either.

Difficulties in coordinating clock signals in synchronous systems differ at the chip, circuit, and system levels. On a chip, a single clock signal is easily distributed to drive every element and data-clock skew is easily controlled – synchronous clocking is a trivial choice on-chip. The situation is more complicated at the circuit (multi-chip) and system (multi-board) levels. Synchronicity requires that either (1) a central clock be distributed across the circuit, (2) independent clocks local to circuit components be frequency locked, or (3) a low frequency reference clock be distributed across the circuit and multiplied up to the data rate at each component. Each of these legacy solutions introduces another layer of problems in component cost, design complexity, increased jitter and noise, and reduced reliability; plus the difficulties increase with component count and separation distance.

There is now a fourth option. New Synchronized Clock Oscillator technology offers a completely different approach. There's no master clock, no need for fanout, no need for Phase-Locked Loop (PLL) frequency locking or multiplication. Plus, in an extraordinary occurrence of good fortune, the system becomes less noisy and more reliable as the component count increases.

In this paper we review the roles that clock signals play in digital systems and the different techniques for distributing clock signals. Through the discussion we emphasize the reasoning behind different techniques and show how Synchronized Clock Oscillator technology improves performance, decreases cost, and simplifies designs.

The Timing Budget

The timing budget [2] sets the ultimate limit on the maximum rate at which a system can operate. It is composed of the sum of the durations of everything that happens between successive clock cycles. In Figure 1 the data is sampled at a time specified by the clock signal, t_{sample}. The setup time, t_{setup}, is the minimum length of time at which the data must be stable prior to being sampled by the receiver. The hold time, t_{hold}, is the minimum time required after sampling. The sum of the setup and hold times is the maximum length of time required by the receiver to determine the logic state of a bit. In an ideal system, setup and hold compose the entire timing budget. The rest of the timing budget accounts for the reality of the system: jitter on the clock and data and the relative skew between the clock and data.

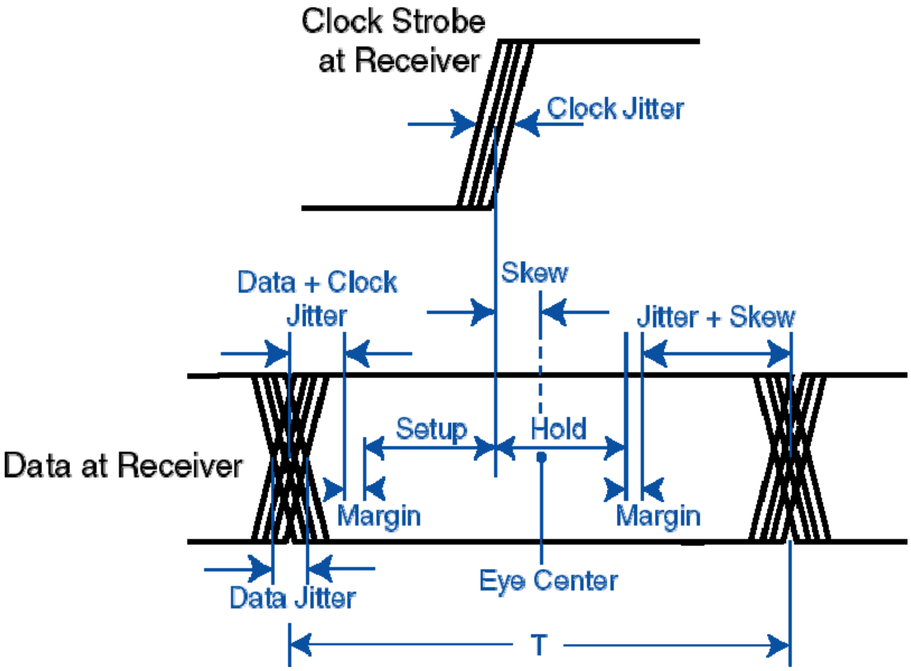


Figure 1: The timing budget is the sum of the duration of every necessary process plus the effect of jitter.

