Application Report
SNAA073G–May 2011–Revised February 2017

AN-2132 Synchronizing Multiple GSPS ADCs in a System: The AutoSync Feature

ABSTRACT

This application report covers an overview of synchronization, implementation of the AutoSync feature, and FAQs on AutoSync.

Contents

List of Figures

List of Tables

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

The AutoSync feature, new on Texas Instruments 10- and 12-bit GSPS ADC family, is capable of synchronizing the data at the output of multiple ADCs in a system. The novel architecture of this feature departs significantly from previous DCLK reset style solutions and has considerable advantages. See Figure 1 for an example. For this application report, "ADC" refers to the ADC12D1800/1600/1000, the ADC10D1500/1000, and the ADC12D1800/1600/1000/800/500RF. It also includes the ADC10D1000QML and ADC12D1600QML.

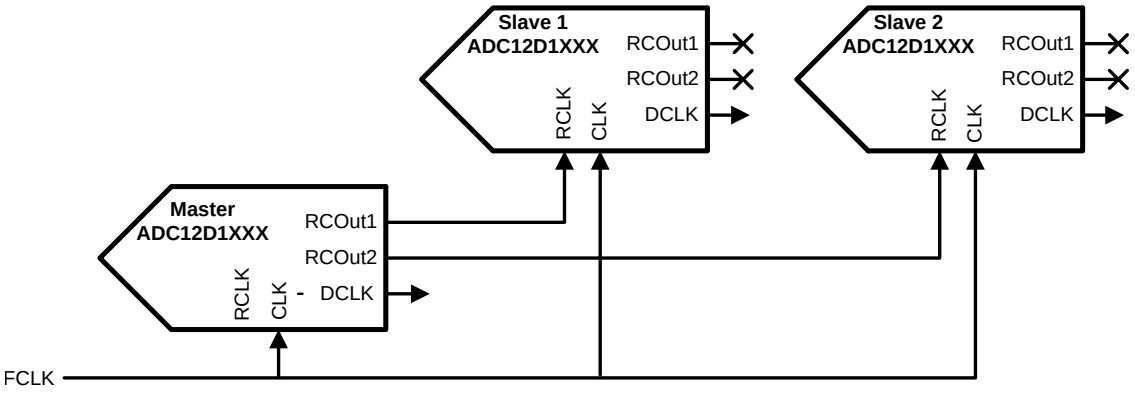

Figure 1. Synchronizing Multiple ADCs Using AutoSync

2 Overview of Synchronization

2.1 The Goal: Synchronizing Multiple ADCs

The purpose of the AutoSync feature is to facilitate aligning the Data and DCLKs of multiple ADCs in a system, so that the data may be easily captured by one FPGA, as illustrated in Figure 2. In this example, data from ADC_1 and ADC_2 are captured by $FPGA_1$, so they must be synchronized.

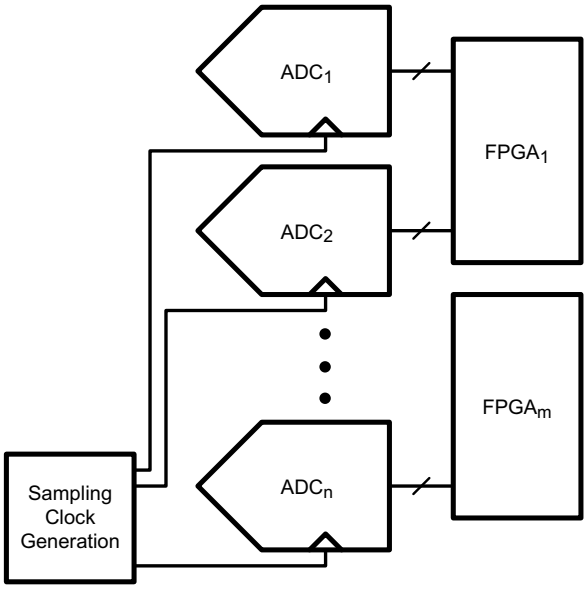

Figure 2. Multiple ADCs in a System

It is only necessary to synchronize the data from multiple ADCs, which data is being captured by the same FPGA. For example, if a system has multiple modules with multiple ADCs and one FPGA per module, then it is only necessary to synchronize the data at the output of the ADCs per module. Synchronizing the FPGAs is a challenge left to the system designer.

In general, this feature is useful in systems where the relationships of the analog inputs to each ADC, with respect to one another, must be known. The two most general cases are:

1. when all ADCs in the system must sample multiple inputs at the same time
2. when the ADCs in the system must sample the input with a known phase relationship with respect to one another

An example of the first case is a 4-channel oscilloscope which must simultaneously sample and display each analog input. For the second case, any system in which the ADCs are interleaved requires that their data output are aligned with a known phase relationship in order to correctly interleave the data in digital post-processing. In both cases, the inputs may be designed to arrive at each ADC with the desired relationship with respect to one another and the Sampling Clock may arrive at each ADC at the same time, but if the converted data at each ADC output has an unknown relationship with respect to the other ADC outputs, then the critical information which was carefully set up at analog inputs and Sampling Clocks, would be lost.

2.2 The Problem: Unsynchronized DCLKs

It is not ensured by design whether the rising Sampling Clock edge which samples the data will generate a rising or falling DCLK transition when data appears at the output. The reason for this is that the Data Clock, DCLK, is a divide-by-2 or divide-by-4 sub-harmonic of the Sampling Clock, FCLK. For the Demux Mode, it is also not certain which one of two Sampling Clock edges will generate the DCLK (and Data). In order to completely synchronize the DCLKs, three requirements must be met:

1. the Sampling Clock must arrive to each ADC at the same instant
2. each DCLK must be generated from the same edge of the Sampling Clock
3. the phase of each DCLK must be the same

DCLK is generated from the Sampling Clock, which is why the Sampling Clock must arrive to each ADC at the same time. Any delta in the arrival of the Sampling Clocks translates to the a similar delta between DCLKs. For 1:2 Demux Mode, in which the output Data is produced on two 12-bit busses, the DCLK runs at ¼ the rate of the Sampling Clock, also known as "FCLK". See Figure 3. $DCLK_1$, $DCLK_2$, $DCLK_3$ and $DCLK_4$ represent the possible DCLKs that might be generated by the same FCLK upon power-up; the same phase DCLK will not be generated at each power-up. When each ADC starts up, there is inherent uncertainty whether the DCLK will start up generated by the first edge of the Sampling Clock, ($DCLK_1$ and $DCLK_3$) or the second edge ($DCLK_2$ and $DCLK_4$) and whether DCLK will be rising ($DCLK_1$ and $DCLK_2$) or falling ($DCLK_3$ and $DCLK_4$). Since the DCLK is a subsample of the Sampling Clock by 4x, there are 4 possibilities for DCLK.

