# Ultralow Jitter, 4.5GHz PLL with 11 Outputs and JESD204B/JESD204C Support

## FEATURES

- **JESD204B/C, Subclass 1 SYSREF Signal Generation**
- **Low Noise Integer-N PLL**
- **Additive Output Jitter < 6fs$_{RMS}$ (Integration BW = 12kHz to 20MHz, f = 4.5GHz)**
- **Additive Output Jitter 65fs$_{RMS}$ (ADC SNR Method)**
- **EZSync™, ParallelSync™ Multichip Synchronization**
- –229dBc/Hz Normalized In-Band Phase Noise Floor
- –281dBc/Hz Normalized In-Band 1/f Noise
- Eleven Independent, Low Noise Outputs with Programmable Coarse Digital and Fine Analog Delays
- Flexible Outputs Can Serve as Either a Device Clock or SYSREF Signal
- Reference Input Frequency up to 500MHz
- LTC6952Wizard™ Software Design Tool Support
- –40°C to 125°C Operating Junction Temperature Range

## APPLICATIONS

- High Performance Data Converter Clocking
- Wireless Infrastructure
- Test and Measurement

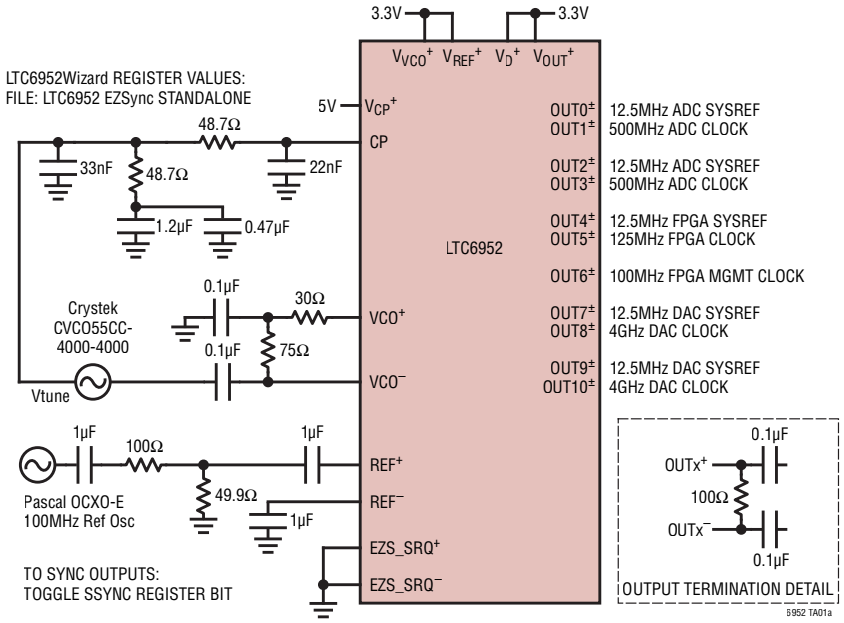
## DESCRIPTION

The LTC®6952 is a high performance, ultralow jitter, JESD204B/C clock generation and distribution IC. It includes a Phase Locked Loop (PLL) core, consisting of a reference divider, phase-frequency detector (PFD) with a phase-lock indicator, ultralow noise charge pump and integer feedback divider. The LTC6952's eleven outputs can be configured as up to five JESD204B/C subclass 1 device clock/SYSREF pairs plus one general purpose output, or simply eleven general purpose clock outputs for non-JESD204B/C applications. Each output has its own individually programmable frequency divider and output driver. All outputs can also be synchronized and set to precise phase alignment using individual coarse half-cycle digital delays and fine analog time delays.

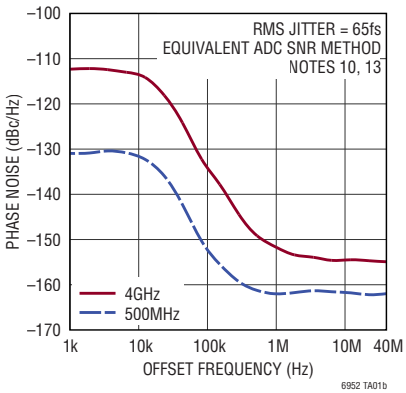
For applications requiring more than eleven total outputs, multiple LTC6952s can be connected together using the EZSync or ParallelSync synchronization protocols.



## TYPICAL APPLICATION




**LTC6952 Phase Noise**



# TABLE OF CONTENTS

# TABLE OF CONTENTS

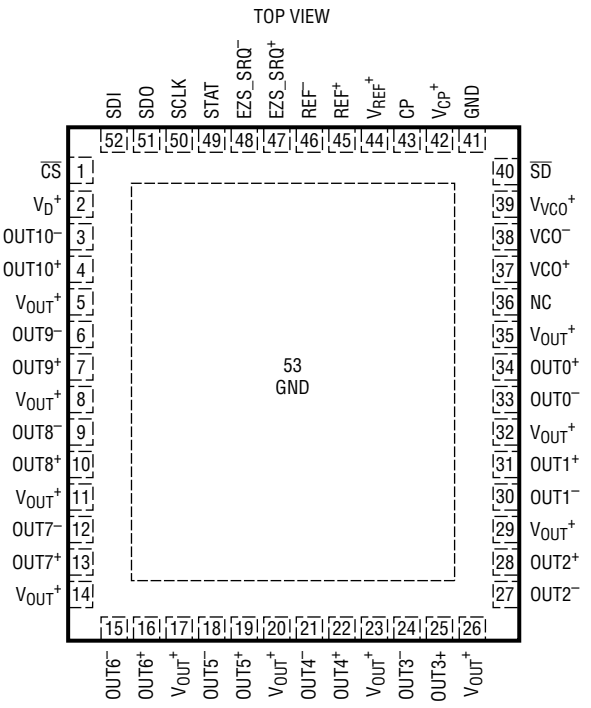
## ABSOLUTE MAXIMUM RATINGS

**(Note 1)**

Supply Voltages

$V^+$ ($V_{REF}^+$, $V_{VCO}^+$, $V_D^+$, $V_{OUT}^+$) to GND .....3.6V
$V_{CP}^+$ to GND.....5.5V
Voltage on CP Pin .....GND - 0.3V to $V_{CP}^+$ + 0.3V
Voltage on all other Pins ..... GND - 0.3V to $V^+$ + 0.3V
Current into OUTx+, OUTx–, (x = 0 to 10).....±25mA
Operating Junction Temperature Range, $T_J$ (Note 2)
LTC6952I ..... –40°C to 125°C
Junction Temperature, $T_{JMAX}$..... 130°C
Storage Temperature Range ..... –65°C to 150°C

## PIN CONFIGURATION




UKG PACKAGE
52-LEAD (7mm × 8mm) PLASTIC QFN

$T_{JMAX}$ = 130°C, $\theta_{JA}$ = 31°C/W, $\theta_{JC}$ = 2°C/W
EXPOSED PAD (PIN #53) IS GND, MUST BE SOLDERED TO PCB

## ORDER INFORMATION

| LEAD FREE FINISH | TAPE AND REEL | PART MARKING | PACKAGE DESCRIPTION | JUNCTION TEMPERATURE RANGE |
|---|---|---|---|---|
| LTC6952IUKG#PBF | LTC6952IUKG#TRPBF | 6952 | 52-Lead (7mm × 8mm) Plastic QFN | –40°C to 125°C |

Contact the factory for parts specified with wider operating temperature ranges.

Tape and reel specifications. Some packages are available in 500 unit reels through designated sales channels with #TRMPBF suffix.

## ELECTRICAL CHARACTERISTICS

**The ● denotes the specifications which apply over the full operating junction temperature range, otherwise specifications are at $T_A$ = 25°C. $V_{REF}^+ = V_D^+ = V_{VCO}^+ = V_{OUT}^+ = 3.3V$, $V_{CP}^+ = 5V$ unless otherwise specified (Note 2). All voltages are with respect to GND.**

| SYMBOL | PARAMETER | CONDITIONS | | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|---|
| **Reference Inputs (REF⁺, REF⁻)** | | | | | | | |
| $f_{REF}$ | Input Frequency | | ● | 1 | | 500 | MHz |
| $V_{REF}$ | Input Signal Level | Single-Ended | ● | 0.5 | 2 | 2.7 | $V_{P-P}$ |
| | Minimum Input Slew Rate | | | | 20 | | V/µs |
| | Input Duty Cycle | | | | 50 | | % |
| | Self-bias Voltage | | ● | 1.65 | 1.85 | 2.25 | V |
| | Minimum Input Signal Detected (REFOK=1) | PDREFPK = 0, $f_{REF}$ = 10MHz, Single-Ended Sine Wave | ● | 350 | | | $mV_{P-P}$ |
| | Maximum Input Signal Not Detected (REFOK=0) | PDREFPK = 0, $f_{REF}$ = 10MHz, Single-Ended Sine Wave | ● | | | 100 | $mV_{P-P}$ |
| | Input Resistance | Differential | ● | 2.4 | 3.8 | 5.8 | kΩ |
| | Input Capacitance | Differential | | | 1.3 | | pF |
| **VCO Input (VCO⁺, VCO⁻)** | | | | | | | |
| $f_{VCO}$ | Frequency Range | | ● | | | 4500 | MHz |
| | Input Signal Level | $R_Z$ = 50Ω, Single-Ended | ● | 0.25 | 0.8 | 1.6 | $V_{P-P}$ |
| | | | ● | –8 | 2 | 8 | dBm |
| | Self-Bias Voltage | | | | 2.05 | | V |
| | Input Common Mode Voltage | 800m$V_{P-P}$ Differential Input | ● | 1.6 | | 2.7 | V |
| | Input Duty Cycle | | | | 50 | | % |
| | Minimum Input Slew Rate | | | | 100 | | V/µs |
| | Minimum Input Signal Detected (VCOOK = 1) | PDVCOPK = 0, $f_{VCO}$ = 100MHz, Single-Ended Sine Wave | ● | 250 | | | $mV_{P-P}$ |
| | Maximum Input Signal Not Detected (VCOOK = 0) | PDVCOPK = 0, $f_{VCO}$ = 100MHz, Single-Ended Sine Wave | ● | | | 40 | $mV_{P-P}$ |
| | Input Resistance | Differential | | | 250 | | Ω |
| | Input Capacitance | Differential | | | 1.0 | | pF |
| **CMOS SYNC/SYSREF Request Input (EZS_SRQ⁺ Only)** | | | | | | | |
| | High-level Input Voltage | EZS_SRQ⁻ tied to GND | ● | 1.3 | | | V |
| | Low-level Input Voltage | EZS_SRQ⁻ tied to GND | ● | | | 0.6 | V |
| | Input Voltage Hysteresis | EZS_SRQ⁻ tied to GND | | | 200 | | mV |
| | Input Current | EZS_SRQ⁻ tied to GND | ● | | | ±1 | µA |
| **Differential SYNC/SYSREF Request Inputs (EZS_SRQ⁺ and EZS_SRQ⁻)** | | | | | | | |
| | Input Signal Level | | ● | 0.5 | 0.8 | 2.7 | $V_{P-P}$ |
| | Self-bias Voltage | | ● | 1.6 | 2.1 | 2.5 | V |
| | Input Common Mode Voltage | 800m$V_{P-P}$ Differential Input | | 1.5 | | 3.0 | V |
| | Input Resistance | Differential | | | 53 | | kΩ |
| | Input Capacitance | Differential | | | 1 | | pF |
| **Phase/Frequency Detector (PFD)** | | | | | | | |
| $f_{PFD}$ | Input Frequency | | ● | | | 167 | MHz |
| **Charge Pump** | | | | | | | |
| $I_{CP}$ | Output Current Range | 20 Settings (see Table 6) | | 0.423 | | 11.2 | mA |
| | Output Current Source/Sink Accuracy | $I_{CP}$ = 423µA to 4.0mA, V(CP) = $V_{CP}^+/2$ | | | | ±8 | % |
| | | $I_{CP}$ = 4.72µA to 11.2mA, V(CP) = $V_{CP}^+/2$ | | | | ±6 | % |
| | Output Current Source/Sink Matching | $I_{CP}$ = 423µA to 4.0mA, V(CP) = $V_{CP}^+/2$ | | | | ±5 | % |
| | | $I_{CP}$ = 4.72mA to 11.2mA, V(CP) = $V_{CP}^+/2$ | | | | ±2.5 | % |
| | Output Current vs Output Voltage Sensitivity | (Note 7) | ● | | 0.25 | 1.0 | %/V |

