Using the LMK03000C to Clean Recovered Clocks

National Semiconductor
Application Note 1734
Dean Banerjee
May 2, 2008

Introduction

In wired communications, there is a need to recover a clock from the data. The advantage of having the clock encoded in the data is that this eliminates the need for a wire to transmit the clock and also helps deal with skew issues.

SERDES (Serializer-Deserializer) parts such as the SCAN25100 can take a set of parallel data and convert it into a series set of data that can be sent on a single wire. Another SERDES part at the receiving end can be used to generate the parallel data from the received series data. This is a very effective technique for sending data over long distances, but the recovered clock will have added phase noise (jitter) relative to the original clock used to generate this serial data.

When data is transmitted over long cables, the signal will become weaker and the signal to noise ratio will be less. If the length of cable is too long, it will not be possible to recover the clock. For this reason, SERDES part can be used as a repeater, where they receive the serial data, recover the clock, and then re-clock the data using the recovered clock or a jitter cleaned clock. In this way, data can be transmitted over much longer distances.

Clock conditioners, such as the LMK03000C can be used to take this dirty (higher phase noise/jitter) clock and generate a clean clock (lower phase noise/jitter) that is the exact same frequency. This application note investigates the impact of cleaning the clock at every stage, and also at the very end.

This application note goes through a practical example using the SCAN25100 SERDES and LMK03001C clock conditioner in order to illustrate how this can be done.

Recovered Clock of the SCAN25100

To understand what a recovered clock looks like, the SCAN25100 was used to generate a recovered clock and it was measured on the E5052 Phase Noise Analyzer. A board was created that used the SCAN25100 in conjunction with the LMK03001C Clock conditioner. In order to measure just the recovered clock with no cleaning, the LMK03001C was powered down.

To start, a 30.72 MHz signal was generated with the HP83712B signal generator and was used as the clean reference clock. From this the SCAN25100 was set in a BIST (Built In Self Test) mode so that it would generate pseudo random data that could be sent. For a practical application, the data is generated in a parallel format, but doing so requires more equipment and complexity which is beyond the scope of this application note.

Once this pseudo random data is created internally in the SCAN25100, the part creates a serial set of data that is sent out the Dout+/Dout- pins of the evaluation board. Then this data is transmitted over coaxial cables to the next board. In this way, the recovered clock of the SCAN25100 can be studied.