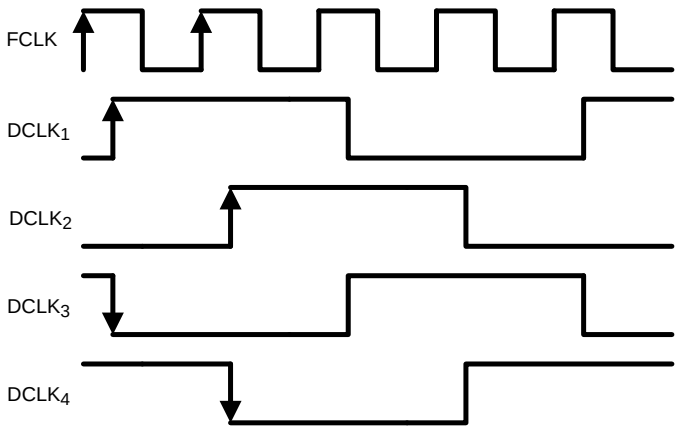

Figure 3. Unsynchronized DCLKs, Demux Mode

For the Non-Demux Mode, the DCLK rate is ½ the Sampling Clock rate; therefore, there are two possibilities for DCLK. See Figure 4.

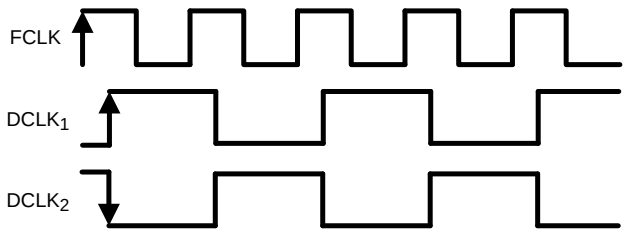

Figure 4. Unsynchronized DCLKs, Non-Demux Mode

Therefore, because the Data and associated DCLK of unsynchronized ADCs are not ensured to be generated at the same time or on the same phase of DCLK, the capture of the Data at the FPGA becomes a difficult task. The solution is to synchronize the DCLKs and Data, which removes any data capture difficulties at the FPGA, via the AutoSync feature.

2.3 The Solution: Autosync

The AutoSync feature operates by configuring one ADC as the Master and all other ADCs as Slaves; the Slave DCLKs are synchronized to the Master DCLK. A Reference Clock, RCLK, runs from the Master ADC to Slave ADC(s) to control the phase of the Slave DCLKs, as shown in Figure 1. Each ADC may be configured as either a Master or a Slave; it is a Master by default. Each ADC can provide up to two Reference Clocks, RCOut1 and RCOut2 to control the phase of Slave ADCs. RCOut1 and RCOut2 are turned off by default; when enabled, they run at FCLK/4 or FCLK/2, depending upon the product. During AutoSync system configuration, each RCLK is configured for a clean capture at each Slave ADC; the AutoSync SP control bits are used to select the correct one of four possible phases of DCLK. The Master and all Slave DCLKs must be observed, for example, by an FPGA, in order to configure the AutoSync feature, as well as verify DCLK synchronization. It is only necessary to configure the AutoSync system for one unit; all other production units (with identical layout, RCLK trace/cable length, and so on) may be simply written with the same AutoSync settings.

Another option is to configure all ADCs in Slave Mode and to drive their RCLKs externally. In this case, one ADC is designated to provide the reference DCLK (similar to the Master), and the other ADCs are synchronized to it.

AutoSync provides significant advantages over previous DCLK reset style solutions:

AutoSync	DCLK Reset
The system recovers automatically from a spontaneous loss of synchronization, for example, caused by a power glitch, without requiring explicit detection. AutoSync is continuously active; any DCLK phase errors quickly propagate out of the system.	If an ADC's DCLK phase loses alignment, loss of synchronization must be detected explicitly at the system level. It can only be corrected by the application of another reset pulse.
No precise setup and hold times are required for RCLK because it is generated by the ADC and configured in a control feedback loop.	Precise setup and hold times are required for the DCLK reset pulse.
There is some flexibility in the system configuration. The Reference Clocks may be configured as a binary tree, daisy chain, or sourced independently.	There is only one system configuration. The DCLK reset pulse must arrive at each ADC at the same time, similar to the Sampling Clock.
When one system is configured, these settings are valid for all production units.	Each unit must be synchronized via a pulse upon power-up.

A simplified version of the design for the Master reference clock generation is shown in Figure 5 and the Slave reference clock generation is shown in Figure 6.

For the Master Mode, the internal reference clocks, FCLK/4 and FCLK/2, are generated, which are synchronous to the sampling clock, FCLK.

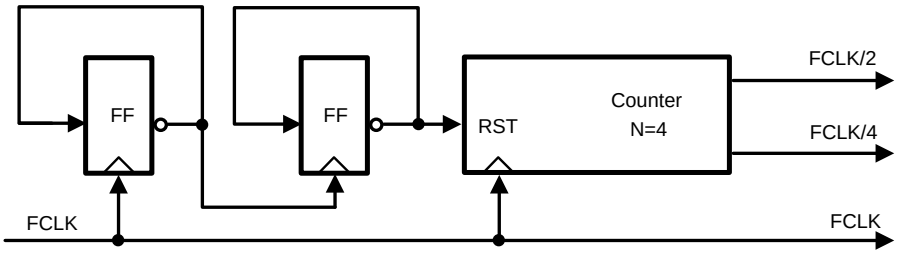

Figure 5. DCLK Generation for Master Mode

For the Slave Mode, instead of independently generating its own reference clocks, the FCLK and counter, which phase are controlled by the RCLK and SP control bits, are used to generate the divide-by-2 and divide-by-4 internal reference clocks, which result in a Slave DCLK that is in-phase to the Master DCLK. By design, its reference clocks are then in a fixed phase relationship to RCLK. T_d represents an adjustable analog delay block, which is used to achieve optimal setup and hold times.