## ELECTRICAL CHARACTERISTICS

**The ● denotes the specifications which apply over the full operating junction temperature range, otherwise specifications are at $T_A$ = 25°C. $V_{REF}^+ = V_D^+ = V_{VCO}^+ = V_{OUT}^+ = 3.3V$, $V_{CP}^+ = 5V$ unless otherwise specified (Note 2). All voltages are with respect to GND.**

| SYMBOL | PARAMETER | CONDITIONS | | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|---|
| | Output Current vs Temperature | V(CP) = $V_{CP}^+$/2 | ● | | 170 | | ppm/°C |
| | Output Hi-Z Leakage Current | $I_{CP}$ = 423µA (Note 7) | | | 0.5 | | nA |
| | | $I_{CP}$ = 11.2mA (Note 7) | | | 5 | | nA |
| $V_{MID}$ | Mid-supply Output Bias Ratio | Referred to ($V_{CP}^+$ - GND) | | | 0.48 | | V/V |
| **Reference Divider (R)** | | | | | | | |
| R | Divide Range | All Integers Included | ● | 1 | | 1023 | Counts |
| **VCO Divider (N)** | | | | | | | |
| N | Divide Range | All Integers Included | ● | 1 | | 65535 | Counts |
| **Digital Pin Specifications** | | | | | | | |
| $V_{IH}$ | High-level Input Voltage | $\overline{CS}$, SDI, SCLK, $\overline{SD}$ | ● | 1.55 | | | V |
| $V_{IL}$ | Low-level Input Voltage | $\overline{CS}$, SDI, SCLK, $\overline{SD}$ | ● | | | 0.8 | V |
| $V_{IHYS}$ | Input Voltage Hysteresis | $\overline{CS}$, SDI, SCLK, $\overline{SD}$ | | | 250 | | mV |
| | Input Current | $\overline{CS}$, SDI, SCLK, $\overline{SD}$ | ● | | | ±1 | µA |
| $I_{OH}$ | High-level Output Current | SDO and STAT, $V_{OH} = V_D^+$ - 400mV | ● | | –3.3 | –1.9 | mA |
| $I_{OL}$ | Low-level Output Current | SDO and STAT, $V_{OL}$ = 400mV | ● | 2.0 | 3.4 | | mA |
| | SDO Hi-Z Current | | ● | | | ±1 | µA |
| **Digital Timing Specifications** | | | | | | | |
| $t_{CKH}$ | SCLK High Time | | ● | 25 | | | ns |
| $t_{CKL}$ | SCLK Low Time | | ● | 25 | | | ns |
| $t_{CSS}$ | $\overline{CS}$ Setup Time | | ● | 10 | | | ns |
| $t_{CSH}$ | $\overline{CS}$ High Time | | ● | 10 | | | ns |
| $t_{CS}$ | SDI to SCLK Setup Time | | ● | 6 | | | ns |
| $t_{CH}$ | SDI to SCLK Hold Time | | ● | 6 | | | ns |
| $t_{DO}$ | SCLK to SDO Time | To $V_{IH}/V_{IL}$/Hi-Z with 30pF load | ● | | | 16 | ns |
| **EZS_SRQ Timing Specifications (See Figure 2)** | | | | | | | |
| $t_{SRQH}$ | EZS_SRQ High Time | | ● | 1 | | | ms |
| $t_{SRQL}$ | EZS_SRQ Low Time | | ● | 1 | | | ms |
| | EZS_SRQ Skew, Part to Part | SRQMD = 0, PARSYNC = 0 | | | | 10 | µs |
| $t_{SS}$ | EZS_SRQ to REF Setup Time | PARSYNC = 1, CMOS EZS_SRQ | ● | 1 | | | ns |
| | | PARSYNC = 1, Differential EZS_SRQ | ● | 0.5 | | | |
| $t_{SH}$ | EZS_SRQ to REF Hold Time | PARSYNC = 1, CMOS EZS_SRQ | ● | 1 | | | ns |
| | | PARSYNC = 1, Differential EZS_SRQ | ● | 0.5 | | | |
| **Output Dividers (M0, M1, M2, M3, M4, M5, M6, M7, M8, M9 and M10)** | | | | | | | |
| Mx | Output Divider Range (x = 0 to 10) | Mx = P × $2^N$, where P = 1 to 32 all integers, N = 0 to 7 | ● | 1 | | 4096 | Counts |
| DDELx | Output Digital Delay (x = 0 to 10) | ½ VCO Cycles (Note 3) | ● | 0 | | 4095 | ½ Cycles |
| $t_{ADELx}$ | Output Analog Delay Time (x = 0 to 10) | ADELx = 0 | | | 0 | | ps |
| | | ADELx = 1, $f_{OUTx} \leq$ 300MHz | | | 90 | | |
| | | ADELx = 63, $f_{OUTx} \leq$ 300MHz | | | 1100 | | |
| | Output Analog Delay Time (x = 0 to 10), Step Size | ADELx = 1 to 31, $f_{OUTx} \leq$ 300MHz | | | 11 | | ps |
| | | ADELx = 32 to 63, $f_{OUTx} \leq$ 300MHz | | | 26 | | |
| | Output Analog Delay (x = 0 to 10) | 300MHz ≤ $f_{OUTx} \leq$ 2.25GHz | | | Note 4 | | |
| | Maximum Output Frequency for Analog Delay | | | | 2.25 | | GHz |
| | Temperature Coefficient of Analog Delay Time | ADELx = 1 to 31 | ● | | 0.06 | | %/°C |
| | | ADELx = 32 to 63 | ● | | 0.06 | | |

## ELECTRICAL CHARACTERISTICS

**The ● denotes the specifications which apply over the full operating junction temperature range, otherwise specifications are at $T_A$ = 25°C. $V_{REF}^+ = V_D^+ = V_{VCO}^+ = V_{OUT}^+ = 3.3V$, $V_{CP}^+ = 5V$ unless otherwise specified (Note 2). All voltages are with respect to GND.**

| SYMBOL | PARAMETER | CONDITIONS | | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|---|
| **CML Clock Outputs (OUT0⁺, OUT0⁻, OUT1⁺, OUT1⁻, OUT2⁺, OUT2⁻, …, OUT10⁺, OUT10⁻)** | | | | | | | |
| $f_{OUT}$ | Output Frequency | Differential Termination = 100Ω, MODEx = 0 (Clock Mode) | ● | 0 | | 4500 | MHz |
| | | Differential Termination = 100Ω, MODEx = 1, 2, or 3 (SYSREF Modes) | ● | 0 | | 150 | |
| $V_{OD}$ | Output Differential Voltage | Differential Termination = 100Ω | ● | 320 | 420 | 550 | mVpk |
| | Output Resistance | Differential | | | 100 | | Ω |
| | Output Common Mode Voltage | Differential Termination = 100Ω | | | $V_{OUT}^+$-1.0 | | V |
| $t_R$ | Output Rise Time, 20% to 80% | Differential Termination = 100Ω | | | 50 | | ps |
| $t_F$ | Output Fall Time, 80% to 20% | Differential Termination = 100Ω | | | 50 | | ps |
| | Output Duty Cycle | Differential Termination = 100Ω | ● | 45 | 50 | 55 | % |
| $t_{PD-VCO}$ | Propagation Delay from VCO± to OUT10 | $f_{VCO}$ = 4500MHz, Mx = 16 | ● | | 335 | | ps |
| | Propagation Delay from VCO± to OUT10, Tem-perature Variation | $f_{VCO}$ = 4500MHz, Mx = 16 | | | 0.35 | | ps/°C |
| $t_{PD-REF}$ | Propagation Delay from REF± to OUT5 | RAO = 0, LOCK = 1 (Note 19) | ● | | 450 | | ps |
| | | RAO = 1, LOCK = 1 (Note 19) | ● | | 20 | | |
| | Propagation Delay from REF± to OUT5, Temperature Variation | RAO = 1, LOCK = 1 (Note 19) | | | 0.2 | | ps/°C |
| $t_{SKEW}$ | Skew, all Outputs (Note 20) | One Part, All Mx the Same, Even, or 1 | ● | | ±10 | ±25 | ps |
| | | One Part, Any Mx | | | ±30 | | |
| | | Across Multiple Parts; RAO = 1, All Mx the Same, Even, or 1; All $T_J$ Within ±10°C | ● | | ±20 | ±50 | |
| | Additional Output Delay, Mx = Odd vs Mx = 1 or Even | Mx = 5, 11, 15, 17, 19, 25, or 27 | | | 4 | | ps |
| | | Mx = 3, 7, 9, 13, 21, 23, 29, or 31 | | | 15 | | |
| **Power Supply Voltages** | | | | | | | |
| | $V_{REF}^+$ Supply Range | | ● | 3.15 | 3.3 | 3.45 | V |
| | $V_{OUT}^+$ Supply Range | | ● | 3.15 | 3.3 | 3.45 | V |
| | $V_D^+$ Supply Range | | ● | 3.15 | 3.3 | 3.45 | V |
| | $V_{VCO}^+$ Supply Range | | ● | 3.15 | 3.3 | 3.45 | V |
| | $V_{CP}^+$ Supply Range | | ● | 3.15 | | 5.25 | V |
| **Power Supply Currents** | | | | | | | |
| $I_{DDOUT}$ | Sum $V_{OUT}^+$ Supply Currents | All Outputs Enabled (Note 5) | ● | | 750 | 850 | mA |
| | | Typical JESD204B/C Application (Note 6) | | | 570 | | mA |
| | | PDALL = 1 | | | 500 | | µA |
| $I_{CCCP}$ | $V_{CP}^+$ Supply Current | $I_{CP}$ = 11.2mA | ● | | 37 | 45 | mA |
| | | $I_{CP}$ = 423µA | | | 10 | | mA |
| | | PDALL = 1 | | | 500 | | µA |
| $I_{DD-3.3V}$ | Sum $V_D^+$, $V_{REF}^+$, $V_{VCO}^+$ Supply Currents | Digital Inputs at Supply Levels | ● | | 176 | 200 | mA |
| | | Digital Inputs at Supply Levels, PDALL = 1 | | | 500 | | µA |
| | Supply Current Deltas from Total Chip Current | PDx[1:0] = 2 (x = 0 to 10), Per Output | | | –34 | | mA |
| | | PDx[1:0] = 3 (x = 0 to 10), Per Output | | | –68 | | mA |
| | | EZS_SRQ± State = 1, SSRQ = 1 or SRQMD = 1 | | | +175 | | mA |
| | | PDPLL = 1 | | | –113 | | mA |
| | | Mx = Odd (Not Mx = 1), Per Output | | | +8.6 | | mA |
| | | ADELx = 1 to 31, Per Output | | | +3.0 | | mA |
| | | ADELx = 32 to 63, Per Output | | | +4.7 | | mA |

## ELECTRICAL CHARACTERISTICS

**The ● denotes the specifications which apply over the full operating junction temperature range, otherwise specifications are at $T_A$ = 25°C. $V_{REF}^+ = V_D^+ = V_{VCO}^+ = V_{OUT}^+ = 3.3V$, $V_{CP}^+ = 5V$ unless otherwise specified (Note 2). All voltages are with respect to GND.**

| SYMBOL | PARAMETER | CONDITIONS | | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|---|
| **Additive Phase Noise, Jitter and Spurious (Note 8)** | | | | | | | |
| | Output Noise/Jitter: Distribution Section Only ($f_{VCO}$ = 4.5GHz, $f_{OUTx}$ = 4.5GHz, Mx = 1) | Phase Noise Floor<br>RMS Jitter, 12kHz to 20MHz Integration BW<br>RMS Jitter, ADC SNR Method (Note 15) | | | –154.3<br>6<br>65 | | dBc/Hz<br>$fs_{RMS}$<br>$fs_{RMS}$ |
| | Output Noise/Jitter: Distribution Section Only ($f_{VCO}$ = 4.5GHz, $f_{OUTx}$ = 2.25GHz, Mx = 2) | Phase Noise Floor<br>RMS Jitter, 12kHz to 20MHz Integration BW<br>RMS Jitter, ADC SNR Method (Note 15) | | | –157.1<br>8<br>66 | | dBc/Hz<br>$fs_{RMS}$<br>$fs_{RMS}$ |
| | Output Noise/Jitter: Distribution Section Only ($f_{VCO}$ = 4.5GHz, $f_{OUTx}$ = 1.125GHz, Mx = 4) | Phase Noise Floor<br>RMS Jitter, 12kHz to 20MHz Integration BW<br>RMS Jitter, ADC SNR Method (Note 15) | | | –160.2<br>9<br>65 | | dBc/Hz<br>$fs_{RMS}$<br>$fs_{RMS}$ |
| | Output Noise/Jitter: Distribution Section Only ($f_{VCO}$ = 3.2GHz, $f_{OUTx}$ = 200MHz, Mx = 16) | Phase Noise Floor<br>RMS Jitter, 12kHz to 20MHz Integration BW<br>RMS Jitter, ADC SNR Method (Note 15) | | | –167.8<br>21<br>65 | | dBc/Hz<br>$fs_{RMS}$<br>$fs_{RMS}$ |
| | Output Noise/Jitter: Distribution Section Only ($f_{VCO}$ = 3.2GHz, $f_{OUTx}$ = 50MHz, Mx = 64) | Phase Noise Floor<br>RMS Jitter, 12kHz to 20MHz Integration BW<br>RMS Jitter, ADC SNR Method (Note 15) | | | –173.8<br>41<br>65 | | dBc/Hz<br>$fs_{RMS}$<br>$fs_{RMS}$ |
| $L_{NORM}$ | Normalized In-Band Phase Noise Floor | $I_{CP}$ = 11.2mA (Note 9, Note 10, Note 11) | | | –229 | | dBc/Hz |
| $L_{1/f}$ | Normalized In-Band 1/f Phase Noise | $I_{CP}$ = 11.2mA (Note 9, Note 12) | | | –281 | | dBc/Hz |
| | Spurious | $f_{OFFSET} = f_{PFD}$, $f_{OUT}$ = 4GHz, PLL Locked (Note 10, Note 13, Note 14, Note 18) | | | –100 | | dBc |

**Note 1:** Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. Exposure to any Absolute Maximum Rating condition for extended periods may affect device reliability and lifetime.