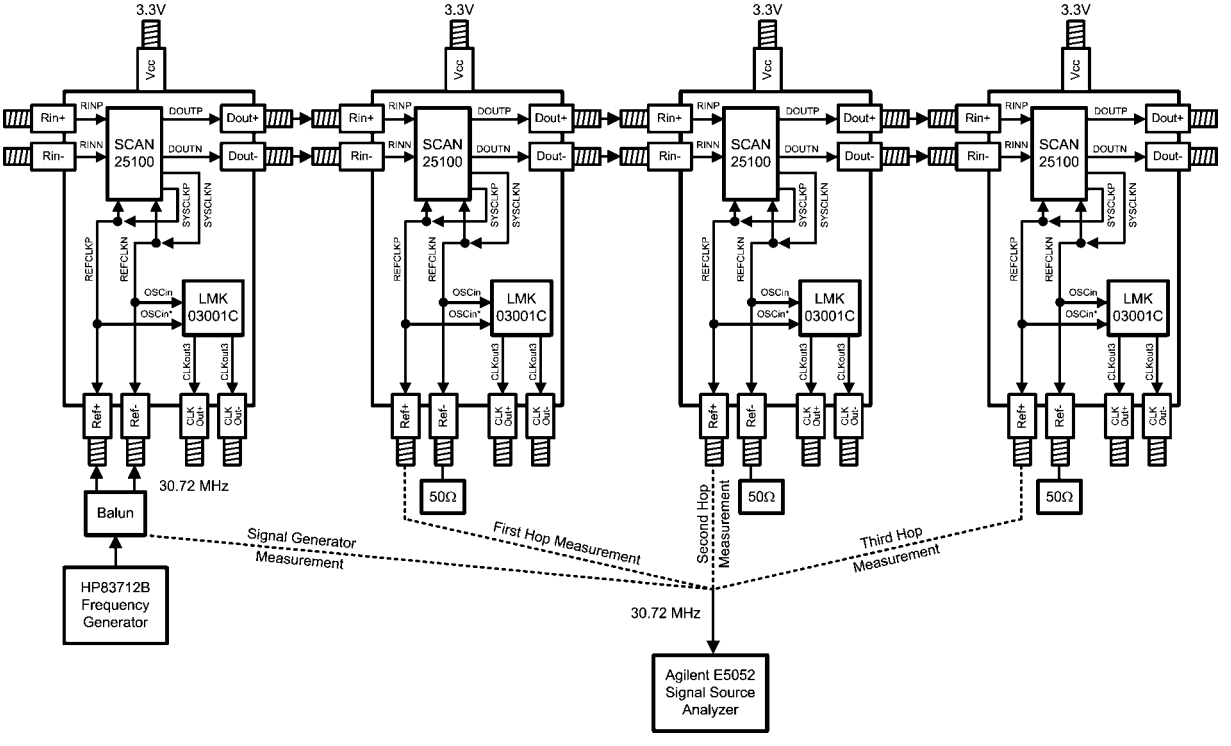

FIGURE 1. Setup for Measuring the SCAN25100 Recovered Clock

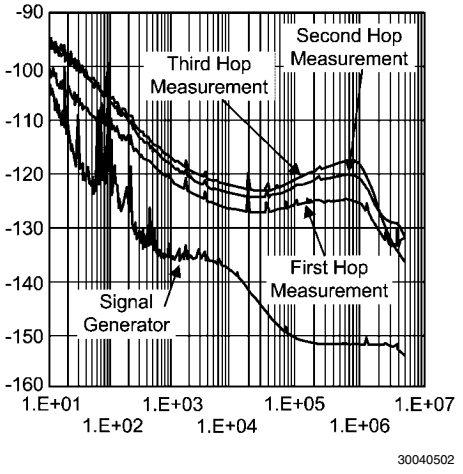

FIGURE 2. SCAN25100 Recovered Clock

Figure 2 shows the raw recovered clock of the SCAN25100. Comparing this to the signal generator source, it is clear that the recovered clock will have more phase noise than the original clock.

By observing the difference between the First Hop Measurement and the Second Hop Measurement one can see the general pattern that doubling the number of hops adds about 3 dB to the phase noise.

JITTER CALCULATION

From this phase noise profile, the jitter can be calculated, once limits for integration are established.

Figure 2 shows the result of one hop, two hops, and three hops. Based on this phase noise, the jitter can be calculated in accordance with the formula shown below. In the calculations for *Table 1* below we are selecting a fixed value of 5 MHz for the **UpperLimit** of the equation below and calculating Jitter for selected **LowerLimit's**:

$$\text{Jitter} = \frac{\sqrt{\int_{\text{LowerLimit}}^{\text{UpperLimit}} 10^{\text{PhaseNoise (Offset)}/10} \bullet \text{dOffset}}}{2 \bullet \pi \bullet \text{Frequency}}$$

TABLE 1. Jitter as a Function of LowerLimit

		Jitter (ps) vs. LowerLimit and Point of Measurement			
		Signal Source	**One Hop**	**Two Hops**	**Three Hops**
LowerLimit	10 Hz	0.467	5.921	9.245	10.887
	100 Hz	0.437	5.913	9.227	10.869
	1 kHz	0.432	5.905	9.216	10.858
	10 kHz	0.420	5.892	9.201	10.840
	100 kHz	0.406	5.793	9.073	10.693
	1 MHz	0.362	4.272	6.261	6.262

One can also observe from the table that the choice of **LowerLimit** has only a small impact of the jitter. For the first hop, the jitter is about 6 ps, provided the choice of **LowerLimit** is 100 kHz or less, which would be true for most applications. Jitter adds in an RMS sense, so this means that the jitter for two sources adds as shown below.

$$\text{Jitter}_{\text{Total}} = \sqrt{(\text{Jitter}_{\text{Source1}})^2 + (\text{Jitter}_{\text{Source2}})^2}$$

In this case, the jitter due to the signal source that runs the first SCAN25100 is small relative to the jitter of the recovered clock it generates. So using these rules, one can generate a rule of thumb for a series of **n** SCAN25100 parts in series.

$$\text{Jitter}_{\text{Total}} \approx 6 \text{ ps} \times \sqrt{n}$$

RESIDUAL PHASE NOISE AND JITTER IN A MULTI-HOP ARCHITECTURE

In addition to knowing the total jitter from all the hops, it is also is useful to know the noise contribution at each hop. This knowledge is useful in order to generalize this result to more hops and to also get some idea of the board to board variation of the LMK-SCAN25100 board. Residual phase noise is the term used to describe the phase noise added by just one device, as opposed to total phase noise, which applies to the whole system.