The design of the Slave Mode represents several key advantages of this feature. Since the RCLK recieved by the Slave ADC has an analog delay block, this removes the requirement to achieve stringent setup and hold times to synchronize the outputs of multiple ADCs. Since the DCLK generated by the Slave ADC is based off of its FCLK, although the phase is determine by the RCLK, the variation through the AutoSync system is not additive. For example, the maximum variation is the variation from one Master plus the variation from one Slave, or, for the case of an externally driven Slave, it is simply the variation from one Slave ADC. Since the final phase of the Slave DCLK is configured via the SPI, this also removes the requirement on the RCLK trace to be any particular length.

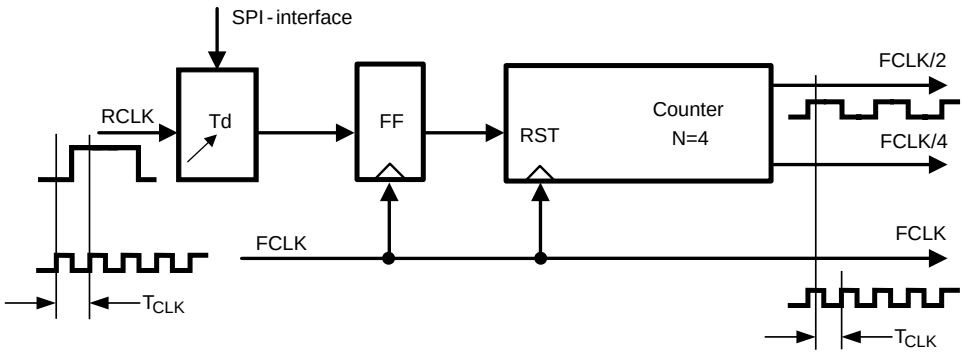

Figure 6. DCLK Generation for Slave Mode

2.3.1 AutoSync Configurations

AutoSync may be configured as a binary tree as in Figure 7, daisy chain as in Figure 8, or independently sourced as in Figure 9, depending upon system requirements. The daisy chain configuration is simply a special case of the binary tree. If RCLK is driven externally, then it must be derived from the Sampling Clock. Note that, due to PVT variation through the Master and Slave ADC AutoSync circuitry, for high sampling clock rates, it may be necessary to drive RCLK externally; see Section 3.3.3 for more information.

