LTC6951 Synchronization Manual Design Examples for EZSync, ParallelSync, EZParallelSync and EZ204Sync

Chris Pearson

INTRODUCTION

For systems demanding a large number of synchronized clock signals, a clock tree using multiple clock devices is often required. Multiple clock devices can add system complexity when compared to a single clock distribution device. One complexity created is the ability to synchronize the clock phases and start times across multiple clock devices. The other challenge created is maintaining the desired low jitter clock performance when cascading multiple clock devices.

Linear Technology's family of PLL/VCO and clock distribution ICs address both the synchronization and performance concerns by providing three multichip synchronization options: EZSync™, ParallelSync™, and EZParallelSync™/EZ204Sync™. This application note provides step by step design examples for each multichip synchronization method.

Table 1 provides a summarized comparison of each synchronization method. Application Note 165 provides detailed descriptions and trade-offs of each synchronization method.

Table 1. Multichip Synchronization Comparison

	Architecture	Jitter	Sync Timing Requirements	Multichip Phase Alignment (All Outputs)
EZSync	Clock Distribution	Low	Easy	Yes, on First Edge*
ParallelSync	Reference Distribution	Ultralow	Moderate	Yes, outputs aligned to reference on First Edge* and have a known latency to sync signal falling edge
EZ204Sync or EZParallelSync	Reference Divide and Distribution	Ultralow	Easy	Yes, aligned to reference per each LTC6951 sync

*First Edge alignment implies all outputs requiring synchronization are phase aligned on the same sync event

TABLE OF CONTENTS

EZSync DESIGN EXAMPLE

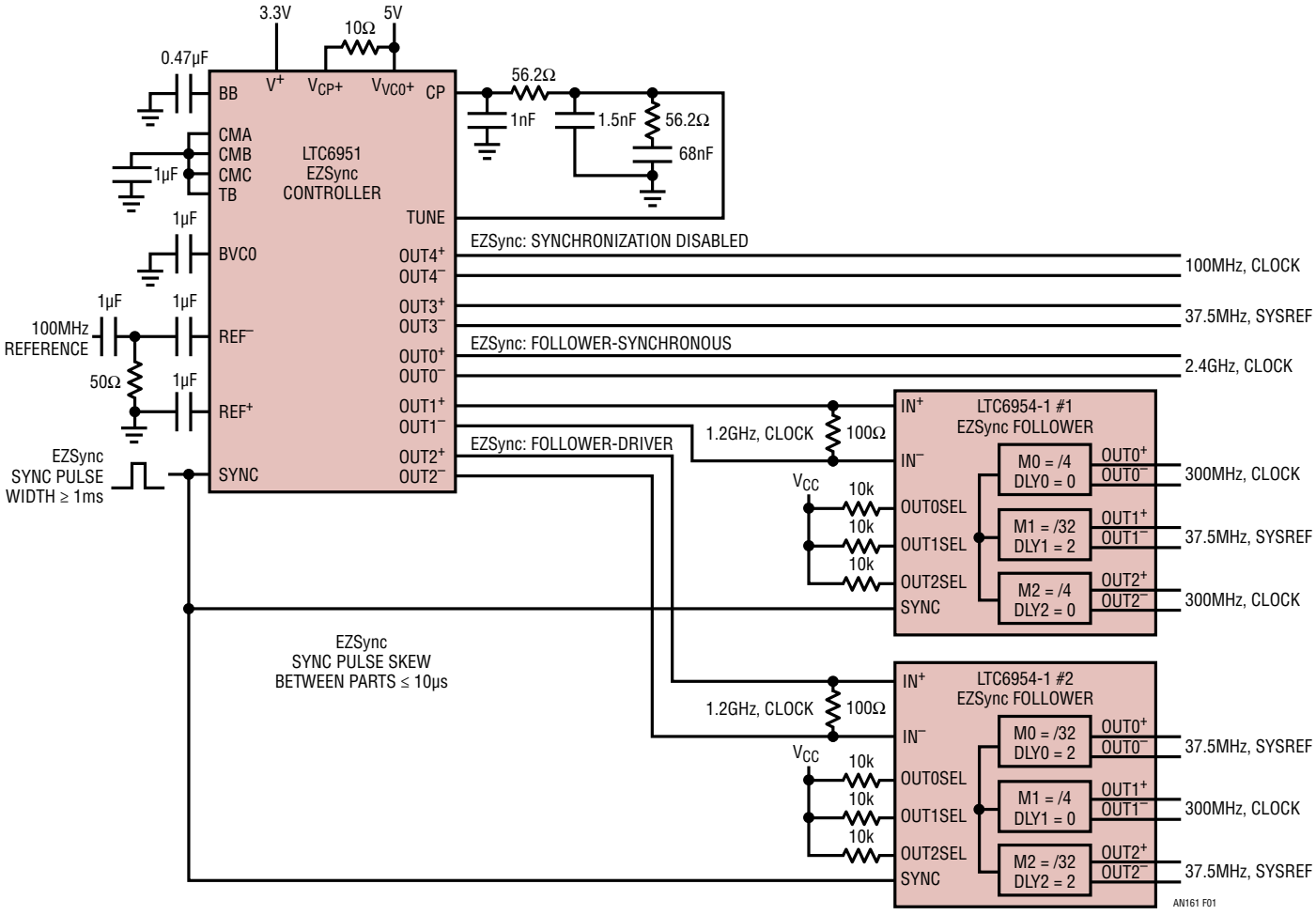


Figure 1. EZSync Design Example

EZSync Design Overview

EZSync is a simple way to generate synchronized clock outputs from multiple cascaded devices requiring only a simple logic signal or serial port interface (SPI) commands to achieve alignment.

Figure 1 introduces the following EZSync terminology: CONTROLLER, FOLLOWER, Follower-Driver, Follower-Synchronous, and Synchronization Disabled which are defined below:

CONTROLLER: EZSync device set to CONTROL mode. A device in CONTROL mode controls the timing for all other EZSync devices.

A CONTROLLER has three EZSync output types: Follower-Driver, Follower-Synchronous and Synchronization Disabled.

For the LTC®6951, to enable CONTROL mode set register bits SN = SR = 0.

FOLLOWER: EZSync device with at least one output set to FOLLOW mode. The FOLLOWER must be DC coupled to a Follower-Driver output.

During a SYNC event an output in FOLLOW mode is set to a logic low. Following a SYNC event, each FOLLOWER requires seven clocks cycles before the outputs in FOLLOW mode output a signal.

An LTC6954 output is set to FOLLOW mode when the respective LTC6954 SYNC_ENx register bit is set high.

Follower-Driver: CONTROLLER's clock output that is connected to a FOLLOWER's clock input. DC coupling is required between the CONTROLLER output and FOLLOWER input.

During a SYNC event, the Follower-Driver outputs are set to a logic low.

Follower-Synchronous: CONTROLLER's clock output that is synchronized to a FOLLOWER device's clock outputs.

During a SYNC event the Follower-Synchronous output is set to a logic low.

Following a SYNC event the Follower-Synchronous and FOLLOWER clock edges are aligned based on the delay settings in the CONTROLLER's and FOLLOWERs' register map.

Synchronization Disabled: Outputs that are not synchronized to Follower-Driver or Follower-Synchronous outputs. These outputs remain active during synchronization.

Synchronizing the LTC6951 outputs in Figure 1 involves sending a common sync signal that meets EZSync timing requirements. These are provided in Figure 1 and the LTC6951 EZSync Design Rules section.

The section titled LTC6951 EZSync Design Rules summarizes the EZSync design rules. The section titled EZSync Design Example section provides the design process used to develop the block diagram in Figure 1. The section titled Delay and Layout Recommendations provides how to minimize skew between parts by accounting for line length delays, FOLLOWER propagation delays and delta's in FOLLOWER and CONTROLLER rise and fall times. The section titled Synchronization Routines, provides initial power-up, power-down and resynchronization sequences. The Expandable Solution section discusses how the block diagram in Figure 1 can expand to support more clock outputs.

LTC6951 EZSync Design Rules

1. CONTROLLER set to CONTROL Mode. For LTC6951 register settings:
   a. SN = 0
   b. SR = 0
2. FOLLOWER outputs set to FOLLOW Mode. For LTC6954 set register setting SYNC_ENX = 1 for each output.
3. EZSync CONTROLLER to FOLLOWER Connection requires DC coupling.
4. EZSync timing requirements:
   a. Sync Pulse width > 1ms
   b. Sync Pulse skew between parts <10µs

EZSync Design Example

This design example will use the LTC6951Wizard™ to aid in the design process. Download LTC6951Wizard at http://www.linear.com/LTC6951Wizard.

This example assumes the following list of design inputs:

Reference	
f_{REF} = 100MHz	
LTC6951: EZSync CONTROLLER	
OUT0	2.4GHz Follower-Synchronous
OUT1	1.2GHz Follower-Driver Outputs
OUT2	1.2GHz Follower-Driver Outputs
OUT3	37.5MHz Follower-Synchronous
OUT4	100MHz Synchronization Disabled FPGA clock
LTC6954: EZSync FOLLOWERs	
$F_{6954\#1\text{-}OUT0}$ = 300MHz	
$F_{6954\#1\text{-}OUT1}$ = 37.5MHz	
$F_{6954\#1\text{-}OUT2}$ = 300MHz	
$F_{6954\#2\text{-}OUT0}$ = 37.5MHz	
$F_{6954\#2\text{-}OUT1}$ = 300MHz	
$F_{6954\#2\text{-}OUT2}$ = 37.5MHz	
Delay settings	Align the rising edge of LTC6951 and LTC6954 SYSREF signal to the falling edge of the clock signals.
Performance Optimization Request	
Design for low jitter.	
Minimize the output skew between devices.	

LTC6951 Setup

Based on the EZSync Design Rules and the above design inputs, the following steps provide input conditions for the LTC6951Wizard.

Step 1: Design input: optimize the LTC6951 charge pump current for low jitter.

Based on the LTC6951 data sheet the best jitter performance is obtained by maximizing the LTC6951 ICP current.

$I_{CLK6951.CP}$ = 11.2mA

Step 2: Design input: align LTC6951 SYSREF rising edge to the LTC6951 2.4GHz falling edge.

Assign the LTC6951 output invert bits as follows:

$OINV0_{6951}$ = 1 (OUT0 = 2.4GHz)

$OINV1_{6951}$ = 0 (OUT1 = SYSREF)

The section Delay and Layout Recommendations will discuss how to determine delay register settings to align LTC6951 Follower-Synchronous outputs and LTC6954 outputs. This example will assume the following inputs to the LTC6951Wizard tool

OUT0 Delay = 0*

OUT3 Delay = 7*

*These values are equivalent to Dx in Equation 2 under the section Delay and Layout Recommendations.

LTC6951Wizard

This section demonstrates the LTC6951Wizard's ability to ease the register setting creation and loop filter design for the LTC6951. Under the LTC6951Wizard's Help Menu a Help Guide is provided that will aid in understanding the operations performed in this section.

The values calculated in Steps 1 and 2 and conditions provided at the start of this design example are summarized below for a quick reference. These values will be used for inputs to the LTC6951Wizard to calculate the register settings and loop filter values for both LTC6951s in this design example.

LTC6951Wizard Inputs for Figure 2:

$f_{6951.REF}$ = 100MHz
$f_{6951.OUT0}$ = 2.4GHz
$f_{6951.OUT1}$ = 1.2GHz
$f_{6951.OUT2}$ = 1.2GHz
$f_{6951.OUT3}$ = 37.5MHz
$f_{6951.OUT4}$ = 100MHz
$I_{6951.CP}$ = 11.2mA
OUT0 Delay = 0
OUT3 Delay = 7

Figures 2 and 3 provide the remaining steps necessary to complete the LTC6951 portion of this design. Several steps in these Figures 2 and 3 require the following additional information.

Importing Reference Noise

Refer to Appendix: Model Reference Noise for LTC6951Wizard Simulations, which describes how to import reference noise into the LTC6951Wizard and the impact of reference noise on loop filter calculations and output noise simulations. The CCHD575_REFNOISE.txt file provided with the LTC6951Wizard should be used for this example.

Loop Filter Selection

Figure 3's step 10 selected Filter 2. Through experimentation Filter 2 was found to be the best option to optimize performance and board space.

1. Set I_{CP} = 11.2mA
2. Select Sync tab. See Steps 2a to 2d on far right
3. Set Fref = 100MHz
4. Set Invert OUT0 = Yes
5. Select All Select
6. Set Fout0 = 2400MHz Fout1= Fout2 = 1200MHz Fout3 = 37.5MHz Fout4 = 100MHz
7. Import Reference Noise File CCHD575-100M.txt (Appendix: Model Reference Noise for LTC6951Wizard Simulations)
8. Select Compute Params

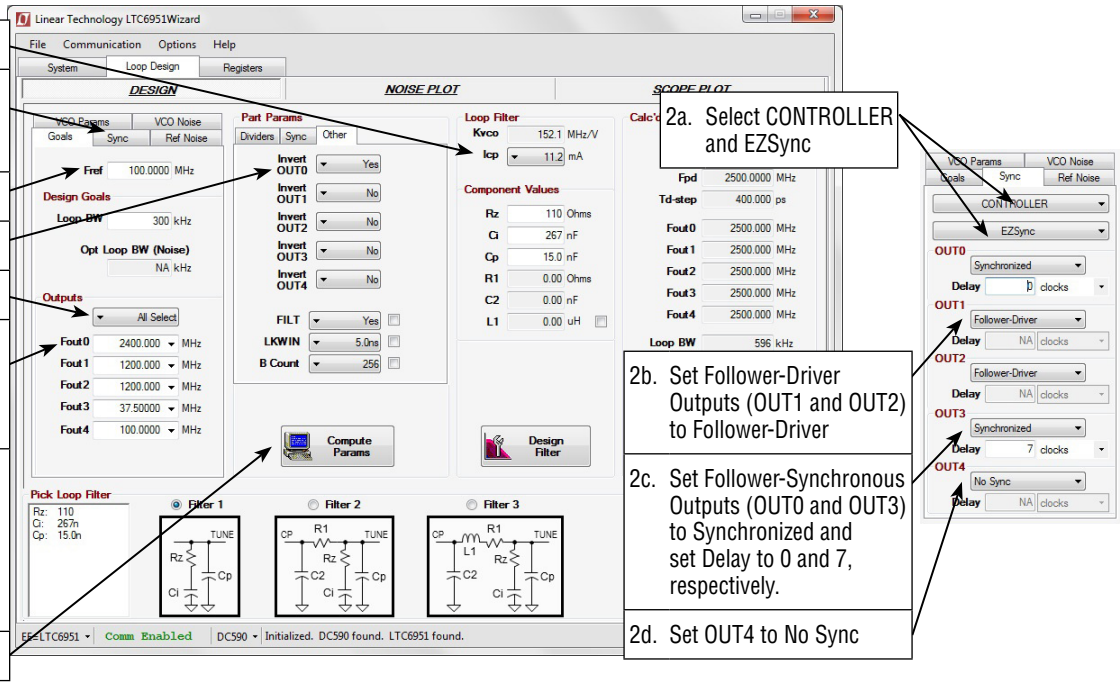


Figure 2. LTC6951Wizard Setup

9. Double click Opt Loop BW (Noise) to copy to Loop BW
10. Select Filter 2 and Design Filter, then set Component Values to closest standard component values.
11. Under File menu, select Save Settings.

    File name = EZSync (see far right)

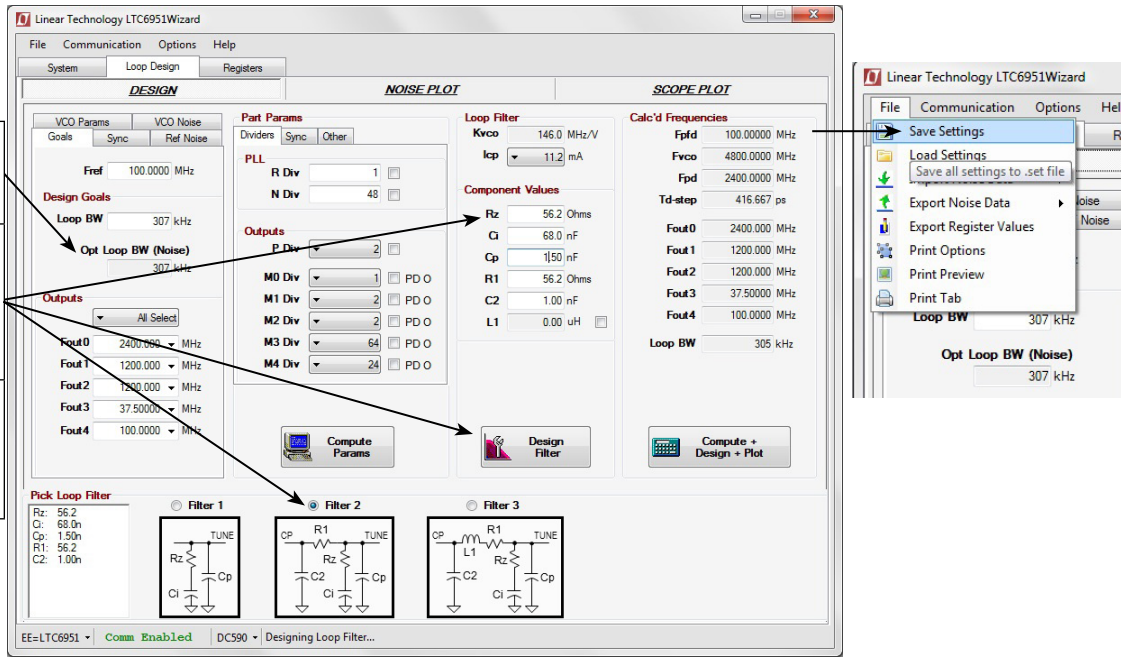


Figure 3. LTC6951Wizard Setup

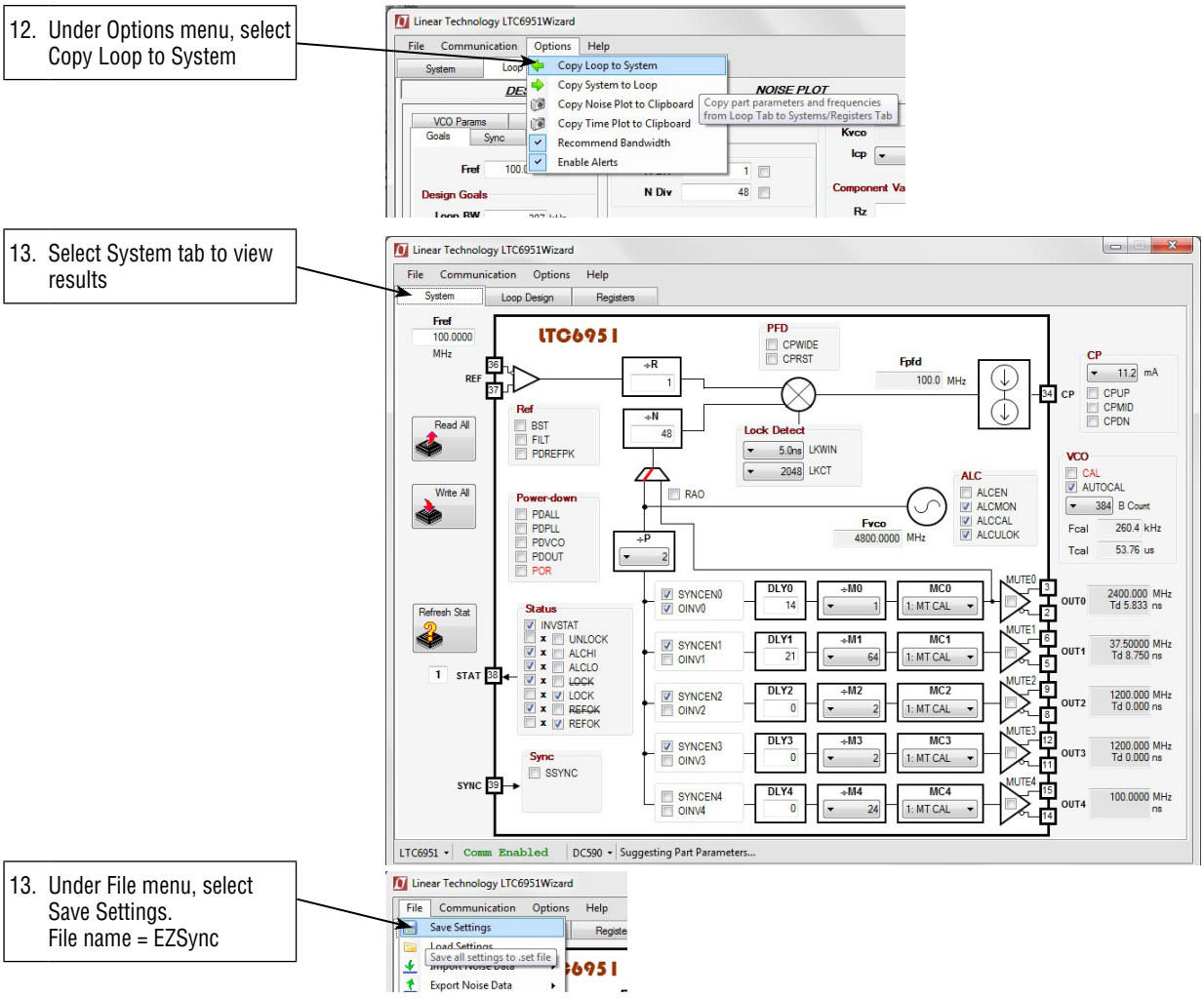


Figure 4. LTC6951Wizard Setup

LTC6954 Setup

Step 4: Design input: selecting LTC6954 FOLLOWER device for low jitter and minimal skew between devices.

The LTC6954 has four device options with various output signal types. The PECL output version, LTC6954-1, was selected over the other versions for the following reasons:

1. Lowest additive jitter
2. Smallest propagation delay variation over temperature

Step 5: Design input: align LTC6954 SYSREF rising edges to the LTC6954 300MHz falling edge.

The LTC6954 does not have an output invert bit. Therefore the LTC6954 input frequency was selected to allow the LTC6954 delay SPI register bits to perform an inversion.

With F_{IN6954} = 1.2GHz and the maximum LTC6954 output frequency = 300MHz, the LTC6954 delay settings can chose between 4 output phases:

$DLYX_{6954}$ = 0 → 0 degrees offset
$DLYX_{6954}$ = 1 → 90 degrees offset
$DLYX_{6954}$ = 2 → 180 degrees offset
$DLYX_{6954}$ = 3 → 270 degrees offset

Setting the 300MHz outputs to a delay of 0 degrees and the 37.5MHz outputs to delay of 180 degrees will properly align the LTC6954 outputs.

$DLY0_{6954\#1}$ = 0
$DLY1_{6954\#1}$ = 2
$DLY2_{6954\#1}$ = 0

$DLY0_{6954\#2} = 2$
$DLY1_{6954\#2} = 0$
$DLY2_{6954\#2} = 2$

Step 6: Design input: minimize the output skew performance between the LTC6954-1 #1 and LTC6954-1 #2.

Per the LTC6954 data sheet output to output skew is best when all LTC6954 divider values are either equal to /1 or all divider values are greater than /1. In this example the LTC6954 input frequency was chosen such that all LTC6954 dividers are greater than /1.

Step 7: LTC6954 settings summarized:

Register settings:

$SYNCENX_{6954} = 1$
$PDIVX_{6954} = 0$
$PDOUTX_{6954} = 0$

$M0_{6954\#1} = 4$
$DEL0_{6954\#1} = 0$
$M1_{6954\#1} = 32$
$DEL1_{6954\#1} = 2$
$M2_{6954\#1} = 4$
$DEL2_{6954\#1} = 0$

$M0_{6954\#2} = 32$
$DEL0_{6954\#2} = 2$
$M1_{6954\#2} = 4$
$DEL1_{6954\#2} = 0$
$M2_{6954\#2} = 32$
$DEL2_{6954\#2} = 2$

Delay and Layout Recommendations

Minimizing output skew between an LTC6951 EZSync CONTROLLER output and an EZSync FOLLOWER output can be performed by solving Equations 2 to 6 (refer to Figure 5).

Equations 1 and 2 align the starting edges of the CONTROLLER's Follower-Synchronous output and Follower's outputs. Equation 1 is provided in the LTC6951 data sheet and shown below for consistency. Equation 2 expands upon Equation 1 by translating the trace lengths, FOLLOWER propagation delay and rise time to the nearest LTC6951 P-divider cycle delay value.

CONTROLLER Follower-Synchronous

$$D_{FSX} = Dx + M_{FDY} \bullet 7 \quad (1)$$

$$D_{FSX} = Dx + M_{FDY} \bullet 7 + \quad (2)$$

$$\text{int}\left(\left(\begin{array}{l}(d2 + d3 - d1)\\ +t_{pd(FOLLOWER)}\\ +t_{r(FOLLOWER)}/2\end{array}\right) \bullet f_{PD} + 0.5\right)$$

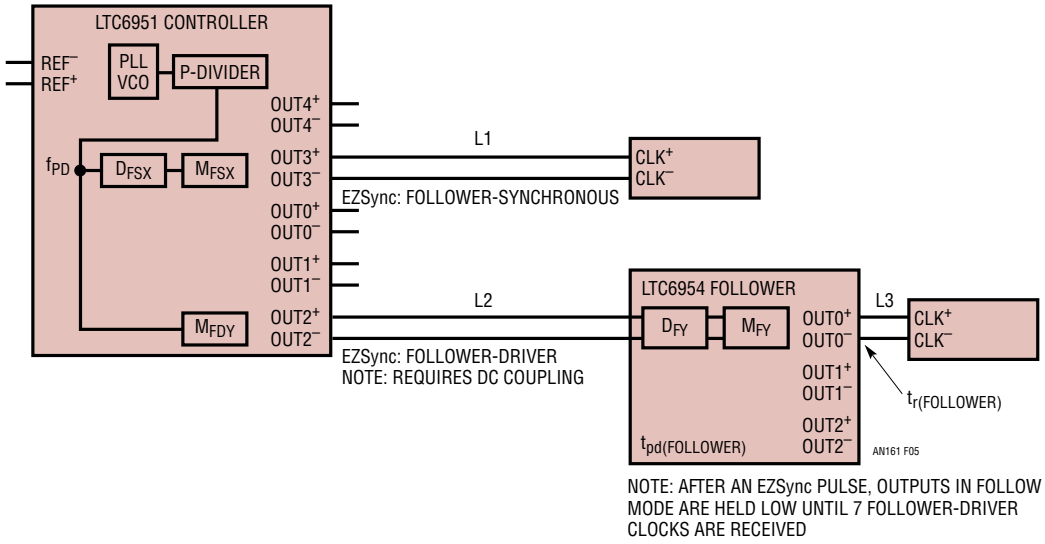


Figure 5. Output Skew

Equations 3 and 4 convert the board's trace length to a signal delay.