The setup for residual phase noise is exactly the same as that for phase noise as shown in *Figure 1*. In fact, all the results concerning residual phase noise were calculated from the results from phase noise in *Figure 2*. Be aware that this calculation is very sensitive to noise.

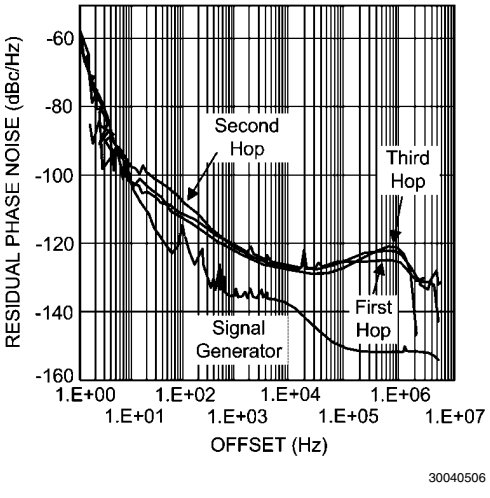

FIGURE 3. Calculated Residual Phase Noise

From these jitter numbers, the added jitter at each stage can be calculated. The formula is:

$$(\text{Jitter}_{\text{Total}})^2 = (\text{Jitter}_{\text{Additive}})^2 + (\text{Jitter}_{\text{Original}})^2$$

$$\Rightarrow \text{Jitter}_{\text{Additive}} = \sqrt{(\text{Jitter}_{\text{Total}})^2 - (\text{Jitter}_{\text{Original}})^2}$$

The table below shows the calculated jitter numbers as a function of the lower offset frequency and the point of measurement. Observe that it is roughly the same for each stage.

TABLE 2. Calculated Residual Jitter

		Residual Jitter (ps) vs. LowerLimit and Point of Measurement			
		Signal Source	**First Hop**	**Second Hop**	**Third Hop**
LowerLimit	10 Hz	0.467	5.902	7.100	5.749
	100 Hz	0.437	5.896	7.083	5.745
	1 kHz	0.432	5.889	7.076	5.740
	10 kHz	0.420	5.878	7.067	5.730
	100 kHz	0.406	5.778	6.983	5.660

An important observation regarding *Figure 3* and *Table 2* is that the shape of the jitter added by each stage is similar, at least below 1 MHz. And even though this shape looks different past 1 MHz, the jitter, which is related to the area under the curve, is about the same order for every stage.

Understanding the Impact of Clock Cleaning

CLOCK CONDITIONERS

Consider the recovered clock phase noise as presented in *Figure 2*. One can calculate the jitter of this phase noise by treating the upper limit for integration (UpperLimit) as a variable. If this is done, the following figure is obtained.

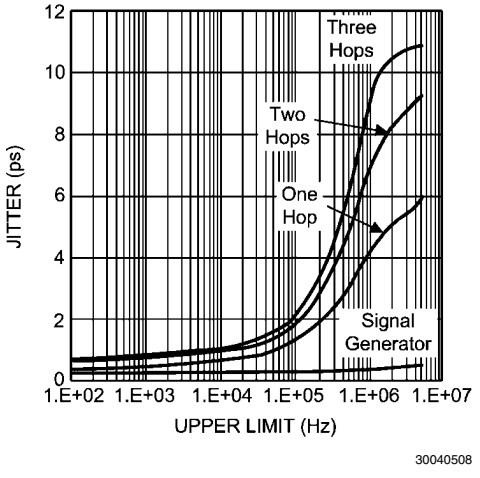

FIGURE 4. Jitter as a Function of UpperLimit

An important observation with *Figure 4* is that if the upper limit of integration could be restricted in some way, then the jitter could be dramatically reduced. This motivates the discussion of a clock cleaner. A clock cleaner is essentially a filter that will filter the noise that is beyond some cutoff frequency. In this case, if a filter could be designed such that one could roll off the noise past 10 kHz or so, the jitter could be substantially reduced. If this filter cleans up the noise, one may wonder what should stop one from setting the cut off point to a much lower frequency, like 1 Hz. The reason is with the practical implementation of the filter. This filter is not a low-pass filter, but rather a band pass which can be programmed to the center frequency of 30.72 MHz. In other words, this filter is a clock conditioner.