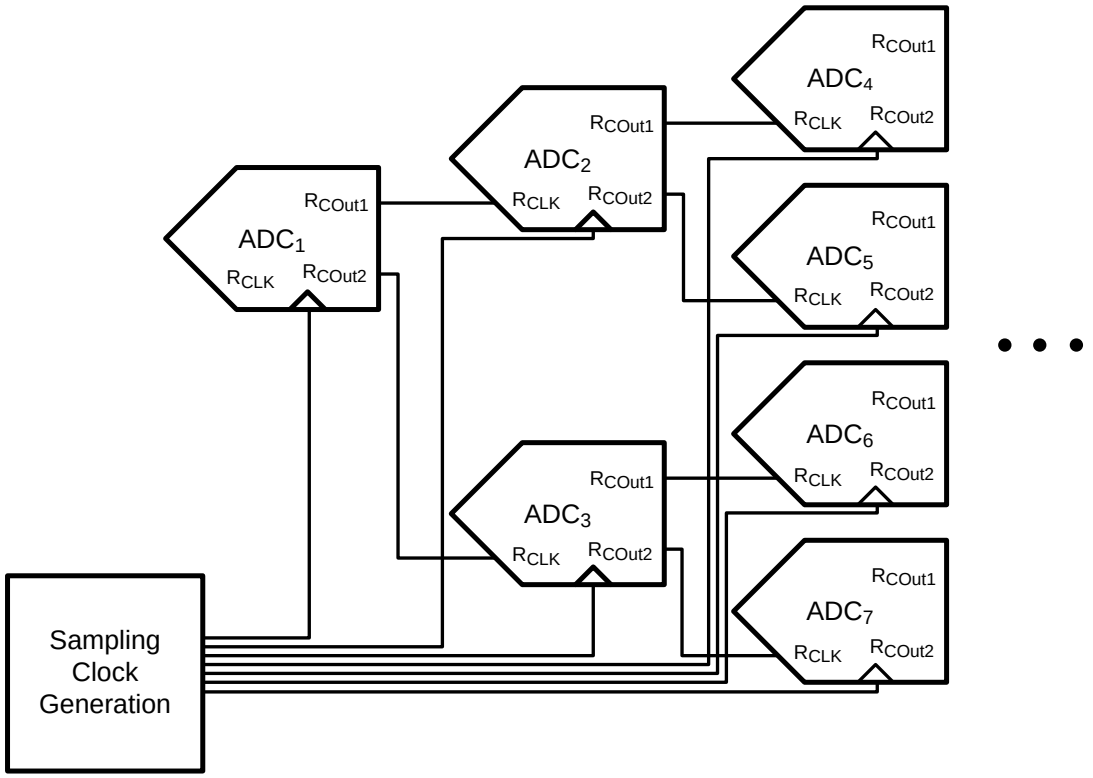

Figure 7. Binary Tree Configuration

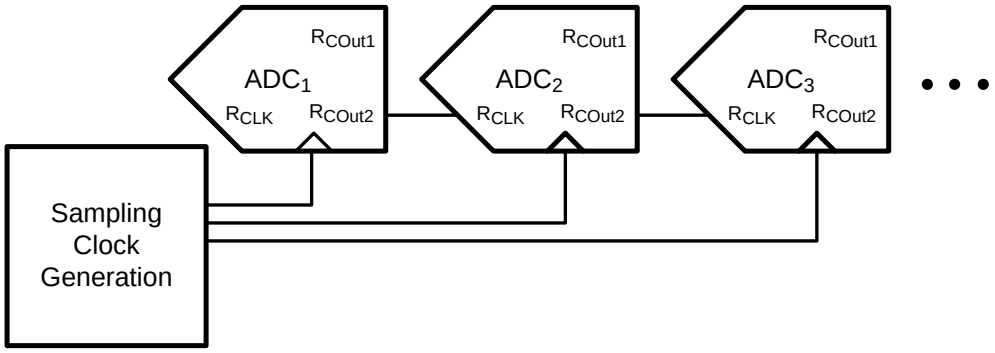

Figure 8. Daisy Chain Configuration

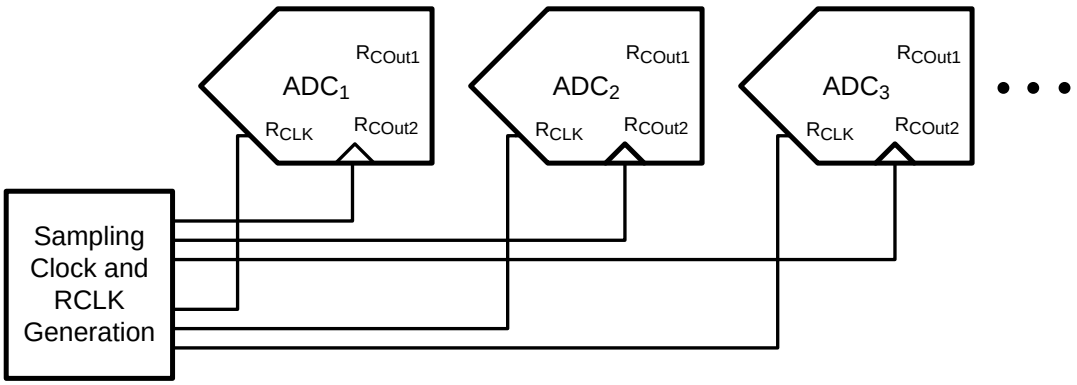

Figure 9. Driving RCLK Externally

3 Implementing AutoSync

The key to implementing AutoSync correctly lies with ensuring that the Sampling Clock arrives to each ADC in the system at the same time, configuring the RCLK to each Slave ADC, and verifying that the synchronization has been correctly configured.

3.1 Implementing the Sampling Clock

Designing the Sampling Clock to arrive at each ADC at the same time is the most difficult part of implementing the AutoSync feature. It is necessary to accomplish this so that each analog input to the ADC is sampled at the same instant. For example, Texas Instruments LMK01xxx family is ideal for clocking the ADC up to 1.6 GHz and the LMK04xxx may be used for higher frequencies.

3.1.1 Selecting a Clocking Chip

A Sampling Clock generation or distribution chip should be selected for at least the following qualities: multiple differential outputs, low jitter and programmable output skew. Single-ended outputs are not recommended; since the Sampling Clock which drives the ADC must be differential, a single-ended to differential conversion stage can introduce unknown skew into the system. For example, a balun is commonly used for single-ended to differential conversion, but some baluns do not have a ensured phase relationship from input-to-output. Simply selecting a clock generator with differential outputs will avoid these potential complications. Low jitter is necessarily a requirement for ultra high-speed ADCs, but this application report will not cover extensive details regarding that topic. Another key is matching the traces from the clock generation chip to each ADC; traces should be well-matched between each + and - output as well as the length of each output pair to each ADC. Programmable output skew is a nice to have feature for tuning out any systematic skew on the board, but care must be taken when using this feature because adding delay typically also adds jitter. The ADC also has a feature for adjusting the incoming Sampling Clock, Sampling Clock Phase Adjust. When using the Sampling Clock Phase Adjust feature, it should be noted that this will also affect the timing of the DCLK. The effect of adjusting the incoming Sampling Clock also adjusts all clocks which are subsequently generated from it. For example, if the incoming Sampling Clock is delayed by Δt, then the DCLK as well as the Data is also delayed by Δt; the RCOut1/2 are not delayed.

3.1.2 Generating DCLK From the Sampling Clock

From the rising edge of the Sampling Clock (FCLK in Figure 10), the DCLK edge transitions after a delay which is composed of three components; the latency, the Sampling Clock-to-data output delay, and the aperture delay. See Figure 10. The latency, t_{LAT}, is the integer number of Sampling Clock cycles (or half cycles), ensured by design, which depends upon the Demux and DES Mode selection, for an analog sample to be converted into a digital value. The actual numbers are the same for all of the ADCs and are available in the datasheet. The Sampling Clock-to-data output delay, t_{OD}, is a fixed time delay in addition to the latency, independent of the Sampling Clock frequency, which is due to gate and parasitic delays. The Aperture Delay, t_{AD}, is the amount of delay, measured from the sampling edge of the clock input, after which the signal present at the input pin is sampled inside the device.

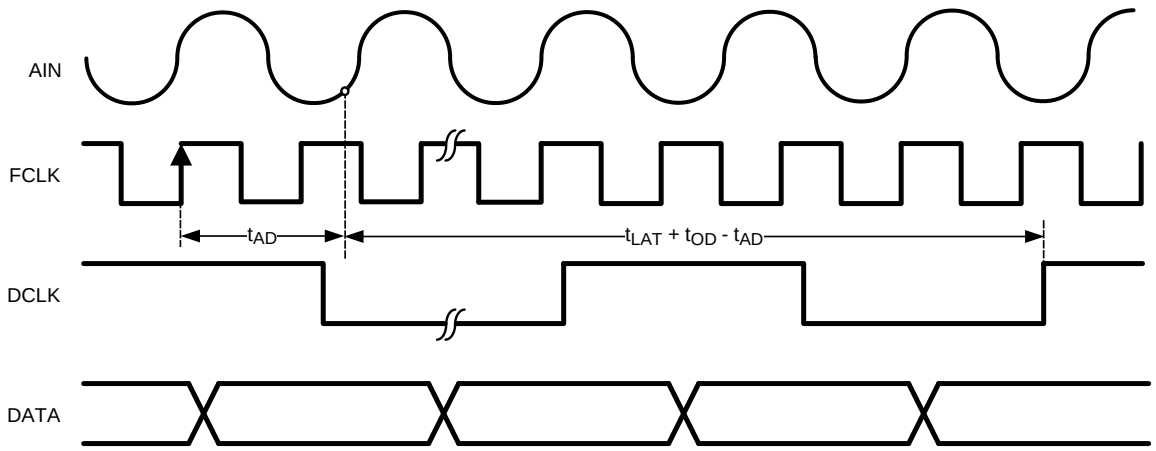

Figure 10. DCLK Generation Timing

If the same Sampling Clock arrives at each ADC at the same time, then the skew between DCLK transitions from ADC to ADC is only dependent upon t_{OD}. These parameters are slightly different for the 10- and 12-bit GSPS ADC families:

	t_{OD}	t_{AD}
ADC10D1x00	2.4 ns (typ)	1.1 ns (typ)
ADC12D1x00	3.2 ns (typ)	1.15 ns (typ)
ADC12Dx00RF	3.15 ns (typ)	1.22 ns (typ)
ADC12D1x00RF	3.2 ns (typ)	1.29 ns (typ)

It is not possible for t_{OD} and t_{AD} to vary so much that they upset the AutoSync scheme. In fact, these parameters do not affect the AutoSync feature at all. See Section 3.3.3 for more information.