Trace Length

$$dx = LX/Vp \quad (3)$$

$$Vp = \frac{c}{\sqrt{\varepsilon r}} \quad (4)$$

Equation 5 calculates the FOLLOWER's delay setting in terms of the CONTROLLER delay settings. Equation 6 calculates the desired delay delta between a Follower-Synchronous output and a FOLLOWER output.

FOLLOWER Output

$$Dy = D_{FY} \bullet M_{FDY} \quad (5)$$

$$D_{Delta(FStoFOLLOWER)} = Dx - Dy \quad (6)$$

where:

c: speed of light (m/s)

D_{FSX}: LTC6951 Follower-Synchronous delay (s)

D_{FY}: FOLLOWER delay (s)

Dx: desired delay of Follower-Synchronous output with respect to a FOLLOWER output when Dy = 0. Dx is the delay value input for the LTC6951Wizard tool in Figure 2.

Dy: desired delay of FOLLOWER output with respect to Follower-Synchronous output when Dx = 0.

$D_{Delta(FStoFOLLOWER)}$: desired delta delay between CONTROLLER and FOLLOWER outputs, in terms of the CONTROLLER delay settings.

dX: signal delay, electrical trace length(s)

f_{PD}: LTC6951 P-Divider output frequency

LX: trace length (m)

M_{FDY}: LTC6951 Follower-Driver divide value

M_{FSY}: LTC6951 Follower-Synchronous divide value

M_{FY}: FOLLOWER divide value

$t_{pd(FOLLOWER)}$: EZSync FOLLOWER propagation delay (s)

$t_{r(FOLLOWER)}$: EZSync FOLLOWER rise time

Vp: velocity of propagation (m/s)

εr: board material dielectric constant (relative permittivity)

Synchronization Routines

On initial power-up:

1. Program LTC6954 and LTC6951 SPI registers
2. Wait for LTC6951 bias voltages to stabilize
3. Calibrate LTC6951 VCO
4. Wait for VCO calibration to complete
5. Send EZSync pulse

Power-down:

1. Power down LTC6951 (PDALL = 1)
2. Power down LTC6954 (PDALL = 1)

Resynchronization (post power-down):

1. Power up LTC6954 (PDALL = 0)
2. Power up LTC6951 (PDALL = 0)
3. Send EZSync pulse

Expandable Solution

EZSync solutions are infinitely expandable. As shown in Figure 6 the EZSync design example can be expanded by adding fanout buffers to distribute additional clocks. The remainder of this section provides considerations when selecting a fanout buffer, the fanout buffer register settings in Figure 6 and comments regarding further expansion.

The fanout buffer in Figure 6 does not require an EZSync pulse. Therefore the fanout buffer does not need to be an EZSync device. However, the fanout buffer must be capable of accepting a DC-coupled input from an EZSync CONTROLLER and driving a DC-coupled input to an EZSync FOLLOWER.

Each stage of a clock distribution architectures produces additive jitter. Referring to Equation 7, the addition of the fanout buffer will increase the total jitter.

$$Jitter_{Total} = \sqrt{\begin{matrix} Jitter_{CONTROLLER}{}^2 + Jitter_{FOLLOWER}{}^2 \\ +Jitter_{FANOUT-BUFFER}{}^2 \end{matrix}} \quad (7)$$

The fanout buffers propagation delay, output rise time and trace lengths to the fanout buffer will need to be included in the delay calculations. Refer to Equation 2 delay calculations in the Delay and Layout Recommendations section.

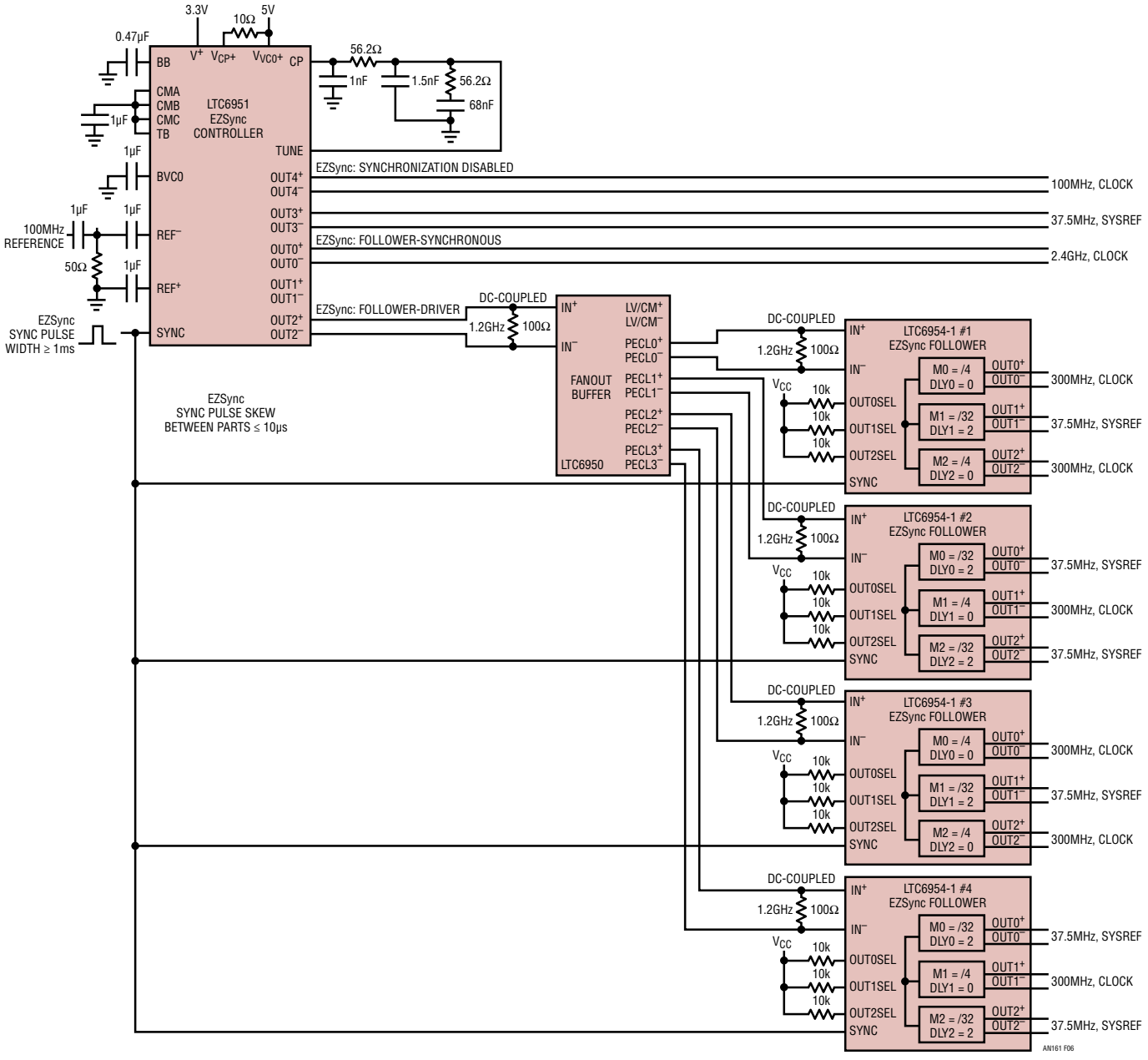


Figure 6. EZSync Expandable Solution

The LTC6950 in Stage 1 is set to distribution only mode, by powering down the PLL circuitry (PDPLL = 1) and connecting the Follower-Driver Signal to the LTC6950 VCO input. Below is a summary of the LTC6950 register settings for Figure 6:

SM1[5] = SM2[5] = 0x20
PDPLL = 1
IBIAS0 = IBIAS1 = IBIAS2 = IBIAS3 = 1
M0 = M1 = M2 = M3 = M4 = 1
PD_DIV4 = 1

All other registers setting can be set to 0.

For further expansion it is possible to choose larger fanout buffers or add additional fanout buffer stages. When designing a multi-stage clock distribution network, take into account the additive properties of

- channel to channel skew
- additive jitter (Equation 7)

ParallelSync DESIGN EXAMPLE

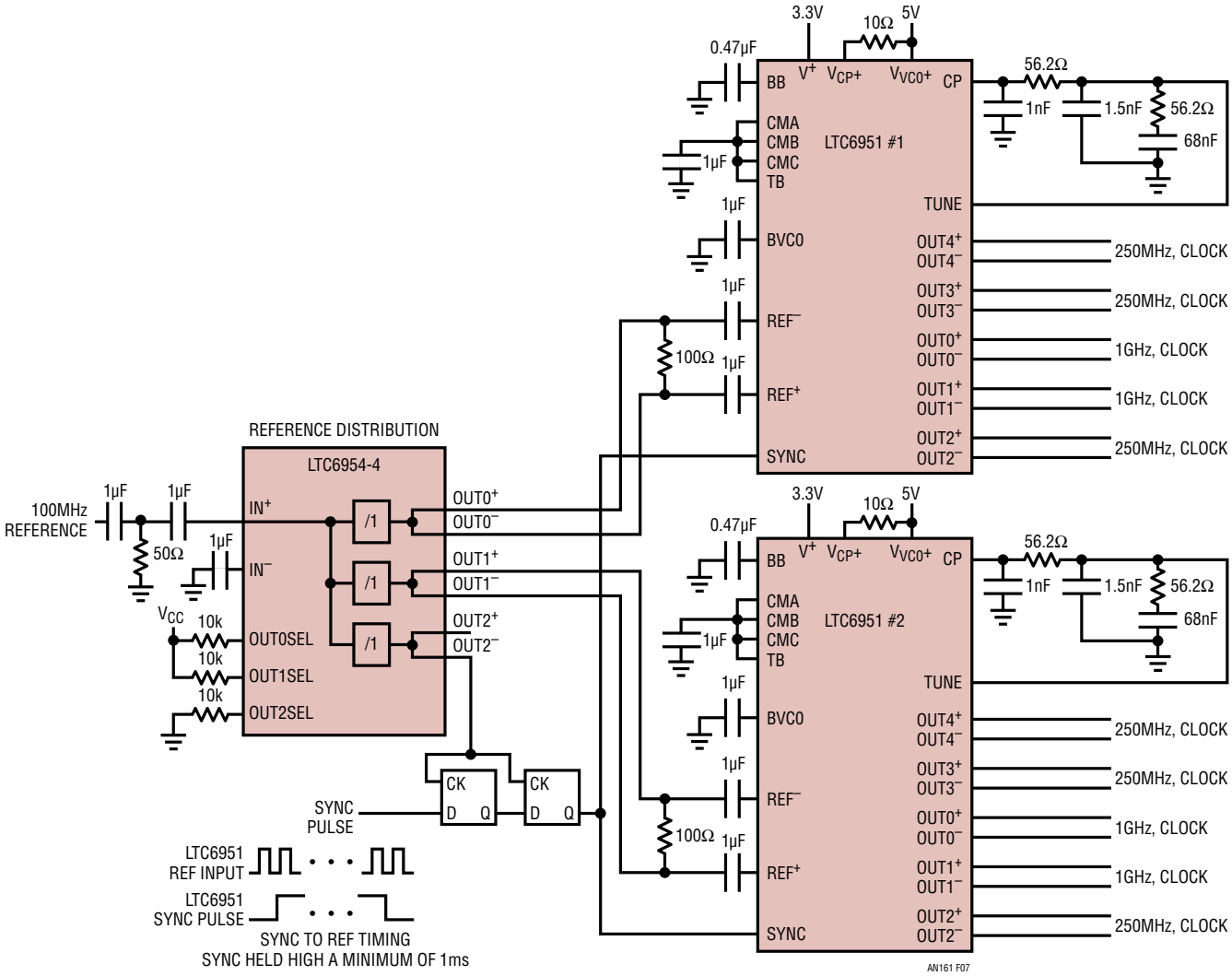


Figure 7. ParallelSync Design Example 1 – Two LTC6951s

ParallelSync Design Overview

ParallelSync is a method to synchronize multiple LTC6951's running in parallel driven by a common reference clock fanout buffer network. Synchronization is achieved through a common reference aligned sync signal.

Synchronizing the LTC6951 outputs in Figure 7 involves sending a common sync signal that meets setup and hold time requirements to the common reference signal. This architecture provides the ability to synchronize all LTC6951 outputs with a known latency to the sync signals falling edge. The ability to synchronize all LTC6951 outputs at a known time is useful in systems that require known and precise initial placement of clock edges.

In Figure 7, the LTC6954-4 is the common reference clock fanout network. The LTC6954-4 was selected to configure OUT0 and OUT1 as LVDS signals to the LTC6951 reference inputs. The LTC6954 LVDS outputs optimize power consumption and LTC9651 performance when compared to the LTC6954's PECL and CMOS options. In addition, OUT2 can be configured as a CMOS signal to drive the D flip-flop circuitry. The LTC6954 OUT2- CMOS output was selected, instead of the OUT2+ CMOS output, because OUT2- can be inverted which adds some SYNC to REF timing flexibility.

The section titled ParallelSync Design Rules summarizes the ParallelSync design rules. The section titled ParallelSync Design Example 1 section provides the design process used to develop the block diagram in Figure 7. For a comparison of synchronization methods, this example mirrors the frequency plan of the EZParallelSync Design Example. Layout Recommendations discusses matching line lengths to minimize skew between parts. The section titled Synchronization Routines, provides initial power-up, power-down and resynchronization sequences. The Expandable Solution section discusses how the block diagram in Figure 7 can expand to support more LTC6951 devices. ParallelSync Design example 2 provides the DC2226 JESD204B frequency plan. The DC2226 is a demo board that includes the LTC6951 and two JESD ADCs (LTC2123).

LTC6951 ParallelSync Design Rules

1. LTC6951 register settings:
   a. RAO = 1 (enabled)
   b. SN = 1
   c. SR = 1
2. Meet LTC6951 data sheet SYNC to REF setup and hold times.

ParallelSync Design Example 1

This design example will use the LTC6951Wizard to aid in the design process. Download LTC6951Wizard at http://www.linear.com/LTC6951Wizard.

This example assumes the following list of design inputs.

Reference	
f_{REF} = 100MHz	
LTC6951s	
Four 1GHz clock signals	
Six 250MHz clock signals	
RDIV = 1	
RAO = 1	
Delay settings	Align LTC6951 outputs rising edge to LTC6951 reference input rising edge.
Performance Optimization Request	
Design for low jitter.	
Minimize the output skew between the LTC6951#1 and LTC6952#2	

LTC6951 Setup

Based on the ParallelSync Design Rules and the above design inputs the following steps provide input conditions for the LTC6951Wizard.

Step 1: Design input: assign output frequencies to optimize the LTC6951 f_{PFD} for low jitter.

ParallelSync's Design Rule 1 sets RAO = 1, making OUT0 part of the PLL feedback loop. As a result OUT0 affects the LTC6951 PLL's PFD frequency (f_{PFD}). Maximizing the LTC6951 f_{PFD} allows for a wider loop bandwidth and as a result optimal jitter performance. For more details, refer to the LTC6951 data sheet sections Reference Source Considerations and In-Band Output Phase Noise. The LTC6951 maximum f_{PFD} frequency is 100MHz.

Referring to the LTC6951 data sheet, Equations 8 and 9 can be derived when RAO_{6951} = 1.

$$f_{6951.PFD} = \frac{f_{6951.OUT0}}{NDIV_{6951}} \quad (8)$$

$$f_{6951.PFD} = \frac{f_{REF}}{RDIV_{6951}} \quad (9)$$

Equations 8 and 9 can be rearranged as follows:

$$\frac{f_{6951.OUT0}}{f_{REF}} = \frac{NDIV_{6951}}{RDIV_{6951}} \quad (10)$$

Substituting the desired output clock frequencies and known f_{REF} = 100MHz into Equation 45, determine the least common multiple for NDIV and RDIV. Then use Equations 8 and 9 to determine f_{PFD}.

If $f_{6951.OUT0}$ = 250MHz:

250MHz/100MHz = $NDIV_{6951}/RDIV_{6951}$

$NDIV_{6951}$ = 5

$RDIV_{6951}$ = 2

f_{PFD} = 50MHz

If $f_{6951.OUT0}$ = 1GHz:

1GHz/100MHz = $NDIV_{6951}/RDIV_{6951}$

$NDIV_{6951}$ = 10

$RDIV_{6951}$ = 1

f_{PFD} = 100MHz

Assigning $f_{6951.OUT0}$ to 1GHz allows for the largest f_{PFD} and as a result minimizes the LTC6951 output jitter.

OUT4 is an LVDS output with an 800MHz maximum output frequency and higher jitter than the other LTC6951 CML outputs. The rest of the output frequencies can be assigned as desired.

$f_{6951\#1.OUT0} = f_{6951\#2.OUT0}$ = 1GHz
$f_{6951\#1.OUT1} = f_{6951\#2.OUT1}$ = 1GHz
$f_{6951\#1.OUT2} = f_{6951\#2.OUT2}$ = 250MHz
$f_{6951\#1.OUT3} = f_{6951\#2.OUT3}$ = 250MHz
$f_{6951\#1.OUT4} = f_{6951\#2.OUT4}$ = 250MHz

Note: The EZParallelSync Design Example uses the same output frequencies as this ParallelSync Design Example 1. However, because EZParallelSync Design Rule 1 limits OUT0's frequency selection a smaller f_{PFD} (50MHz) was used. The smaller f_{PFD} resulted in the EZParallelSync example having ~10fs degraded jitter performance when compared to this example. The frequencies in these two examples were chosen specifically to highlight this difference. Depending on the desired reference and output frequencies, differences in f_{PFD} between these two synchronization methods may or may not result. This note is directed at the LTC6951, as other Linear Technology PLL/VCOs may not have the LTC6951's pre-scalar divider architecture. As a result the LTC6951 EZParallelSync Design rule #1 may not apply to other PLL/VCOs.

Step 2: Design input: optimize the LTC6951 charge pump current for low jitter.

Based on the LTC6951 data sheet the best jitter performance is obtained by maximizing the LTC6951 ICP current.

$I_{CLK6951.CP}$ = 11.2mA

Step 3: Design input: minimize the output skew performance between the LTC6951#1 and LTC6952#2.

The LTC6951 device to device skew is best when the LTC6951 register value FILT = 0.

LTC6951Wizard

This section demonstrates the LTC6951Wizard's ability to ease the register setting creation and loop filter design for the LTC6951. Under the LTC6951Wizard's Help Menu a Help Guide is provided that will aid in understanding the operations performed in this section.

The values calculated in Steps 1-3 and conditions provided at the start of this design example are summarized below for a quick reference. These values will be used for inputs to the LTC6951Wizard to calculate the register settings and loop filter values for both LTC6951s in this design example.

LTC6951Wizard inputs for Figure 9:

$f_{6951.REF}$ = 100MHz
$f_{6951.OUT0}$ = 1GHz
$f_{6951.OUT1}$ = 1GHz
$f_{6951.OUT2}$ = 250MHz
$f_{6951.OUT3}$ = 250MHz
$f_{6951.OUT4}$ = 250MHz
$I_{6951.CP}$ = 11.2mA
$NDIV_{6951}$ = 10
$RDIV_{6951}$ = 1
$FILT_{6951}$ = 0
RAO_{6951} = 1

Figures 9 and 10 provide the remaining steps necessary to complete the LTC6951 portion of this design. Several steps in these Figures 9 and 10 require the following additional information.

Importing Reference Noise

Refer to Appendix: Model Reference Noise for LTC6951Wizard Simulations, which describes how to import reference noise into the LTC6951Wizard and the impact of reference noise on loop filter calculations and output noise simulations. Example 1 in the appendix creates the reference noise profile for this example.

Delay setting: DLYX BITS

For this example, the request was made to align the rising edge of the LTC6951 outputs with the rising edge of the LTC6951 reference input. The LTC6951 Wizard automatically calculates the DLYX bits based off of Equation 11.

Figure 9, step 2b sets the Delay value = 0. A LTC6951Wizazd Delay value = 0 forces the LTC6951Wizard to calculate the LTC6951 DLYX settings to align the LTC6951 output and reference input rising edges. Figure 10 shows the DLYX bits = 2 based off the wizard calculation. Figure 11 shows that the LTC6951 output and reference inputs rising edges are aligned.

Solving Equation 11 for Dx (DLYX) from the values provided below match the LTC6951Wizard Delay results in Figures 10 and 11.

$$Dx = Dxi + CEILING\left(\frac{18}{N \bullet M0}\right) \bullet N \bullet M0 - 18 \qquad (11)$$

Dxi = 0 (aligns to reference)

18, number of PDIV cycles

N = 5

M0 = 8

$$Dx = 0 + CEILING\left(\frac{18}{10 \bullet 2}\right) \bullet 2 \bullet 10 - 18$$

Dx = 2 (DLYX delay settings)

SYNCENX Bits

Ensure the LTC6951 SYNCENX bits are set to a 1 for all signals that require synchronization. Refer to Figure 9, Step 2b.

LTC6951 OINV Bit

Figure 9, step 6 sets the OINVX values for each output. Figure 8 provides a recommendation for OINVX settings based on schematic connections to the device being clock. In this example all LTC6951 outputs will use the Standard OUTX Connection, setting OINVX = 0 (not inverted).

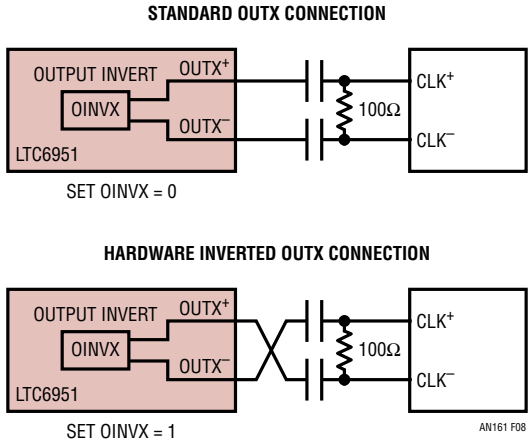


Figure 8. LTC6951 OINVX STATE

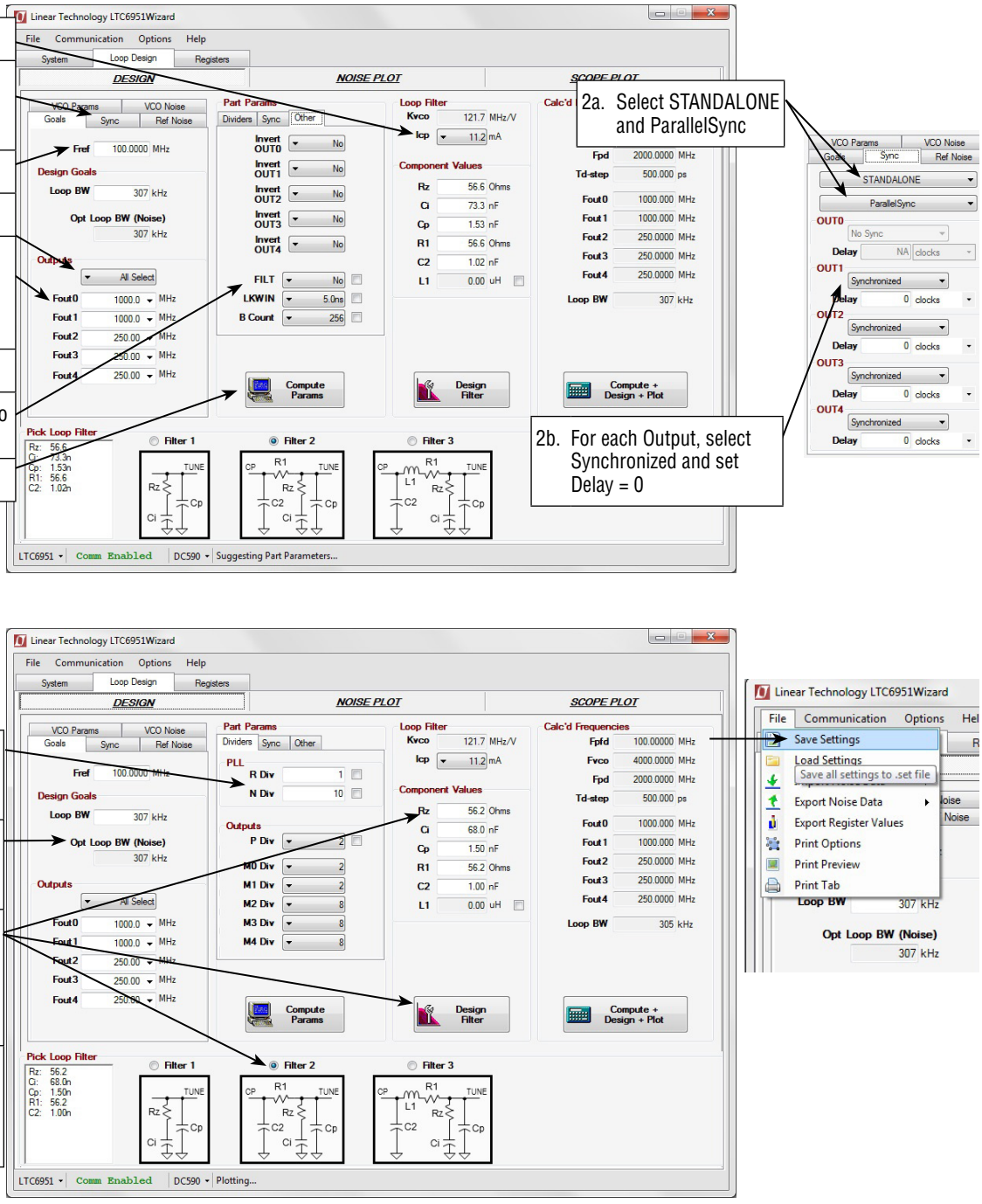
Loop Filter Selection

Figure 9's step 11 selected Filter 2. Through experimentation Filter 2 was found to be the best option to optimize performance and board space.