A clock conditioner consists of a PLL, VCO, and Distribution block. In order to clean the clock, the 30.72 MHz recovered clock is divided by R (2 in this case) in order to get a phase detector frequency of 15.36 MHz. There is a Voltage Controlled Oscillator (VCO) at 1474.56 MHz. This VCO frequency is divided down by N (96 in this case) in order to generate a feedback signal of 15.36 MHz. The outputs of the N and R counters are compared by the Phase Frequency Detector (PFD). The PFD outputs a correction current to the loop filter that converts this correction current into a voltage that can steer the VCO. The loop filter can be thought of as a low pass filter that integrates these correction currents from the PFD.

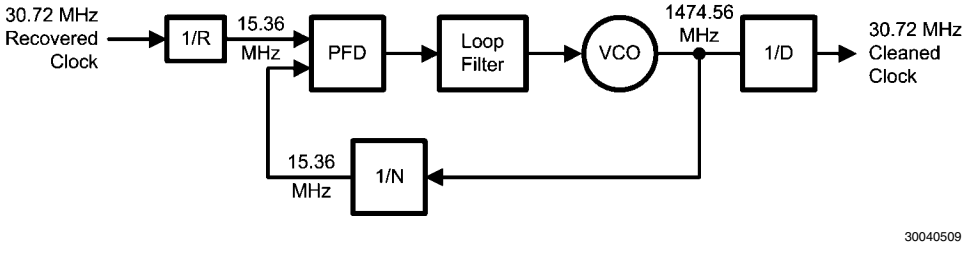

FIGURE 5. Clock Conditioner Architecture

CHOOSING THE OPTIMAL LOOP BANDWIDTH

The loop filter can be implemented with one external resistor and two external capacitors. These components can be chosen to adjust the loop bandwidth of the system. At frequencies below the loop bandwidth, noise from the 30.72 MHz recovered clock pass through, but the VCO noise is attenuated. At frequencies above the loop bandwidth, the noise of the 30.72 MHz recovered clock is attenuated, but the VCO noise passes through. Therefore, the optimal design choice for the loop bandwidth depends on the noise of the VCO and recovered clock. In this case, the VCO used was the internal VCO of the LMK03001C clock conditioner. The recovered clock was from the SCAN25100, and is shown for one hop, two hops, or three hops in *Figure 6*. Normally, one finds the frequency at which the free-running VCO noise is equal to the other in-band noise sources and adds 25% to this in order to find the optimal jitter. The extra 25% comes because the loop filter is not a brick wall filter and there tends to be some peaking near the loop bandwidth. In the case of a recovered clock, it is probably better to not add this 25% factor because more hops could be added. Therefore, one should just pick the frequency at where these curves intersect.

From *Figure 6*, this implies that if one hop is used, then a loop bandwidth of about 11 kHz is optimal, for two hops the optimal loop bandwidth is about 8.5 kHz, and for three hops, the optimal loop bandwidth is about 7.2 kHz.