Legacy Synchronous Architecture

In an ideal synchronous circuit [3], changes in the logic levels of every component are simultaneously defined by transition of a common clock signal. The advantage of a synchronous design is that the timing of all events can be safely assumed; there is no need for active components to monitor and coordinate the timing of different events. In practice, of course, logic transitions have finite rise/fall times, signal propagation has delay, and registers have nonzero latch times which combine to absorb the timing budget and reduce the maximum possible system speed.

At the chip level, Figure 2, synchronous clocking is the obvious choice to provide the most efficient data flow. Adjacent registers, R_i and R_j are separated by a logic block L_{ij}. R_i is clocked by C_i and R_j by C_j. R_i switches when C_i transitions and the signal propagates through L_{ij} to R_j. The combination of the clock quality and the delay of each component set a limit on the maximum clock speed. The timing budget, in addition to the requirements shown in Figure 1, must also include the total delay of the signal – the time required for data to leave the initial register once the clock signal arrives, plus the propagation time of the signal through the logic and interconnect.

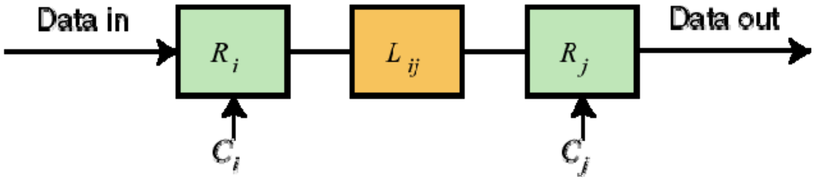


Figure 2: Chip-level synchronous timing.

At the circuit and system levels, things are different. All events may not be simultaneous, but the timing of every event is coordinated in one place at the system level. For example, in a circuit with mixed signals, multiple chips at a common frequency but different logic requirement, e.g., LVDS and CMOS, separate clock signals must be frequency locked. Whether fanned out or locked with a PLL, the clock quality degrades.

In inter-board systems, e.g., server blade applications, a module operating in one clock domain inevitably needs to send data to another module operating in a second clock domain. Legacy synchronous designs present problems that have motivated asynchronous designs but, as we'll see, most of these problems are solved by applying Synchronized Crystal Oscillator technology.

Clock Distribution in Legacy Synchronous Systems

Legacy synchronous systems distribute a common clock signal by fanning out a master clock to each component of the system, Figure 3. A single input clock signal is redriven by several output buffers. The buffers have propagation delay, though fanouts are available that incorporate PLLs to eliminate skew between the outputs. The tradeoff is that the PLLs introduce jitter. When more than one fanout part is required, it is important to include adjustable delay in the circuit to eliminate skew between fanout modules.

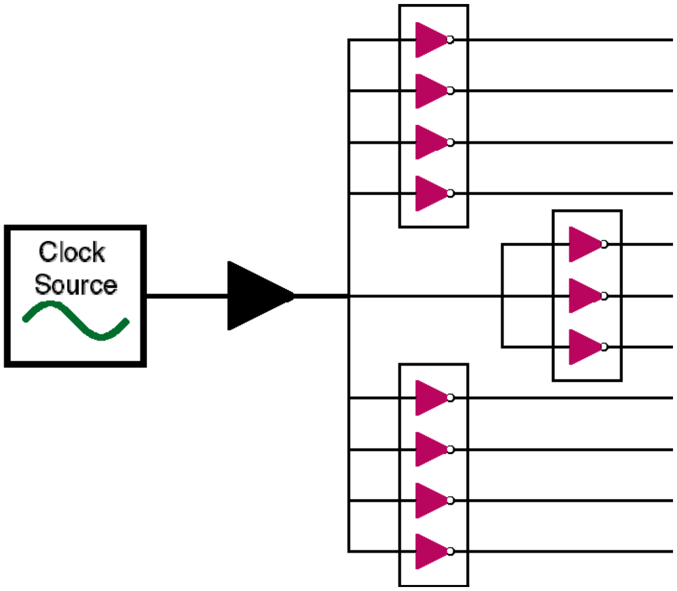


Figure 3: Master clock fanout.