**Note 2.** The LTC6952 is guaranteed to meet specified performance limits over the full operating junction temperature range of –40°C to 125°C. Under maximum operation conditions, air flow or heat sinking may be required to maintain a junction temperature of 125°C or lower. It is required that the Exposed Pad (Pin 53) be soldered directly to the ground plane with an array of thermal vias as described in the Applications Information section.

**Note 3.** Absolute maximum time of digital delay is limited to 100µs.

**Note 4.** For $f_{OUT} \geq$ 300MHz, analog delay time vs ADELx varies. See Typical Performance Characteristics plot and the Operations section.

**Note 5.** All outputs configured as enabled clocks: all PDx[1:0]=0, EZS_SRQ± pin state=0, SSRQ=0, SRQMD=0, BST=1, PDALL=0, PDVCOPK=0, PDREFPK=0.

**Note 6.** Configured with six enabled clock outputs and five SYSREF outputs with output drivers disabled: PD0, PD2, PD4, PD6, PD8, and PD10 = 0; PD1, PD3, PD5, PD7, and PD9 = 2; EZS_SRQ± pin state=0; SSRQ=0; SRQMD=0; BST=1; PDALL=0; PDVCOPK=0; PDREFPK=0

**Note 7.** For 1.0V < V(CP) < $V_{CP}^+$ – 1.1V.

**Note 8.** Additive phase noise and jitter from LTC6952 distribution section only. External VCO, reference and PLL noise is not included.

**Note 9.** Measured inside the loop bandwidth with the loop locked.

**Note 10.** Reference frequency supplied by Pascal OCXO-E, $f_{REF}$ = 100MHz, $P_{REF}$ = 6dBm.

**Note 11.** Output Phase Noise Floor is calculated from Normalized Phase Noise Floor by $L_{OUT} = L_{NORM} + 10\log_{10}(f_{PFD}) + 20\log10(f_{RF}/f_{PFD})$.

**Note 12.** Output 1/f Noise is calculated from Normalized 1/f Phase Noise by $L_{OUT(1/f)} = L_{1/f} + 20\log_{10}(f_{RF}) - 10\log_{10}(f_{OFFSET})$.

**Note 13.** $I_{CP}$ = 11.2mA, $f_{PFD}$ = 100MHz, FILTR = 0, Loop BW = 16kHz, $f_{VCO}$ = 4GHz

**Note 14.** Measured using DC2609

**Note 15.** Additive RMS jitter (ADC SNR method) is calculated by integrating the distribution section's measured additive phase noise floor out to $f_{CLK}$. Actual ADC SNR measurements show good agreement with this method.

**Note 16.** Measured with 36" cables from output of DC2609 to measurement instrument. Cable loss is NOT accounted for in this plot.

**Note 17.** Statistics calculated from 640 total measured parts from two process lots.

**Note 18.** Measured using differential LTC6952 outputs driving LTC6955. LTC6955 provides differential to single-ended conversion for rejection of common mode spurious signals. See the Applications Information section for details.

**Note 19.** Measured on OUT5. $f_{REF}$ = 100MHz, $f_{VCO}$ = 4400MHz, $f_{OUT}$ = 275MHz, RD = 1

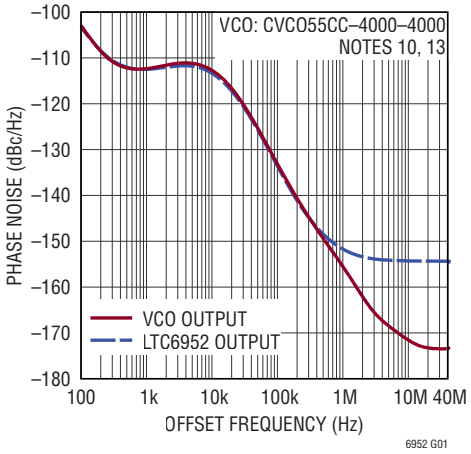
**Note 20.** Skew is defined as the difference between the zero-crossing time of a given output and the average zero-crossing time of all outputs.

**Note 21.** Measured VCO input power is de-embedded to the pins of the LTC6952.

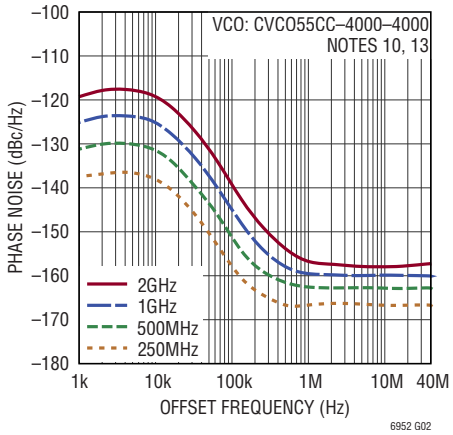
## TYPICAL PERFORMANCE CHARACTERISTICS

$T_A$ = 25°C, $V_{REF}^+ = V_{OUT}^+ = V_D^+ = V_{VCO}^+ = 3.3V$, $V_{CP}^+ = 5V$, unless otherwise noted.

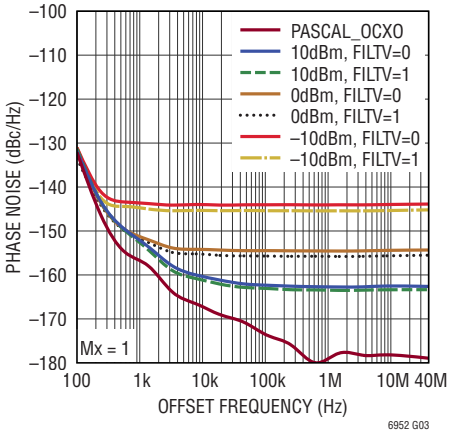
### Total Closed Loop Phase Noise $f_{VCO}$= 4GHz, Mx = 1



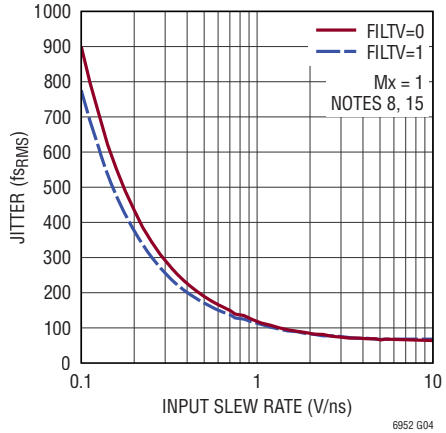

### Total Closed Loop Phase Noise, $f_{VCO}$ = 4GHz, Mx = 2, 4, 8, and 16



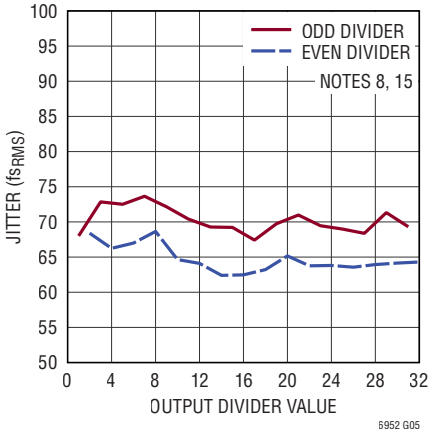

### Total Open Loop Phase Noise $F_{IN}$ = 100MHz Sine Wave



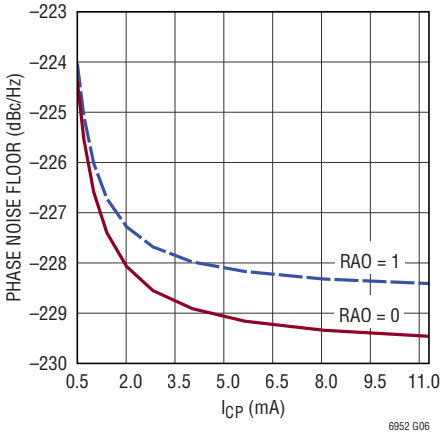

### Additive Jitter vs Input Slew Rate, ADC SNR Method



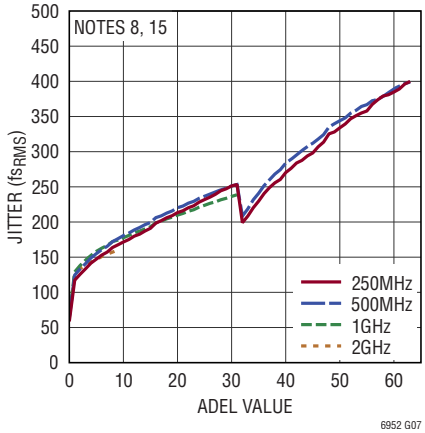

### Additive Jitter vs Divider Setting, ADC SNR Method



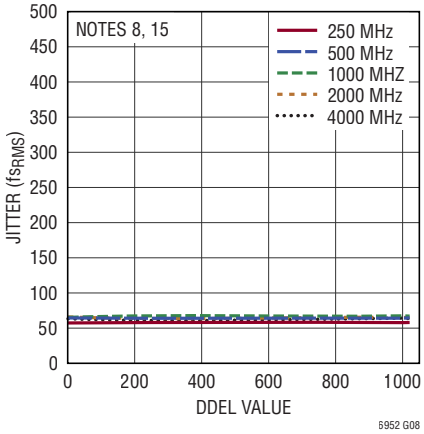

### Normalized In-Band Phase Noise Floor vs CP Current



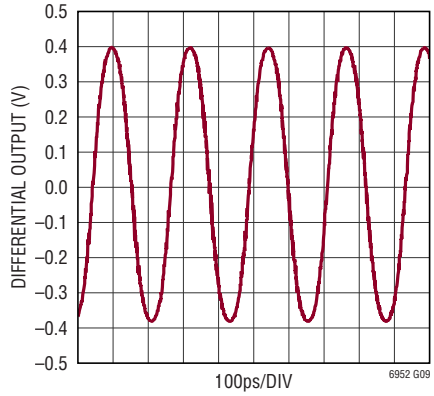

### Additive Jitter vs ADEL Value, ADC SNR Method $f_{VCO}$ = 4GHz, Mx = 2, 4, 8, and 16




### Additive Jitter vs DDEL Value, ADC SNR Method $f_{VCO}$ = 4GHz, Mx = 1, 2, 4, 8, and 16



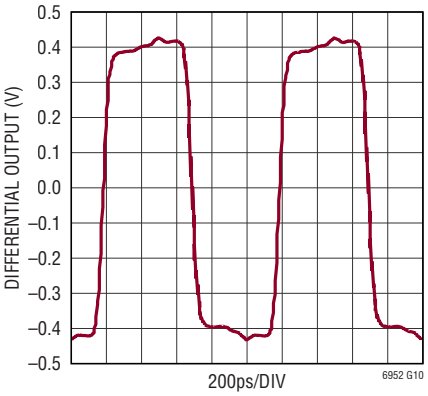

### Differential Output at 4.5GHz



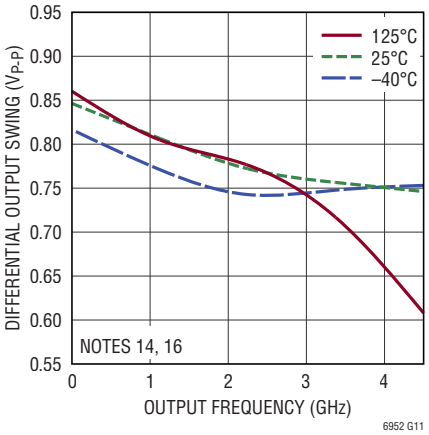
## TYPICAL PERFORMANCE CHARACTERISTICS

$T_A$ = 25°C, $V_{REF}^+ = V_{OUT}^+ = V_D^+ = V_{VCO}^+$ = 3.3V, $V_{CP}^+$ = 5V, unless otherwise noted.