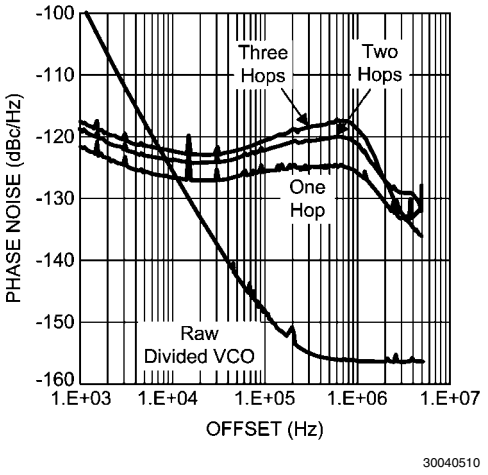

FIGURE 6. Determining Which Loop Bandwidth to Use

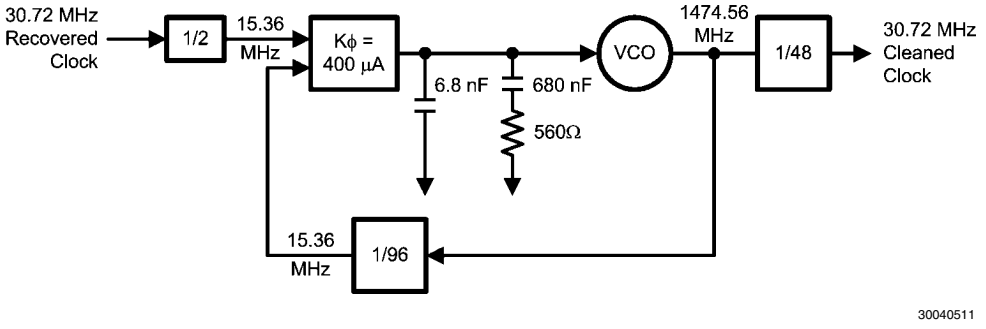

FIGURE 7. LMK03001C Setup for Cleaning a 30.72 MHz Recovered Clock

IMPLEMENTING THESE DESIGN TECHNIQUES IN A PRACTICAL DESIGN

In order to show all these techniques, three hops with the SCAN25100 were used and the final stage was cleaned with the LMK03001C clock conditioner. From *Figure 6*, we see that the optimal loop bandwidth for three hops is about 7.2 kHz. However, the choice of three hops is arbitrary, and perhaps more hops could be used.

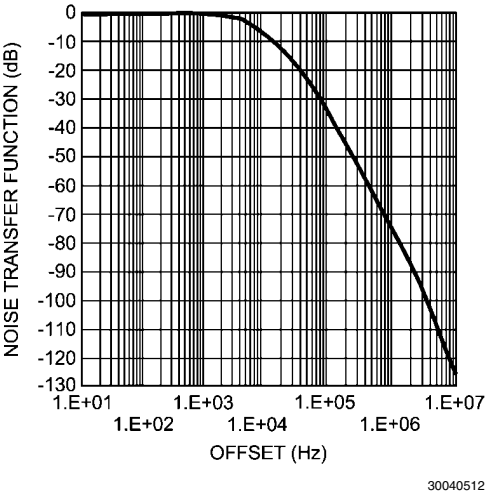

FIGURE 8. Recovered Clock Noise Suppression Due to the Loop Filter

For this reason, a narrower loop bandwidth of 4 kHz was used to be sure that this solution would be robust if more hops were used. Also, it makes the design less dependent on the data

patterns used in the recovered clock and the spectral purity of the crystal used to drive the SCAN25100.

Figure 8 shows how much the loop filter of the system suppresses the recovered clock noise. At offsets greater than 4 kHz, the recovered clock noise is attenuated. According to *Figure 6*, the optimal value for the loop bandwidth would be closer to 7.2 kHz, but it was reduced to 4 kHz because the results for the three hop analysis with the 3 hops were sufficient, and narrower loop bandwidth makes the design more flexible if more hops are added.

The LMK03000C could also be used for this application, but the LMK03001C was chosen because a VCO frequency of 1474.56 MHz could be used. For the LMK series of parts, the VCO phase noise, adjusted for output frequency, is best at the lower frequency range because there is more internal VCO capacitance switched in to generate this lower VCO frequency. Since the LMK03001C has a lower VCO limit of 1474.56 MHz, this makes it a good choice.

The LMK03001C also has 16 levels of charge pump current. If the charge pump current is below about 400 uA, then there could be some slight degradation to phase noise. Increasing the charge pump current beyond 400 uA has no measurable improvement in phase noise, but it increases the size of the loop filter capacitors. For this reason 400 uA was chosen for the charge pump current. As for the phase detector frequency, the choice was motivated to balance the PLL phase noise contribution, loop filter capacitor sizes, and cycle slipping. At 15.36 MHz phase detector frequency, this makes it such that the LMK03001C PLL phase noise is far below that of the recovered clock. Although a 30.72 MHz phase detector frequency could be used, it would not really improve the in-band phase noise, since it was dominated by other sources. Also, it would increase the capacitor sizes and also degrade lock time due to cycle slipping.