In many applications a low frequency clock is fanned out across a system and the clock is multiplied to the data rate at each component. The jitter of the PLL multiplier's Voltage Controlled Oscillator (VCO) is added to the clock signal and, as a result of multiplication, the jitter of the clock itself increases as the square of the multiplication factor [4].

Another legacy technique for clock distribution is to simply daisy chain a single clock signal across the system, Figure 4. At each component, a well-tuned delay must be provided to synchronize the system. In principle, we could hope that each component bleeds off an identical copy of the clock signal. In practice, it is difficult to match impedances so perfectly that the clock signal isn't reflected at each tap. Multiple reflections interfere with the signal and introduce noise and jitter.

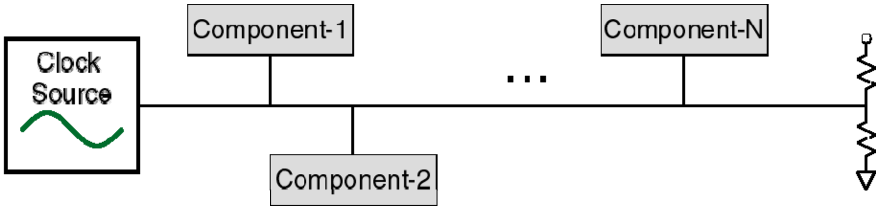


Figure 4: Daisy chained clock distribution.

Skew and Jitter

In the timing budget, Figure 1, the setup and hold times of the receiver are determined by the choice of component, but skew and jitter are imperfections that can be reduced by careful design.

Skew is the fixed timing difference between two signals. The primary cause of skew is the difference in trace length, but anything that affects signal propagation speed can contribute – trace width and impedance, variations in dielectric constants, and temperature. The skew of two traces of identical length, but different media and configuration can be as high as 20%; temperature differences rarely cause skew of more than a few percent.

If the receiver samples the data on the rising edge of the clock signal, as in Figure 1, then as long as the clock provides the receiver a rising edge (any rising edge) at the right time, there is no relevant skew. If we're considering jitter though, assuring that the same clock edge that was used to generate a data transition is also used to strobe that transition at the receiver can dramatically decrease the effective jitter of the system. If the data signal and the clock signal both have the same jitter, then they track each other.

Insuring that the clock used in a receiver has the same jitter as the data is one of the driving motivations for adopting asynchronous architectures.

Asynchronous Architecture

Where the components of synchronous systems must act in concert, the components of asynchronous systems can be autonomous,

Figure 5. At the transmitter, the clock signal determines the timing of logic transitions and, at the receiver, rather than sample incoming data with the trivial assumption of synchronous timing, a separate clock must be at least temporarily phase and frequency locked so that bits can be sampled at their centers.

Asynchronous architectures have several advantages over legacy synchronous designs at the inter-board level, few advantages at the circuit level, and, except in the most rare cases, no advantages at the chip level.

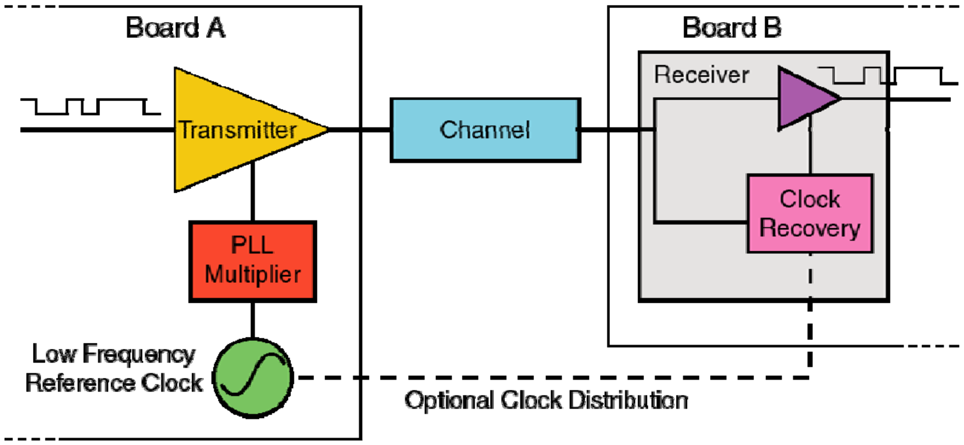


Figure 5: A simple diagram of an asynchronous system.