1. Set I_{CP} = 11.2mA
2. Select Sync tab. See Steps 2a and 2b on far right
3. Set Fref = 100MHz
4. Select All Select
5. Set Fout0 = Fout1 = 1000MHz Set Fout2 = Fout3 = Fout4 = 250MHz
6. Set Invert OUTx = No
7. Set FILT = No, check box to lock value
8. Select Compute Params

2a. Select STANDALONE and ParallelSync

2b. For each Output, select Synchronized and set Delay = 0

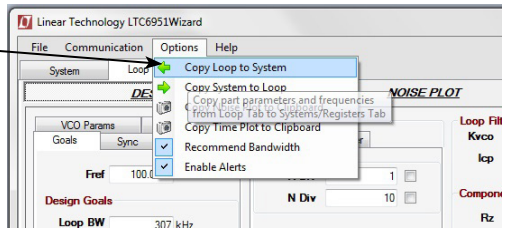


9. Verify R Div = 1 and N Div matches previous calculation
10. Double click Opt Loop BW (Noise) to copy to Loop BW
11. Select Filter 2 and Design Filter, then set Component Values to closest standard component values.
12. Under File menu, select Save Settings.

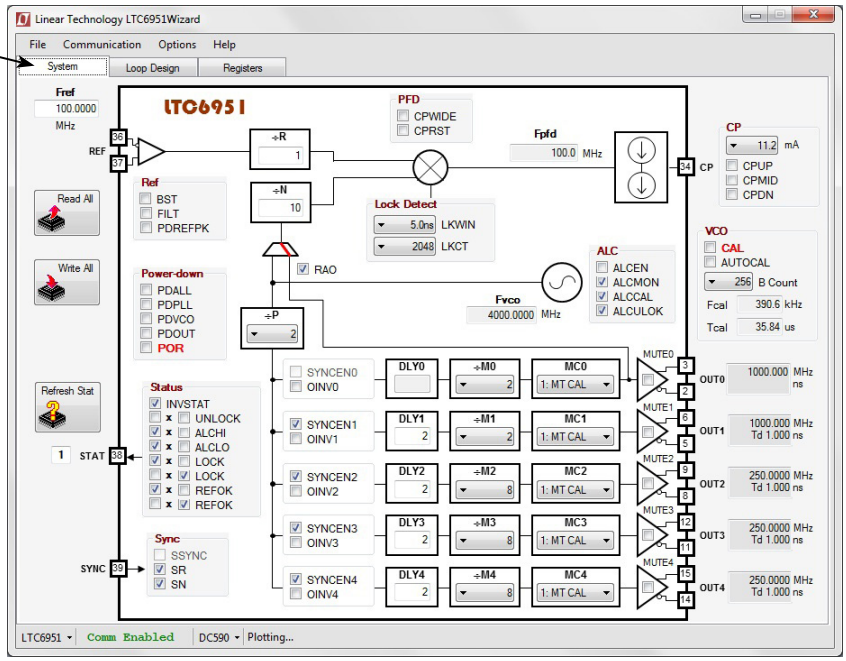
    File name = ParallelSync (see far right)

Figure 9. LTC6951Wizard Setup

13. Under Options menu, select Copy Loop to System



14. Select System tab to view results



15. Under File menu, select Save Settings.
File name = ParallelSync

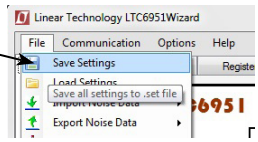


Figure 10. LTC6951Wizard Setup Continued

16. Select Loop Design

17. Select Scope Plot

18. Select Plot

1 RDIV Cycle = M0 • N • R PDIV Cycles
M0=2, N=10, R = 1
1 RDIV Cycle = 20 PDIV Cycles

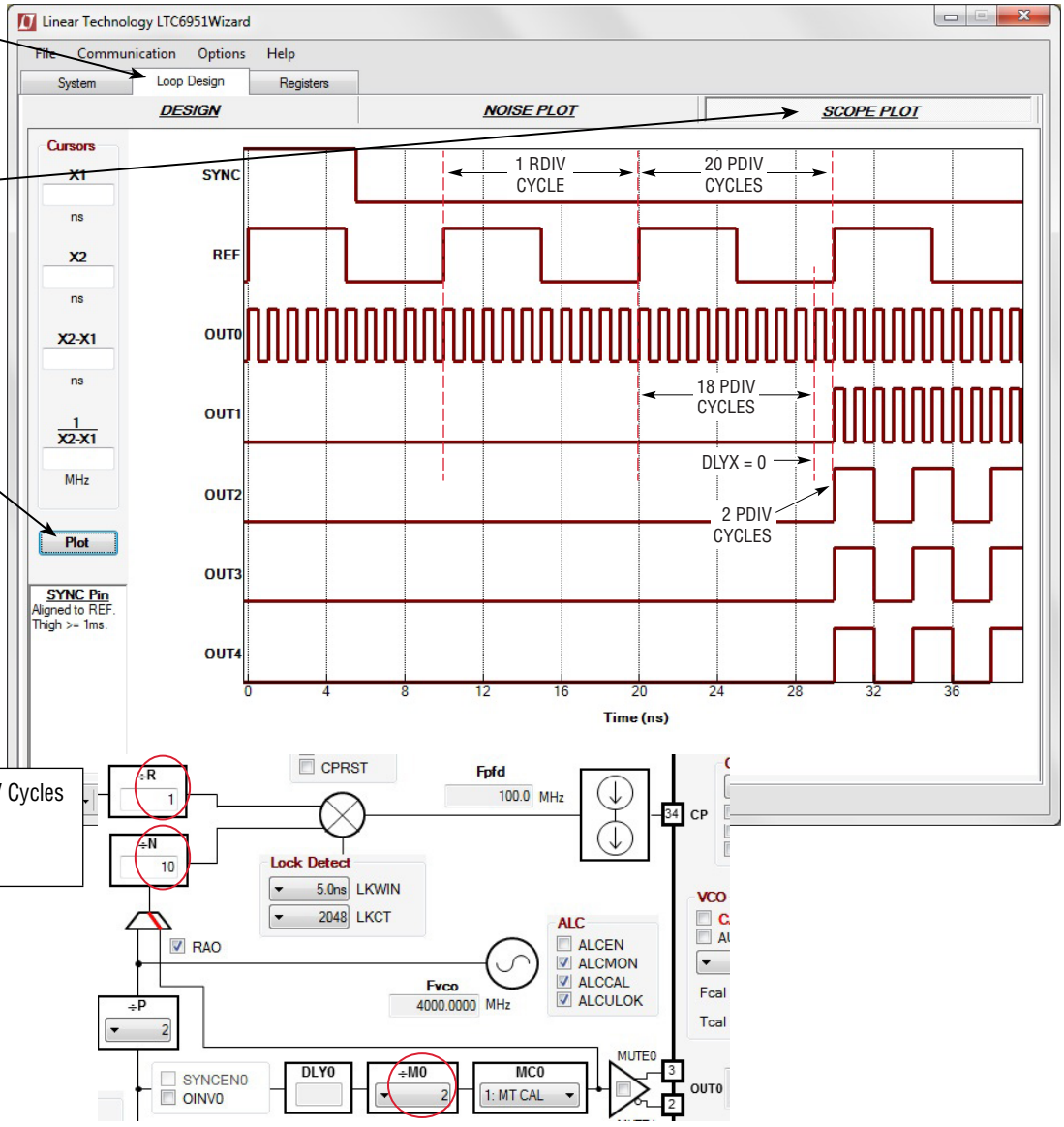


Figure 11. LTC6951Wizard Delay Settings

LTC6954 Setup

Step 4: Design input: minimize the output skew performance between the LTC6951#1 and LTC6952#2.

Skew in reference signals will result in skew between LTC6951s. Therefore, it is recommended to design the reference distribution device for low skew and match trace lengths on the reference signals during board layout.

With ParallelSync, the LTC6951 outputs are phase aligned to the LTC6951's internal reference divider's output. As a result, a fanout buffer can be used for reference distribution. In this example the LTC6954 was chosen for the fanout buffer with dividers set to 1.

$M0_{6954} = 1$

$M1_{6954} = 1$

$M2_{6954} = 1$

According to the LTC6954 data sheet, best skew performance is obtained when either one of following two conditions are met:

- Condition 1: all LTC6954 output divider settings equal 1
- Condition 2: all LTC6954 output divider settings are >1.

Step 5: Verify LTC6954 output to LTC6951 connection.

It is required to choose an identical reference schematic from Figure 12 for both LTC6951s. This ensures both LTC6951 PLLs align to the same reference edge.

For this example, both LTC6951 reference inputs can use Figure 12's Hardware Inverted Reference Connection with a LTC6954 divide value equal to 1. Figure 12's Sync to Ref Timing Circuit should use a divide value of 1. It is also recommended to use the LTC6954 OUTX- CMOS output, which has an output invert bit, for the Sync to Ref Timing Circuit.

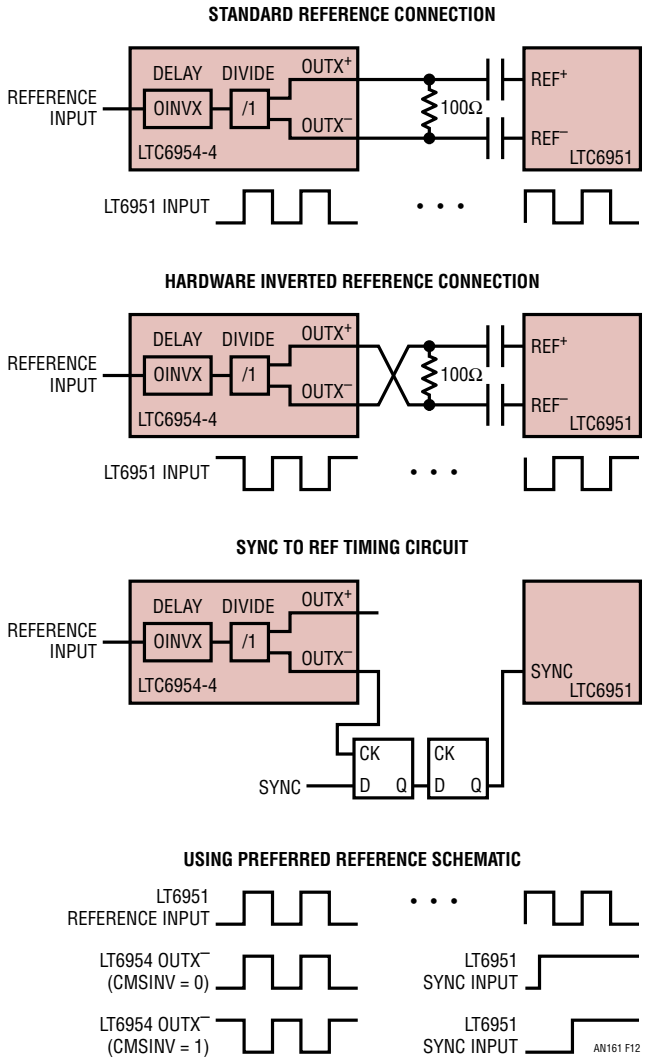


Figure 12. Reference Distribution Connection

Step 6: LTC6954 settings summarized:

OUTXSEL Pin Settings
OUT0SEL = V_{CC} (LVDS)
OUT1SEL = V_{CC} (LVDS)
OUT2SEL = GND (CMOS)

Register Settings

When all LTC6954 divide values equal 1, the LTC6954 DLYX and SYNCEN settings have no effect phase relationship and can be set to any value.

$SYNCEN0_{6954} = 1$
$M0_{6954} = 1$
$DEL0_{6954} = 0$
$PDIV0_{6954} = 0$
$PDOUT0_{6954} = 0$

$SYNCEN1_{6954} = 1$
$M2_{6954} = 1$
$DEL1_{6954} = 0$
$PDIV1_{6954} = 0$
$PDOUT1_{6954} = 0$

$SYNCEN2_{6954} = 1$
$M2_{6954} = 1$
$DEL2_{6954} = 0$
$PDIV2_{6954} = 0$
$PDOUT2_{6954} = 0$
$CMSINV2_{6954} = 0$

Layout Recommendations

To minimize LTC6951 output skew match electrical trace lengths as shown in Equations 12 and 13 (refer to Figure 13).

$$L_{REF\#1} = L_{REF\#2} \quad (12)$$

$$L_{\#1.OUTX} = L_{\#2.OUTX} \quad (13)$$

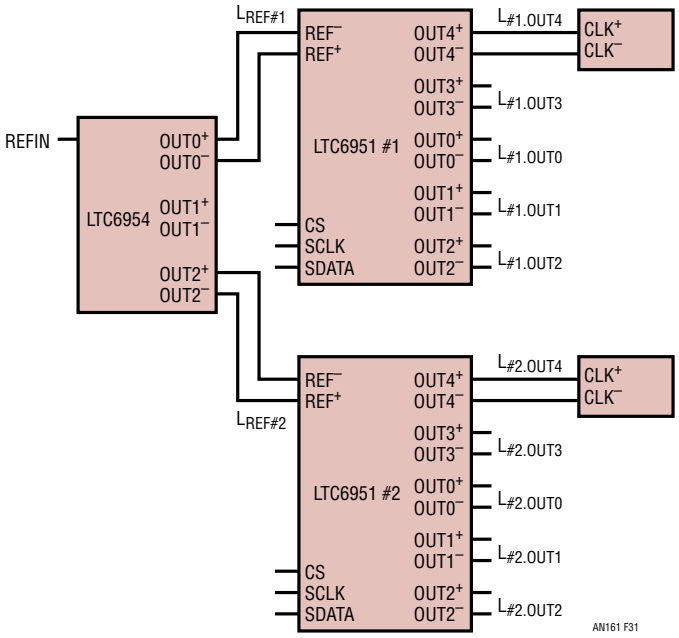


Figure 13. Trace Length Matching

Synchronization Routines

On initial power-up:

1. Program LTC6954 and LTC6951 SPI registers
2. If $MX_{6954} > 1$, Toggle LTC6954 SYNC pin (minimum 1ms)
3. Wait for LTC6951 bias voltages to stabilize
4. Calibrate all LTC6951 VCOs
5. Send SYNC pulse to LTC6951 Sync pins (see LTC6951 SYNC Pulse Width section)

Power-down:

1. Power down LTC6951 (PDALL = 1)
2. Power down LTC6954 (PDALL = 1)

Resynchronization(post power-down):

1. Power Up LTC6954 (PDALL = 0)
2. Power Up LTC6951 (PDALL = 0)
3. If $MX_{6954} > 1$, Toggle LTC6954 SYNC pin (minimum 1ms)
4. Send SYNC pulse to LTC6951 Sync pins (see LTC6951 SYNC Pulse Width section)

LTC6951 Sync Pulse Width

The requirements for the sync pulse width depend on the LTC6951 Rdivider setting. When the LTC6951 internal reference divider equals 1, the latency from the reference input to any output will be consistent. In this configuration the sync pulse width should be greater than 1ms.

When R = 1

$$\text{Sync Pulse Width} > 1\text{ms} \quad (12)$$

When the LTC6951 internal reference divider is > 1, the latency from the reference input to any output has R different possibilities depending on where SYNC falls relative to R DIV. By creating a SYNC pulse exactly REFCYCLES wide, all outputs will begin with the same latency to the reference input every time a synchronization event occurs. Equations 14 and 15 calculate the SYNC pulse width that allows for consistent latency, when R > 1.

When R>1

$$\text{REFCYCLES} = R \bullet \text{Ceiling}(1\text{ms} \bullet f_{REF/R}) + 1 \quad (14)$$

$$\text{Sync Pulse Width} = \text{REFCYCLES}/f_{REF} \quad (15)$$

Refer to the LTC6951 data sheet for more details.

Expandable Solution

The ParallelSync solution is infinitely expandable. As shown in Figure 14 the ParallelSync design example 1 can be repeated by adding an additional fanout buffer to distribute the reference.

Distributing a reference aligned synchronization signal in a multi-stage fanout architecture across multiple daughter cards is an additional challenge with the ParallelSync architecture. Each stage in the reference fanout network has a propagation delay that should be accounted for. Figure 14 accounts for propagation delays by retiming the sync signal in both reference distribution stages.

The LTC6950 in Stage 1 is set to distribution only mode, by powering down the PLL circuitry (PDPLL = 1) and connecting the reference to the LTC6950 VCO input. It is also recommended to use the LTC6950 LVCM– CMOS output, which has an output invert bit, for the Sync to Ref timing circuit.

Below is a summary of the LTC6950 register settings for Figure 14:

SM1[5] = SM2[5] = 0x20
PDPLL = 1
IBIAS0 = IBIAS1 = IBIAS2 = IBIAS3 = 1
M0 = M1 = M2 = M3 = M4 = 1

All other registers setting can be set to 0.

For further expansion it is possible to choose larger fanout buffers in Stages 1 or 2 and/or cascade additional reference distribution stages between Stage 1 and Stage 2. When designing a multi-stage reference distribution network, take into account the additive properties of

- channel to channel skew
- noise floor at frequency offsets less than the LTC6951 loop filter's pass-band.

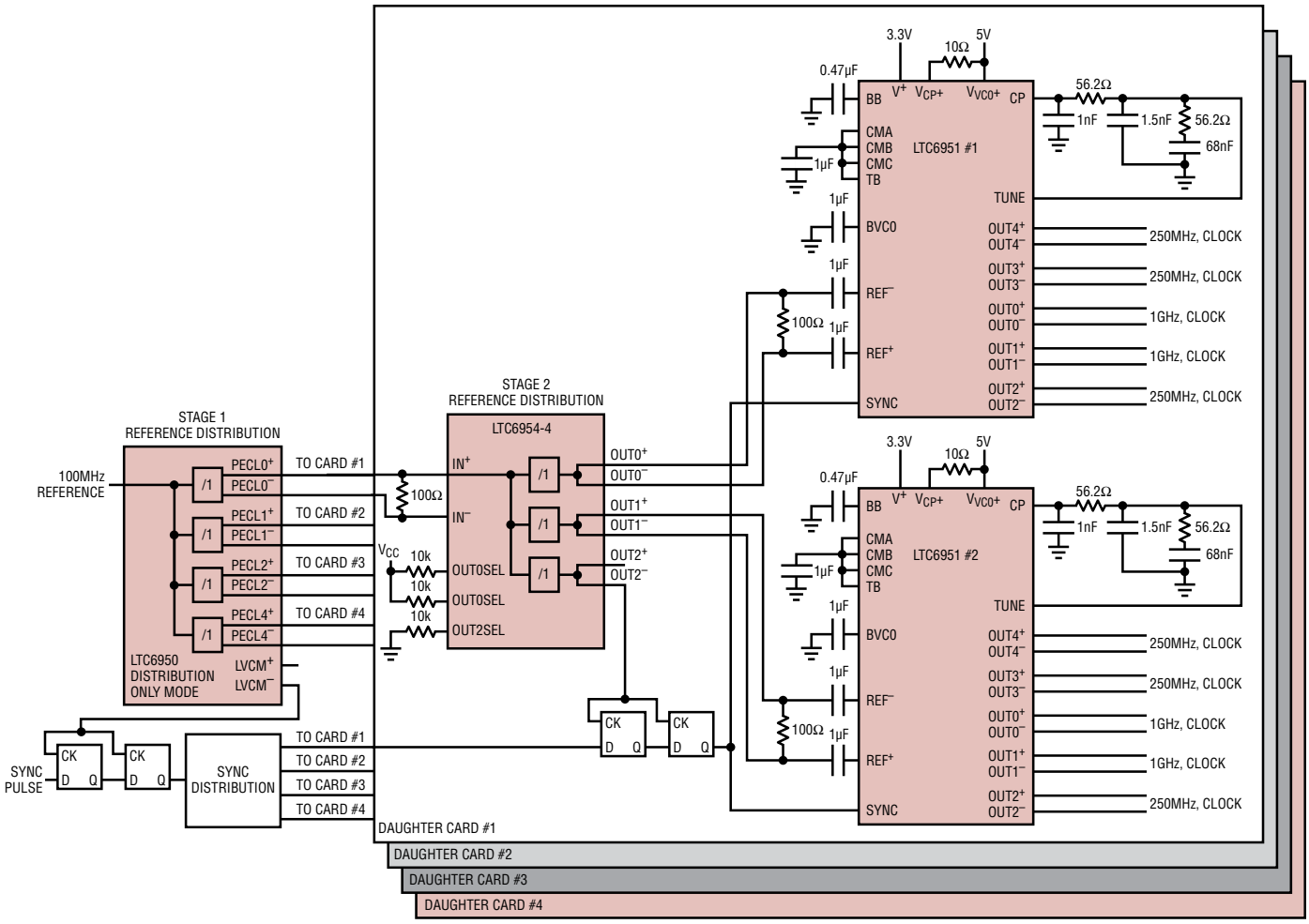


Figure 14. ParallelSync Expandable Solution

JESD204B ParallelSync DESIGN EXAMPLE 2

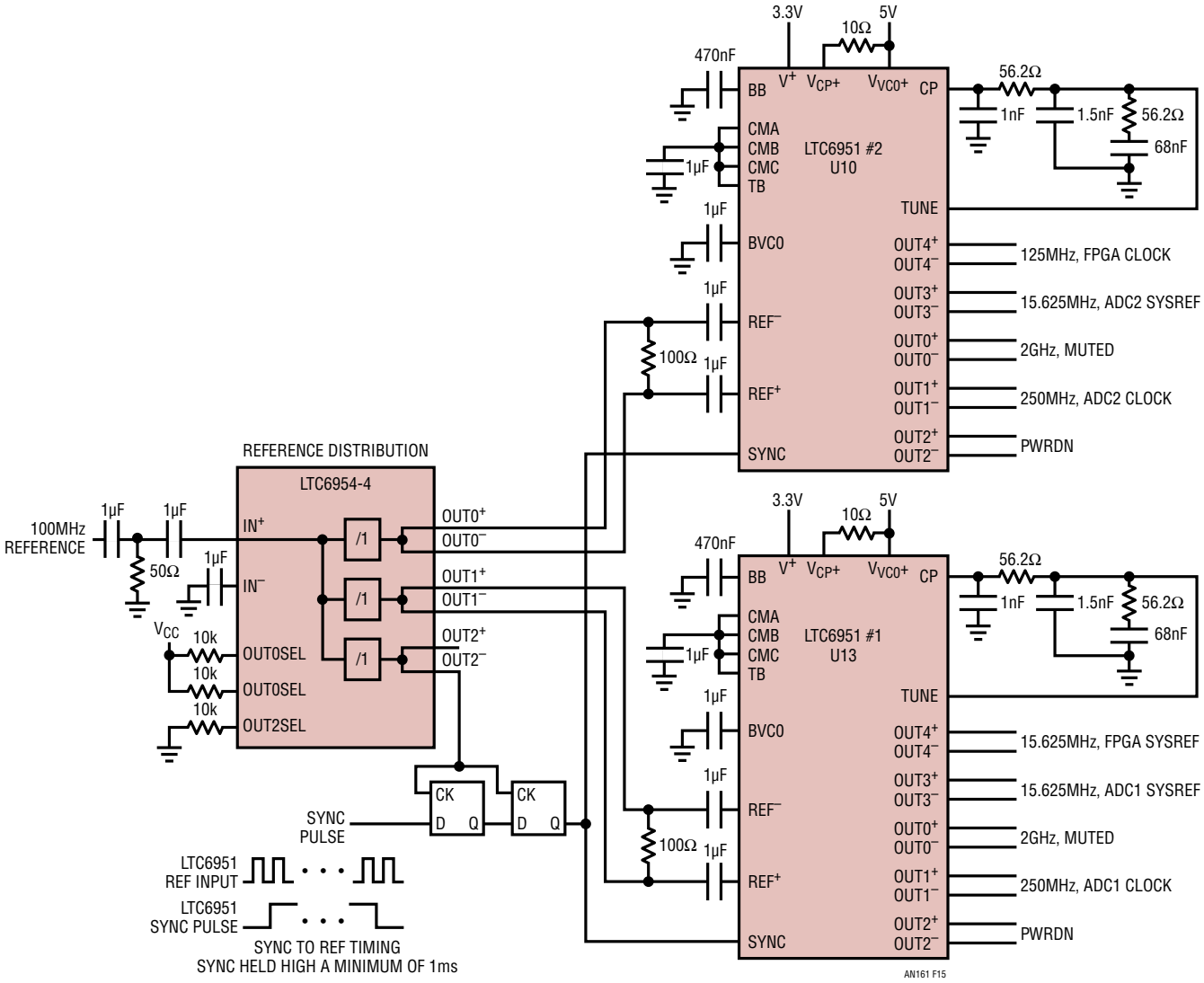


Figure 15. ParallelSync Design Example 2 – JESD204B

JESD204B ParallelSync Design Overview

Figure 15 demonstrates the LTC6951 in JESD204B subclass 1 ParallelSync configuration. The frequencies chosen mirror that of the DC2226 JESD204B subclass 1 demonstration board which includes two LTC2123 JESD204B ADCs.

The reference and synchronization sections of this design are identical to the ParallelSync Design Example 1. Design information for these sections will refer back to the relevant section in ParallelSync Design Example 1.

This example provides CLOCK and SYSREF signals to two ADC's and one FPGA. The selection of LTC6951 CLOCK and SYSREF output pins were selected to ease board layout and to optimize performance.

Board Layout

On the DC2226, an LTC6951 was placed next to each ADC to minimize the ADC CLOCK and SYSREF trace lengths. Minimizing the trace lengths between the LTC6951 and the ADC has the effect of increasing the reference trace length between the LTC6954 and the LTC6951, or vice

versa. Longer trace lengths typically increase the probability of unwanted signals or noise coupling onto the signal of interest.

Unwanted noise or signals coupling onto either the LTC6951 reference input or LTC6951 clock output can create random or deterministic clock jitter, respectively. Random ADC clock jitter degrades the ADC SNR, where deterministic jitter degrades the ADC SFDR. For this reason the LTC6951 reference input and LTC6951 clock outputs are both critical signals. However, when making trade-offs on trace length, it is recommended to treat the ADC clock signal as the more sensitive signal for the following reasons:

Filtering: LTC6951 reference inputs signals are naturally filtered by the existing narrowband PLL loop filter. The PLL loop filter does not affect LTC6951 output to output skew. Conversely, unwanted signals coupling onto an ADC clock can only be removed by adding a clock filter, which increases BOM cost and degrades clock to clock skew.

Impedance Matching: PLL reference frequencies are typically slower than ADC clock frequencies. Slower frequencies ease impedance matching and signal integrity concerns.