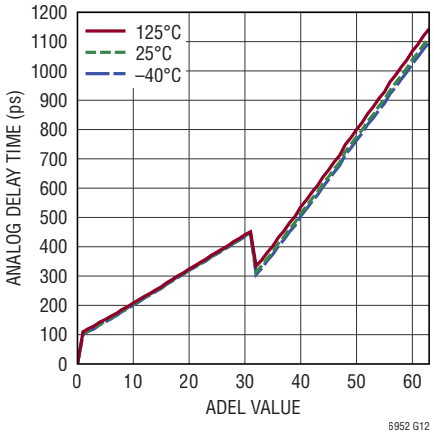
### Differential Output at 1GHz



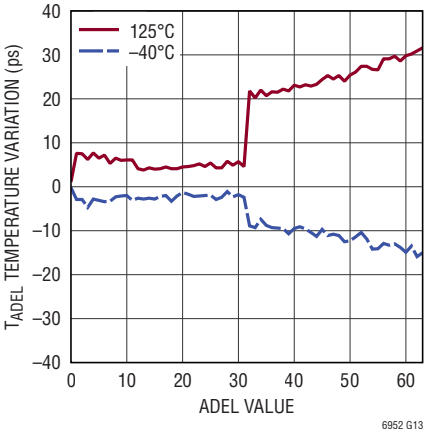

### Differential Output Swing vs Frequency, Temperature



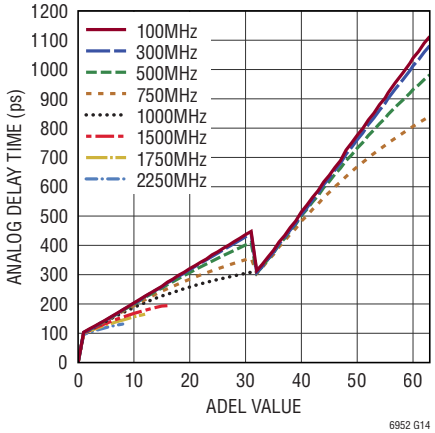

### Analog Delay Time vs ADEL Value, Temperature



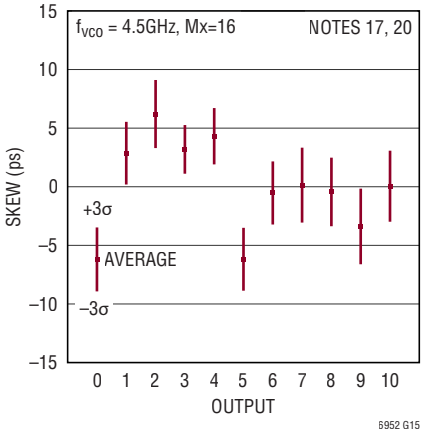

### Analog Delay Time Temperature Variation from 25°C



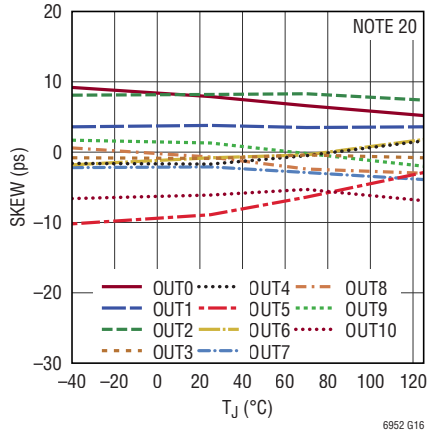

### Analog Delay Time vs ADEL Value Over Multiple Output Frequencies



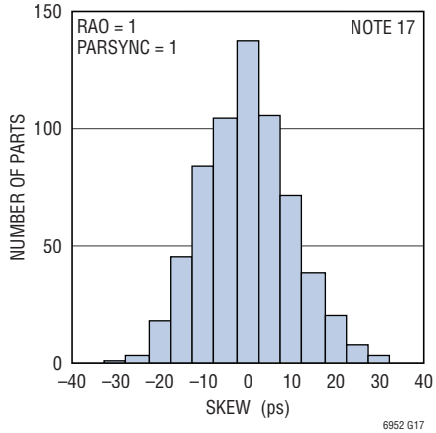

### Expected Skew Variation for a Single Part



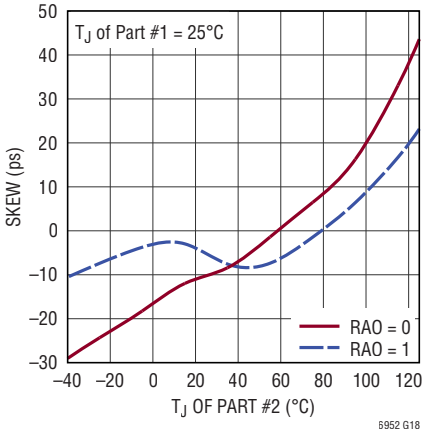

### Skew Variation with Temperature For a Single Typical Part




### Expected Skew Variation Across Multiple Parts in ParallelSync Mode



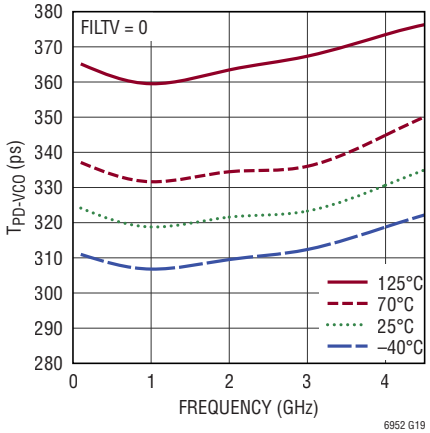

### Skew Between Two Parts at Different Temperatures



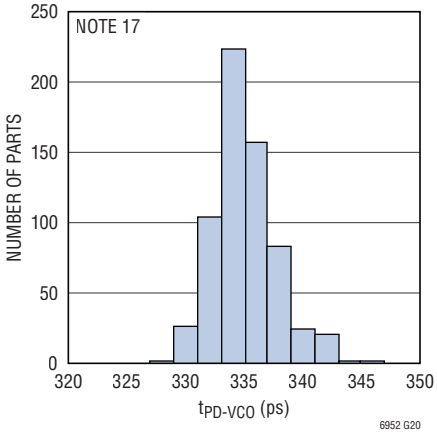
## TYPICAL PERFORMANCE CHARACTERISTICS

$T_A$ = 25°C, $V_{REF}^+ = V_{OUT}^+ = V_D^+ = V_{VCO}^+$ = 3.3V, $V_{CP}^+$ = 5V, unless otherwise noted.

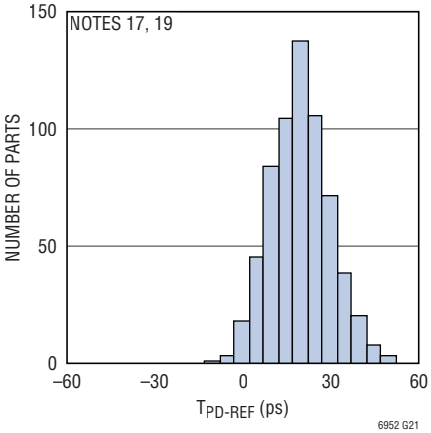
### VCO Propagation Delay vs Frequency, Temperature



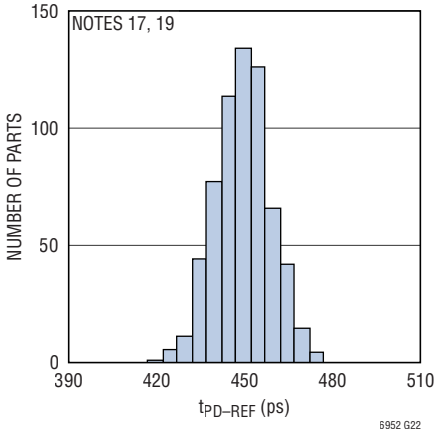

### Propagation Delay Variation, VCO Input to OUT4



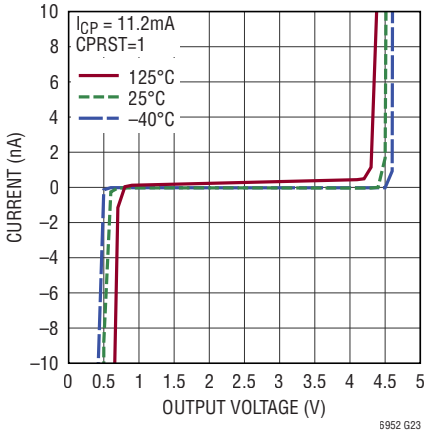

### Closed Loop Reference to Output Propagation Delay, RAO = 1



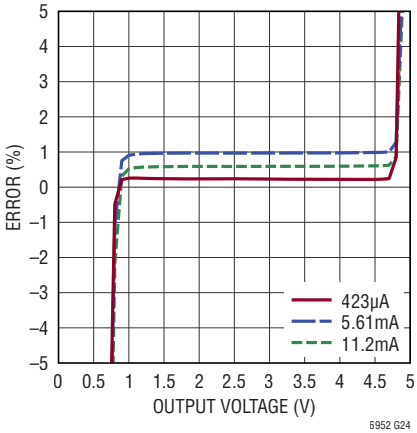

### Closed Loop Reference to Output Propagation Delay, RAO = 0



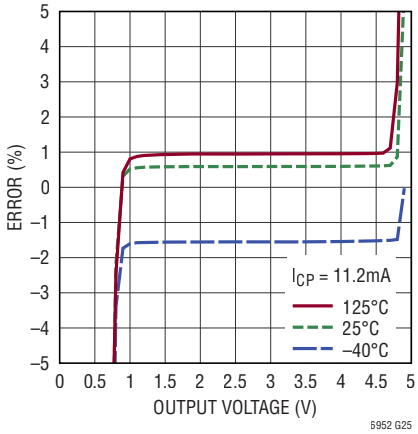

### Charge Pump Hi–Z Current vs Voltage, Temperature



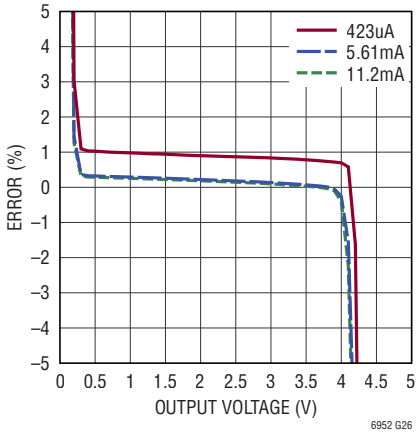

### Charge Pump Sink Current Error vs Voltage, Output Current



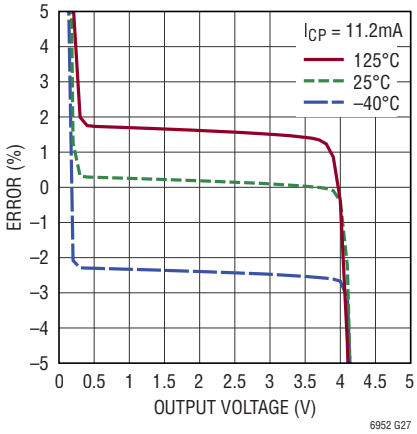

### Charge Pump Sink Current Error vs Voltage, Temperature




### Charge Pump Source Current Error vs Voltage, Output Current



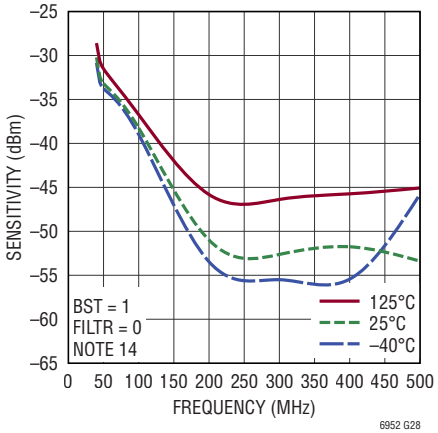

### Charge Pump Source Current Error vs Voltage, Temperature



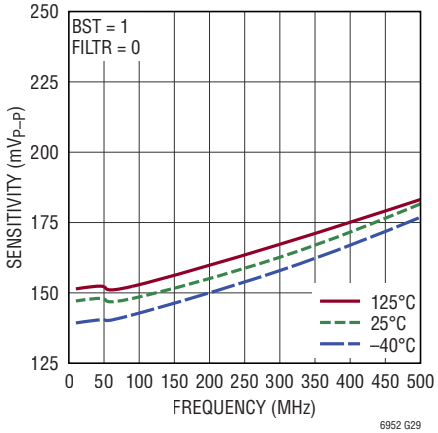
## TYPICAL PERFORMANCE CHARACTERISTICS

**$T_A$ = 25°C, $V_{REF}^+ = V_{OUT}^+ = V_D^+ = V_{VCO}^+$ = 3.3V, $V_{CP}^+$ = 5V, unless otherwise noted.**

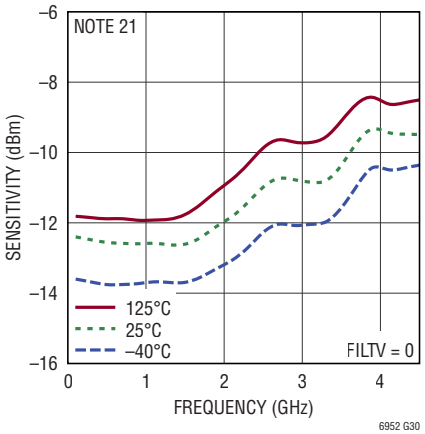
**REF Input Sensitivity vs Frequency, Temperature**