Asynchronous architectures solve several of the problems we've seen in legacy synchronous systems: fanout and the associated increase in jitter is not an issue, skew is no longer a problem, and having multiple clocks reduces the possibility of catastrophic central clock failure.

The autonomous nature of asynchronous architecture provides scalability and redundancy. The less coordination between boards, the easier it is to add or subtract them as needed.

Of course the benefits of asynchronous design come at a cost.

Clock Recovery

In the systems that concern us, different components must communicate and, to do so, an element of synchronicity is required.

The first sacrifice made in moving from a synchronous to an asynchronous architecture is the seamlessly transparent timing of each event in the system. This sacrifice amounts to surrendering the ultra-high performance that can only be attained in a system where every event occurs in, not just figurative but, literal harmony.

There are several ways for asynchronous systems to achieve the level of synchronization necessary for communication [5]. In

Figure 5 the transmission of a data signal is controlled by one clock and its reception is controlled by another.

Figure 6 shows a clock recovery system. The VCO of a PLL is locked to the transitions of the incoming data and used to strobe the receiver as in Figure 1. These are also called embedded clock systems because the clock used to reconstruct incoming data is embedded in the data itself. Other than within the clock recovery circuit where the positioning of the strobe must be well centered in the setup and hold comfort zone of the receiver, problems with skew are eliminated.

An advantage of embedded clocking is that the wider the bandwidth of the clock recovery circuit, the more that jitter on the clock tracks jitter on the data.

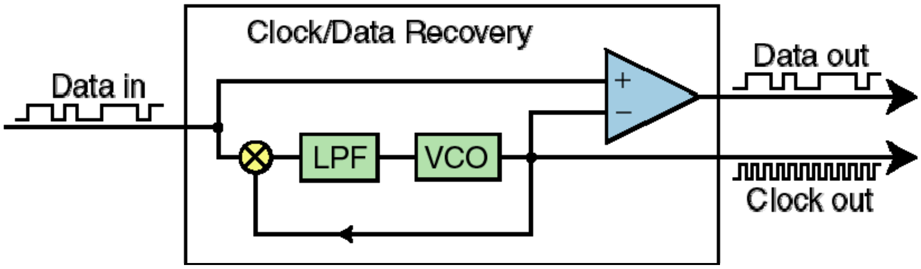


Figure 6: Sampling bits at the receiver.

In some designs, a low frequency clock signal is distributed to the receiver to aid the clock recovery circuit. Distributing the clock not only eliminates some of the advantages of the asynchronous design but indicates how difficult clock recovery can be. PLL based clock recovery circuits are expensive components. The digital alternative, a Phase Interpolator (PI), is less expensive but more difficult to characterize. PIs are also more likely to suffer nonlinear effects and usually require a distributed clock.

There is another cost to clock recovery. The data signal must have transitions with sufficient frequency to prevent the clock recovery circuit from losing lock. In a long string of consecutive identical bits there is effectively no clock signal embedded in the data. Similarly, if there is not an equal number of high and low logic voltages, the recovered clock will drift, moving the strobe position away from the setup and hold comfort zone.

The requirements of transition density and DC balance can be resolved by encoding the data. 8B/10B coding, for example, has a minimum of two transitions for every eight bits and is DC balanced. It too comes with a cost. First, the transmitter must have a coding shift register, and second, since eight data bits are encoded in a total of ten transmitted bits, the effective data rate is reduced by 25% - which is the same cost as a 25% reduction in timing budget.

Obviously, if the disadvantages of the synchronous architectures could be overcome, there would be little advantage to using an asynchronous design.

Plesiochronous Clocking

Plesiochronous (pronounced “please-ee-ah-krun-us”) systems have components with separate clocks whose frequencies are nominally the same but are not frequency locked. Transmitted bits arrive within a specified window of the time slot in which they were sent, but not necessarily within the clock interval required of a synchronous system. Components of plesiochronous systems rely on buffering to cope with the lack of synchronization. Data are buffered prior to retransmission timed by the local clock. The timing budget neither benefits from the autonomy of components that asynchronous systems have nor the component coherence that synchronous systems have.