LTC6951 Output Selection

With ParallelSync, The LTC6951 OUT0 path is part of the PLL feedback loop (RAO = 1), which disables OUT0's delay functionality. Since not all LTC6951 outputs were needed and OUT0 delay feature is disabled, the OUT0 output buffer is powered down. The OUT0 divider network remains enabled to support the PLL feedback loop. Refer to the LTC6951 data sheet for more information regarding how the RAO bit affects the OUT0 operation. In addition, by not selecting OUT0 as a device clock this allowed for more flexibility in selecting the LTC6951 f_{PFD}. Careful selection of f_{PFD} optimizes the LTC6951 jitter performance. This point will be discussed more in the design example.

The LTC6951's OUT4 is an LVDS output. The other four LTC6951 outputs are CML outputs. LTC6951#2's and LTC6951#1's OUT4 pins were selected for the FPGA CLOCK and SYSREF signal, because the FPGA accepted LVDS signal levels.

The remaining three CML LTC6951 outputs, OUT1, OUT2, and OUT3 are identical in operation and performance. A CML output was chosen to drive the ADC Clock inputs, because the LTC6951 CML outputs have lower jitter than the LTC6951 OUT4 LVDS output. OUT1 and OUT3 were chosen to drive the ADC CLOCK and SYSREF due to layout considerations. OUT2 is closer to the LTC6951 reference input. Since not all outputs were required in this example, OUT2 was powered down to limit board coupling concerns between OUT2 and the LTC6951 reference input.

The section titled ParallelSync Design Example 2 section provides the design process used to develop the block diagram in Figure 15. Layout Recommendations discusses matching line lengths to minimize skew between parts. The section titled Synchronization Routines, provides initial power-up, power-down and resynchronization sequences. The Expandable Solution section discusses how the block diagram in Figure 7 can expand to support more LTC6951 devices.

ParallelSync Design Example 2

This design example will use the LTC6951Wizard to aid in the design process. Download LTC6951Wizard at http://www.linear.com/LTC6951Wizard.

This example assumes the following list of design inputs.

Reference	
f_{REF}	100MHz
LTC6951s	
$f_{6951\#1.OUT1}$ = 250MHz	
$f_{6951\#1.OUT3}$ = 15.625MHz	
$f_{6951\#1.OUT4}$ = 15.625MHz	
$f_{6951\#2.OUT1}$ = 250MHz	
$f_{6951\#2.OUT3}$ = 15.625MHz	
$f_{6951\#2.OUT4}$ = 125MHz	
RAO = 1	
Delay settings	Best performance
Performance Optimization Request	
Design for low jitter.	
Minimize the output skew between the LTC6951#1 and LTC6952#2	

LTC6951 Setup

Based on the ParallelSync Design Rules and the above design inputs, the following steps provide input conditions for the LTC6951Wizard.

Step 1: Design input: determine OUT0's frequency to optimize the LTC6951 f_{PFD} for low jitter.

ParallelSync's Design Rule 1 sets RAO = 1, making OUT0 part of the PLL feedback loop. As a result OUT0 affects the LTC6951 PLL's PFD frequency (f_{PFD}). Maximizing the LTC6951 f_{PFD} allows for a wider loop bandwidth and as a result optimal jitter performance. For more details, refer to the LTC6951 data sheet sections Reference Source Considerations and In-Band Output Phase Noise. The LTC6951 specified maximum f_{PFD} frequency is 100MHz.

The LTC6951Wizard automatically calculates the optimal OUT0 frequency in Figures 18 and 20, when PwrDown is chosen for OUT0.

$f_{6951\#1.OUT0}$ = PwrDown
$f_{6951\#2.OUT0}$ = PwrDown

Step 2: Design input: optimize the LTC6951 charge pump current for low jitter.

Refer to Step 2 in the ParallelSync Design Example 1

Step 3: Design input: minimize the output skew performance between the LTC6951#1 and LTC6952#2.

Refer to Step 3 in the ParallelSync Design Example 1

LTC6951Wizard

This section demonstrates the LTC6951Wizard's ability to ease the register setting creation and loop filter design for the LTC6951. Under the LTC6951Wizard's Help Menu a Help Guide is provided that will aid in understanding the operations performed in this section.

The values calculated in Steps 1-3 and conditions provided at the start of this design example are summarized below for a quick reference. These values will be used for inputs to the LTC6951Wizard to calculate the register settings and loop filter values for both LTC6951s in this design example.

LTC6951Wizard inputs for Figures 18 and 19:

$f_{6951.REF}$ = 100MHz
$f_{6951\#1.OUT0}$ = PwrDown
$f_{6951\#1.OUT1}$ = 250MHz
$f_{6951\#1.OUT3}$ = 15.625MHz
$f_{6951\#1.OUT4}$ = 15.625MHz

$f_{6951\#2.OUT0}$ = PwrDown
$f_{6951\#2.OUT1}$ = 250MHz
$f_{6951\#2.OUT3}$ = 15.625MHz
$f_{6951\#2.OUT4}$ = 125MHz

$FILT_{6951}$ = 0
RAO_{6951} = 1

Figures 18 to 24 provide the remaining steps necessary to complete the LTC6951 portion of this design. Several steps in these figures require the following additional information.

Importing Reference Noise

Refer to Appendix: Model Reference Noise for LTC6951Wizard Simulations, which describes how to import reference noise into the LTC6951Wizard and the impact of reference noise on loop filter calculations and output noise simulations. Example 1 in the appendix creates the reference noise profile for this example.

Delay setting: CLOCK DLYX Bits

For this example, the request was made to set delays values of the LTC6951 outputs for best performance. Less than optimal performance can result if the reference frequency mixes with an LTC6951 output frequency on the board. An initial attempt to avoid mixing produce will set the LTC6951 outputs delays so that the LTC6951 input reference edges and the LTC6951 output clock edges occur at different times.

Step 3a: Determine the LTC6951 PDIV cycles that coincide with the LTC6951 OUT1's 250MHz clock rising and falling edges.

$$Clock\_CYCLE = \frac{f_{PIV}}{f_{OUT1}} \quad (16)$$

$$Clock\_CYCLE = \frac{2GHz}{250MHz}$$

$$Clock\_CYCLE = 8\ PDIV\ Cycles$$

$$Clock\ Rising\ Edge = 8\ PDIV\ Cycles \bullet x + Dxi \quad (17)$$

$$Clock\ Falling\ Edge = 8\ PDIV\ Cycles \bullet x + 4 + Dxi \quad (18)$$

Where,

x is any integer,

Dxi adjusted delay setting, when Dxi = 0 the output aligns to the reference.

Step 3b: Determine the number of LTC6951 PDIV cycles with respect to LTC6951 Reference input frequency.

$$REF\_CYCLE = \frac{f_{PIV}}{f_{REF}} \quad (19)$$

$$REF\_CYCLE = \frac{2GHz}{100MHz}$$

$$REF\_CYCLE = 20\ PDIV\ cycles$$

$$REF\ Rising\ Edge = 20\ PDIV\ Cycles \bullet y \quad (20)$$

$$REF\ Falling\ Edge = 20\ PDIV\ Cycles \bullet y + 10 \quad (21)$$

Step 3c: Determine the LTC6951 PDIV cycles that coincide with the LTC6951 OUT3's 250MHz SYSREF rising edge.

$$SYSREF\_CYCLE = \frac{f_{PIV}}{f_{REF}} \quad (22)$$

$$SYSREF\_CYCLE = \frac{2GHz}{15.625MHz}$$

$$SYSREF\_CYCLE = 128\ PDIV\ cycles$$

$$SYSREF\ Rising\ Edge = 128\ PDIV\ Cycles \bullet x + Dxi \quad (23)$$

To maximize JESD204B SYSREF to CLOCK setup and hold times the SYSREF signals rising edge should occur on the falling edge of the CLOCK signal.

Step 3d: Determine Dxi by plotting results as shown below in Figure 16

Dxi = 1 or Dxi = 3 meet the desired criteria of not having the clock edges coincide with the reference edges. A Dxi = 3 for the Clock and Dxi = 7 for the SYSREF were chosen for Step 10 in Figure 20.

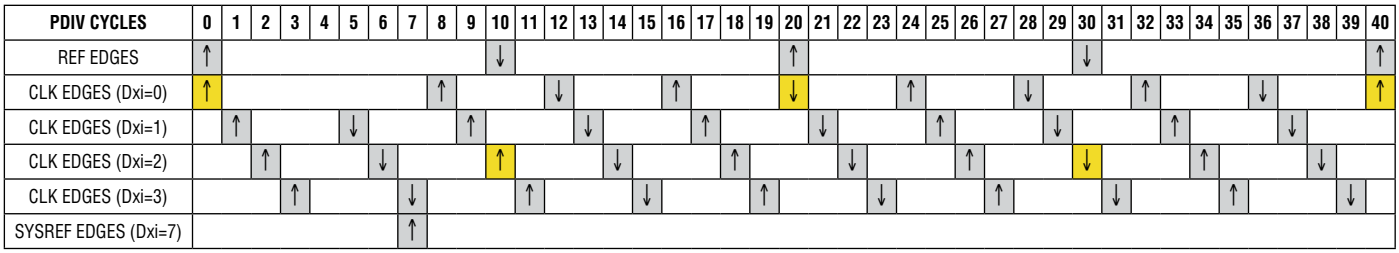


Figure 16. LTC6951 Reference, Clock and SYSREF Edge Location

Step 3e: Verify the LTC6951Wizard DLYx calculation.

The LTC6951 Wizard automatically calculates the DLYX bits based off of Equation 24. This same equation is found in the LTC6951 data sheet.

Solving Equation 24 for Dx (DLYX) from the values provided below match the LTC6951Wizard Delay results in Figure 22.

$$Dx = Dxi + CEILING\left(\frac{18}{N \bullet M0}\right) \bullet N \bullet M0 - 18 \qquad (24)$$

Dx = DLYx value in LTC6951 SPI map

Dxi = adjusted delay setting, a 0 aligns output rising edge to reference rising edge

18, number of PDIV cycles

N = 20

M0 = 1

For Clock:

Dxi = 3

$$Dx = 3 + CEILING\left(\frac{18}{20 \bullet 1}\right) \bullet 20 \bullet 1 - 18$$

Dx = 5 (Clock DLYX delay settings)

For SYSREF:

Dxi = 7

$$Dx = 7 + CEILING\left(\frac{18}{20 \bullet 1}\right) \bullet 20 \bullet 1 - 18$$

Dx = 9 (SYSREF DLYX delay settings)

SYNCENX Bits

Ensure the LTC6951 SYNCENX bits are set to a 1 for all signals that require synchronization. Refer to Figure 18, Step 2b.

LTC6951 OINV Bit

Figure 20, step 13 sets the OINVX values for each output. Figure 17 provides a recommendation for OINVX settings based on schematic connections to the device being clock. In this example all LTC6951 outputs will use the Standard OUTX Connection, setting OINVX = 0 (not inverted).

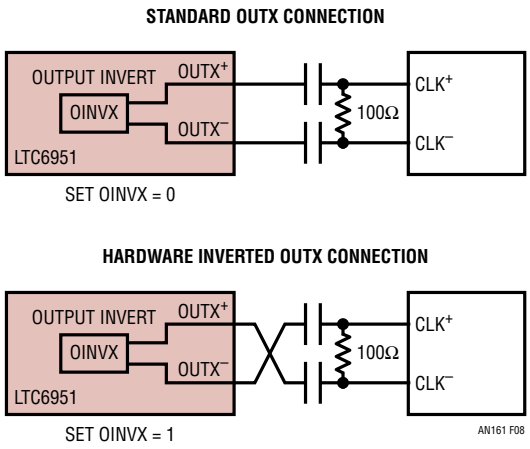


Figure 17. LTC6951 OINVX STATE

Loop Filter Selection

Figure 21's step 18 selected Filter 2. Through experimentation Filter 2 was found to be the best option to optimize performance and board space.

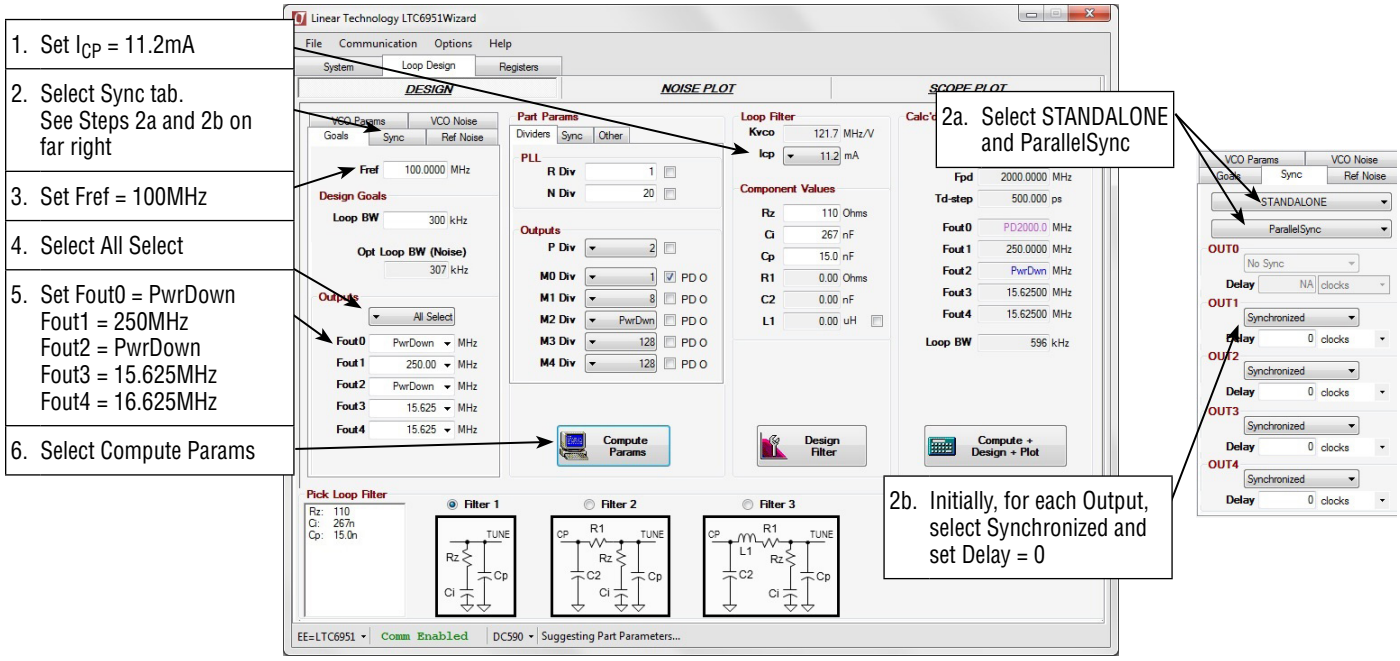


Figure 18. ParallelSync Design Example 2, DC2226 U13

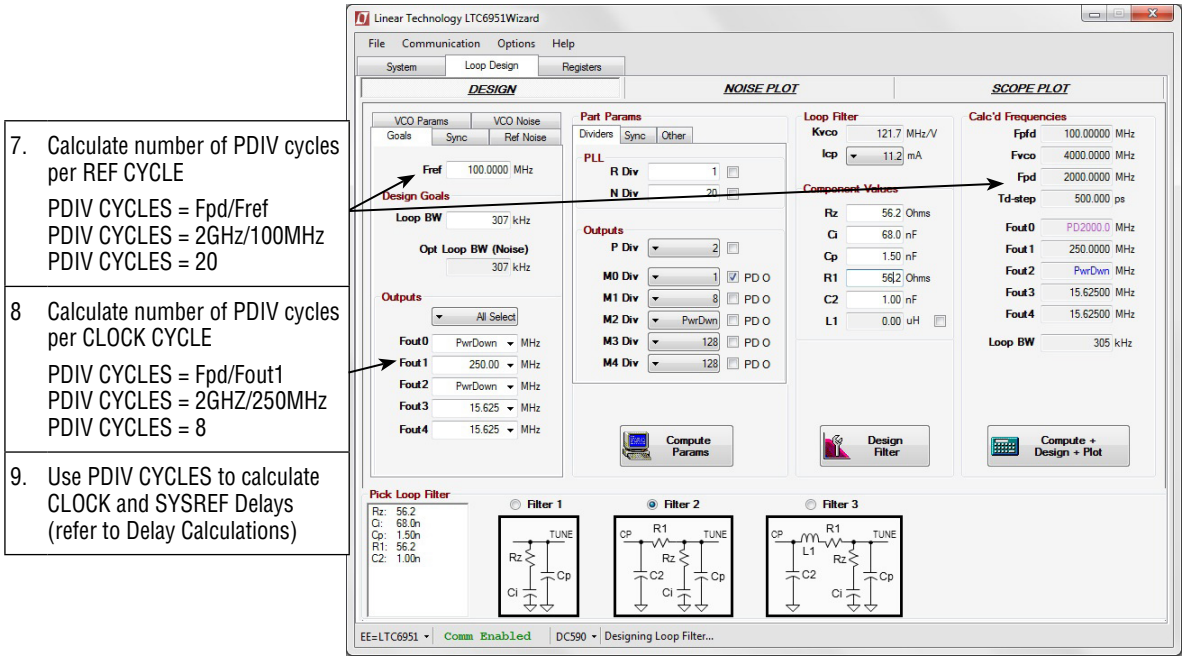


Figure 19. ParallelSync Design Example 2, DC2226 U13 (Continued)

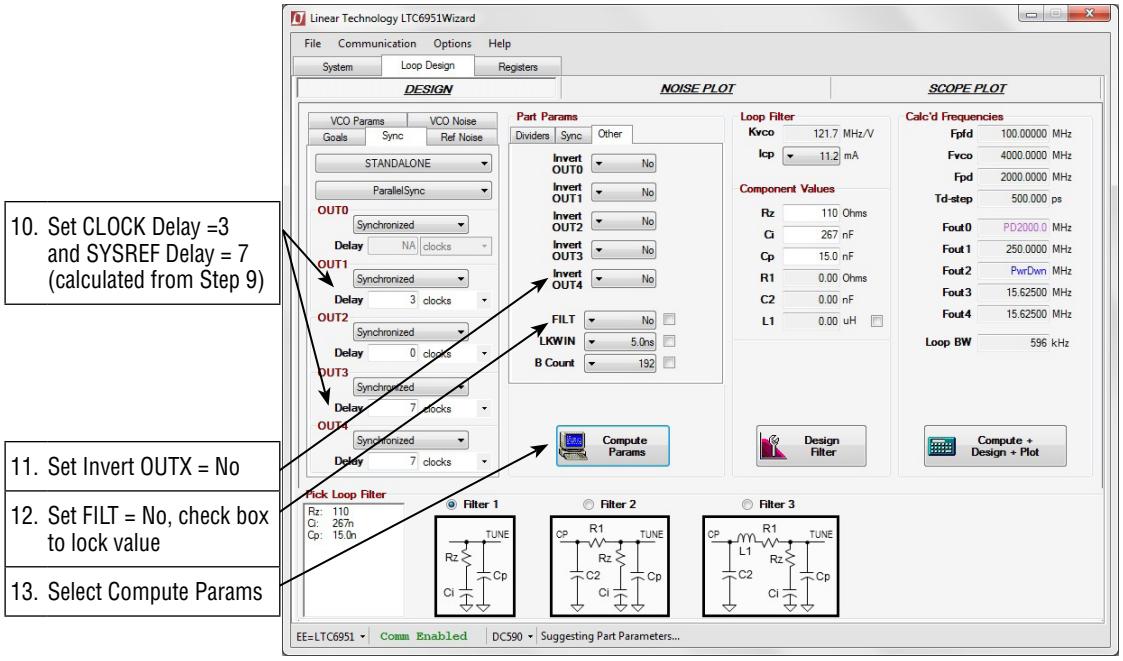


Figure 20. ParallelSync Design Example 2, DC2226 U13 (Continued)

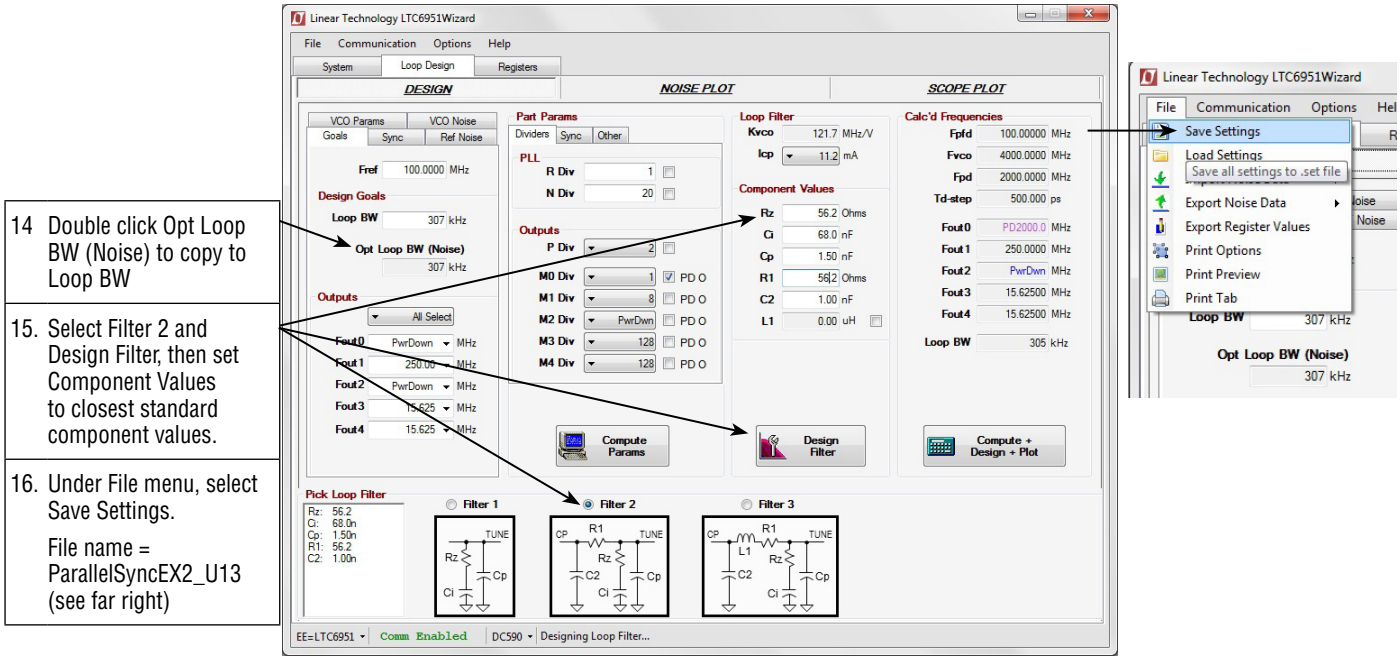


Figure 21. ParallelSync Design Example 2, DC2226 U13 (Continued)

17. Under Options menu, select Copy Loop to System

18. Select System tab to view results

19. Under File menu, select Save Settings.
File name = ParallelSyncEX2_U13

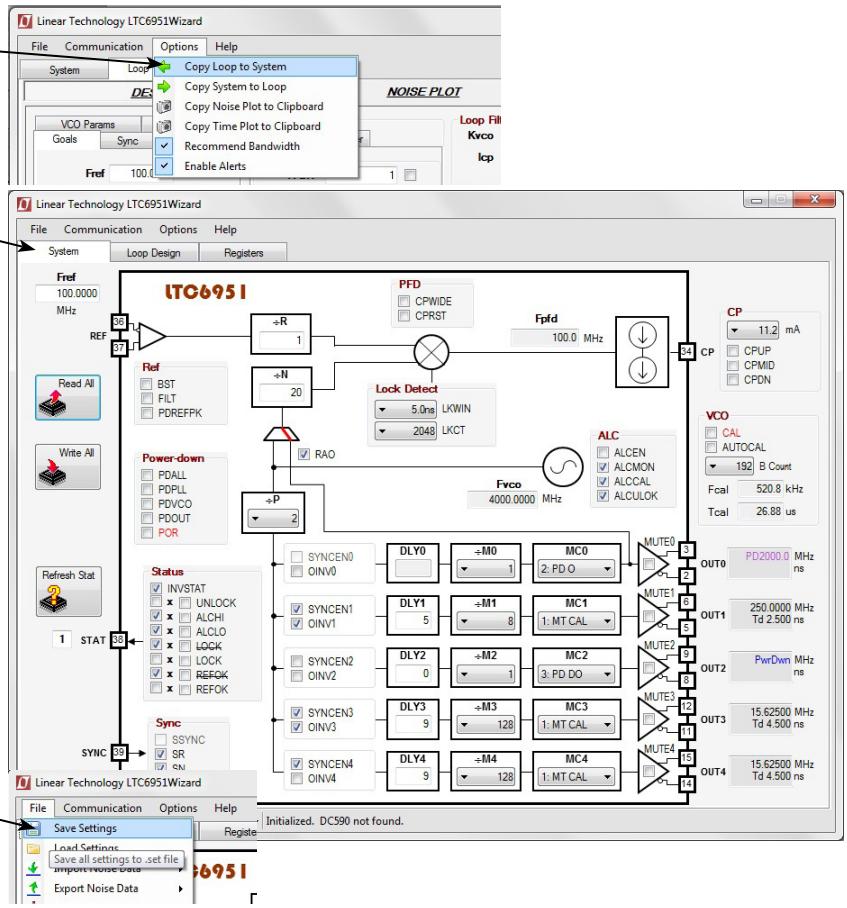


Figure 22. ParallelSync Design Example 2, DC2226 U13 (Continued)