3.2 Implementing the Reference Clock

The Reference Clock must be routed from RCOut1/2 on the Master ADC to the RCLK on the Slave ADC and then configured. It is also possible to drive the Reference Clock externally.

The routing for the Reference Clock does not have any stringent requirements, which is one of the advantages of the AutoSync feature. It is only necessary to match the length of the differential outputs to each other. In the case that the Slave ADC is on the same board, the trace may be any length; in the case that the Slave ADC is on a different board, the cable may be any length. The length of any Reference Clock trace / cable pair does not need to be the same length as any other pair. This is because the Reference Clock can be adjusted for clean capture at each Slave ADC. If the Slave Reference Clock is being driven by another ADC, then the trace / cable may be DC-coupled. The trace impedance should be 50Ω single-ended or 100Ω differential.

For routing the Reference Clock from one board to another, a shielded, twisted-pair, 50Ω single-ended cable which is rated up to the RCLK frequency is sufficient. Some off the shelf examples include: USB2.0, USB3.0, Firewire, Display Port, HDMI, Ethernet/RJ-45 for CAT-5/6/7, PCI-Express, and SATA/SAS. An example of an automotive grade cable and connector with a locking mechanism is Rosenburger HSD.

3.2.1 Configuring the Reference Clock

The AutoSync configuration register from the ADC datasheet is provided below as a reference: Note that Bit 6 is different between the ADC10D1x00 and the ADC12D1x00. For the ADC10D1x00, Bit 6 is available as part of the DRC control bits and for the ADC12D1x00, Bit 6 is reserved.

Table 1. ADC10D1x00 AutoSync Register

Address: E**h** (1110**b**)																POR state: 0003**h**
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DRC(9:0)										Res	SP(1:0)		ES	DOC	DR
POR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Table 2. ADC12D1x00(RF) AutoSync Register

Address: E**h** (1110**b**)																POR state: 0003**h**
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DRC(8:0)									Res		SP(1:0)		ES	DOC	DR
POR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

ADC10D1x00	
Bits 15:6	DRC(9:0): Delay Reference Clock. These bits may be used to increase the delay on the input Reference Clock when synchronizing multiple ADCs. The delay range is 0s (0**d**) to 1200 ps (639**d**). The delay remains the maximum of 1200 ps for any codes above or equal to 639**d**. Each bit equates to approximately 1.9 ps delay.
ADC12D1x00	
Bits 15:7	DRC(8:0): Delay Reference Clock. These bits may be used to increase the delay on the input Reference Clock when synchronizing multiple ADCs. The delay range is 0s (0**d**) to 1200 ps (319**d**). The delay remains the maximum of 1200 ps for any codes above or equal to 319**d**. Each bit equates to approximately 3.8 ps delay.
Bit 6	Reserved. Must be set as shown.
ADC10D1x00 / ADC12D1x00	
Bit 5	Reserved. Must be set as shown.
Bits 4:3	SP(1:0): Select Phase. These bits select the phase of the Reference Clock which is latched. The codes correspond to the following phase shift: 00**b** = 0°; 01**b** =90°; 10**b** = 180°; 11**b** = 270°
Bit 2	ES: Enable Slave. Set this bit to 1**b** to enable the Slave Mode of operation. In this mode, the internal divided clocks are synchronized with the Reference Clock coming from the master ADC. The master clock is applied on the input pins RCLK. If this bit is set to 0**b**, then the devices is in Master Mode.
Bit 1	DOC: Disable Output reference Clocks. Setting this bit to 0**b** sends a CLK/4 or CLK/2 signal on RCOut1 and RCOut2. The default setting of 1b disables these output drivers. This bit functions as described, regardless of whether the devices is operating in Master or Slave Mode, as determined by ES (Bit 2).
Bit 0	DR: Disable Reset. The default setting of 1**b** leaves the DCLK_RST functionality disabled. Set this bit to 0**b** to enable DCLK_RST functionality.

Step #1 - Configure the ADC into Master / Slave Mode

Configuring the ADC into Master / Slave is accomplished via Bit 2 (Enable Slave). By default, each ADC is configured as a Master, so that it is generating its own unsynchronized DCLK. Each ADC whose DCLK will be synchronized to another DCLK should be configured into Slave Mode. For a system which is configured as either a binary tree or daisy chain, there will be one ADC in Master Mode and the rest of the ADCs will be in Slave Mode. Generally, for a system in which RCLK is driven externally, all ADCs will be in Slave Mode. In this case, the DCLK of any one of the ADCs should be designated as a reference, similar to the Master ADC, to which the phase of the other ADCs will be synchronized.

Step #2 - Enable the Reference Clocks

Enabling / disabling the output Reference Clocks is accomplished via Bit 1 (Disable Output reference Clocks). Leaving the Reference Clocks disabled will save a small amount of power and reduce any spurious energy at that frequency, so it is generally recommended to leave the output Reference Clocks disabled if they are not used. Both Reference Clocks are enabled / disabled together; it is not possible to enable just one or the other.

Step #3 - Adjust each Slave ADC's Reference Clock for a clean capture

The incoming Reference Clock to each ADC may be delayed by using the Delay Reference Clock (DRC) Bits. As the Reference Clock is delayed by increasing DRC from 0 to the maximum code, the behavior of the Slave ADC's DCLK will change. An example is shown in Figure 11. The codes in certain regions will provide a stable DCLK output, as shown by the regions labeled: "DCLK", "DCLK + 90°", and "DCLK + 180°". The codes in the shaded region will not produce a stable DCLK output; instead, DCLK will generally be logic-high in the unstable region. There may also be 1 or 2 codes on the border of the stable / unstable region which produce an intermittent DCLK. As the codes increase from 0 to the max code, the DCLK produced will be 90° delayed as compared to the DCLK produced in the previous stable region.

These are the priorities for selecting the DRC code:

- If there are no observable unstable regions, set DRC to the mid-code setting.
- If there is only one observable unstable region, set DRC to the middle of the largest available stable region.
- If there are two or more observable unstable regions, set DRC to the middle of the largest stable region which uses the lowest DRC code.

For an example of selecting the DRC code, see the arrow in the center of the "DCLK + 90°" region in Figure 11.