The New Synchronous Architecture

The new concept of Synchronized Crystal Oscillators provides a new technique for clock synchronization. It is similar in layout to the daisy-chain architecture of Figure 4, but with several major improvements.

Figure 7 shows the new architecture. A dual bus is used with alternating input and output synchronization signals maintaining complete synchronization of every clock in the system. Each Synchronized Crystal Oscillator (SXO) has both a synchronization input and an output. The synchronization input sets the crystal frequency of the SXO. For example, in Figure 7, consider the first Synchronized Crystal Oscillator, SXO-1. Its frequency is set by the signal on bus-1 and provides a synchronization output to bus-2. The next Synchronized Crystal Oscillator, SXO-2, has its synchronization input/output with the opposite orientation: the frequency of SXO-2 is set by the signal on bus-2 and transmits a synchronization output to bus-1. The chain continues. SXO-3 is synchronized by the signal on bus-1 and synchronizes SXOs on bus-2, and so on. It's as though all odd numbered SXOs are synchronized by all even numbered SXOs and vice versa. Instead of a single crystal oscillator operating within its own closed loop, every Synchronized Crystal Oscillatorin the system is synchronized within a common feedback loop. The result is a system with an arbitrary number of clocks in perfect synchronization.

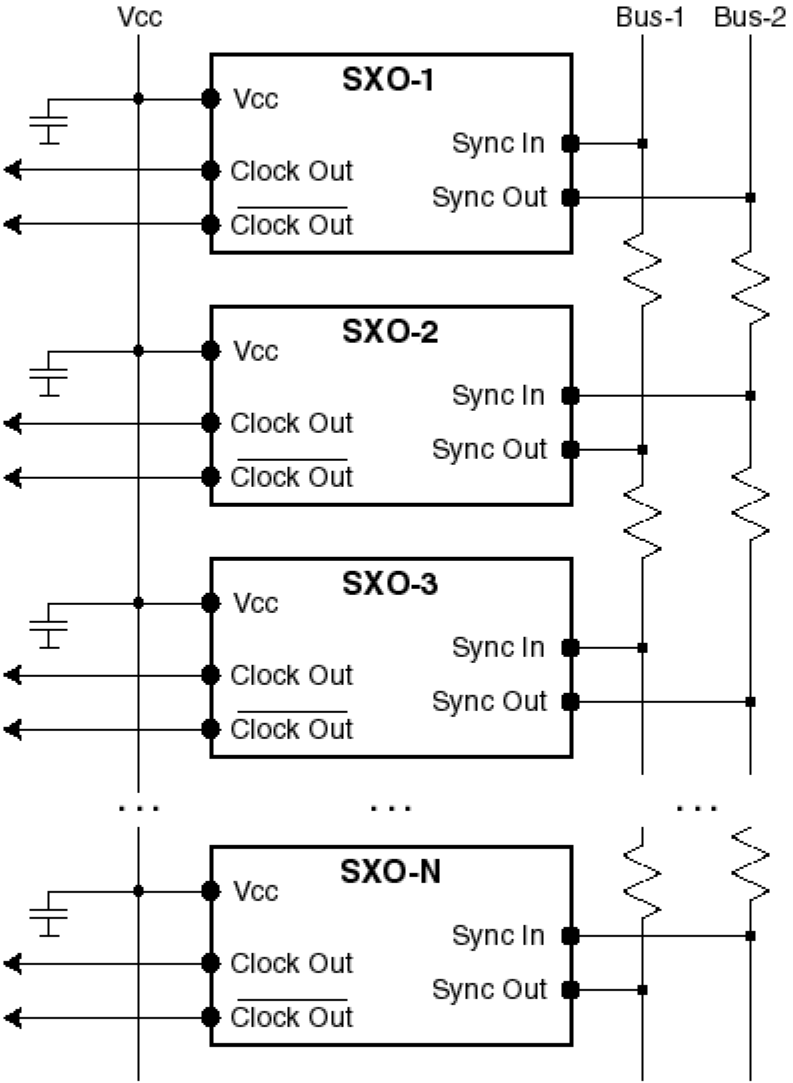
Notice that there is no central clock, no fanout or buffers, no PLLs, and none of the problems associated with these, now, extra parts. Plus, since there is no master, system synchronization is independent of the power application sequence.