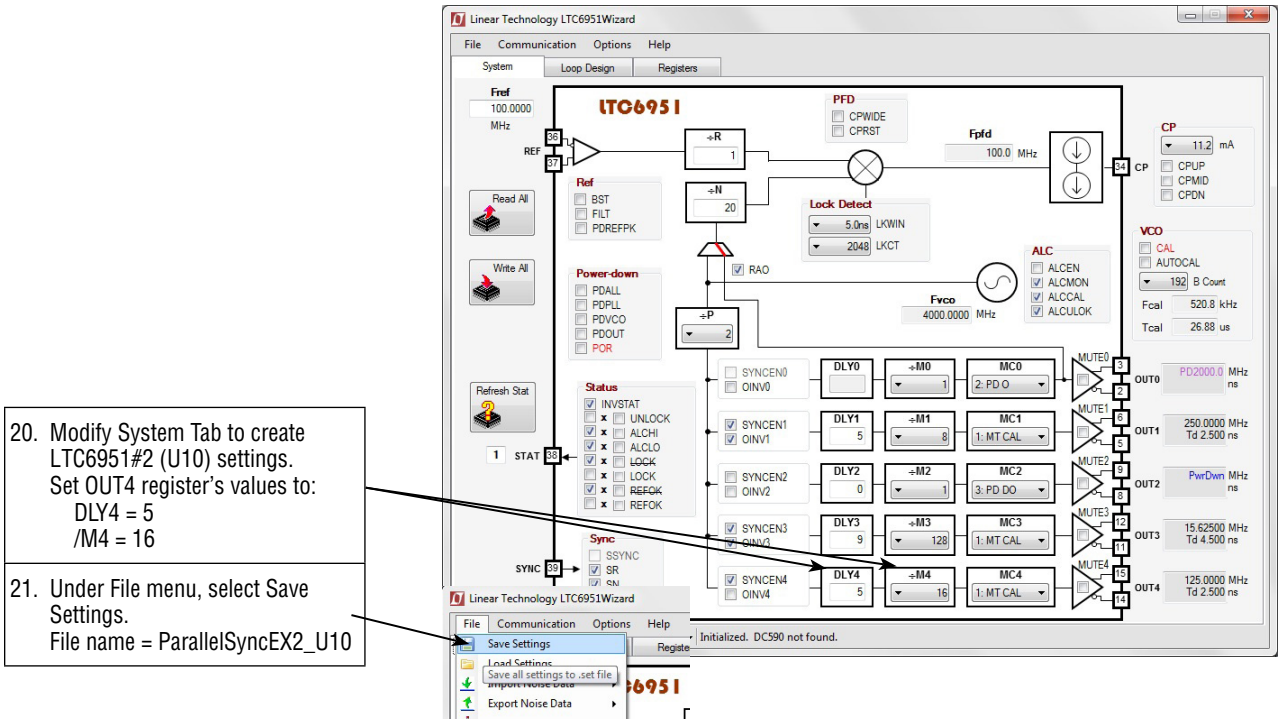


Figure 23. ParallelSync Design Example 2, DC2226 U10 (Continued)

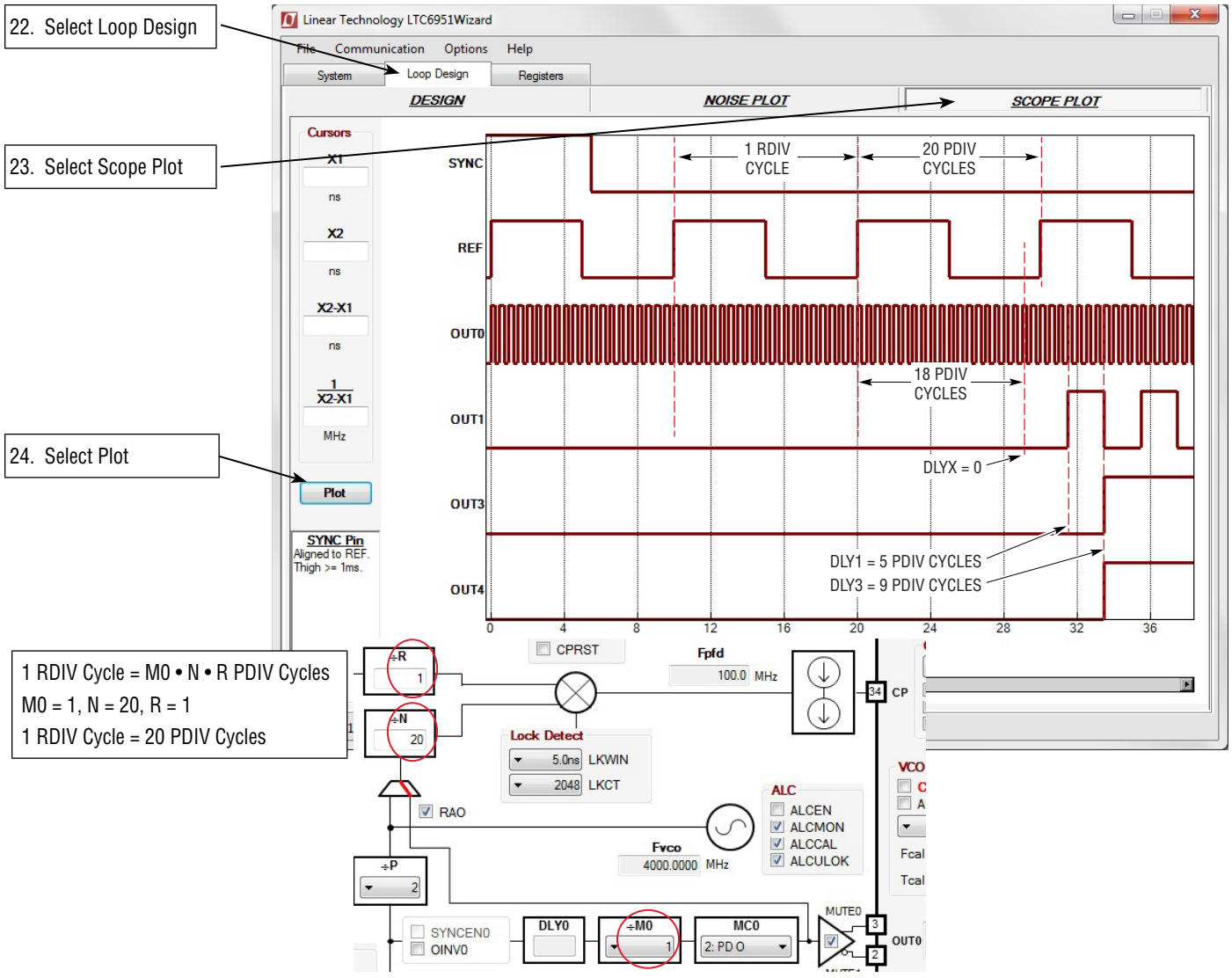


Figure 24. ParallelSync Design Example 2 (Continued)

LTC6954 Setup

Step 4: Design input: minimize the output skew performance between the LTC6951#1 and LTC6952#2.

Step 5: Verify LTC6954 output to LTC6951 connection.

Step 6: LTC6954 settings summarized:

Refer to Step 4 to Step 6 in the ParallelSync Design Example 1

Layout Recommendations

Refer to the Layout Recommendation section in the ParallelSync Design Example 1

Synchronization Routines

On initial power-up:

1. Program LTC6954 and LTC6951 SPI registers
2. If $MX_{6954} > 1$, Toggle LTC6954 SYNC pin (minimum 1ms)
3. Wait for LTC6951 bias voltages to stabilize
4. Calibrate all LTC6951 VCOs
5. Send SYNC pulse to LTC6951 Sync pins (see LTC6951 SYNC Pulse Width section)

Power-down after JESD204b alignment sequence is complete:

1. Power down LTC6951 SYSREF OUTX (MCX = 2)

   (NOTE: power down LTC6951 output, but leave LTC6951 output divider enable to avoid resynchronizing all clocks)

If JESD204 requires re-alignment:

1. Power up LTC6951 SYSREF OUTX (MCX = 1)

LTC6951 Sync Pulse Width

Refer to the LTC6951 Sync Pulse Width section in the ParallelSync Design Example 1. For this example the LTC6951 RDIV = 1.

Expandable Solution

The ParallelSync solution is infinitely expandable, refer ParallelSync Design Example 1 Section titled Expandable Solution and Figure 14 in for more details

EZParallelSync DESIGN EXAMPLE

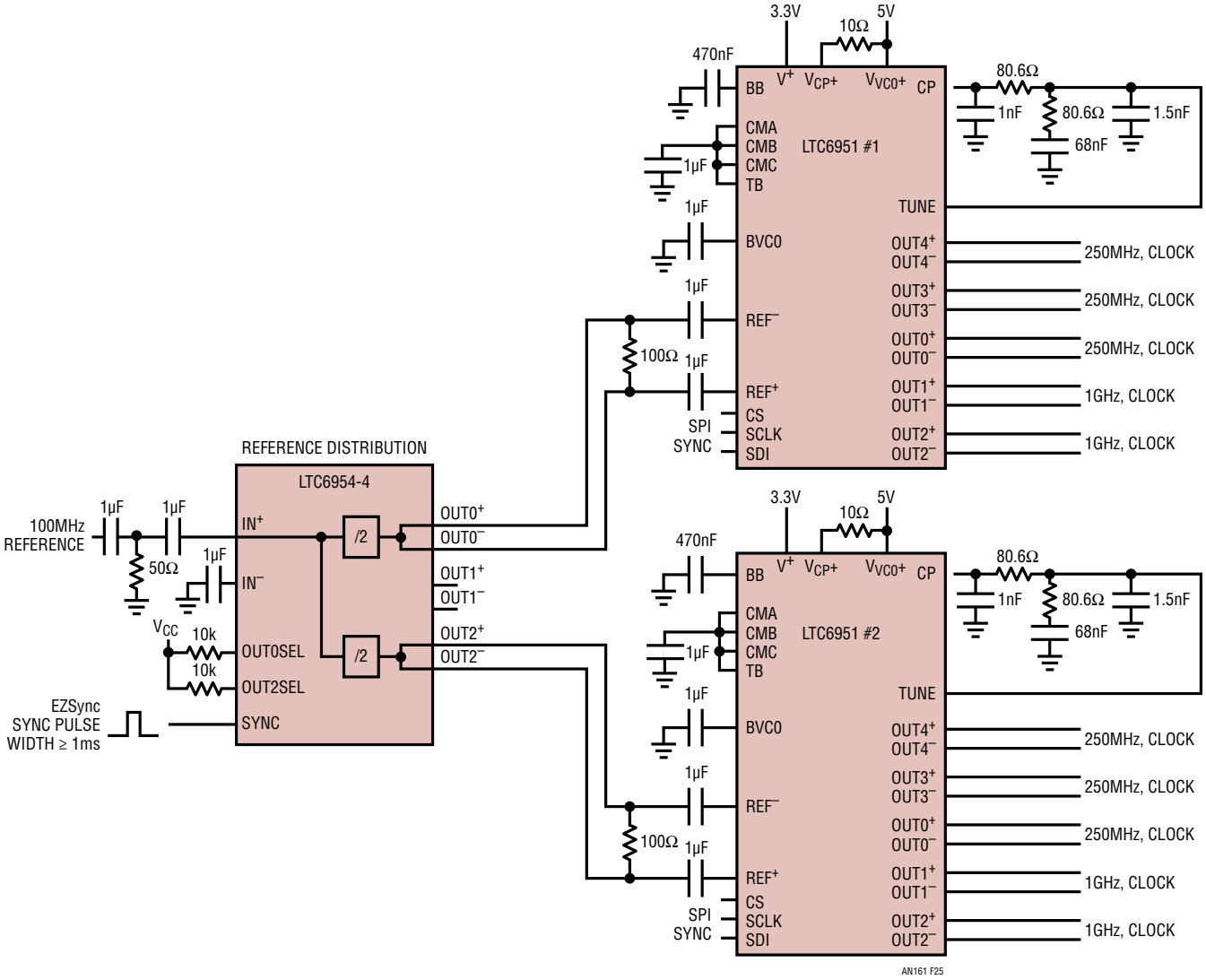


Figure 25. EZParallelSync Design Example – Two LTC6951s

EZParallelSync Design Overview

EZParallelSync is a simple way to synchronize multiple LTC6951's running in parallel driven by a common reference clock divider/distribution network. Synchronization is easily achieved through SPI commands. The LTC6951 SYNC pin can be used in lieu of the SPI command.

In Figure 25, the LTC6954 is the common reference clock divider/distribution network. The LTC6954 acts as an external reference divider (RDIV) to provide two LTC6951s with phase aligned reference frequencies. The external RDIV allows the LTC6951s internal RDIVs to equal 1. Setting the LTC6951 RDIV = 1 allows for output phase alignment across multiple LTC6951s to a common reference edge.

This architecture provides the ability to synchronize any LTC6951 to any other LTC6951 at any time. As a result if any LTC6951 is not used continuously then the unused LTC6951 can be completely powered down. When needed the powered down LTC6951 can be powered up and resynchronized independently without recalibrating

the LTC6951 VCO and without performing a full system clock synchronization. This ability to asynchronously synchronize independent LTC6951 with EZParallelSync is also useful in plug and play (hot plug) applications.

This example uses the same f_{OUT0} and the same f_{REF} for all LTC6951s. A variant of EZParallelSync is EZ204Sync. The EZ204Sync Design Example provides an example where different OUT0 or REF frequencies are used.

The section titled EZParallelSync Design Rules summarizes the EZParallelSync design rules. The section titled EZParallelSync Design Example section provides the design process used to develop the block diagram in Figure 25. Layout Recommendations discusses matching line lengths to minimize skew between parts. The section titled Synchronization Routines, provides power-up, power-down and resynchronization sequences. The Expandable Solution section discusses how the block diagram in Figure 25 can expand to support more LTC6951 devices.

LTC6951 EZParallelSync Design Rules

When compared to ParallelSync, EZParallelSync has the additional design rule that the LTC6951 OUT0 pin is assigned to the lowest output frequency per LTC6951. Refer to the LTC6951 data sheet for details related to PDIV and RAO.

1. LTC6951 OUT0 pin assigned to the lowest output frequency
2. LTC6951 register settings:
   a. RDIV = 1
   b. RAO = 1 (enabled)
3. $$X \bullet \frac{f_{6951\#1.OUT0}}{f_{6951\#1.REF}} = \frac{f_{6951\#N.OUT0}}{f_{6951\#N.REF}} \quad (25)$$

   Which can also be written as

   $$X \bullet NDIV_{6951\#1} = NDIV_{6951\#N} \quad (26)$$

   Where N can be any integer > 1 and X is an integer. In most cases X = 1.

EZParallelSync Design Example

This design example will use the LTC6951Wizard to aid in the design process. Download LTC6951Wizard at http://www.linear.com/LTC6951Wizard.

This example assumes the following list of design inputs.

Reference	
f_{REF} = 100MHz	
LTC6951s	
$f_{6951\#1.OUT0} = f_{6951\#2.OUT0}$ = 250MHz	
$f_{6951\#1.OUT1} = f_{6951\#2.OUT1}$ = 1GHz	
$f_{6951\#1.OUT2} = f_{6951\#2.OUT2}$ = 1GHz	
$f_{6951\#1.OUT3} = f_{6951\#2.OUT3}$ = 250MHz	
$f_{6951\#1.OUT4} = f_{6951\#2.OUT4}$ = 250MHz	
RDIV = 1	
RAO = 1	
Delay settings	Align LTC6951 outputs rising edge to LTC6951 reference input rising edge.
Performance Optimization Request	
Design LTC6951 for low jitter.	
Minimize the output skew between the LTC6951#1 and LTC6952#2	
Part Placement and Routing	
The LTC6954 and both LTC6951s will be placed on the top side of the board. For the most direct routing connect: • LTC6954 $OUTX^+$ to LTC6951 IN^- • LTC6954 $OUTX^-$ to LTC6951 IN^+	
This creates a reference signal inversion at the LTC6951 inputs.	

LTC6951 Setup

Based on the EZParallelSync Design Rules and the above design inputs, the following steps provide input conditions for the LTC6951Wizard.

Step 1: Design Rule 1 verification

The first design rule is met since the slowest output frequency, 250MHz, is assigned to OUT0.

Step 2: Design Rule 2 verification

To ensure Design Rule 2's RDIV = 1 requirement is met, calculate $M0_{6954}$, $M2_{6954}$, and both LTC6951's $f_{6951.REF}$ and $NDIV_{6951}$ values. $M0_{6954}$ and $M2_{6954}$ refer to the LTC6954 divide value.

The second design rule states that $RDIV_{6951\#1}$ = 1 and RAO_{6951} = 1. Referring to the LTC6951 data sheet the following two equations are provided when RAO_{6951} = 1.

$$f_{6951\#1.PFD} = \frac{f_{6951\#1.REF}}{RDIV_{6951\#1}} \quad (27)$$

$$f_{6951\#1.PFD} = \frac{f_{6951\#1.OUT0}}{NDIV_{6951\#1}} \quad (28)$$

Referring to the LTC6954 data sheet the following equation is provided.

$$f_{6951\#1.REF} = \frac{f_{REF}}{M0_{6954}} \quad (29)$$

Since $RDIV_{6951\#1}$ = 1, Equations 27, 28, and 29 can be rearranged as follows:

$$NDIV_{6951\#1} = \frac{M0_{6954} \bullet f_{6951\#1.OUT0}}{f_{REF}} \quad (30)$$

Substituting the known values for f_{REF} = 100MHz and $f_{6951\#1.OUT0}$ = 250MHz into Equation 30 results in

$$NDIV_{6951\#1} = \frac{M0_{6954} \bullet 250MHz}{100MHz} \quad (31)$$

Which simplifies to

$$NDIV_{6951\#1} = 2.5 \bullet M0_{6954} \quad (32)$$

Based on the LTC6951 data sheet, to optimize for the lowest jitter possible $f_{6951\#1.PFD}$ should be as large as possible, which allows for a wider bandwidth loop filter. This statement assumes the reference input signal noise level is not limiting the LTC6951's in-band noise performance. Therefore, solve Equation 32 for the least common integer multiple for NDIV and M0.

$NDIV_{6951\#1}$ = 5

$M0_{6954}$ = 2

Next solve Equation 29 and 27.

$f_{6951\#1.REF}$ = 50MHz

$f_{6951\#1.PFD}$ = 50MHz

Since LTC6951#1 and LTC6951#2 have an identical frequency plan.

$NDIV_{6951\#1} = NDIV_{6951\#2}$ = 5

$M0_{6954} = M2_{6954}$ = 2

$f_{6951\#1.REF} = f_{6951\#2.REF}$ = 50MHz

Step 3: Design Rule 3 verification

Based on the results from step 2, Design Rule 3 is met.

$f_{6951\#1.OUT0}/f_{6951\#1.REF} = f_{6951\#2.OUT0}/f_{6951\#2.REF}$

Step 4: Design input: optimize the LTC6951 for low jitter.

Based on the LTC6951 data sheet, the jitter performance is obtained by maximizing the f_{PFD} frequency (see Step 2) and maximizing the LTC6951 ICP current.

$ICLK_{6951.CP}$ = 11.2mA

Note: The ParallelSync Design Example #1 uses the same output frequencies as this EZParallelSync Design Example. However, because ParallelSync does not require the lowest output frequency on OUT0 (EZParallelSync Design Rule 1) a larger f_{PFD} (100MHz) could be obtained. The larger f_{PFD} resulted in the ParallelSync example having ~10fs improved jitter performance when compared to this example. The frequencies in these two examples were chosen specifically to highlight this difference. Depending on the desired reference and output frequencies, differences in f_{PFD} between these two synchronization methods may or may not result. This note is directed at the LTC6951, as other Linear Technology PLL/VCOs may not have the LTC6951's pre-scalar divider architecture. As a result the LTC6951 EZParallelSync Design rule #1 may not apply to other PLL/VCOs.

Step 5: Design input: minimize the output skew performance between the LTC6951#1 and LTC6952#2.

The LTC6951 device to device skew is best when the LTC6951 register value FILT = 0.

LTC6951Wizard

This section demonstrates the LTC6951Wizard's ability to ease the register setting creation and loop filter design for the LTC6951. Under the LTC6951Wizard's Help Menu a Help Guide is provided that will aid in understanding the operations performed in this section.

The values calculated in Steps 1 to 5 and conditions provided at the start of this design example are summarized below for a quick reference. These values will be used for inputs to the LTC6951Wizard to calculate the register settings and loop filter values for both LTC6951s in this design example.

LTC6951Wizard inputs for Figure 27:

$f_{6951.REF}$ = 50MHz
$f_{6951.OUT0}$ = 250MHz
$f_{6951.OUT1}$ = 1GHz
$f_{6951.OUT2}$ = 1GHz
$f_{6951.OUT3}$ = 250MHz
$f_{6951.OUT4}$ = 250MHz
$I_{6951.CP}$ = 11.2mA
$NDIV_{6951}$ = 5
$RDIV_{6951}$ = 1
$FILT_{6951}$ = 0
RAO_{6951} = 1

Figures 27 and 28 provide the remaining steps necessary to complete the LTC6951 portion of this design. Several steps in Figures 27 and 28 require the following additional information.

Importing Reference Noise

Refer to Appendix: Model Reference Noise for LTC6951Wizard Simulations, which describes how to import reference noise into the LTC6951Wizard and the impact of reference noise on loop filter calculations and output noise simulations. Example 2 in the appendix creates the reference noise profile for this example.

Delay setting: DLYX BITS

For this example, the request was made to align the rising edge of the LTC6951 outputs with the rising edge of the LTC6951 reference input. The LTC6951 Wizard automatically calculates the DLYX bits based off of Equation 33. This same equation is found in the LTC6951 data sheet.

Figure 27, step 2b sets the Delay value = 0. An LTC6951Wizard Delay value = 0 forces the LTC6951Wizard to calculate the LTC6951 DLYX settings to align the LTC6951 output and reference input rising edges. Figure 28 shows the DLYX bits = 22 based off the wizard calculation. Figure 29 shows that the LTC6951 output and reference inputs rising edges are aligned.

$$Dx = Dxi + CEILING\left(\frac{18}{N \bullet M0}\right) \bullet N \bullet M0 - 18 \qquad (33),$$

Solving Equation 33 from the values shown match the LTC6951Wizard results, shown in Figures 28 and 29.

Dxi = 0 (align to reference)
18, number of PDIV cycles
N = 5
M0 = 8

$$Dx = 0 + CEILING\left(\frac{18}{5 \bullet 8}\right) \bullet 5 \bullet 8 - 18$$

Dx = 22 (delay settings)

SYNCENX BITS

Ensure the LTC6951 SYNCENX bits are set to a 1 for all signals that require synchronization. Refer to Figure 27, Step 2b.

LTC6951 OINV Bit

Figure 27, step 7 sets the OINVX values for each output. Figure 26 provides a recommendation for OINVX settings based on schematic connections. In this example all LTC6951 outputs will use the Standard OUTX Connection, setting OINVX = 0 (not inverted).

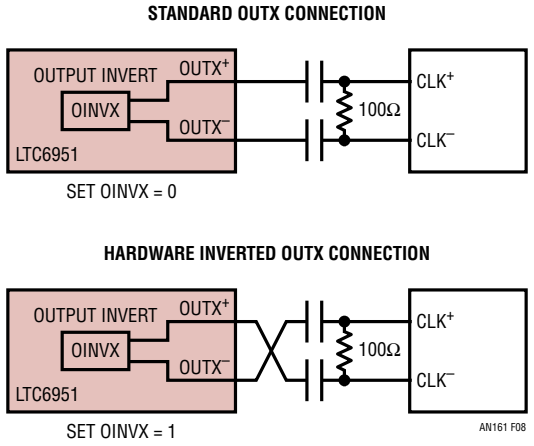
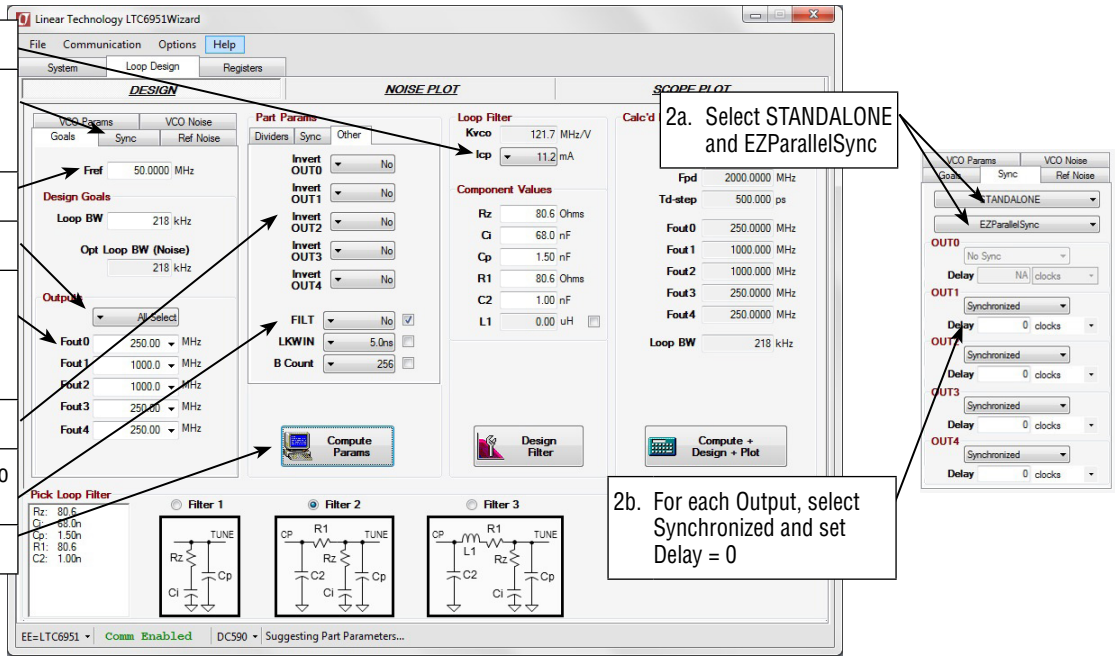


Figure 26. LTC6951 OINVX State

Loop Filter Selection

Figure 27's step 11 selected Filter 2. Through experimentation Filter 2 was found to be the best option to optimize performance and board space.