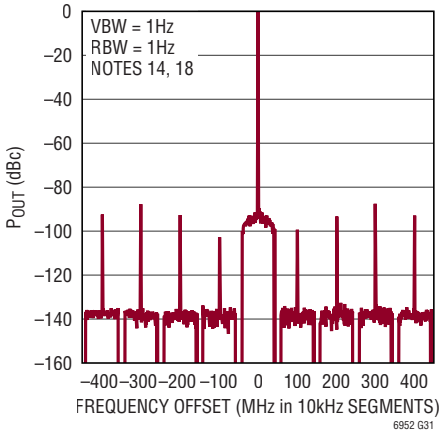

**REF Input Signal Detected vs Frequency, Temperature**



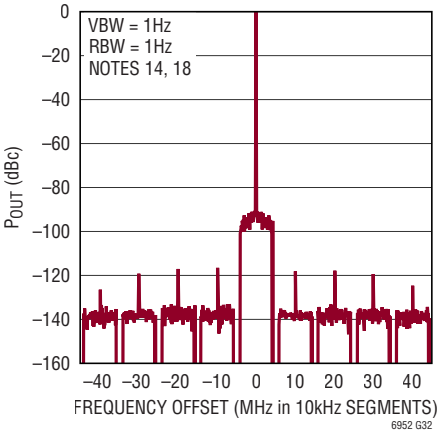

**VCO Input Signal Detected vs Frequency, Temperature**



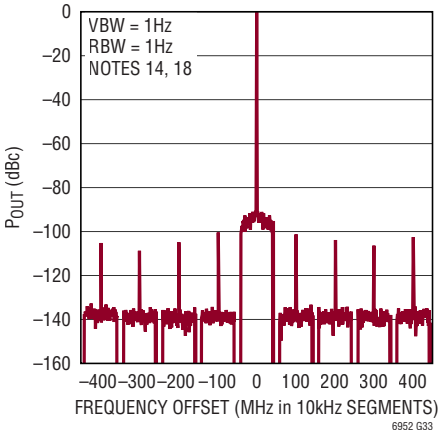

**Spurious Response, $f_{OUT}$ = 4GHz, $f_{VCO}$ = 4GHz, $f_{REF}$ = 100MHz, $f_{PFD}$ = 100MHz, Loop BW = 16kHz**



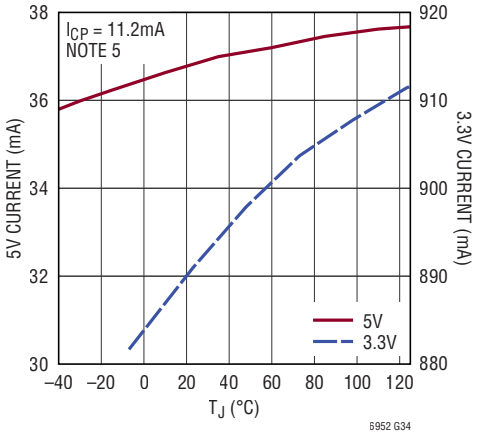

**Spurious Response, $f_{OUT}$ = 4GHz, $f_{VCO}$ = 4GHz, $f_{REF}$ = 100MHz, $f_{PFD}$ = 10MHz, Loop BW = 5kHz**



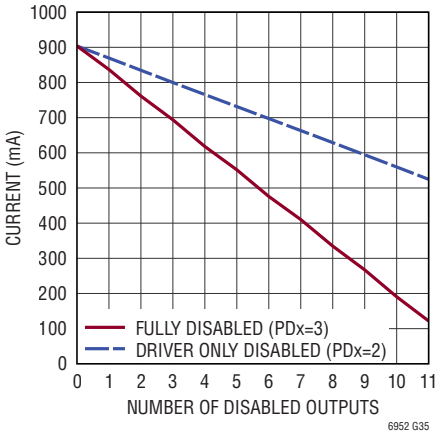

**Spurious Response, $f_{OUT}$ = 4GHz, $f_{VCO}$ = 4GHz, $f_{REF}$ = 100MHz, $f_{PFD}$ = 10MHz, Loop BW = 5kHz**




**Supply Current vs Junction Temperature, All Outputs Enabled**




**3.3V Supply Current vs Number of Disabled Outputs**



## PIN FUNCTIONS

**$\overline{CS}$ (Pin 1):** Serial Port Chip Select. This CMOS input initiates a serial port communication burst when driven low, ending the burst when driven back high. See the Operation section for more details.

**$V_D^+$ (Pin 2):** 3.15V to 3.45V positive supply pins for synchronization/SYSREF request functions and serial port. This pin should be bypassed directly to the ground plane using a 0.01µF ceramic capacitor as close to the pin as possible.

**OUT0$^+$, OUT0$^-$ (Pins 34, 33):** Output Signals. The output divider is buffered and presented differentially on these pins. The outputs have 50Ω (typical) output resistance per side (100Ω differential). The far end of the transmission line is typically terminated with 100Ω connected across the outputs. See the Operation and Applications Information section for more details.

**OUT1$^+$, OUT1$^-$ (Pins 31, 30):** Same as OUT0.

**OUT2$^+$, OUT2$^-$ (Pins 28, 27):** Same as OUT0.

**OUT3$^+$, OUT3$^-$ (Pins 25, 24):** Same as OUT0.

**OUT4$^+$, OUT4$^-$ (Pins 22, 21):** Same as OUT0.

**OUT5$^+$, OUT5$^-$ (Pins 19, 18):** Same as OUT0.

**OUT6$^+$, OUT6$^-$ (Pins 16, 15):** Same as OUT0.

**OUT7$^+$, OUT7$^-$ (Pins 13, 12):** Same as OUT0.

**OUT8$^+$, OUT8$^-$ (Pins 10, 9):** Same as OUT0.

**OUT9$^+$, OUT9$^-$ (Pins 7, 6):** Same as OUT0.

**OUT10$^+$, OUT10$^-$ (Pins 4, 3):** Same as OUT0.

**$V_{OUT}^+$ (Pins 5, 8, 11, 14, 17, 20, 23, 26, 29, 32, 35):** 3.15V to 3.45V positive supply pins for output dividers. Each pin should be separately bypassed directly to the ground plane using a 0.01µF ceramic capacitor as close to the pin as possible.

**NC (Pin 36):** Not Connected Internally. It is recommended that this pin be connected to the ground pad (Pin 53).

**VCO$^+$, VCO$^-$ (Pins 37, 38):** VCO Input Signals. The differential signal placed on these pins is buffered with a low noise amplifier and fed to the internal distribution path and feedback dividers. These self-biased inputs present a differential 250Ω (typical) resistance to aid impedance matching. They may also be driven single-ended by using the matching circuit in the Applications Information section.

**$V_{VCO}^+$ (Pins 39):** 3.15V to 3.45V positive supply pin for VCO circuitry. This pin should be bypassed directly to the ground plane using a 0.01µF ceramic capacitor as close to the pin as possible.

**$\overline{SD}$ (Pin 40):** Chip Shutdown Pin. When tied to GND, this CMOS input disables all blocks in the chip. This is the same function as PDALL in the serial interface.

**GND (Pin 41):** Negative Power Supply (Ground). This pin should be tied directly to the ground pad (Pin 53).

**$V_{CP}^+$ (Pin 42):** 3.15V to 5.25V positive supply pin for charge pump circuitry. This pin should be bypassed directly to the ground plane using a 0.1µF ceramic capacitor as close to the pin as possible.

**CP (Pin 43):** Charge Pump Output. This bidirectional current output is normally connected to the external loop filter. See the Applications Information section for more details.

**$V_{REF}^+$ (Pin 44):** 3.15V to 3.45V positive supply pin for reference input circuitry. This pin should be bypassed directly to the ground plane using a 0.1µF ceramic capacitor as close to the pin as possible.

**REF$^+$, REF$^-$ (Pins 45, 46):** Reference Input Signals. This differential input is buffered with a low noise amplifier, which feeds the reference divider. They are self-biased and must be AC coupled with 1µF capacitors. If used single-ended with V(REF$^+$) ≤ 2.7$V_{P-P}$, bypass REF$^-$ to GND with a 100nF capacitor. If used single-ended with V(REF$^+$) > 2.7$V_{P-P}$, bypass REF$^-$ to GND with a 47pF capacitor.

## PIN FUNCTIONS

**EZS_SRQ+, EZS_SRQ– (Pins 47, 48):** Synchronization or SYSREF Request Input. Bit SRQMD defines this input as an EZSync request or SYSREF request. It can operate as a differential input, or EZS_SRQ– can be tied to GND and EZS_SRQ+ driven with a single-ended CMOS signal. See the Operation and Applications Information section for more details.

**STAT (Pin 49):** Status Output. This signal is a configurable logical OR combination of the UNLOCK, $\overline{\text{VCOOK}}$, VCOOK, $\overline{\text{LOCK}}$, LOCK, $\overline{\text{REFOK}}$, and REFOK status bits, programmable via the STATUS register. It can also be configured to present a diode voltage for temperature measurement. See the Operation section for more details.

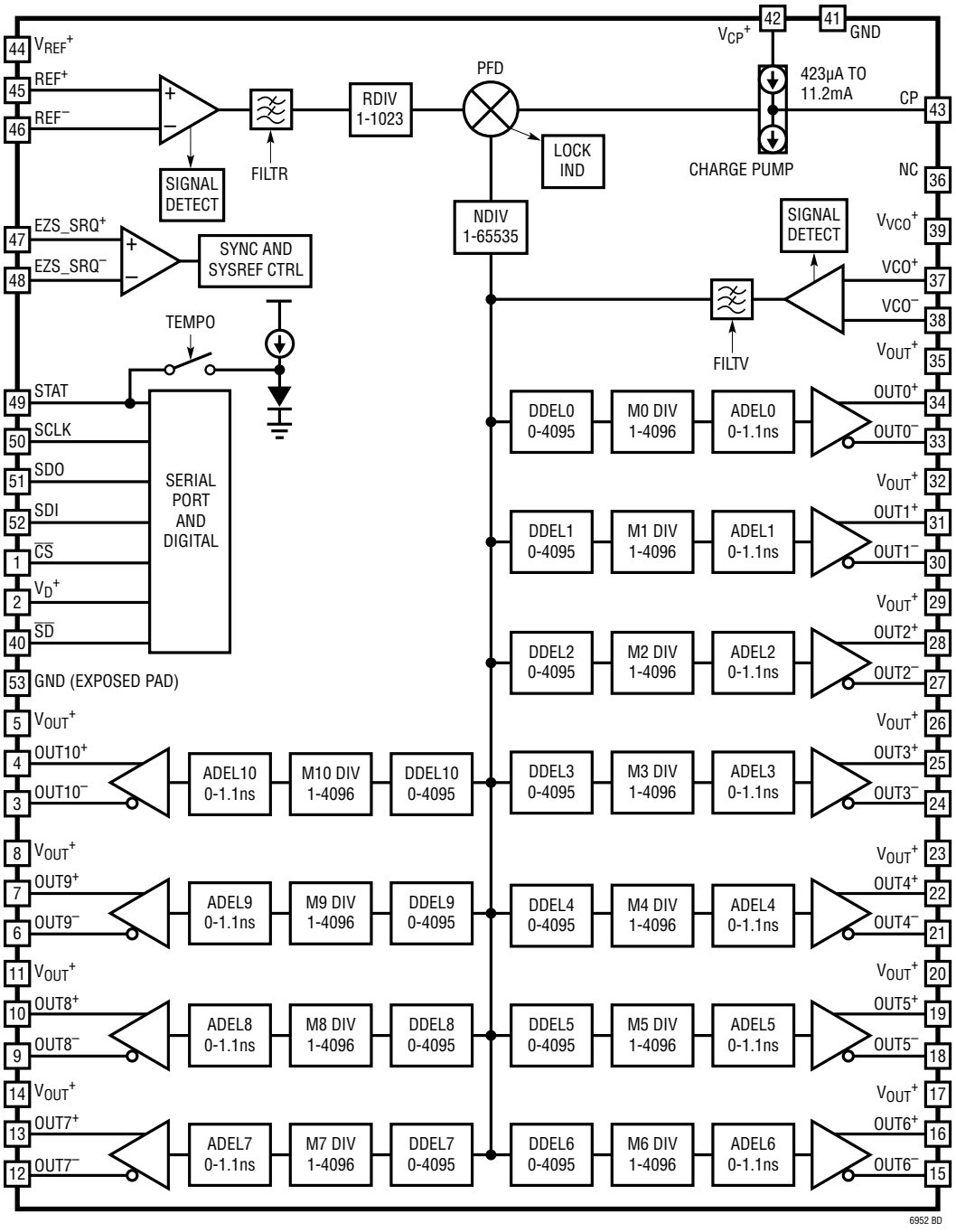
**SCLK (Pin 50):** Serial Port Clock. This CMOS input clocks serial port input data on its rising edge. See the Operation section for more details.

**SDO (Pin 51):** Serial Port Data Output. This CMOS three-state output presents data from the serial port during a read communication burst. Optionally attach a resistor of > 200kΩ to GND to prevent a floating output. See the Operation section for more details.

**SDI (Pin 52):** Serial Port Data Input. The serial port uses this CMOS input for data. See the Operation section for more details.