The new Synchronized Crystal Oscillator architecture has most of the benefits of asynchronous architecture and all of the benefits of synchronous architecture:

- **Synchronous Timing:** There is no need for active components to monitor and coordinate the timing of different events. The timing of all events in the system is governed by frequency locked clocks.
- **Cost Savings and Redundancy:** The architecture is intrinsically redundant – there is no need to build in back-up clocking. If a clock fails on a given board, that board continues to receive the clock signal on the SXO bus from the combination of every Synchronized Crystal Oscillator in the system.

  The application of multiple clocks all but eliminates the possibility of catastrophic central clock failure.
- **Scalability:** Synchronized systems (i.e., boards) can be added without addition of fanouts or buffers permitting unlimited scalability.
- The coordination of timing between separate boards is rendered trivial; the dual bus architecture makes adding or removing boards a plug-and-play task.
- Synchronization of the clocks is independent of the order in which they are powered-up – sub-systems can be added or removed with impunity.

The only remnant of legacy synchronous architecture that remains in the new Synchronized Crystal Oscillator architecture is skew. While the frequency of every clock is locked to the signals on the dual bus architecture, the phase is not locked. The tradeoff between synchronous and asynchronous architectures is careful design of skew and delay versus painstaking design of clock recovery. On the one hand, skew and delay are easy to simulate and tuning them needn't involve active components. Neither is true of clock recovery circuits.



The above figure represents NEL Frequency Controls' Dual Alternating BUS Architecture for Clock Synchronization purposes. Use of this architecture for clock synchronization purposes requires a license from NEL, as NEL has filed a patent application covering this design.

Figure 7: Synchronous architecture new Synchronized Crystal Oscillator technology (patent application has been filed).

Breakthrough Jitter Performance

The sum of a Synchronized Crystal Oscillator system is truly greater than its parts. The multiple feedback between SXOs upon which the technology is based not only frequency locks every clock, but it reduces the jitter on every clock. The more Synchronized Clock Oscillator modules that a system employs, the less jitter every component of the system experiences.

Figure 8 shows the phase noise spectrum [6] of one, two, and twelve SXOs. Phase noise is equal to half the Single Side Band (SSB) spectrum, hence the horizontal axis of Figure

8 is the offset frequency of the phase noise relative to the clock frequency, or equivalently, the frequency of the jitter.

Table 1 shows phase noise values at several specific offset frequencies. Notice that in every case, introduction of more oscillators reduces the phase noise of the system.

The rms jitter is given by integrating the phase noise spectrum over the offset frequency:

From 100 Hz to 20 MHz:

1 SXO → rms Jitter = 115 fs

2 or more SXOs → rms Jitter = 91 fs

In legacy synchronous systems the jitter on each clock increases with the number of signals fanned out from the master, Figure 3. In some cases, for example, if PLL multipliers are used, the jitter on signals high in the tree can increase dramatically. Signal degradation was one of the driving motivations for adoption of asynchronous architecture. Obviously new synchronous systems no longer have this disadvantage.