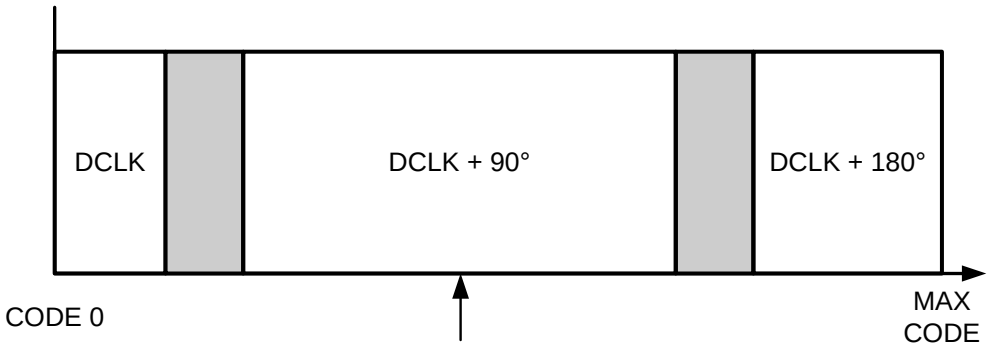

Figure 11. DRC Code vs DCLK Behavior

At this step, due to the nature of the AutoSync feature, it is not actually relevant what the phase of the Slave DCLK is (as compared to the Master / Reference DCLK), only that the DCLK is generated by selecting a DRC code in the center of the stable region. Therefore, the length of trace or cable from where RCLK is generated to each Slave ADC may be any length.

The minimum code of DRC = 0**d** translates to a delay of 0 ps and the maximum code translates to a delay of 1200 ps for both the ADC10D1x00 and the ADC12D1x00. The amount of delay scales linearly with the code. Therefore, the code in the center of the stable region will be the average value of the codes which trigger the start of the unstable region on either side of the stable region. The delay is an absolute amount of time and is not related to the Sample Clock frequency. A faster Sample Clock frequency and, consequently, faster RCLK, will result in more transitions from stable to unstable region (and vice versa) over the full range of codes.

Step #4 - Select the correct DCLK phase for each Slave ADC to match the Master DCLK phase

The phase of the Slave DCLK can be adjusted to match that of the Master DCLK using Bits 4:3 (Select Phase). There are only four possible settings: 0°, 90°, 180°, and 270°. Select the phase that matches the Master DCLK, for example, SP<1:0> = 01**b** as shown in Figure 12. The phase shifted DCLK for the other three SP codes are also shown on the same plot.

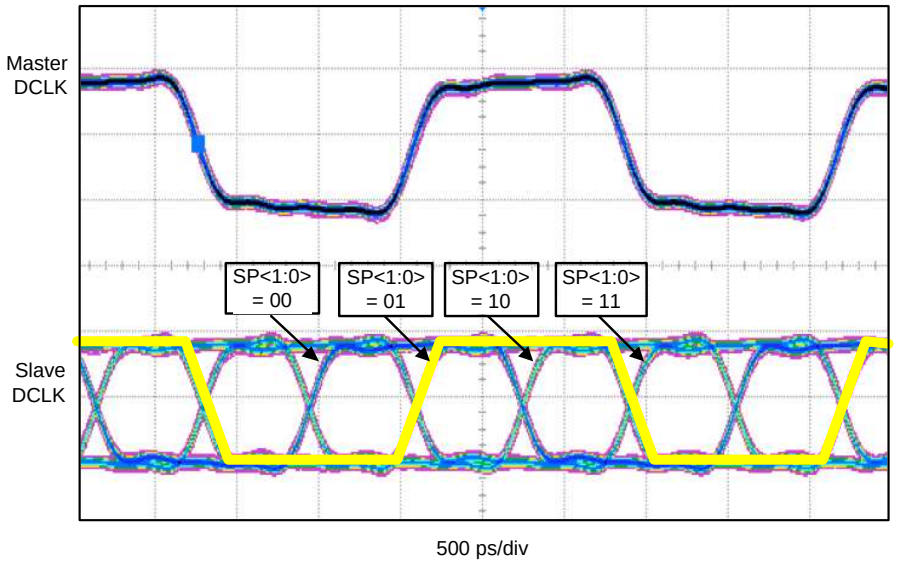

Figure 12. Slave DCLK vs. SP<1:0> Selection

3.2.2 Driving the Reference Clock Externally

Depending upon what architecture is most convenient for the system, it is also possible to drive the Reference Clocks externally. There are two options; in either case, the RCLK must be phase locked to the Sampling Clock. In the first case, one clock generation and distribution chip can drive all the Slave ADC Sampling Clocks and Reference Clocks, as in Figure 9. In the second case, the Reference Clock from one Master ADC is distributed to the Slave ADCs, for example, Texas Instruments LMK01000, as in Figure 13.

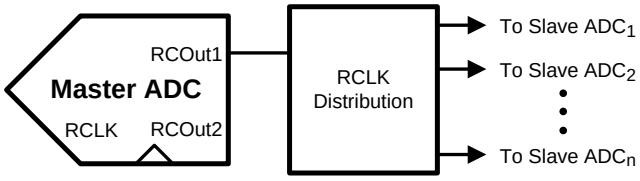

Figure 13. RCLK Distribution and External Drive

The Equivalent Circuit for RCLK is shown in Figure 14. When driving RCLK externally, it must be done differentially. It should be AC-coupled, as the input buffer will create its own bias internally. The differential RCLK should have Vpp > 250 mV. The maximum limits are given in the datasheet. RCLK, as generated by the ADC, has the same characteristics as the DCLK: it is a square wave with duty-cycle 50% and frequency as shown in Table 3. RCLK may not be a pulse.

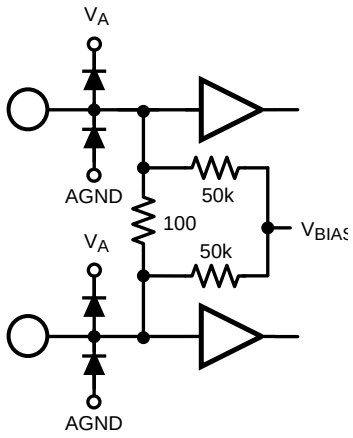

Figure 14. RCLK Equivalent Circuit

Table 3. RCLK Frequency by Product

Product	RCLK Frequency
ADC10D1000	FCLK / 4
ADC10D1500	FCLK / 4
ADC12D1000	FCLK / 4
ADC12D1600	FCLK / 4
ADC12D1800	FCLK / 4
ADC12D500RF	FCLK / 2
ADC12D800RF	FCLK / 2
ADC12D1000RF	FCLK / 4
ADC12D1600RF	FCLK / 4
ADC12D1800RF	FCLK / 4
ADC12D1600QML-SP in LSPSM	FCLK / 2
ADC12D1600QML-SP in non-LSPSM	FCLK / 4

3.3 Verifying the System

The final steps include implementing the DCLK and verifying the closed loop system.