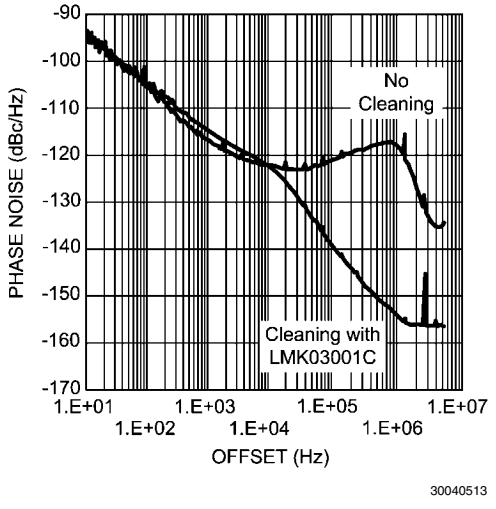

FIGURE 9. LMK03001C Setup for Cleaning a 30.72 MHz Recovered Clock

In *Figure 9*, the recovered clock has lower jitter if it is cleaned. If one uses an integration limit of 100 Hz to 5 MHz, then the clock that is not cleaned has a jitter of 5.3 ps and the cleaned clock has a jitter of 1.4 ps. Note that in the range of 100 Hz to about 8 kHz, the clock conditioner actually adds a little phase noise. This is because the VCO noise is contributing here. If this is a major concern, the loop bandwidth should be increased. This design was done with a wider loop bandwidth as well and the jitter was reduced to around 900 fs.

Conclusion

The SCAN25100 and LMK03001C family are an excellent choice of parts for sending and recovering data. In a multi-hop architecture, the SCAN25100 has a built-in reference clock that makes it such that there is no reference clock necessary for any of the SCAN25100 parts, except for the first in the series that is used to serialize the data.

LMK03001C is an ideal choice for cleaning this clock, since it has excellent phase noise performance at higher offset frequencies. This is important because this noise at higher offset frequencies the recovered clock noise will be attenuated.

The noise of the reference clock of the SCAN25100 deteriorates primarily at higher phase noise offset frequencies when increasing the number of hops, so filtering at every hop, as opposed to just the last hop, only gives a marginal benefit to jitter cleaning, since this noise will all be filtered at the last hop anyways. However, there could be reason to clean the clock at every hop if there was need to use this recovered clock at every hop, instead of just at the very end.

The LMK02000 clock conditioner allows the user to supply a VCXO in order to obtain improved phase noise performance close to the carrier if that is required for system reasons other than data clocking.

Notes

For more National Semiconductor product information and proven design tools, visit the following Web sites at:

Products		Design Support	
Amplifiers	www.national.com/amplifiers	WEBENCH	www.national.com/webench
Audio	www.national.com/audio	Analog University	www.national.com/AU
Clock Conditioners	www.national.com/timing	App Notes	www.national.com/appnotes
Data Converters	www.national.com/adc	Distributors	www.national.com/contacts
Displays	www.national.com/displays	Green Compliance	www.national.com/quality/green
Ethernet	www.national.com/ethernet	Packaging	www.national.com/packaging
Interface	www.national.com/interface	Quality and Reliability	www.national.com/quality
LVDS	www.national.com/lvds	Reference Designs	www.national.com/refdesigns
Power Management	www.national.com/power	Feedback	www.national.com/feedback
Switching Regulators	www.national.com/switchers		
LDOs	www.national.com/ldo		
LED Lighting	www.national.com/led		
PowerWise	www.national.com/powerwise		
Serial Digital Interface (SDI)	www.national.com/sdi		
Temperature Sensors	www.national.com/tempsensors		
Wireless (PLL/VCO)	www.national.com/wireless		

National Semiconductor Americas Technical Support Center
Email: support@nsc.com
Tel: 1-800-272-9959

National Semiconductor Europe Technical Support Center
Email: europe.support@nsc.com
German Tel: +49 (0) 180 5010 771
English Tel: +44 (0) 870 850 4288

National Semiconductor Asia Pacific Technical Support Center
Email: ap.support@nsc.com

National Semiconductor Japan Technical Support Center
Email: jpn.feedback@nsc.com