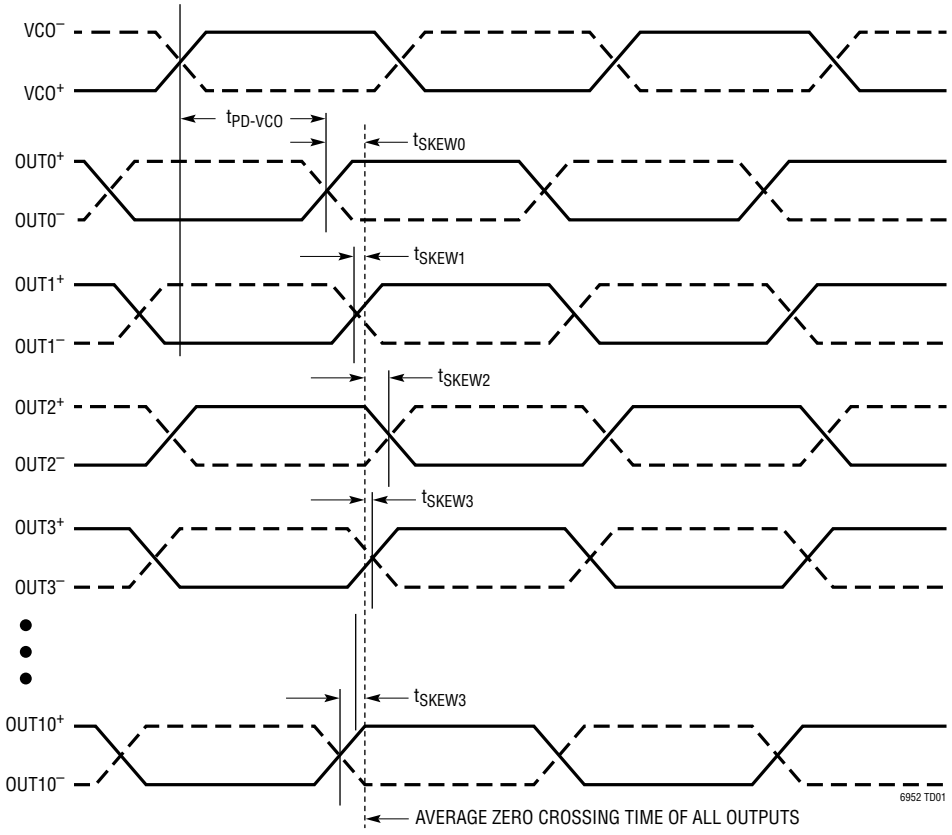
**GND (Exposed Pad Pin 53):** Negative Power Supply (Ground). The package exposed pad must be soldered directly to the PCB land. The PCB land pattern should have multiple thermal vias to the ground plane for both low ground inductance and also low thermal resistance.

# BLOCK DIAGRAM

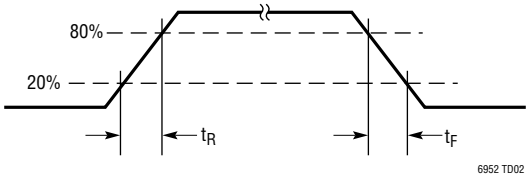


# TIMING DIAGRAMS

**Propagation Delay and Output Skew**




**Differential CML Rise/Fall Times**



# OPERATION

The LTC6952 is a high-performance integer-N PLL and multi-output clock generator that operates up to 4.5GHz. Utilizing Analog Devices' proprietary EZSync and ParallelSync standards, users of the LTC6952 can synchronize clocks across multiple outputs and multiple chips. Using an external low-noise VCO, the device is able to achieve superior integrated jitter performance by the combination of its extremely low in-band phase noise and excellent output noise floor. For JESD204B/C subclass 1 applications, the LTC6952 also provides several convenient methods to generate SYSREF pulses.

## REFERENCE INPUT BUFFER

The PLL's reference frequency is applied differentially on pins $REF^+$ and $REF^-$. These high-impedance inputs are self-biased and should be AC coupled with 1µF capacitors (see Figure 1 for a simplified schematic). Alternatively, the inputs may be used single-ended by applying the reference frequency at $REF^-$ and bypassing $REF^+$ to GND with a 1µF capacitor. If the single-ended signal is greater than $2.7V_{P-P}$, then use a 47pF capacitor for the GND bypass.

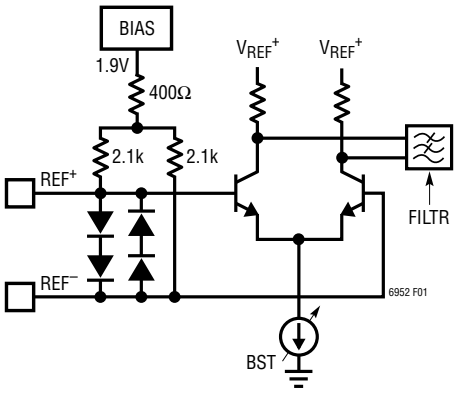



**Figure 1. Simplified REF Interface Schematic**

A high quality signal must be applied to the $REF^{\pm}$ inputs as they provide the frequency reference to the entire PLL. To achieve the part's in-band phase noise performance, apply a sine wave signal of at least 6dBm into 50Ω, or a square wave of at least $0.5V_{P-P}$, with slew rate of at least 20V/µs. See the Applications Information section for more information on reference input signal requirements and interfacing.

Additional options are available through serial port register h02 to further refine the application. Bit FILTR controls the reference input buffer's low-pass filter, and should be set for sine wave signals based upon $f_{REF}$ to limit the reference's wideband noise. The FILTR bit must be set correctly to reach the $L_{NORM}$ normalized in-band phase noise floor. See Table 1 for recommended settings. Square wave inputs will have FILTR set to a "0".

**Table 1. FILTR Programming**

| FILTR | Sine Wave $f_{REF}$ | Square Wave $f_{REF}$ |
|---|---|---|
| 1 | < 20MHz | N/A |
| 0 | ≥ 20MHz | All $f_{REF}$ |

The BST bit should be set based upon the input signal level to prevent the reference input buffer from saturating. The BST programming is the same whether the input is a sine wave or a square wave. See Table 2 for recommended settings and the Applications Information section for programming examples.

**Table 2. BST Programming**

| BST | $V_{REF}$ |
|---|---|
| 1 | < 1.6 $V_{P-P}$ |
| 0 | ≥ 1.6 $V_{P-P}$ |

### Reference Peak Detector

A reference input peak detection circuit is provided on the $REF^{\pm}$ inputs to detect the presence of a reference signal and provides the REFOK and $\overline{REFOK}$ status flags available through both the STAT output and serial port register h00. $\overline{REFOK}$ is the logical inverse of REFOK. The circuit has hysteresis to prevent the REFOK flag from chattering at the detection threshold. The reference peak detector may be powered-down using the PDREFPK bit found in register h02.

The peak detector approximates an RMS detector, therefore sine and square wave inputs will give different detection thresholds by a factor of 4/π. See Table 3 for REFOK detection values.

**Table 3. REFOK, $\overline{REFOK}$ Status Output vs REF Input**

| REFOK | $\overline{REFOK}$ | Sine Wave $f_{REF}$ | Square Wave $f_{REF}$ |
|---|---|---|---|
| 1 | 0 | ≥ 250$mV_{P-P}$ | ≥ 200$mV_{P-P}$ |
| 0 | 1 | < 100$mV_{P-P}$ | < 75$mV_{P-P}$ |

# OPERATION

## REFERENCE DIVIDER (R)

A 10-bit divider is used to reduce the frequency seen at the phase/frequency detector (PFD). Its divide ratio R may be set to any integer from 1 to 1023. Use the RD[9:0] bits found in registers h06 and h07 to directly program the R divide ratio. See the Applications Information section for the relationship between R and the $f_{REF}$, $f_{PFD}$, $f_{VCO}$, and $f_{OUTx}$ frequencies.

A mode to provide synchronization of the Reference inputs to the R divider output (R ≥ 2) using the rising edge of the EZS_SRQ± pins is enabled when the PARSYNC bit in register h06 is set to "1". This synchronization is critical for output alignment in ParallelSync Mode, as described later in this section. The EZS_SRQ± rising edge must meet setup and hold timing to the rising edge of the Reference input. See Figure 2 for the timing relationships between the Reference input, EZS_SRQ± and the R divider output. Note that changing the R divider output edge timing will force the PLL to lose phase lock but will return to normal operation after several loop time constants. See Reference Signal and EZS_SRQ Timing for ParallelSync Mode in the Applications Information section for the timing requirements of EZS_SRQ± to REF in this mode.

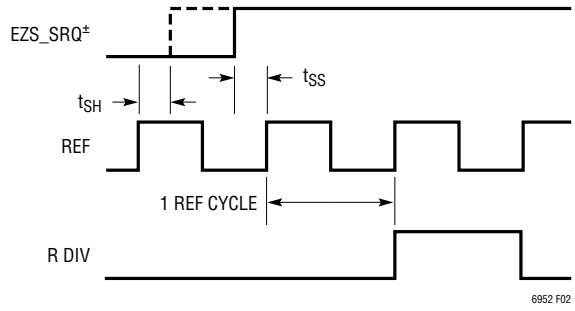



**Figure 2. EZS_SRQ± to REF Timing (PARSYNC = 1)**

## PHASE/FREQUENCY DETECTOR (PFD)

The phase/frequency detector (PFD), in conjunction with the charge pump, produces source and sink current pulses proportional to the phase difference between the outputs of the R and N dividers. This action provides the necessary feedback to phase-lock the loop, forcing a phase alignment at the PFD's inputs. The PFD may be disabled with the CPRST bit which prevents UP and DOWN pulses from being produced. See Figure 3 for a simplified schematic of the PFD.

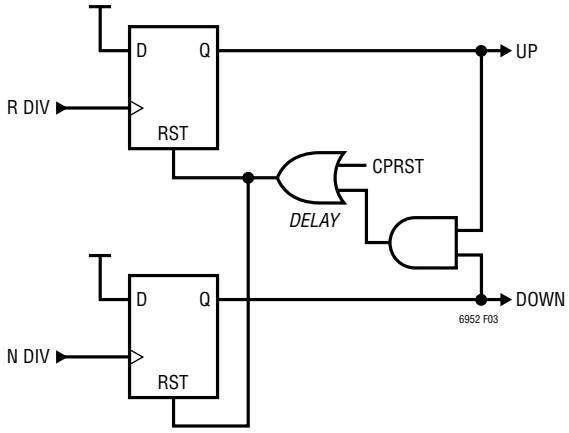



**Figure 3. Simplified PFD Schematic**

## LOCK INDICATOR

The lock indicator uses internal signals from the PFD to measure phase coincidence between the R and N divider output signals. It is enabled by programming LKCT[1:0] in the serial port register h06 (see Table 5), and produces LOCK, $\overline{\text{LOCK}}$ and UNLOCK status flags, available through both the STAT output and serial port register h00. $\overline{\text{LOCK}}$ is the logical inverse of LOCK.

The user sets the phase difference lock window time $t_{LWW}$ for a valid LOCK condition with the LKWIN bit found in register h06. Table 4 contains recommended settings for different $f_{PFD}$ frequencies. See the Applications Information section for examples.

**Table 4. LKWIN Programming**

| LKWIN | $t_{LWW}$ | $f_{PFD}$ |
|---|---|---|
| 0 | 3ns | > 5MHz |
| 1 | 10ns | ≤ 5MHz |

The PFD phase difference must be less than $t_{LWW}$ for the COUNTS number of successive counts before the lock indicator asserts the LOCK flag. The LKCT[1:0] bits are used to set COUNTS depending upon the application.

# OPERATION

Larger values of COUNTS lead to more accurate and stable lock indications at the expense of longer lock indication times. Set LKCT[1:0] = 0 to disable the lock indicator. See Table 5 for LKCT[1:0] programming and the Applications Information section for examples.

**Table 5. LKCT[1:0] Programming**

| LKCT[1:0] | COUNTS |
|---|---|
| 0 | Lock Indicator Disabled |
| 1 | 32 |
| 2 | 256 |
| 3 | 2048 |

When the PFD phase difference is greater than $t_{LWW}$, the lock indicator immediately asserts the UNLOCK status flag and clears the LOCK flag, indicating an out-of-lock condition. The UNLOCK flag is immediately de-asserted when the phase difference is less than $t_{LWW}$. See Figure 4 below for more details.

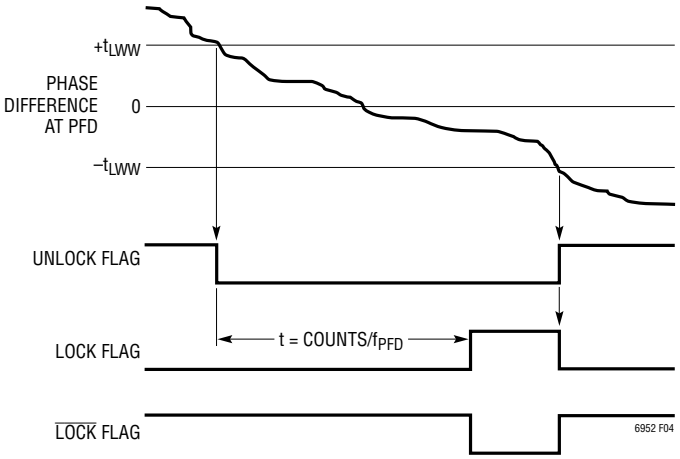



**Figure 4. UNLOCK, LOCK, and $\overline{\text{LOCK}}$ Timing**

## CHARGE PUMP

The charge pump, controlled by the PFD, forces sink (DOWN) or source (UP) current pulses onto the CP pin, which should be connected to an appropriate loop filter. See Figure 5 for a simplified schematic of the charge pump.