1. Set I_{CP} = 11.2mA
2. Select Sync tab. See Steps 2a and 2b on far right
3. Set Fref = 50MHz
4. Select All Select
5. Set Fout0 = Fout3 = Fout4 = 250MHz Set Fout1 = Fout2 = 1000MHz
6. Set Invert OUTx = No
7. Set FILT = No, check box to lock value
8. Select Compute Params

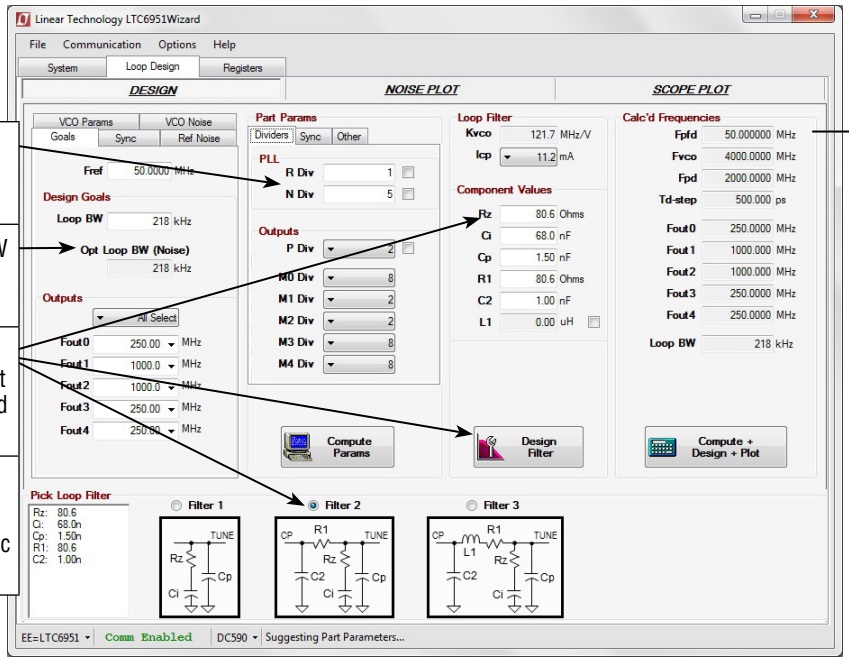


2a. Select STANDALONE and EZParallelSync

2b. For each Output, select Synchronized and set Delay = 0

9. Verify R Div = 1 and N Div matches previous calculation
10. Double click Opt Loop BW (Noise) to copy to Loop BW
11. Select Filter 2 and Design Filter, then set Component Values to closest standard component values.
12. Under File menu, select Save Settings.

    File name = EZParallelSync (see far right)



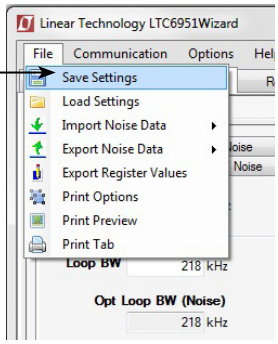


Figure 27. LTC6951Wizard Setup

12. Under Options menu, select Copy Loop to System

13. Select System tab to view results

13. Under File menu, select Save Settings.
File name = EZParallelSync

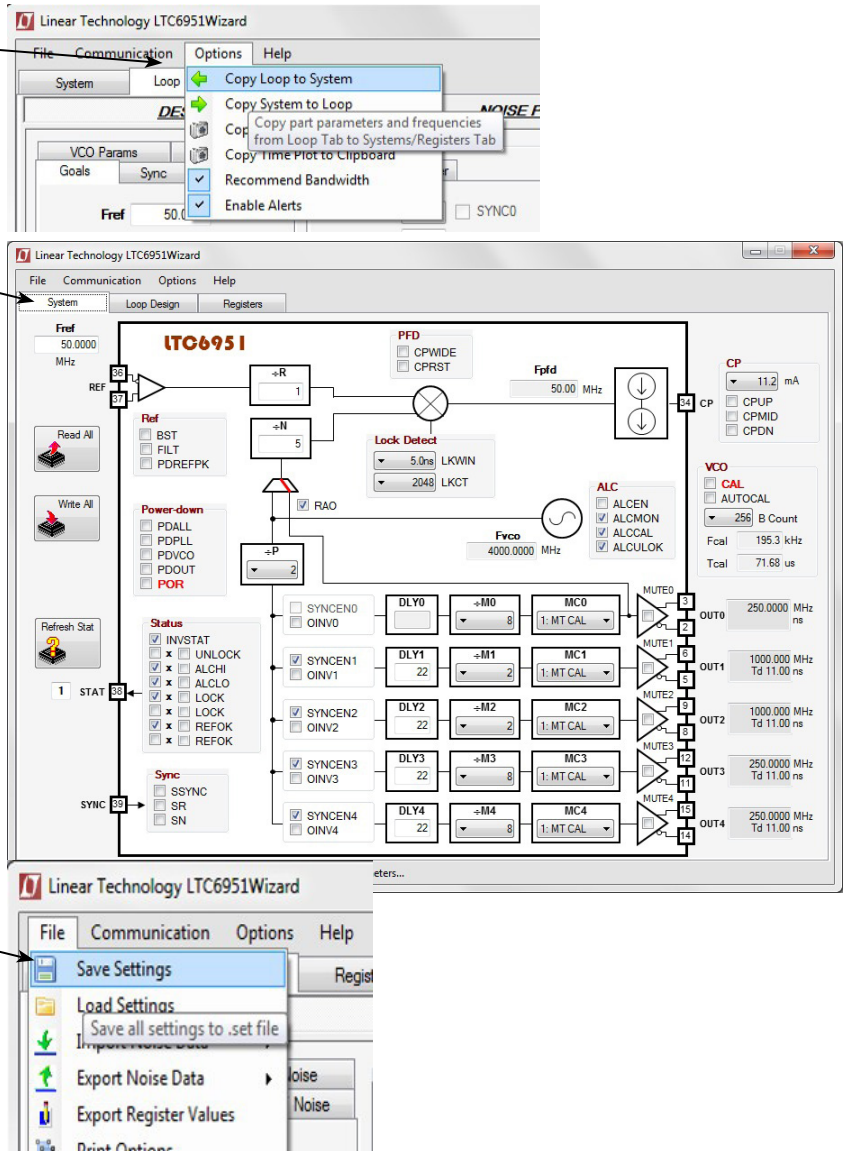


Figure 28. LTC6951Wizard Setup Continued

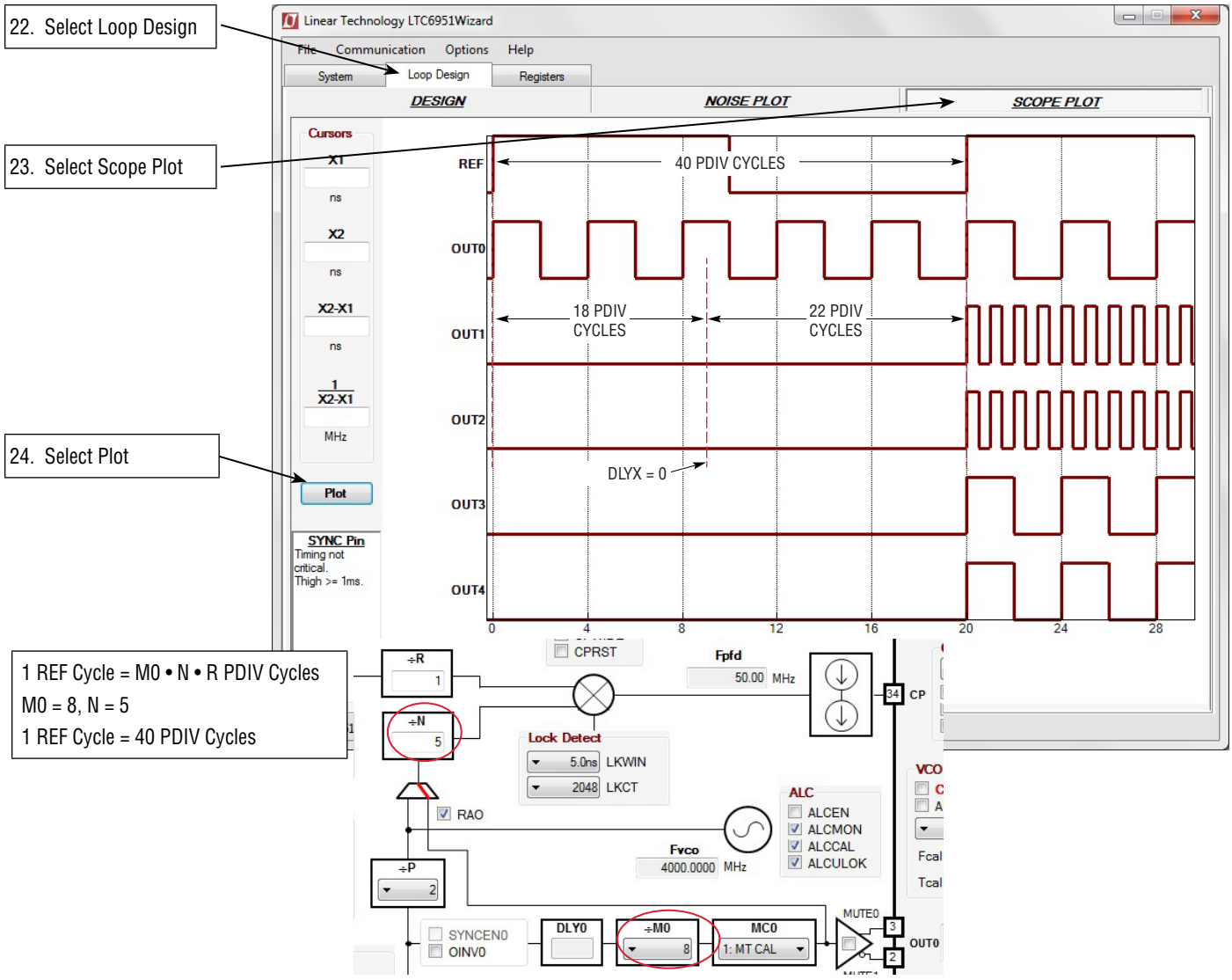


Figure 29. LTC6951Wizard Delay Settings

LTC6954 Setup

Step 6: Design input: minimize the output skew between the LTC6951#1 and LTC6952#2.

The LTC6951 outputs are phase aligned to the reference input. Skew in reference signals will result in skew between LTC6951s. Therefore, it is recommended to match trace lengths on the reference signals during board layout.

According to the LTC6954 data sheet best skew performance is obtained when either one of following two conditions are met:

- Condition 1: all LTC6954 output divider settings equal 1
- Condition 2: all LTC6954 output divider settings are >1.

In Step 2 the LTC6954 was design for optimal outputs skew, since Condition 2 was met.

$M0_{6954} = 2$

$M2_{6954} = 2$

Step 7: Verify LTC6954 output to LTC6951 connection.

It is recommended to choose an identical reference schematic from Figure 30 for both LTC6951s. This ensures both LTC6951 PLLs align to the same reference edge.

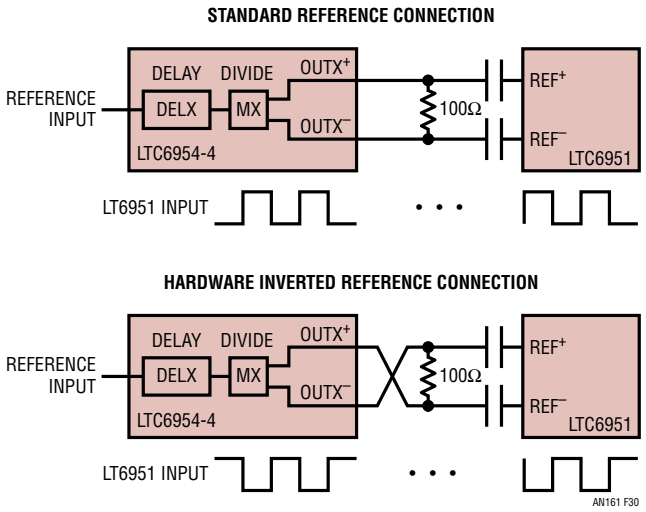


Figure 30. Reference Distribution Connection

For this example, both reference inputs were required to use Figure 30's Hardware Inverted Reference Connection. Therefore, set $DEL0_{6954} = DEL2_{6954}$.

If it is desired to connect LTC6951#1 to Figure 30's Standard Reference Connection and LTC6951#2 to the Hardware Inverted Reference Connection, then the reference inputs will be inverted with respect to each other.

To account for this schematic inversion, invert one of the LTC6954 output signals by delaying one LTC6954 output a ½ cycle using the LTC6954 delay bits (see Table C1). Using the LTC6954 delay registers in this manner is only possible when all LTC6954 divide values are even numbers.

Table 2. LTC6954 Register Settings, when Schematic Chooses Different Reference Connections

	LTC6954 Register Settings	
LTC6954 OUTX	**MX**	**DELX**
LTC6951: Standard Reference Connection	Even Number	Y*
LTC6951: Hardware Inverted Reference Connection	Even Number	$\frac{\min(MX)}{2} + Y^*$

* Y, integer, same value for all DELX

Step 8: LTC6954 register settings summarized:

$SYNCEN0_{6954} = 1$
$M0_{6954} = 2$
$DEL0_{6954} = 0$
$PDIV0_{6954} = 0$
$PDOUT0_{6954} = 0$

$PDIV1_{6954} = 1$

$SYNCEN2_{6954} = 1$
$M2_{6954} = 2$
$DEL2_{6954} = 0$
$PDIV2_{6954} = 0$
$PDOUT2_{6954} = 0$

Layout Recommendations

To minimize LTC6951 output skew match electrical trace lengths as shown in Equations 34 and 35 (refer to Figure 31).

$$L_{REF\#1} = L_{REF\#2} \quad (34)$$

$$L_{\#1.0UTX} = L_{\#2.0UTX} \quad (35)$$

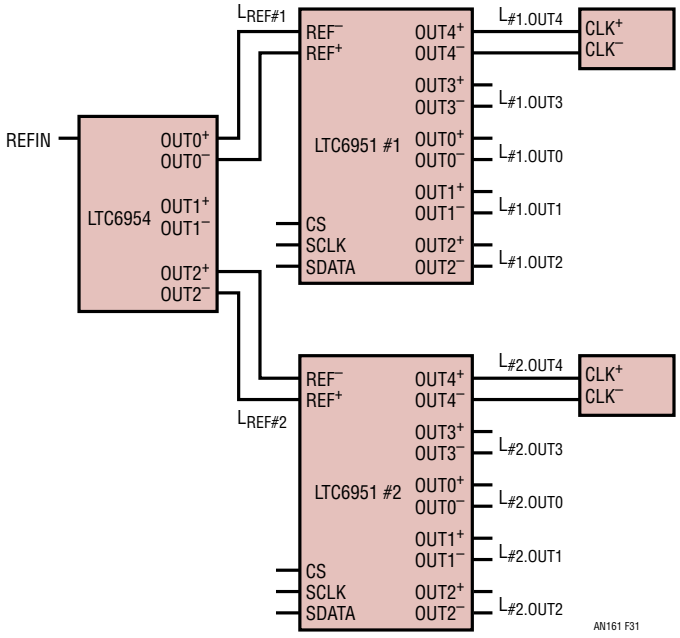


Figure 31. Trace Length Matching

Synchronization Routines

On initial power-up:

1. Program LTC6954 and LTC6951 SPI registers
2. Toggle LTC6954 SYNC pin (minimum 1ms)
3. Wait for LTC6951 bias voltages to stabilize
4. Calibrate all LTC6951 VCOs
5. Toggle all LTC6951 SPI SSYNC bits or SYNC pins (minimum 1ms)

Power down Idle LTC6951:

1. Power down idle LTC6951 (PDALL = 1)
2. Power down LTC6954 OUTX connected to idle LTC6951 (PD_OUTX = 1)

Resynchronization of Idle LTC6951:

1. Power up idle LTC6951 (PDALL = 0)
2. Power up LTC6954 OUT X connected to idle LTC6951 (PD_OUTX = 0)
3. Toggle LTC6951 SPI SSYNC bit or SYNC pin (minimum 1ms)

Expandable Solution

The EZParallelSync solution is infinitely expandable. As shown in Figure 32, the EZParallelSync design example can be repeated by adding an EZSync CONTROLLER to distribute the reference.

For ease of synchronization, in Figure 32 the divide by two function was moved from Stage 2 to the LTC6950 in Stage 1. As a result, Stage 2 can now use a low noise fanout buffer such as the LTC6957.

For further expansion, it is possible to cascade additional reference distribution stages between Stage 1 and Stage 2. When designing a multi-stage reference divider/distribution network, take into account the additive properties of:

- channel to channel skew
- noise floor at frequency offsets less than the LTC6951 loop filter's pass-band. Refer to Appendix: Model Reference Noise for LTC6951Wizard Simulations.

If some LTC6951's can power down during operation, then selecting reference distribution parts with the ability to power down individual outputs can save additional power.

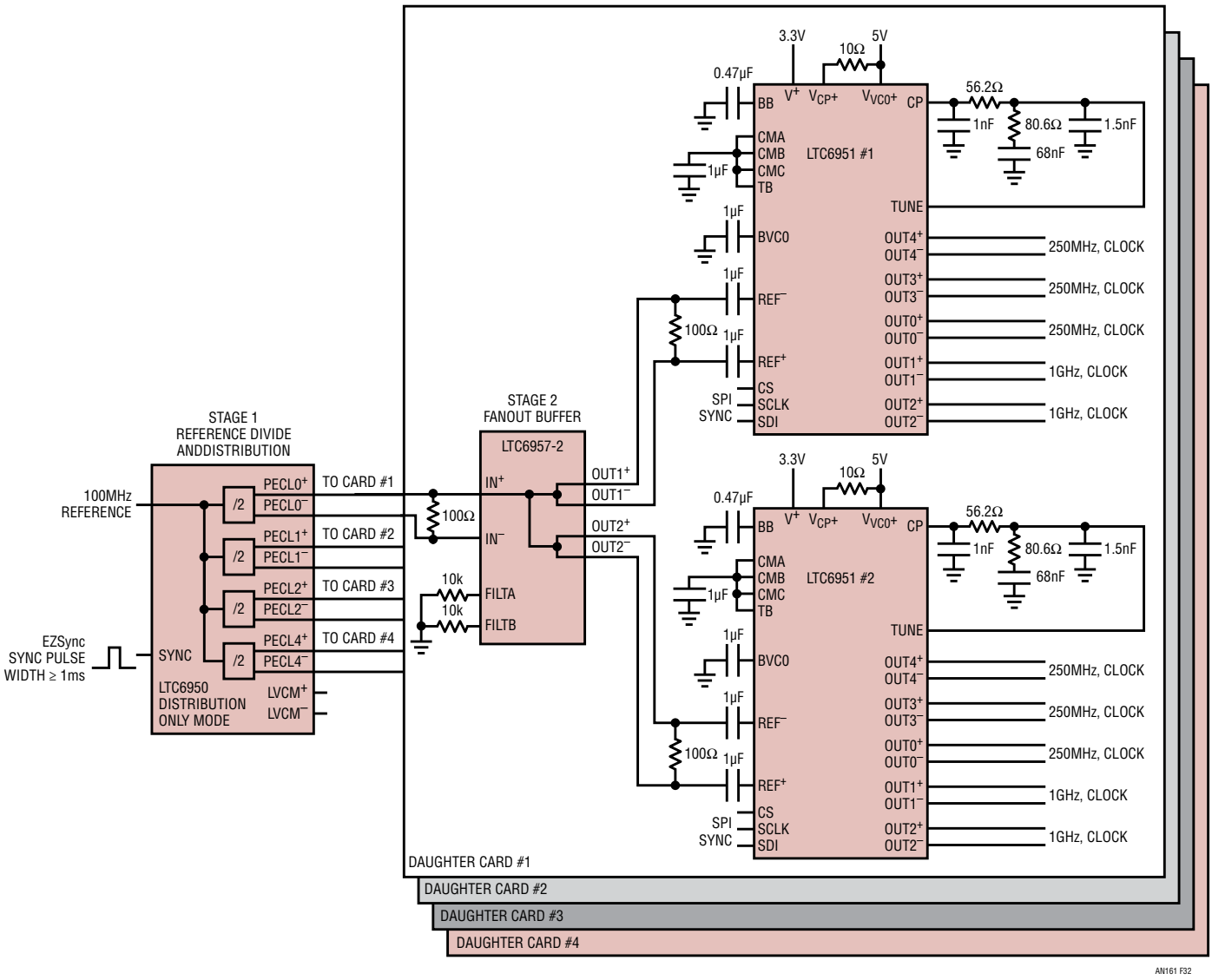


Figure 32. EZParallelSync Expandable Solution

EZ204Sync DESIGN EXAMPLE

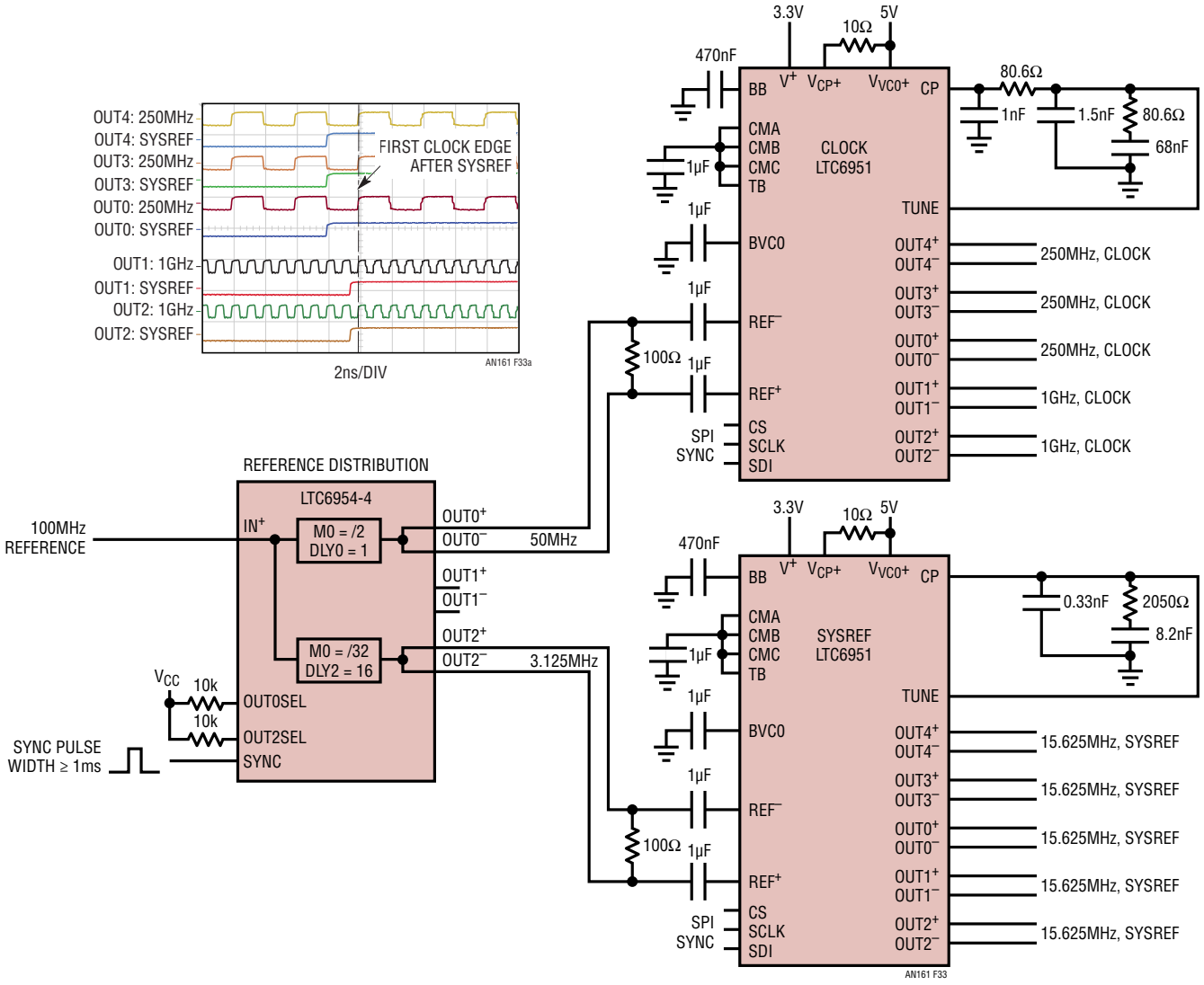


Figure 33. EZ204Sync Design Example – Two LTC6951s

EZ204Sync Design Overview

EZ204Sync is a subset of EZParallelSync, which is optimized to provide a method to synchronize JESD204B CLOCK and SYSREF signals with asynchronous SPI SYNC commands to an unlimited number of LTC6951s. The LTC6951 SYNC pin can also be used in lieu of the SPI sync command.

Figure 33 separates the CLOCK and SYSREF signals onto separate LTC6951s. In Figure 33, these are denoted as CLOCK LTC6951 and SYSREF LTC6951. This architecture requires an external EZSync reference divider/distribution network, such as the LTC6954 shown in Figure 33.

By using an EZSync reference divider, the CLOCK and SYSREF reference phases are phase aligned following an EZSync event. The external reference divider (RDIV) allows for the LTC6951s' internal RDIV to equal 1. Setting the LTC6951 RDIV = 1 allows for output phase alignment across multiple LTC6951s to a common reference edge.

This architecture provides the ability to synchronize any LTC6951 to any other LTC6951 at any time. As a result,

if any LTC6951 is not used continuously, then the unused LTC6951s can be completely powered down. When needed, the powered down LTC6951s can be powered up and resynchronized independently without recalibrating its VCO and without performing a full system clock synchronization. This ability to asynchronously synchronize independent LTC6951s with EZ204Sync is useful for JESD204B subclass 1 applications.

The section titled EZ204Sync Design Guidelines summarizes the EZ204Sync design rules. The section titled EZ204Sync Design Example section provides the design process used to develop the block diagram in Figure 33. Layout Recommendations discusses matching line lengths to minimize skew between parts. The section titled Synchronization Routines provides the initial power-up, power-down and resynchronization sequences. The Expandable Solution section discusses how the block diagram in Figure 33 can expand to support more LTC6951 devices.

EZ204Sync Design Rules

When compared to ParallelSync, EZ204Sync has the additional design rule that the LTC6951 OUT0 pin is assigned to the lowest output frequency per LTC6951. Refer to the LTC6951 data sheet for details related to PDIV and RAO.

1. LTC6951 OUT0 pin assigned to the lowest output frequency per LTC6951

2. LTC6951 register settings:

a. RDIV = 1

b. RAO = 1 (enabled)

3. $$X \bullet \frac{f_{SYS6951.OUT0}}{f_{SYS6951.REF}} = \frac{f_{CLK6951.OUT0}}{f_{CLK6951.REF}} \quad (36)$$

Which can also be written as

$$X \bullet NDIV_{SYS6951} = NDIV_{CLK6951} \quad (37)$$

Where X is an integer. In most cases X = 1.

EZ204Sync Design Example

This design example will use the LTC6951Wizard to aid in the design process. Download LTC6951Wizard at http://www.linear.com/LTC6951Wizard.

This example assumes the following list of design inputs.