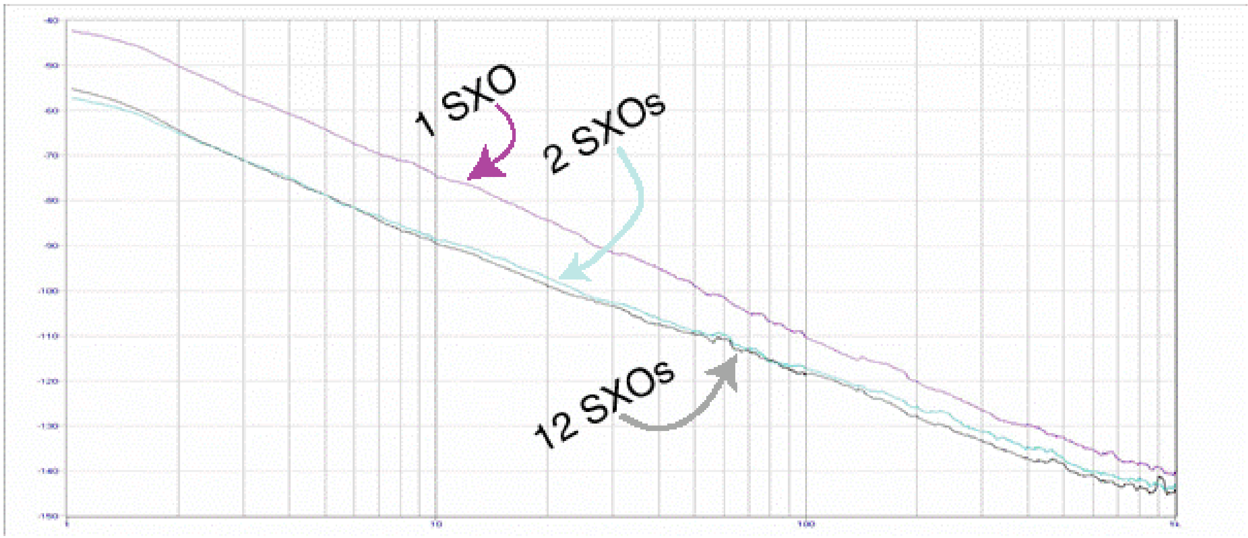


Figure 8: Phase noise spectrum with one, two, and twelve oscillators.

Offset frequency:	10 Hz	100 Hz	1 kHz	10 kHz	100 kHz	1 MHz
One SXO	-75 dB	-110	-140	-160	-165	-165
Two SXOs	-90	-117	-143	-160	-165	-165
12 SXOs	-90	-118	-145	-160	-165	-165

Table 1: Phase noise measurements in dB at several different frequency offsets.

Conclusion

The advantages of legacy synchronous architecture diminished with increasing data rates to the point where it made sense to adopt asynchronous architecture, despite its cumbersome handling of event timing and the need for robust timing interfaces. The introduction of Synchronized Crystal Oscillator technology heralds a return to the simplicity and high performance of full synchronization at today's clock speeds.

The advantages and features of NEL's SXO modules are:

- High reliability with multiple synchronous clocks.
- No master clock, no PLL required for the system
- Eliminates additive jitter degradation associated with clock distribution
- Linear scalability
- Complete system-wide clock redundancy
- "Hot" – swappable
- Synchronization is independent of power application sequence
- Jitter is reduced at every node
- While synchronized, all units exhibit identical phase noise characteristics

[1] Ransom Stephens, "The Rules of Jitter Analysis," AnalogZone, http://www.analogzone.com/nett0927.pdf, 2004; Ransom Stephens, "Analyzing Jitter at High Data Rates," IEEE Communications Magazine, vol. 42, no. 2, February 2004, pp. 6-10; NIST Technical Note 1337, "Characterization of Clocks and Oscillators," edited by D.B. Sullivan, D.W. Allan, D.A. Howe, F.L. Walls, 1990.

[2] For a general reference, see H.W. Johnson and M. Graham, *High Speed Digital Design*, Prentice Hall, 1993 and Howard Johnson, *High speed signal propagation: advanced black magic*, Prentice Hall, 2003.

[3] Eby G. Friedman, "Clock Distribution Networks in Synchronous Digital Integrated Circuits," *Proceedings of the IEEE*, Vol. 89, No. 5, May 2001.

[4] W. P. Robins, *Phase Noise in Signal Sources Theory and Applications,* (Peter Peregrinus Ltd., 1982).

[5] Ransom Stephens, *Jitter 360, Part 5: Clock Recovery in Serial Data Systems*, available at www.jitter360.com.

[6] Ransom Stephens, “Using Clock Jitter Analysis to Reduce BER in Serial Data Applications,” Agilent Technologies Application Note, Literature Number 5989-5718EN, available at www.agilent.com.