The output current magnitude $I_{CP}$ may be set from 423µA to 11.2mA using the CP[4:0] bits found in serial port register h0A. A larger $I_{CP}$ can result in lower in-band noise due to the lower impedance of the loop filter components. See Table 6 for programming specifics and the Applications Information section for loop filter examples.

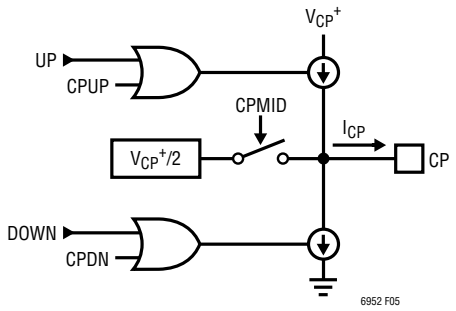



**Figure 5. Simplified Charge Pump Schematic**

**Table 6. CP[4:0] Programming**

| CP[4:0] | $I_{CP}$ |
|---|---|
| 0 | 423µA |
| 1 | 500µA |
| 2 | 592µA |
| 3 | 700µA |
| 4 | 842µA |
| 5 | 1.00mA |
| 6 | 1.18mA |
| 7 | 1.40mA |
| 8 | 1.68mA |
| 9 | 2.00mA |
| 10 | 2.36mA |
| 11 | 2.81mA |
| 12 | 3.37mA |
| 13 | 4.00mA |
| 14 | 4.72mA |
| 15 | 5.61mA |
| 16 | 6.73mA |
| 17 | 8.02mA |
| 18 | 9.43mA |
| 19 | 11.20mA |
| 20 to 31 | Invalid |

### Charge Pump Functions

The charge pump contains additional features to aid in system startup. See Table 7 for a summary.

## OPERATION

**Table 7. Charge Pump Function Bit Descriptions**

| BIT | DESCRIPTION |
|---|---|
| CPDN | Force Sink Current |
| CPINV | Invert PFD Phase |
| CPMID | Enable Mid-Voltage Bias |
| CPRST | Reset PFD, Hi-Z CP |
| CPUP | Force Source Current |
| CPWIDE | Extend Current Pulse Width |

The CPMID bit found in register h0B enables a resistive $V_{CP}^{+}/2$ output bias which may be used to pre-bias troublesome loop filters into a valid voltage range. When using CPMID, it is recommended to also assert the CPRST bit, forcing a PFD reset which puts the charge pump in a Hi-Z state. Both CPMID and CPRST must be set to "0" for normal operation.

The CPUP and CPDN bits force a constant $I_{CP}$ source or sink current, respectively, on the CP pin. The CPRST bit may also be used in conjunction with the CPUP and CPDN bits, allowing a pre-charge of the loop to a known state, if required. CPUP, CPDN, and CPRST must be set to "0" to allow the loop to lock.

The CPWIDE bit extends the charge pump output current pulse width by increasing the PFD reset path's delay value. CPWIDE is normally set to "0". Setting CPWIDE = 0 provides the best in-band phase noise performance.

The CPINV bit found in register h0B should be set for applications requiring signal inversion from the PFD, such as for loops using negative-slope tuning oscillators or for complex external loops using an inverting op amp in conjunction with a positive-slope tuning oscillator. A passive loop filter used in conjunction with a positive-slope VCO requires no inversion, so CPINV = 0.

### REFERENCE ALIGNED OUTPUT (RAO) MODE

The RAO mode is activated by setting the RAO bit in register h06 to "1". It aligns internal delays such that the output rising edge will always occur at an exact integer number of VCO clock cycles from the incoming reference signal. This mode is only used for closed loop PLL applications (PDPLL = 0), and is useful when a known delay between outputs and the incoming reference signal is required. It is also used to minimize skew between parallel LTC6952s in ParallelSync applications (see the ParallelSync section). The tradeoff for using the RAO mode is slightly degraded PLL in-band noise (<1.0dB).

### VCO INPUT BUFFER

The LTC6952's VCO input buffer provides a flexible interface to either differential or single-ended frequency sources. The inputs are self-biased, and AC-coupling is recommended for applications using external VCO/VCXO/VCSOs. However, the VCO input can also be driven DC-coupled by LVPECL, CML, or any other driver type within the input's specified common mode range. See the Applications Information section for more information on common VCO input interface configurations, noting that the LTC6952's VCO input buffer has an internal differential resistance of 250Ω as shown in Figure 6.

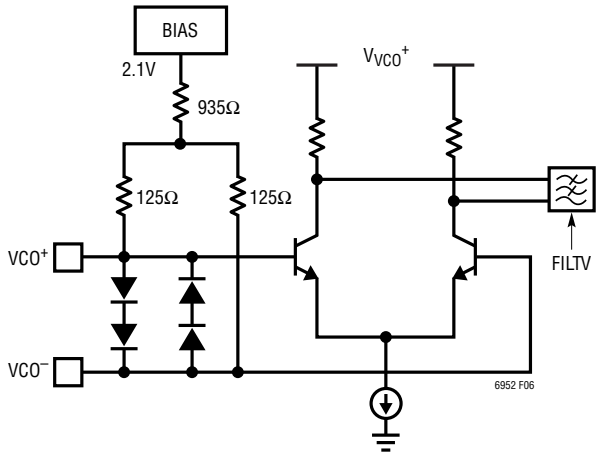



**Figure 6. Simplified VCO Interface Schematic**

The maximum input frequency for the VCO buffer is 4.5GHz, and the maximum amplitude is $1.6V_{P-P}$. It is also important that the VCO± input signal be low noise and have a slew rate of at least 100V/µs, although better performance will be achieved with a higher slew rate. For applications with VCO input slew rates less than 2V/ns, better phase noise performance will be achieved by enabling the internal broadband noise filtering circuit within the VCO input buffer. This is accomplished by asserting the configuration bit FILTV in serial port register h02. Note that setting FILTV = 1 when the slew rate of the VCO input is greater than 2V/ns will degrade

## OPERATION

the overall PLL phase noise performance. See Table 8 for recommended settings of FILTV.

**Table 8. FILTV Programming**

| FILTV | Slew Rate of VCO Input |
|---|---|
| 1 | < 2V/ns |
| 0 | ≥ 2V/ns |

### VCO Peak Detector

A VCO input peak detection circuit is provided on the VCO± inputs to detect the presence of a VCO signal and provides the VCOOK and $\overline{\text{VCOOK}}$ status flags available through both the STAT output and serial port register h00. $\overline{\text{VCOOK}}$ is the logical inverse of VCOOK. The circuit has hysteresis to prevent the VCOOK flag from chattering at the detection threshold. The reference peak detector may be powered-down using the PDVCOPK bit found in register h02.

The peak detector approximates an RMS detector, therefore sine and square wave inputs will give different detection thresholds by a factor of 4/π. See Table 9 for VCOOK detection values.

**Table 9. VCOOK, $\overline{\text{VCOOK}}$ Status Output vs VCO Input**

| VCOOK | $\overline{\text{VCOOK}}$ | Sine Wave $f_{VCO}$ | Square Wave $f_{VCO}$ |
|---|---|---|---|
| 1 | 0 | ≥ 350mV$_{P-P}$ | ≥ 275mV$_{P-P}$ |
| 0 | 1 | < 100mV$_{P-P}$ | < 75mV$_{P-P}$ |

### VCO DIVIDER (N)

The 16-bit N divider provides the feedback from the external VCO to the PFD. The divide ratio may be directly programmed from 1 to 65535 using the ND[15:0] bits found in registers h08 and h09. See the Applications Information section for the relationship between N and the $f_{REF}$, $f_{PFD}$, and $f_{VCO}$ frequencies.

### OUTPUT DIVIDERS (M0 TO M10)

The eleven independent, identical output dividers are driven directly from the VCO input buffer. They divide the incoming VCO frequency $f_{VCO}$ by the divide value Mx to produce a 50% duty cycle output signal at frequency $f_{OUTx}$. The Mx value is set by the MPx[4:0] and the MDx[2:0] bits using the following equation:

$$Mx = (MPx + 1) \bullet 2^{MDx} \quad (1)$$

*For proper operation, MDx must be 0 if Mx is less than or equal to 32.*

Any divider can be muted or powered down to save current by adjusting its corresponding PDx[1:0] bits. The description of the PDx[1:0] bits is shown in Table 10.

**Table 10. PDx[1:0] Programming**

| PDx[1:0] | DESCRIPTION |
|---|---|
| 0 | Normal Operation |
| 1 | Mute Output (OUTx=0 if OINVx=0, OUTx=1 if OINVx=1), Internal Divider Remains Running and Synchronized |
| 2 | Power Down Output (Both + and – Outputs Go to $V_{OUT}^+$), Internal Divider Remains Running and Synchronized |
| 3 | Power Down Divider and Output (Both + and – Outputs Go to $V_{OUT}^+$), Internal Divider Stops and Must Be Re-Synchronized Upon Return to Normal Operation |

### DIGITAL OUTPUT DELAYS (DDEL0 TO DDEL10)

Each output divider can have the start time of the output delayed by integer multiples of ½ of the VCO period after a synchronization event. The digital delay value is programmed into the DDELx[11:0] bits and can be any value from 0 to 4095. Digital delays are only enabled when the synchronization bits SRQENx are set to "1", and any changes to the output digital delays will not be reflected until after synchronization. Digital delay can be used with no degradation to clock jitter performance. See the Operation section on Synchronization and the Applications Information section for details on the use of the digital delay settings.

### ANALOG OUTPUT DELAYS (ADEL0 TO ADEL10)

Each output has a fine analog delay feature to further adjust its output delay time ($t_{ADELx}$) in small steps controlled by the ADELx[5:0] bits. For output frequencies less than 300MHz, absolute time delays range from 0 to 1.1ns. Above 300MHz, the time delay is output frequency dependent, and the valid useful range of ADELx is reduced according to Table 11.

**Table 11. Maximum ADELx vs Output Frequency Range**

| $f_{OUT}$ Range | Maximum ADELx |
|---|---|
| $f_{OUTx}$ ≤ 750MHz | 63 |
| 750MHz < $f_{OUTx}$ ≤ 1GHz | 31 |
| 1GHz < $f_{OUTx}$ ≤ 1.5GHz | 16 |
| 1.5GHz < $f_{OUTx}$ ≤ 1.75GHz | 12 |
| 1.75GHz < $f_{OUTx}$ ≤ 2.25GHz | 8 |
| $f_{OUTx}$ > 2.25GHz | 0 |

## OPERATION

Figure 7 shows the approximate analog delay time ($t_{ADELx}$) vs ADELx and output frequency. Note that the y-axis is logarithmic scale, and that analog delay is zero for ADEL = 0. See the Applications Information section for a more comprehensive method of calculating expected analog delay.

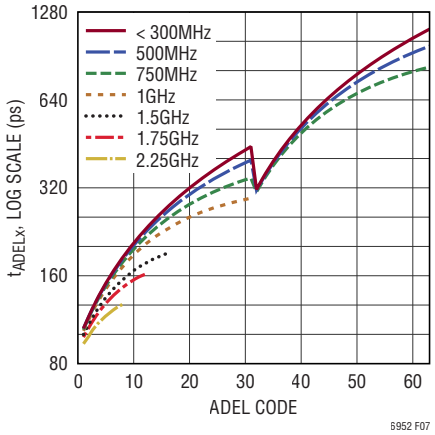



**Figure 7. Analog Delay vs ADEL Code and Output Frequency**

*Use caution when using analog delay on device clocks as this will degrade jitter. Digital delay should be used whenever possible since it does not impact performance. The maximum value of analog delay will never need to be more than half of a VCO input period.*

Analog delays are always enabled regardless of the value of the SRQENx bits, and they take effect immediately upon a write to the ADELx registers. However, changes in ADEL can cause the output to glitch temporarily, especially switching between ADEL=0 and ADEL≠0. See the Applications Information section for details on the use of the analog delay settings. The LTC6952Wizard may be used for ADEL calculation and visualization.

### CML OUTPUT BUFFERS (OUT0 TO OUT10)

All of the outputs are very low noise, low skew 2.5V CML buffers. Each output can be either AC or DC coupled, and terminated with 100Ω differentially. If a single-ended output is desired, each side of the CML output can be individually AC coupled and terminated with 50Ω. The OINVx bits can selectively invert the sense of each output to facilitate board routing without having to cross matched length traces. OINVx also determines the state of the output in a muted condition (PDx = 1) as shown in Table 10. See Figure 8 for circuit details.