Reference
f_{REF} = 100MHz
CLOCK-LTC6951
$f_{CLK6951.OUT0}$ = 250MHz
$f_{CLK6951.OUT1}$ = 1GHz
$f_{CLK6951.OUT2}$ = 1GHz
$f_{CLK6951.OUT3}$ = 250MHz
$f_{CLK6951.OUT4}$ = 250MHz
RDIV = 1
RAO = 1
SYSREF-LTC6951
$f_{SYS6951.OUTX} = \frac{f_{CLK6951.OUT0}}{16}$ (38)
RDIV = 1
RAO = 1
Performance Optimization Request
Design CLOCK LTC6951 for low jitter.
Design SYSREF LTC6951 for low power.
Optimize skew between CLOCK LTC6951 to SYSREF LTC6951 outputs.
Part Placement And Routing
The LTC6954 and both LTC6951s will be placed on the top side of the board. For the most direct routing connect:
• LTC6954 $OUTX^+$ to LTC6951 IN^-
• LTC6954 $OUTX^-$ to LTC6951 IN^+
This creates a reference signal inversion at the LTC6951 inputs.

Clock LTC6951 Setup

Based on the EZ204Sync Design Rules and the above design inputs the following steps provide input conditions for the LTC6951Wizard.

Step 1: Design Rule 1 verification

The first design rule is met since the slowest clock frequency, 250MHz, is assigned to OUT0.

Step 2: Design Rules 2 and 3 verification

Calculate $f_{CLK6951.REF}$, MO_{6954}, and $NDIV_{CLK6951}$ to meet Design Rule 3.

The second design rule states that RDIV = 1 and RAO = 1. Referring to the LTC6951 data sheet the following two equations are provided when RAO = 1.

$$f_{CLK6951.PFD} = \frac{f_{CLK6951.REF}}{RDIV_{CLK6951}} \quad (39)$$

$$f_{CLK6951.PFD} = \frac{f_{CLK6951.OUT0}}{NDIV_{CLK6951}} \quad (40)$$

Referring to the LTC6954 data sheet, the following equation is provided.

$$f_{CLK6951.REF} = \frac{f_{REF}}{M0_{6954}} \quad (41)$$

Since $RDIV_{CLK6951}$ = 1, Equations 39, 40, and 41 can be rearranged as follows:

$$NDIV_{CLK6951} = M0_{6954} \bullet \frac{f_{CLK6951.OUT0}}{f_{REF}} \quad (42)$$

Substituting the known values for f_{REF} = 100MHz, $f_{CLK6951.OUT0}$ = 250MHz into Equation 41 results in

$$NDIV_{CLK6951} = M0_{6954} \bullet \frac{250MHz}{100MHz} \quad (43)$$

Which simplifies to

$$NDIV_{CLK6951} = 2.5 \bullet M0_{6954} \quad (44)$$

Based on the LTC6951 data sheet, to optimize for the lowest jitter possible, $f_{CLK6951.PFD}$ should be as large as possible, which allows for a wider bandwidth loop filter. This statement assumes the reference input signal noise level is not limiting the LTC6951's in-band noise performance. Therefore, solve Equation 44 for the least common integer multiple of NDIV and M0.

$NDIV_{CLK6951}$ = 5

$M0_{6954}$ = 2

Next solve Equation 41.

$f_{CLK6951.REF}$ = 50MHz

Step 3: Design input: optimize the CLOCK LTC6951 for low jitter.

Based on the LTC6951 data sheet the jitter performance is obtained by maximizing the LTC6951 ICP current.

$I_{CLK6951.CP}$ = 11.2mA

Step 4: Design input: optimize the skew performance between the CLOCK LTC6951 and SYSREF LTC6951.

The LTC6951 device to device skew is best when the LTC6951 register value FILT = 0.

SYSREF LTC6951 Setup

Step 5: Calculate the SYSREF frequencies, using Equation 38.

$f_{SYS6951.OUTX}$ = 15.625MHz

In other applications there may be more than one SYSREF frequency. If this is the case assign the lowest SYSREF frequency to OUT0 to meet Design Rule 1.

Step 6: Design Rules 2 and 3 verification

Calculate $f_{SYS6951.REF}$, $M2_{6954}$, and $NDIV_{SYS6951}$ to meet Design Rules 2 and 3.

Solve Equations 36 and 37 from Design Rule 3 for $f_{SYS6951.REF}$ and $NDIV_{SYS6951}$, by using known values for $f_{SYS6951.OUT0}$ = 15.625MHz, $f_{CLK6951.OUT0}$ = 250MHz, $f_{CLK6951.REF}$ = 50MHz, and $NDIV_{CLK6951}$ = 5

$$X \bullet \frac{f_{SYS6951.OUT0}}{f_{SYS6951.REF}} = \frac{f_{CLK6951.OUT0}}{f_{CLK6951.REF}} \quad (36)$$

By setting X = 1,

$1 \bullet (15.625MHz/f_{SYS6951.REF}) = (250MHz/50MHz)$

$f_{SYS6951.REF}$ = 3.125MHz

$$X \bullet NDIV_{SYS6951} = NDIV_{CLK6951} \quad (37)$$

$1 \bullet NDIV_{SYS6951}$ = 5

$NDIV_{SYS6951} = NDIV_{CLK6951}$ = 5

Rewriting Equation 41 for the SYSREF LTC6951, solve for M2.

$$M2_{6954} = \frac{f_{REF}}{f_{SYS6951.REF}} \quad (45)$$

$M2_{6954}$ = 100MHz/3.125MHz

$M2_{6954}$ = 32

Step 7: Design input: optimize the SYSREF LTC6951 for low power.

SYSREF is only used for the JESD204B alignment routines and is not a high performance clock. Therefore, the SYSREF LTC6951 can be placed in a lower power mode than the CLOCK LTC6951.

Set $I_{SYS6951.CP}$ = 2mA

LTC6951Wizard

This section demonstrates the LTC6951Wizard's ability to ease the register setting creation and loop filter design for both the SYSREF LTC6951 and CLOCK LTC6951. Under the LTC6951Wizard's Help Menu a Help Guide is provided that will aid in understanding the operations performed in this section.

The values calculated in Steps 1 to 7 and conditions provided at the start of this design example are summarized below for a quick reference.

LTC6951Wizard inputs for the SYSREF LTC6951 in Figure 35

$f_{SYS6951.REF}$ = 3.125MHz
$f_{SYS6951.OUTX}$ = 15.625MHz
$I_{SYS6951.CP}$ = 2mA
$NDIV_{SYS6951}$ = 5
$RDIV_{SYS6951}$ = 1
$FILT_{SYS6951}$ = 0
$RAO_{SYS6951}$ = 1

LTC6951Wizard inputs for the CLOCK LTC6951 in Figure 36

$f_{CLK6951.REF}$ = 50MHz
$f_{CLK6951.OUT0}$ = 250MHz
$f_{CLK6951.OUT1}$ = 1GHz
$f_{CLK6951.OUT2}$ = 1GHz
$f_{CLK6951.OUT3}$ = 250MHz
$f_{CLK6951.OUT4}$ = 250MHz
$I_{CLK6951.CP}$ = 11.2mA
$NDIV_{CLK6951}$ = 5
$RDIV_{CLK6951}$ = 1
$FILT_{CLK6951}$ = 0
$RAO_{CLK6951}$ = 1

Figures 35, 36 and 37 provide the remaining steps necessary to complete the LTC6951 portion of this design. Several steps in Figures 35 to 37 require the following additional information.

Importing Reference Noise

Refer to Appendix: Model Reference Noise for LTC6951Wizard Simulations, which describes how to import reference noise into the LTC6951Wizard and the impact of reference noise on loop filter calculations and output noise simulations. Examples 2 & 3 in the appendix create the reference noise profile for this example.

SYNCENX BITS

Ensure the CLOCK LTC6951 and SYSREF LTC6951 SYNCENX bits are set to a 1 for all signals that require synchronization. Refer to Figure 35 Step 2b and Figure 36 Step 15b.

Loop Filter Selection

Figure 35's step 11 selected Filter 1 for the SYSREF LTC6951. The simplest filter was selected for board space and cost reasons, because SYSREF jitter performance is not important.

Figure 36's step 24 selected Filter 2 for the CLOCK LTC6951. Through experimentation Filter 2 was found to be the best option to optimize performance and board space.

LTC6951 OINV Bit

Figure 35, step 6 and Figure 36, step 19 set the OINVX values for each output. The LTC6951 OINV value can be determined by referring to the schematics shown Figure 34. These OINVX settings, along with the DEL[X] settings above, program the LTC6951's SYSREF rising edge to start a ½ CLOCK cycle before its paired LTC6951 CLOCK's rising edge.

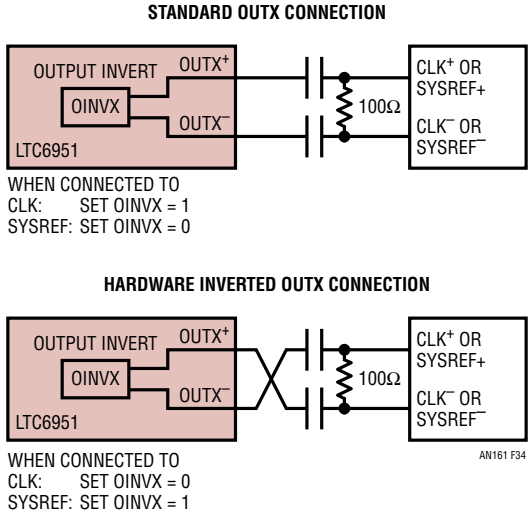


Figure 34. LTC6951 OINVX State

DELAY Settings

Figure 37, step 28 references Equation 46. Equation 46 calculates DEL[X] in PDIV cycles. The same DEL[x] value will be used for both SYSREF LTC6951 and CLOCK LTC6951 pairs. In Equation 46 the number 18 refers to the number of PDIV cycles when DEL[X] = 0. For more information refer to the LTC6951 data sheet OUTPUT SYNCHRONIZATION section titled Synchronization Events.

$$DEL[X] = Y \bullet M0_{SYS6951} - 18 + \frac{M0_{CLK6951}}{M[X]_{CLK6951}} - 1 \quad (46)$$

where Y is smallest integer that ensures DEL[X]>0.

As shown in Figure 37, step 28

DEL1 = 1 • 128 – 18 + (8/2) – 1 = 113

DEL2 = 1 • 128 – 18 + (8/2) – 1 = 113

DEL3 = 1 • 128 – 18 + (8/8) – 1 = 110

DEL4 = 1 • 128 – 18 + (8/8) – 1 = 110

If Equation 46 calculates a DEL[X] > 255 PDIV cycles (maximum delay setting), try increasing the LTC6951 PDIV value used in Figures 35 and 36. This will create a larger delay range. Repeat the steps in Figures 35 and 36 after the PDIV value is increased (check the box next to PDIV in the LTC6951Wizard Design tab to avoid auto-calculating a new PDIV).

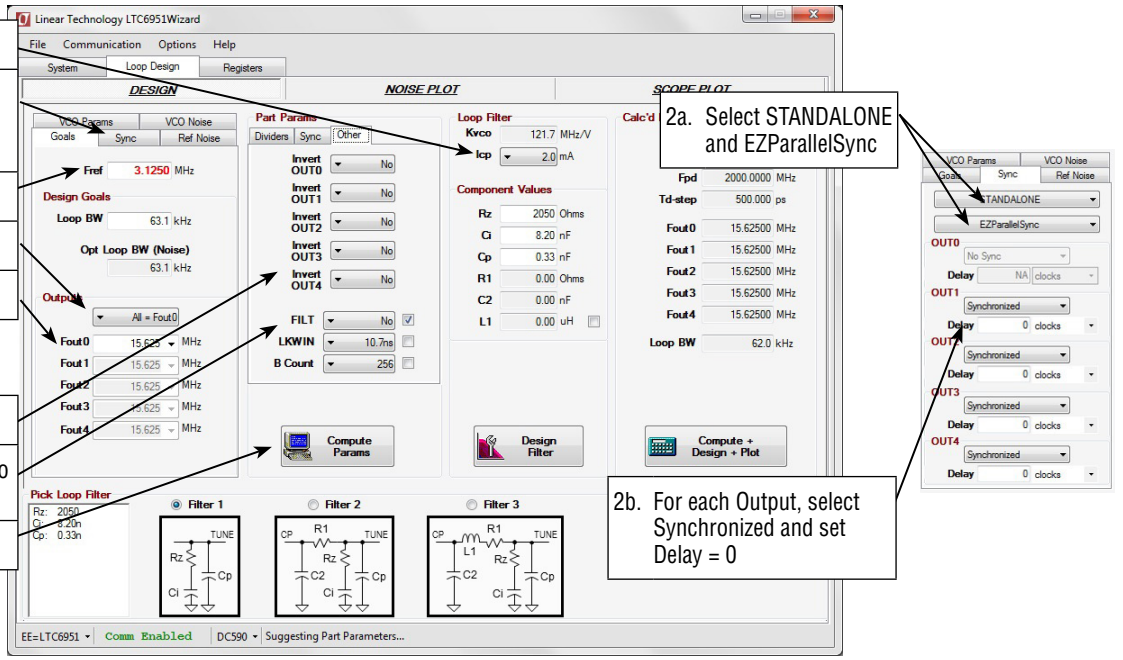
SYSREF LTC6951 ALCEN Bit

Figure 37, step 33 sets the SYSREF LTC6951 ALCEN bit high. If ALCHI or ALCLO are being monitored, then ALCEN is set high to reset the ALC flags during power-down and power-up routines for the SYSREF LTC6951. Refer to the Synchronization Routines section.

1. Set I_{CP} = 2mA
2. Select Sync tab. See Steps 2a and 2b on far right
3. Set Fref = 3.125MHz
4. Select All = Fout0
5. Set Fout = 15.625MHz
6. Set Invert OUTx = No
7. Set FILT = No, check box to lock value
8. Select Compute Params

2a. Select STANDALONE and EZParallelSync

2b. For each Output, select Synchronized and set Delay = 0



9. Verify R Div = 1 and N Div matches previous calculation
10. Double click Opt Loop BW (Noise) to copy to Loop BW
11. Select Filter 1 and Design Filter, then set Component Values to closest standard component values.
12. Record M0 Div Value.

    In this example: M0 Div = 128
13. Under File menu, select Save Settings.

    File name = SYSREF6951 (see far right)

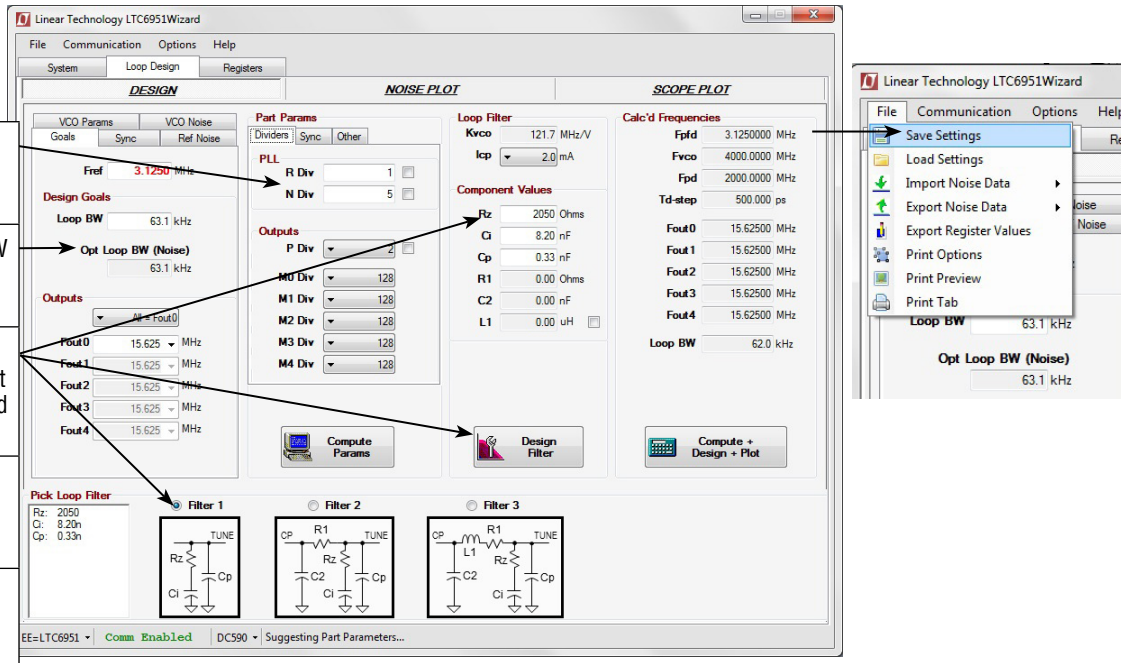
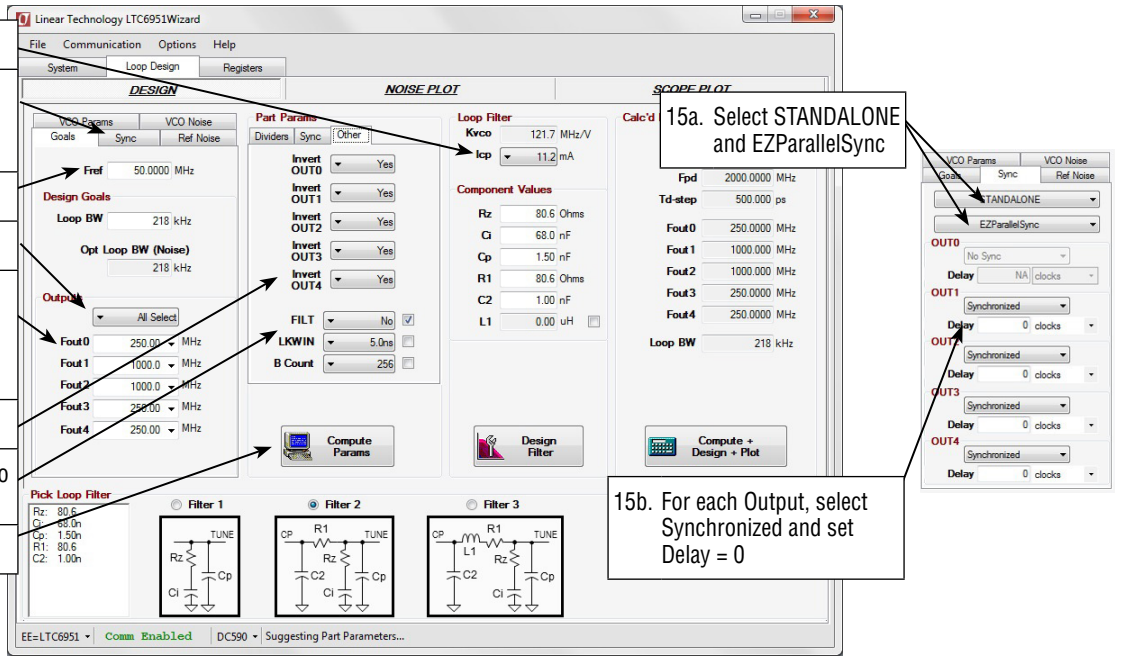


Figure 35. LTC6951Wizard: SYSREF LTC6951

14. Set I_{CP} = 11.2mA
15. Select Sync tab. See Steps 15a and 15b on far right
16. Set Fref = 50MHz
17. Select All Select
18. Set Fout0 = Fout3 = Fout4 = 250MHz Set Fout1 = Fout2 = 1000MHz
19 Set Invert OUTx = Yes
20. Set FILT = No, check box to lock value
21. Select Compute Params

15a. Select STANDALONE and EZParallelSync

15b. For each Output, select Synchronized and set Delay = 0



22. Verify R Div = 1 and N Div matches previous calculation
23. Double click Opt Loop BW (Noise) to copy to Loop BW
24. Select Filter 2 and Design Filter, then set Component Values to closest standard component values.
25. Record all Mx Div values.

    In this example:
    M0 Div = 8
    M3 Div = M4 Div = 8
    M1 Div = M2 Div = 2
26. Under File menu, select Save Settings.

    File name = CLKREF6951 (see far right)

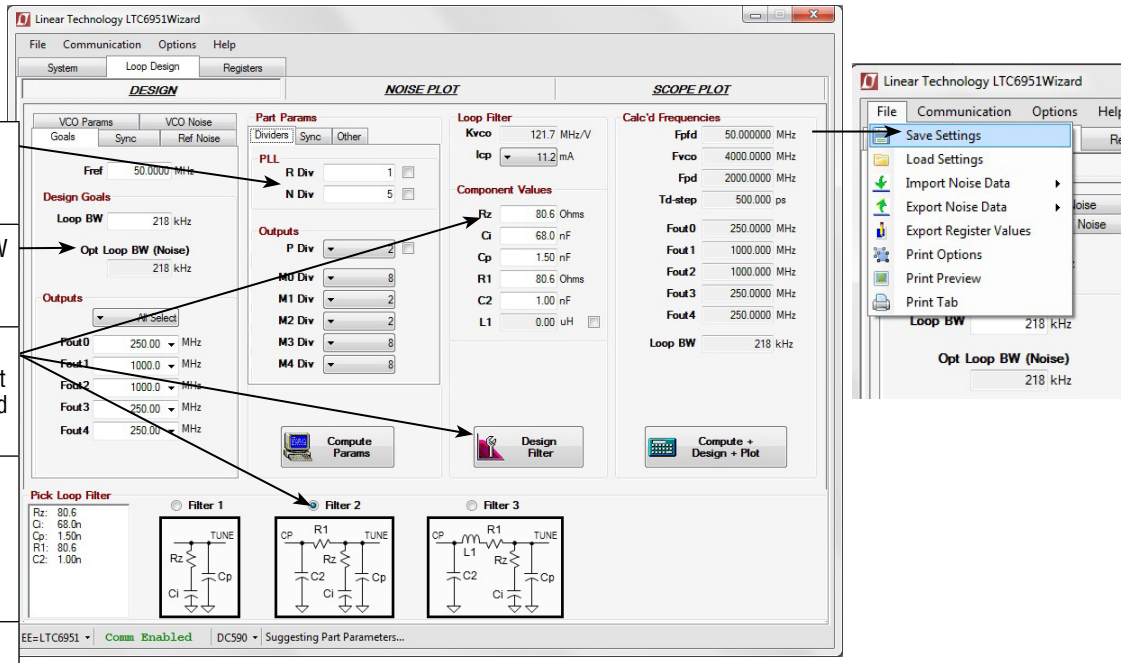


Figure 36. LTC6951Wizard: CLOCK LTC6951

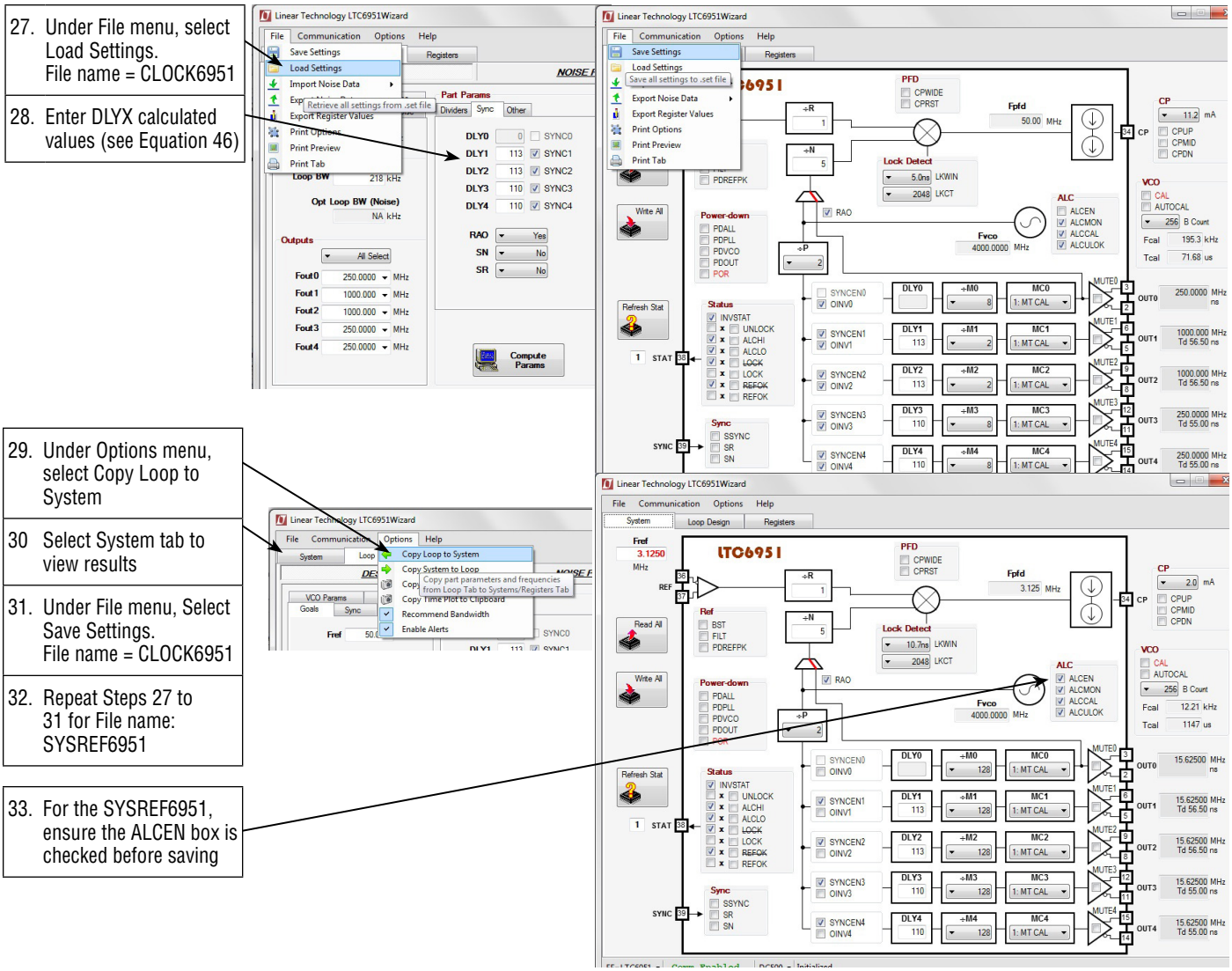


Figure 37. LTC6951Wizard Delay Settings and System Tab

LTC6954 Setup

Step 8: Design input: optimize the skew performance between the CLOCK LTC6951 and SYSREF LTC6951.

According to the LTC6954 data sheet best skew performance is obtained when all output divider settings are >1, as was determined in Steps 2 and 5 ($M0_{6954}$ = 2, $M2_{6954}$ = 32).

As an aside, if $M0_{6954}$ had been set to 1 and $M2_{6954}$ > 2 than the LTC6954 output skew would have degraded 25ps. One work around to improve this would be to double f_{REF} so $M0_{6954}$ could be set to 2.