3.3.1 Implementing the DCLK

Since the DCLKs must be synchronized when they reach the FPGA, the DCLK routing from each ADC to the FPGA should be matched in length. If the ADCs are mounted on separate boards, then it is a system-level design challenge to match the DCLK routing to the FPGA. Alternately, the data captured at each FPGA may be synchronized at a different levels in the system. It is strongly recommended to add probe points on each DCLK, at the same distance from each ADC DCLK output to each probe point. This will enable verification of AutoSync via an oscilloscope during the process of system debug if the FPGA does not initially work to accomplish this.

3.3.2 Verifying AutoSync

The DCLKs from all ADC must be monitored, at least to configure the AutoSync feature, for example, via an FPGA or oscilloscope. Once the feature is configured, if all DCLKs are on the same phase, then the feature has been correctly implemented. An exception to this is the setting of DRC; it is possible to choose a code for DRC which is not maximally ideal, that is, not centered in the stable region, and the feature will still function.

It is not possible to use the Test Pattern Mode (TPM) to verify any part of the AutoSync configuration. This is because the TPM does not start up in a synchronized manner. Therefore, it is possible that the multiple ADCs in the system are synchronized, but the output of their Test Pattern Modes may not be. However, the TimeStamp feature can be used to verify AutoSync. When enabled, this feature uses the LSB of the Data output to add a time stamp from a signal input at the DCLK_RST+/– pins. For example, for the ADC12D1800 in DES Mode, the resolution of this signal can be as low as 1/3.6GHz, which results in an uncertainty of only ±0.14 ns.

3.3.3 AutoSync PVT System Variation

If the AutoSync feature has been correctly implemented and configured, then RCLK cannot shift enough over Process, Voltage, and Temperature (PVT) corners to disrupt the phase of DCLK. However, if DRC is not placed in the middle of a stable region, but instead is placed close to an unstable region boundary, then it is possible that the Reference Clock could shift enough over temperature to cross the boundary into the unstable region and disrupt the correct DCLK phase. The PVT variation for the Master and Slave are as follows, from simulation:

- $dt(PVT)_{MASTER}$ = ±170 ps
- $dt(PVT)_{SLAVE}$ = ±100 ps

The total PVT AutoSync variation for a Master/Slave configuration is ±270 ps. Since the RCLK is always resynchronized to the Sampling Clock, even multiple daisy chained Slave ADCs will have a maximum PVT variation of only ±270 ps. In order for the AutoSync feature to be ensured, the total system variation must be less than one clock period, for example, for Fclk = 1.8 GHz, it must be < 555 ps. The timing budget must include the variation through the Master, the variation through the Slave, and the variation of the RCLK trace with respect to the FCLK trace, from board to board. This is because AutoSync is designed to be configured on one system, and is ensured to function on all systems. However, if AutoSync is explicitly configured on each system, then the RCLK trace variation can be ignored.

If the RCLK is externally driven, then the timing budget only includes the variation in the timing chip which is driving the Slave RCLKs, the variation of the RCLK trace with respect to the FCLK trace, and the variation through the Slave. This is how driving RCLK externally can make meeting the timing budget for a particular FCLK frequency easier to achieve. For FCLK > 1 GHz, it is recommended to drive RCLK externally. If not, it is strongly recommended to carefully control the RCLK variation with respect to FCLK from board to board - or to configure AutoSync for each system.

The path for the AutoSync feature PVT variation is from the Sampling Clock through the AutoSync circuitry to the RCOut1 of the Master and from the RCLK through the AutoSync circuitry to the RCOut of the Slave, see Figure 15.

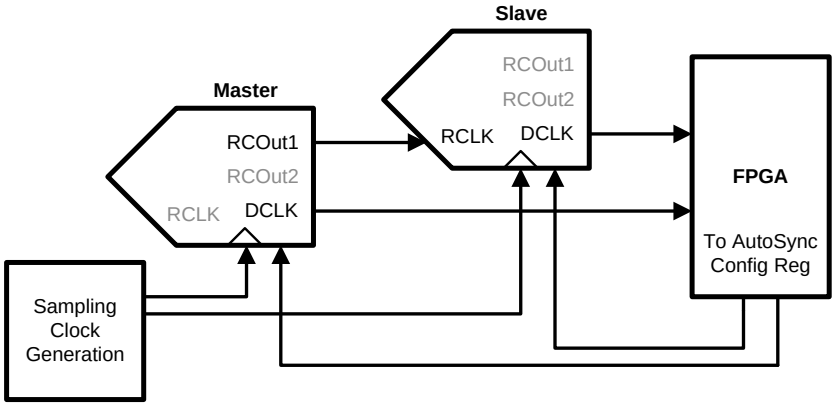

Figure 15. AutoSync Closed Loop Configuration

3.3.4 Sample Instant PVT System Variation

It is important to keep in mind that AutoSync is designed to solve one system issue: to align in phase all sub-harmonic clock generation for multiple, independent ADC devices within a single system. However, due to t_{OD} and t_{AD} PVT variation, AutoSync cannot ensure that the data which was sampled at the same instant at multiple, independent ADC devices will be present at the same index sample at the output of each ADC. t_{OD} and t_{AD} PVT variation can affect when the DCLK is generated, but they do not actually affect the functionality of AutoSync system.

From simulation, the largest expected variation is as follows:

- t_{OD} and t_{AD} vs Temperature: ±8.3%
- t_{OD} and t_{AD} vs Supply: ±3.6%
- t_{OD} and t_{AD} vs Process: ±19.2%
- t_{OD} and t_{AD} vs Composite PVT: ±29.7%

For example, consider the ADC12D1600RF; here are some of the relevant numbers:

- t_{OD} = 3.2 ns
- t_{AD} = 1.29 ns
- Fs = 1600 MHz; T = 625 ps
- Max PVT variation = ±29.7%

The time from sampling instant to data output is $t_{LAT} + t_{OD} – t_{AD}$. Since the latency for each ADC is the same, and the sampling clock to each is the same, t_{LAT} may be discounted. The maximum variation is:

$= (t_{OD} – t_{AD}) \times dt(PVT)$

= (3.2 ns - 1.29 ns) × ±0.297

= ±0.57 ns = 1.13 ns

Since the max PVT variation > T (1.13 ns > 625 ps), AutoSync cannot ensure that the same analog input, sampled at two different ADCs will be present at the same index output sample. However, this can be ensured for Fclk < 880 MHz (T = 1.13 ns). The second task is to align the samples, in time, which can be accomplished via the TimeStamp feature.