Step 9: Verify LTC6954 output to LTC6951 connection.

The design inputs above required this example to use the Hardware Inverted Reference Connection for the distributed reference signals, see Figure 38. Hardware reference inversions create an extra design consideration, because both LTC6954 output reference signals will be synchronized on the falling edge, while the LTC6951 PLL locks to the reference rising edge, also shown in Figure 38.

To account for the hardware reference signal inversion, the LTC6954 output signals should be inverted. To invert the LTC6954 output, delay the LTC6954 outputs a ½

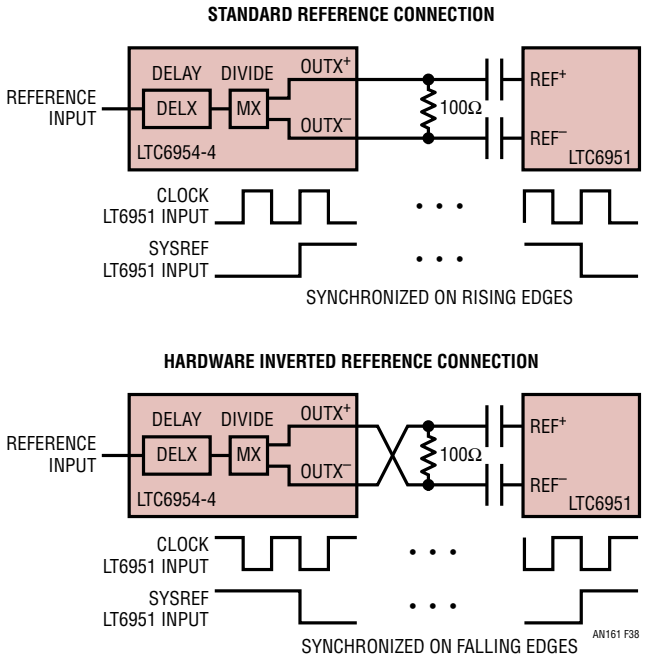


Figure 38. Reference Distribution Connection

output cycle using the LTC6954 delay bits. Inverting the LTC6954 outputs with the delay register creates a design constraint because inversion is only possible when the LTC6954 divider value is set to an even number. In this example both $M0_{6954}$ and $M2_{6954}$ are even numbers, which makes inverting the outputs possible with the LTC6954.

As an aside, if $M0_{6954}$ had been set to 1 the LTC6954 output could not be inverted, because the LTC6954 does not have an output inversion bit. To resolve this issue the designer would either need to change to the preferred schematic in Figure 38 or double f_{REF} which allows $M0_{6954}$ to be set to 2.

Table 3. Standard Reference Connection

	LTC6954 Register Settings	
LTC6954 OUTX	**MX**	**DELX**
To CLOCK LTC6951	Any	Y*
To SYSREF LTC6951	Any	Y*

*Y, integer, same value for all DELX

Table 4. Hardware Inverted Reference Connection

	LTC6954 Register Settings	
LTC6954 OUTX	**MX**	**DELX**
To CLOCK LTC6951	Even Number	$\frac{MX_{CLOCK}}{2} + Y^*$
To SYSREF LTC6951	Even Number	$\frac{MX_{SYSREF}}{2} + Y^*$

*Y, integer, same value for all DELX

Both reference signals are inverted for this example, refer to Table 4.

LTC6954.M0 = 2
LTC6954.DEL0 = LTC6954.M0/2 = 1
LTC6954.M2 = 32
LTC6954.DEL2 = LTC6954.M2/2 = 16

Step 10: LTC6954 register settings summarized:

$SYNCEN0_{6954}$ = 1
$M0_{6954}$ = 2
$DEL0_{6954}$ = 1
$PDIV0_{6954}$ = 0
$PDOUT0_{6954}$ = 0

$PDIV1_{6954}$ = 1

$SYNCEN0_{6954}$ = 1
$M2_{6954}$ = 32
$DEL2_{6954}$ = 16
$PDIV2_{6954}$ = 0
$PDOUT2_{6954} = PDALL_{SYS6951}$

Layout Recommendations

Minimizing SYSREF to CLOCK skew is recommended to ensure setup and hold times are met for proper JESD204B alignment sequence functionality. To minimize skew match electrical trace lengths in the CLOCK and SYSREF paths as shown in Equation 45 (refer to Figure 39).

$$L_{REF.CLK} + L_{CLK.OUTX} = L_{REF.SYS} + L_{SYS.OUTX} \quad (45)$$

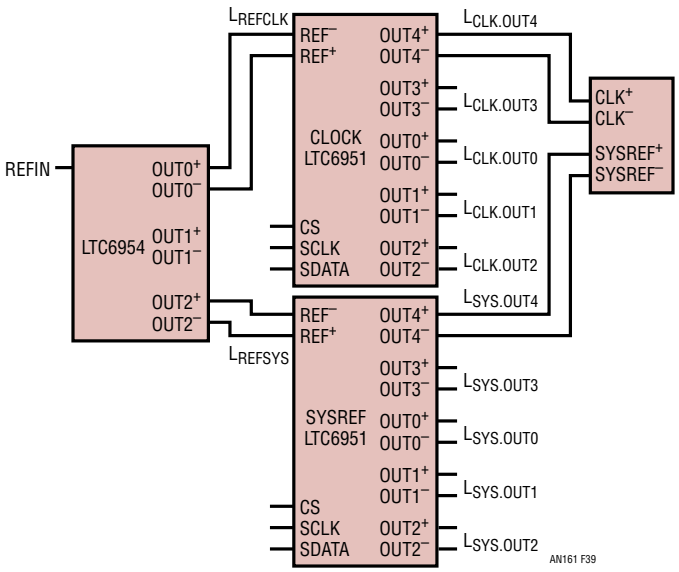


Figure 39. Trace Length Matching

Synchronization Routines

On initial power-up:

1. Program LTC6954 and LTC6951 SPI registers
2. Toggle LTC6954 SYNC pin (minimum 1ms)
3. Wait for LTC6951 bias voltages to stabilize
4. Calibrate all LTC6951 VCOs
5. Toggle all LTC6951 SPI SSYNC bits or SYNC pins (minimum 1ms)

Power down SYSREF after JESD204b alignment sequence is complete:

1. Power down SYSREF LTC6951 (PDALL = 1)
2. Power down LTC6954 OUT2 (PD_OUT2 = 1)

If JESD204 requires re-alignment:

1. Power up SYSREF LTC6951 (PDALL = 0)
2. Power up LTC6954 OUT2 (PD_OUT2 = 0)
3. Toggle SYSREF LTC6951 SPI SSYNC bit or SYNC pin (minimum 1ms)

Expandable Solution

The EZ204Sync solution is infinitely expandable. As shown in Figure 40 the EZ204Sync design example can be repeated by adding an EZSync CONTROLLER to distribute the reference to multiple LTC6954s.

For further expansion it is possible to cascade additional reference distribution stages between Stage 1 and Stage 2. Refer to the EZSync Design Example for more details.

When designing a multi-stage reference divider/distribution network, take into account the additive properties of

- channel to channel skew
- noise floor at frequency offsets less than the CLOCK LTC6951 loop filter's pass-band. Refer to Appendix: Model Reference Noise for LTC6951 Wizard Simulations.

If some LTC6951's can power down during operation, then selecting reference distribution parts with the ability to power down individual outputs can save additional power.

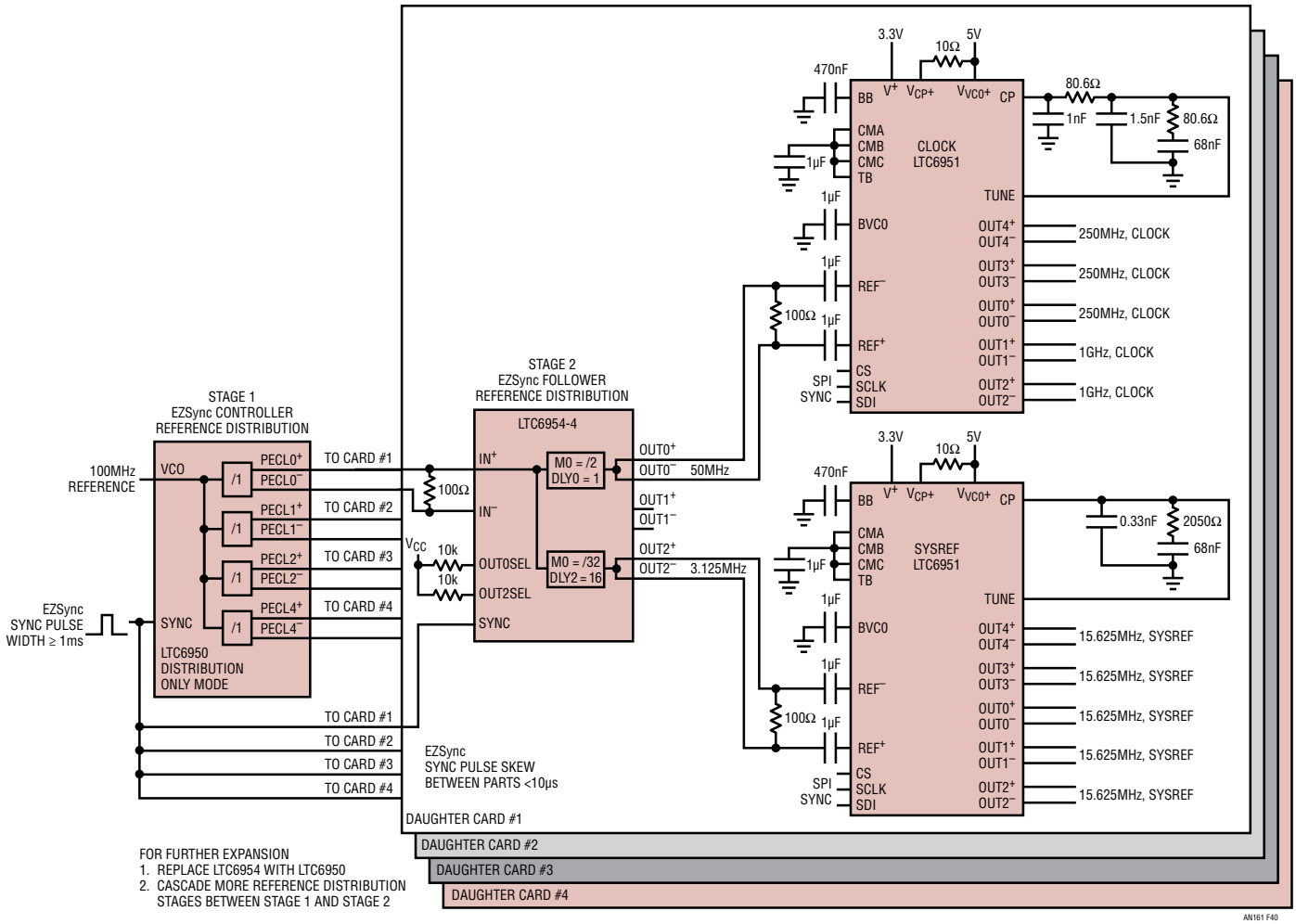


Figure 40. EZ204Sync Expandable Solution

APPENDIX: MODEL REFERENCE NOISE FOR LTC6951Wizard SIMULATIONS

Reference Noise

Noise profiles of different reference oscillators can change the calculated results of the loop filter bandwidth and LTC6951 jitter performance. Significant changes in these results occur when the reference oscillator noise floor is greater than the LTC6951's normalized in-band noise (see Figure 41). In-band noise refers to the noise at offset frequencies less than the loop filters bandwidth. Refer to the LTC6951 data sheet for more information on reference noise characteristics.

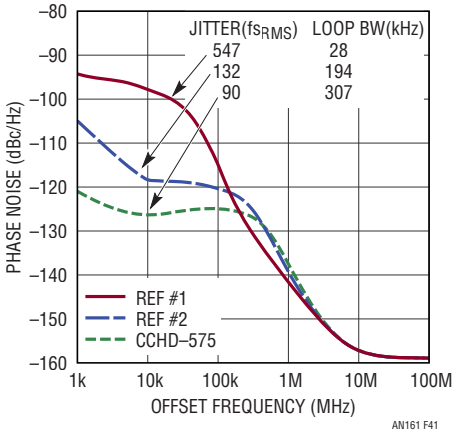


Figure 41. Comparing LTC6951Wizard Results with Different Reference Profiles

When designing loop filters and simulating jitter performance with the LTC6951Wizard, it is recommended to import the noise profile of the desired reference network for best results. Figures 42 and 43 provide the information required to import a reference noise file into the LTC6951Wizard. The remainder of this appendix provides information and examples to aid in estimating the phase noise of a reference distribution network. The examples in this appendix are based off the reference distribution networks provided in the LTC6951 Sync Manual's design examples.

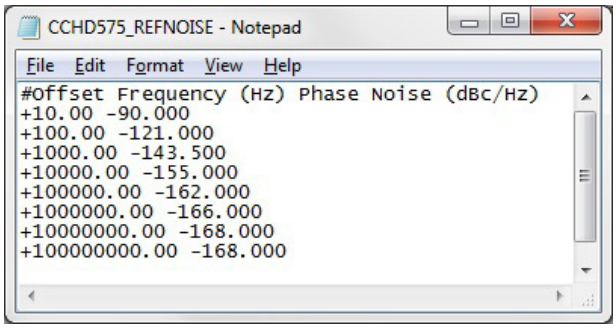


Figure 42. LTC6951Wizard Reference Noise File Format

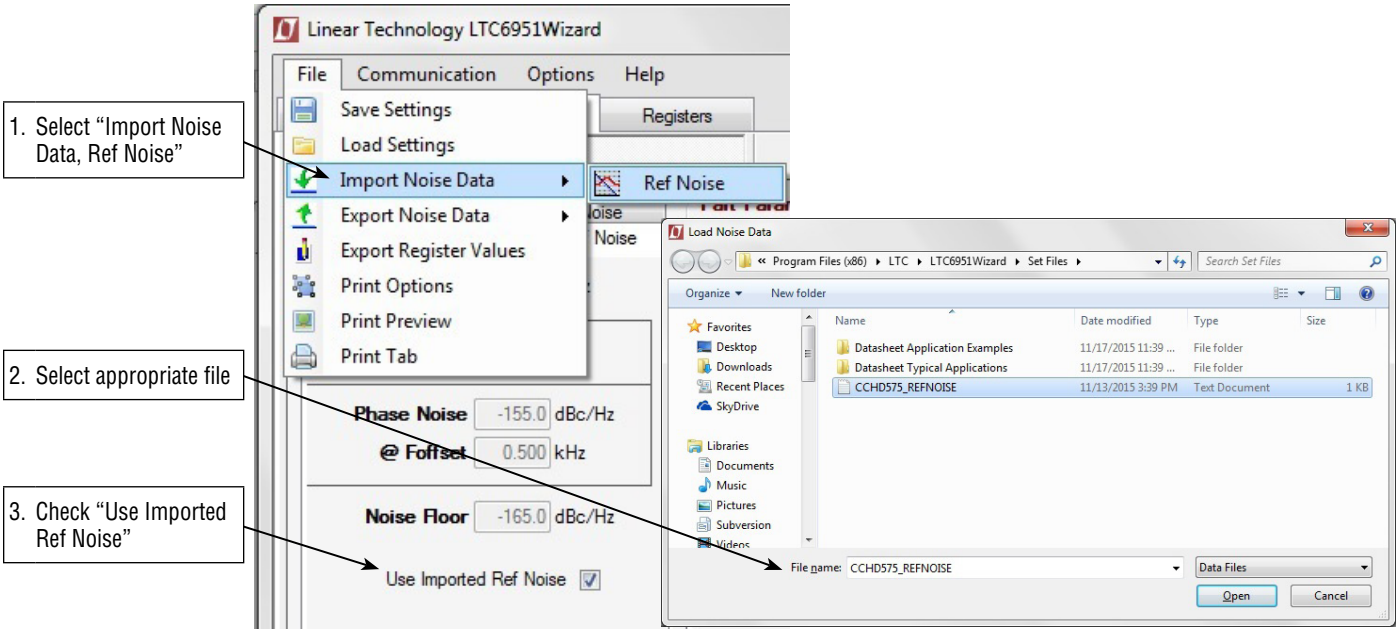


Figure 43. Importing Reference Noise Profiles into LTC6951Wizard

Distributed Reference Noise

A distributed reference's phase noise is the product of the reference phase noise and the additive phase noise of the fanout buffer.

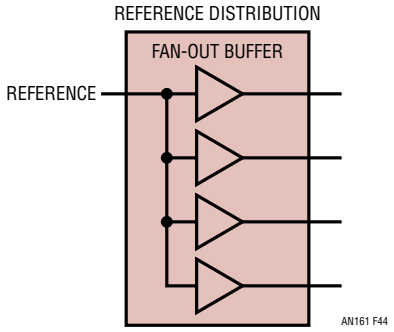


Figure 44. Reference Distribution

Equation 47 calculates the estimated total phase noise of the reference distribution network.

$$PN_{TOT}(X) = 20 \bullet LOG\sqrt{\left(10^{\left(\frac{PN_{REF}(X)}{20}\right)}\right)^2 + \left(10^{\left(\frac{PN_{DIS}(X)}{20}\right)}\right)^2} \quad (47)$$

Where:

PN_{DIS}, Reference distribution IC phase noise at f_{REF}

PN_{REF}, Reference phase noise at f_{REF}

PN_{TOT}, Combined PN_{DIS} and PN_{REF} phase noise at f_{REF}

X, offset frequency in Hz

To minimize the fanout buffer phase noise contribution, choose a fanout buffer whose additive in-band phase noise is 6dB lower than the in-band reference phase noise. In most cases the out-of-band phase noise can be neglected, since the PLL's loop filter removes out-of-band noise.

Low noise reference and fanout buffer data sheets usually provide phase noise plots at specific carrier frequencies. From these phase noise plots it is possible to estimate the combined phase noise profile of the reference and fanout buffer. If phase noise plots at the desired frequency are not available, Equation 48 scales the phase noise from the nearest available frequency to the desired frequency.

$$PN_Y(X) = PN_{Y-DS}(X) - 20 \bullet LOG\left(\frac{f_{Y-DS}}{f_{REF}}\right) \quad (48)$$

Where:

f_{Y-DS}, carrier frequency of phase noise curve (PN_{Y-DS})

f_{REF}, desired reference frequency at LTC6951 reference input

PN_Y, phase noise at f_{REF}

PN_{Y-DS}, phase noise at f_{Y-DS} provided in reference's data sheet or distribution IC's data sheet

X, offset frequency in Hz

Example 1: Distributed Reference (ParallelSync)

Estimate the reference phase noise curve to import into LTC6951Wizard using Crystek's CCHD-575 100MHz reference and the LTC6954-4 as the LVDS fanout buffer.

Step 1: Refer to vendor's data sheets for phase noise curves provided in table below.

Offset Frequency (Hz)	CCHD-575 100MHz (dBc/Hz)	LTC6954-4 122.88MHz (dBc/Hz)
10	–90	–140
100	–121	–148
1k	–143.5	–156
10k	–155	–161
100k	–162	–162
1M	–166	–162
10M	–168	–162
100M	–168	–162

Step 2: Use Equation 48 to adjust the LTC6954-4 phase noise profile from 122.88MHz to 100MHz:

$$PN_{6954}(X) = PN_{6954-DS}(X) - 20 \bullet LOG\left(\frac{122.88MHz}{100MHz}\right)$$

$$PN_{6954}(X) = PN_{6954\text{-}DS}(X) - 1.8$$

Offset Frequency (Hz)	CCHD-575 100MHz (dBc/Hz)	LTC6954-4 100MHz (dBc/Hz)
10	–90	–141.8
100	–121	–149.8
1k	–143.5	–157.8
10k	–155	–162.8
100k	–162	–162.8
1M	–166	–163.8
10M	–168	–163.8
100M	–168	–163.8

Step 3: Calculate the total reference distribution network phase noise using the values in Step 2 with Equation 47.

Offset Frequency (Hz)	CCHD-575 100MHz (dBc/Hz)	LTC6954-4 100MHz (dBc/Hz)	PN_{TOT} 100MHz (dBc/Hz)
10	–90	–141.8	–90
100	–121	–149.8	–121
1k	–143.5	–157.8	–143.3
10k	–155	–162.8	–154.3
100k	–162	–162.8	–159.8
1M	–166	–163.8	–161.8
10M	–168	–163.8	–162.4
100M	–168	–163.8	–162.4

Step 4: Create an LTC6951Wizard reference input file (see Figure 42) from data in the Offset Frequency and PN_{TOT} columns in Step 3.

Divided And Distributed Reference Noise

Selecting a reference divide and distribution IC has the same concerns as selecting a fanout buffer. Additional noise effects with this architecture due to aliasing can be observed as the divider value increases. As a result the accuracy of the following examples may degrade with larger divide values.

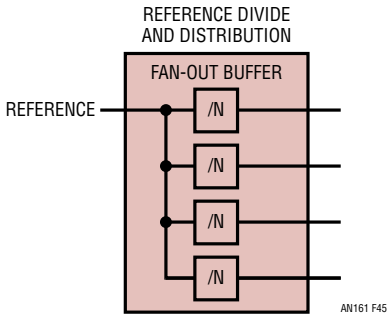


Figure 45. Reference Divide and Distribution

Example 2: Divided And Distributed Reference (EZParallelSync and EZ204Sync CLOCK6951)

Estimate the reference phase noise curve to import into LTC6951Wizard using Crystek's CCHD-575 100MHz reference and the LTC6954-4 as the LVDS reference divide/distribution IC. The LTC6954-4 divider will be set to 2 to create a 50MHz reference at the LTC6951 input.

Step 1: Refer to vendor's data sheets for phase noise curves provided in table below

Offset Frequency (Hz)	CCHD-575 100MHz (dBc/Hz)	LTC6954-4 30.72MHz (122.88MHz/4) (dBc/Hz)
10	–90	–140
100	–121	–152
1k	–143.5	–161
10k	–155	–166
100k	–162	–166
1M	–166	–166
10M	–168	–166
100M	–168	–166

Step 2: Use Equation 48 to adjust the CCHD-575 and LTC6954-4 phase noise profiles to 50MHz

CCHD-575:

$$PN_{CCHD}(X) = PN_{CCHD-DS}(X) - 20 \bullet LOG\left(\frac{100MHz}{50MHz}\right)$$

$$PN_{CCHD}(X) = PN_{CCHD\text{-}DS}(X) - 6$$

LTC6954-4:

$$PN_{6954}(X) = PN_{6954-DS}(X) - 20 \bullet LOG\left(\frac{30.72MHz}{50MHz}\right)$$

$$PN_{6954}(X) = PN_{6954\text{-}DS}(X) + 4.2$$

Offset Frequency (Hz)	CCHD-575 50MHz (dBc/Hz)	LTC6954-4 50MHz (dBc/Hz)
10	–96	–135.8
100	–127	–147.8
1k	–149.5	–156.8
10k	–161	–161.8
100k	–168	–161.8
1M	–172	–161.8
10M	–174	–161.8
100M	–174	–161.8

Step 3: Calculate the total reference distribution network phase noise using the values in step 2 with Equation 47.

Offset Frequency (Hz)	CCHD-575 50MHz (dBc/Hz)	LTC6954-4 50MHz (dBc/Hz)	PN_{TOT} 50MHz (dBc/Hz)
10	–96	–135.8	–96
100	–127	–143.8	–126.9
1k	–149.5	–151.8	–148.8
10k	–161	–156.8	–158.4
100k	–168	–157.8	–160.9
1M	–172	–157.8	–161.4
10M	–174	–157.8	–161.5
100M	–174	–157.8	–161.5

Step 4: Create an LTC6951Wizard reference input file (see Figure 42) from data in the Offset Frequency and PN_{TOT} columns in Step 3.

Example 3: Divided and Distributed Reference (EZ204Sync SYSREF6951)

Estimate the reference phase noise curve to import into LTC6951Wizard using Crystek's CCHD-575 100MHz reference and the LTC6954-4 as the LVDS reference divide/distribution IC. The LTC6954-4 divider will be set to 32 to create a 3.125MHz reference at the SYSREF LTC6951 input.

Step 1: Refer to vendor's data sheets for phase noise curves provided in table below.

Offset Frequency (Hz)	CCHD-575 100MHz (dBc/Hz)	LTC6954-4 30.72MHz (122.88MHz/4) (dBc/Hz)
10	–90	–140
100	–121	–152
1k	–143.5	–161
10k	–155	–166
100k	–162	–166
1M	–166	–166
10M	–168	–166
100M	–168	–166

Step 2: Use Equation 48 to adjust the CCHD-575 and LTC6954-4 phase noise profiles to 3.125MHz

CCHD-575:

$$PN_{CCHD}(X) = PN_{CCHD-DS}(X) - 20 \bullet LOG\left(\frac{100MHz}{3.125MHz}\right)$$

$$PN_{CCHD}(X) = PN_{CCHD\text{-}DS}(X) - 30.1$$

LTC6954-4

$$PN_{6954}(X) = PN_{6954-DS}(X) - 20 \bullet LOG\left(\frac{30.72MHz}{3.125MHz}\right)$$

$$PN_{6954}(X) = PN_{6954\text{-}DS}(X) - 19.9$$

Offset Frequency (Hz)	CCHD-575 3.125MHz (dBc/Hz)	LTC6954-4 3.125MHz (dBc/Hz)
10	–120.1	–159.9
100	–151.1	–171.9
1k	–173.6	–180.9
10k	–185.1	–185.9
100k	–192.1	–185.9
1M	–196.1	–185.9
10M	–198.1	–185.9
100M	–198.1	–185.9

Step 3: Calculate the total reference distribution network phase noise using the values in step 2 with Equation 47.

Offset Frequency (Hz)	CCHD-575 3.125MHz (dBc/Hz)	LTC6954-4 3.125MHz (dBc/Hz)	PN_{TOT} 3.125MHz (dBc/Hz)
10	–120.1	–159.9	–120.1
100	–151.1	–171.9	–151.1
1k	–173.6	–180.9	–172.9
10k	–185.1	–185.9	–182.5
100k	–192.1	–185.9	–184.9
1M	–196.1	–185.9	–185.5
10M	–198.1	–185.9	–185.6
100M	–198.1	–185.9	–185.6

Step 4: Create an LTC6951Wizard reference input file (see Figure 42) from data in the Offset Frequency and PNTOT columns in Step 3.

LT 0317 • PRINTED IN USA