For a sample which is made at the same instant at two different ADCs, the maximum delta is shown in Table 4.

Table 4. Maximum Delta Number of Sample Instants

	N < 1	N < 2	N < 3
ADC10D1x00	{150 MHz, 1295 MHz}	{1295 MHz, 1500 MHz}	N/A
ADC12D1x00	{150 MHz, 821 MHz}	{821 MHz, 1642 MHz}	{1642 MHz, 1800 MHz}
ADC12Dx00RF	{150 MHz, 800 MHz}	N/A	N/A
ADC12D1600QML-SP LSPSM			
ADC12D1x00RF	{150 MHz, 881 MHz}	{881 MHz, 1762 MHz}	{1762 MHz, 1800 MHz}
ADC12D1600QML-SP non-LSPSM			

4 Acknowledgments

The author would like to thank the design team for their innovative work on this new feature and support for writing this AN, the GSPS ADC team for their support, and customers for challenging and relevant feedback. This technology has been patented; see US Patent Publication No. US 2010/0201559 AI; Publication Date Aug. 12, 2010.

5 Frequently Asked Questions

1. How should unused I/O on the AutoSync and DCLK reset features be terminated?

Table 5. Unused Synchronization Pin Recommendation

Pin(s)	Unused Termination
RCLK+/–	Do not connect.
RCOUT1+/–	Do not connect.
RCOUT2+/–	Do not connect.
DCLK_RST+	Connect to GND via 1-kΩ resistor.
DCLK_RST–	Connect to V_A via 1-kΩ resistor.

2. Is it okay to drive the RCLK on a Slave ADC that is powered down?

Driving the RCLK of an ADC which is not powered on will activate the ESD diodes of each RCLK+/- input. Also, an ADC which is driving a powered off RCLK will burn an additional 48mA across the 100Ω input termination. In order to achieve the maximum lifetime of the product, it is not recommended to do this. The Reference Clocks of an upstream ADC, RCOut1 and RCOut2, may be turned on or off via the Configuration Register (Addr: E**h**, Bit 1). For system operation, all ADCs should be first powered up, and then the Slave ADCs may be safely driven with RCLK.

3. What happens to DCLK if the Slave ADC loses its RCLK?

An ADC in Slave Mode which receives no RCLK will produce a static logic level at its DCLK output; DCLK will not transition. The AutoSync feature cannot be turned on or off, but rather, the ADC is configured into either Master Mode or Slave Mode. If the ADC is in Master Mode, then it is not synchronized to any other ADCs. If it is in Slave Mode, then it must receive a RCLK signal else it produces no DCLK.

4. Is it necessary to synchronize DCLKI and DCLKQ?

By design, the DCLKI and DCLKQ from each ADC are phase aligned with respect to one another; no additional effort is required to align them. For this reason, either DCLKI or DCLKQ may be used to capture data by the FPGA. Which DCLK is selected may simply be a matter of which is more convenient from a layout perspective. The other DCLK may be used as a system clock or left unused. As long as the ADC is receiving power and a Sampling Clock, and the channel is not powered-down (set via PDI or PDQ), and either configured in Master Mode or configured in Slave Mode and properly receiving an RCLK signal, then that channel will generate a DCLK. In the description and conceptual block diagrams, DCLK1, DCLK2, and so on, always refers to the DCLKs from ADC1, ADC2, and so on, and not to DCLKI and DCLKQ.

5. What is the phase relationship between RCLK and DCLK?

This relationship is uncharacterized because it is not essential to the functionality of the AutoSync feature. RCLK is configured empirically at each Slave ADC.

6. Does AutoSync have any limitations?

AutoSync relies on the Sampling Clock arriving to each ADC at the same time. If this timing is too seriously compromised, the AutoSync feature is not able to compensate for it.

Although AutoSync is very convenient to use because the RCLK trace / cable may be any length, it is still necessary to monitor each Slave DCLK as compared to the Master / Reference DCLK while configuring the feature. In the case that the ADCs to be synchronized are located on multiple boards, it is a system design challenge to ensure that each DCLK is routed back to a single FPGA with minimal skew for comparison.

The purpose of AutoSync is to align in phase all sub-harmonic clock generation for multiple, independent ADC devices within a single system. However, as a result of t_{OD} and t_{AD} PVT variation, AutoSync cannot ensure that the data that was sampled at the same instant at multiple, independent ADC devices will be present at the same time on DCLK.

7. Is RCLK the same frequency for both Demux and Non-Demux Mode?

Yes, for all devices except the ADC12D1600QML-SP. See Table 3 for more details on this device.

8. How is the DDR Clock Phase related to AutoSync? Should it be configured before or after AutoSync is configured?

All ADCs in the system should be configured in the same DDR Clock Phase, either 0° or 90°. Theoretically, it should make no difference to AutoSync if the DDR Phase is set before or after AutoSync is configured. Practically, it is more reasonable to configure the DDR Phase before configuring AutoSync.

9. Is the order of AutoSync setting configuration important for the Slave ADCs?

Yes, it is important for topologies in which the Slave ADCs receive their RCLK from an upstream ADC. For such topologies, the Slave ADCs should be configured starting from the Master ADC and progressing to each downstream Slave ADC in order. For the case where RCLK is externally driven, the order of Slave ADC AutoSync setting configuration is irrelevant.

10. Does calibration affect the AutoSync system?

No. Calibration affects the input impedance of the analog input and the sampling clock, and internal bias currents which affect the linearity of the device. AutoSync affects the timing of the DCLK and data. Calibrating before or after configuring AutoSync will not affect the timing of DCLK and data.

11. Does AutoSync work for the Hirel ADCs?

Yes. AutoSync is also available on the ADC10D1000QML and ADC12D1600QML.

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from F Revision (April 2013) to G Revision	Page
• Added row to table 3 to clarify low-sampling power-saving mode (LSPSM) for ADC12D1600QML-SP device	12
• Added ADC12D1600QML-SP device (with and without LSPSM) to table 4	14
• Added text to question 7 that clarifies use for ADC12D1600QML-SP device	15

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265