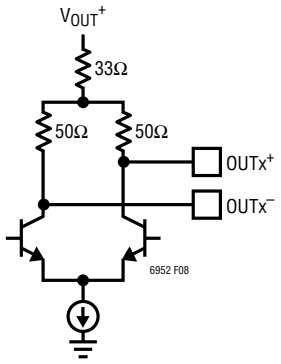



**Figure 8. Simplified CML Interface Schematic (All OUTx)**

### OUTPUT SYNCHRONIZATION AND SYSREF GENERATION

The LTC6952 has circuitry to allow all outputs to be synchronized into known phase alignments in multiple ways to suit different applications using the EZSync and ParallelSync Multichip Clock Edge Synchronization protocols. Synchronization can be between any combination of outputs on the same chip (EZSync Standalone), across multiple cascaded follower chips (EZSync Multi-Chip), or even across multiple parallel chips on the same reference domain (ParallelSync). Once the outputs are at the correct frequency and synchronized, the LTC6952 also has the ability to produce free-running, gated, or finitely pulsed SYSREF signals as indicated by the JESD204B/C subclass 1 specification.

#### EZS_SRQ Input Buffer

Both synchronization and SYSREF requests are achieved by either a software signal (bit SSRQ in register h0B) or a voltage signal on the EZS_SRQ± pins. The voltage on these pins may be any differential signal within the specifications in the Electrical Characteristics, or alternatively a CMOS signal on EZS_SRQ+ while EZS_SRQ– is tied to GND. A simplified schematic of the EZS_SRQ input is shown in Figure 9. When using the SSRQ bit, the state of the EZS_SRQ± pins must be a logic “0”, easily achieved by setting both EZS_SRQ± pins to GND. Likewise, when using the EZS_SRQ± pins, bit SSRQ must be set to “0”. Table 12 shows the use of the EZS_SRQ± pins and SSRQ bit vs the SRQMD and PARSYNC bits. SSRQ bit control

## OPERATION

is disabled when PARSYNC is “1” due to setup and hold time requirements to the REF input.

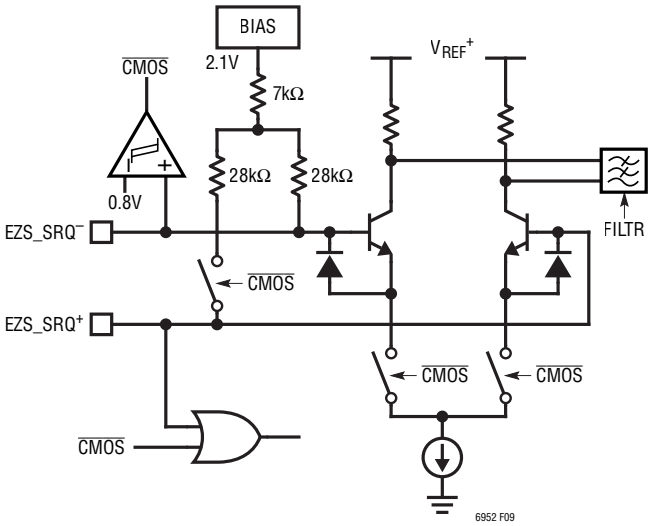



**Figure 9. Simplified EZS_SRQ Interface Schematic**

*Note that synchronization MUST be performed before a SYSREF request. The synchronization must be repeated only if the divider setting is changed, or if the divider is powered down.*

**Table 12. Purpose of EZS_SRQ± pins and SSRQ bit**

| SRQMD | EZS_SRQ PINS | | SSRQ BIT | |
|---|---|---|---|---|
| | PARSYNC=0 | PARSYNC=1 | PARSYNC=0 | PARSYNC=1 |
| 0 | Synchronization Request (SYNC) | Synchronization Request (SYNC) | Synchronization Request (SYNC) | Disabled |
| 1 | SYSREF Request (SYSREQ) | SYSREF Request (SYSREQ) | SYSREF Request (SYSREQ) | Disabled |

### Synchronization Overview

The goal of synchronization is to align all output dividers on single or multiple LTC6952s (or other EZSync or ParallelSync Analog Devices clock parts) into a known phase relationship. At initial power-up, after a power-on reset (POR), or any time the output divide values are changed, the outputs will not be synchronized. Any changes to the output digital delays (DDELx) will not be reflected until after synchronization. Although the outputs will be at the correct frequency without synchronization, the phases will have an unknown relationship until a synchronization event occurs.

To enable synchronization on the LTC6952, the SRQMD bit in register h0B must be set to “0”. Synchronization begins either with the EZS_SRQ input driven to a high state or by writing “1” to the SSRQ bit (only if PARSYNC = 0). For any output with its SRQENx bit set to “1”, the output divider will stop running and return to a logic “0” state after an internal timing delay of greater than 100µs. *The EZS_SRQ input state or SSRQ bit must remain high for a minimum of 1ms.*

Additionally, if bit PARSYNC is set to “1” when the EZS_SRQ input is driven high, the R divider for R ≥ 2 is reset as shown in Figure 2 and explained in the Reference Divider (R) section. This synchronizes the internal PFD reference inputs on multiple parallel LTC6952s.

When the EZS_SRQ input is driven back low, or “0” is written to the SSRQ bit (only if PARSYNC = 0), the synchronized internal dividers will start after an initial latency dependent on the settings of bits PDPLL and PARSYNC , as shown in Table 13 and Table 14. Outputs with DDELx≠0 will be delayed by an extra DDELx/2 VCO cycles. The behavior of each output will be defined individually by the output’s corresponding SRQENx and MODEx bits also shown in Table 13 and Table 14. All dividers with the same DDELx delay setting will have their output rising edge occur within the skew times as defined in the Electrical Characteristics table. The range of each delay is 0 to 4095 VCO half cycles and is independent of the divide ratio setting of each divider. See the Applications Information section for synchronization programming examples. Additionally, the LTC6952Wizard may be used to visualize these timing relationships.

### SYSREF Generation Overview

The JESD204B/C subclass 1 specification describes a method to align multiple data converter devices (ADCs or DACs) in time and provide repeatable and programmable latency across the serial link with a logic device (FPGA). The Local Multi-Frame Clocks (LMFC) and internal clock dividers on all devices in the system are synchronized by a pulse (or pulse train) named SYSREF. Care must be taken to make sure the SYSREF signal remains synchronized to the ADC, DAC, and FPGA clocks and meets setup and hold timing as specified by the devices.

## OPERATION

**Table 13. Synchronization (SRQMD = 0) Output Behavior vs Device Settings for with PLL Enabled (PDPLL = 0)**

| SRQENx | PARSYNC | MODEx | Internal Divider Sync to Other Dividers | Internal Divider Sync to RDIV | Internal Divider Start Latency from SYNC Signal Falling Edge (DDELx = 0) | Output Behavior |
|---|---|---|---|---|---|---|
| 0 | N/A | 0 | No | No | N/A | Free Running |
| | | 1, 2, or 3 | | | | Muted |
| 1 | 0 | 0 | Yes | Yes | ~45µs + $R/f_{REF}$ + $8/f_{VCO}$ | Mute on SYNC High, Run on SYNC Low |
| | | 1 or 3 | | | | Muted |
| | | 2 | | | | SYNC Signal Pass Through |
| | 1 | 0 | Yes | Yes | $(x + R)/f_{REF}$ + $8/f_{VCO}$* | Mute on SYNC High, Run on SYNC Low |
| | | 1 or 3 | | | | Muted |
| | | 2 | | | | SYNC Signal Pass Through |

* Latency depends on the duration of the SYNC pulse, specifically x = floor ($t_{SYNC} \times f_{REF}$ - 1) mod R. All outputs will be synchronized correctly regardless of x

**Table 14. Synchronization (SRQMD = 0) Output Behavior vs Device Settings for with PLL Disabled (PDPLL = 1)**

| SRQENx | PARSYNC | MODEx | Internal Divider Sync to Other Dividers | Internal Divider Sync to RDIV | Internal Divider Start Latency from SYNC Signal Falling Edge (DDELx = 0) | Output Behavior |
|---|---|---|---|---|---|---|
| 0 | N/A | 0 | No | No | N/A | Free Running |
| | | 1, 2, or 3 | | | | Muted |
| 1 | 0 | 0 | Yes | No | ~45µs + $7/f_{VCO}$ | Mute on SYNC High, Run on SYNC Low |
| | | 1 or 3 | | | | Muted |
| | | 2 | | | | SYNC Signal Pass Through |
| | 1 | 0 | Yes | No | $7/f_{VCO}$ | Mute on SYNC High, Run on SYNC Low |
| | | 1 or 3 | | | | Muted |
| | | 2 | | | | SYNC Signal Pass Through |

The LTC6952 supports three different methods of SYSREF generation as described in the JESD204B/C specification:

- Free Running
- Gated On/Off by a SYSREF Request Signal
- One, Two, Four, or Eight SYSREF Pulses After the Rising Edge of a SYSREF Request Signal

These modes are defined by each output's individually programmable MODEx bits. In order to generate SYSREF pulses, bit SRQMD must be set to "1" as shown in Table 12, and MPx must be greater than 0. SYSREF requests (SYSREQ) are applied on the EZS_SRQ± pins or by setting the SSRQ bit to "1" (unless PARSYNC = 1). Table 15 describes the output behavior in SYSREF generation mode. Bits SYSCT[1:0] can be found in register h0B.

*Note that synchronization MUST be completed prior to SYSREF generation as described in the Synchronization Overview.*

# OPERATION

**Table 15. Output Behavior in SYSREF Generation Mode (SRQMD = 1)**

| SRQENx | MODEx | Output Behavior |
|---|---|---|
| 0 | 0 | Free Run, Ignore SYSREQ |
| | 1, 2, or 3 | Muted, Ignore SYSREQ |
| 1 | 0 | Free Run, Ignore SYSREQ |
| | 1 | Gated pulses: Run on SYSREQ High, Mute on Low |
| | 2 | SYSREQ Pass Through |
| | 3 | Output $2^{SYSCT}$ Pulses After SYSREQ Goes High |

## Multi-Chip Synchronization and SYSREF Generation

Using one LTC6952 in EZSync Standalone configuration (Figure 10), up to eleven clock signals or SYSREFs can be generated and synchronized. For applications requiring more than eleven clock outputs, the LTC6952 supports two methods of multi-chip synchronization and SYSREF generation: EZSync Multi-Chip and ParallelSync. The synchronization configuration is determined by bits EZMD and PARSYNC, and their required settings are shown in Table 16. Table 17 introduces the important attributes of these methods and their variants, with further details provided in the following paragraphs. Note that this table only refers to two-stage applications. Many more outputs are possible by using more stages.

**Table 16. Settings of EZMD and PARSYNC for Different Synchronization Topologies**

| Control Bit | EZSync Standalone (see Figure 10) | EZSync Multi-Chip (see Figure 11 and Figure 12) | | ParallelSync Multi-Chip (see Figure 13 and Figure 14) |
|---|---|---|---|---|
| | | CONTROLLER | FOLLOWER | |
| PARSYNC | 0 | 0 | 0 | 1 |
| EZMD | 0 | 0 | 1 | 0 |

**Table 17. Parameters and Limitations of EZSync and ParallelSync (Two Stages Only)**

| | EZSync Standalone (see Figure 10) | EZSync Multi-Chip | | | | ParallelSync Multi-Chip | |
|---|---|---|---|---|---|---|---|
| | | Pin Controlled Requests (see Figure 11) | | Request Passthough (see Figure 12) | | General Reference Distribution (see Figure 13) | LTC6952 Reference Distribution (see Figure 14) |
| | | Controller | Follower | Controller | Follower | | |
| RMS Jitter[a] | ~75fs | ~75fs | ~105fs | ~75fs | ~105fs | ~75fs | ~75fs |
| Possible Number of Followers (Nfol) | - | 1 to 11 | | 1 to 5 | | - | - |
| Possible Number of Parallel Parts (Npar) | - | - | | - | | Unlimited[b] | 1 to 5 |
| Total Number of Outputs | 11 | 11 - Nfol | 11 · Nfol | 11 – 2Nfol | 11 · Nfol | 11 · Npar | 11 · Npar |
| Maximum Number of Outputs | 11 | 121 | | 56 | | Unlimited[b] | 55 |
| Maximum Skew | $t_{SKEW}$ | ~ $t_{SKEW}$ + $t_{PD}$[c] | | ~ $t_{SKEW}$ + $t_{PD}$[c] | | ~$t_{SKEW}$[d] | ~$t_{SKEW}$[d] |
| SYNC Timing | Easy | Easy | | Easy | | Moderate | Easy |
| SYSREF Request Timing | Easy | Moderate | | Easy | | Moderate | Easy |
| Number of External VCOs | 1 | 1 | | 1 | | Npar | Npar |
| Software SYNC/SYSREF Request? | Yes | No | | Yes | | No | Yes |

[a] Assumes ADC SNR equivalent integrated PLL/VCO RMS jitter contribution of 27fs and additive jitter for distribution-only parts of 70fs.

[b] The only limitation is the ability to distribute the reference accurately.

[c] Assumes worst case skew between controller and follower outputs. Dependent on propagation delay of follower and skew of controller-to-follower routing.

[d] Dependent on skew of reference distribution parts, reference routing, and individual part-to-part skew.

# OPERATION

### EZSync Multi-Chip

When using EZSync Multi-Chip, compatible devices are cascaded together, with the clock output of a CONTROLLER device driving the VCO inputs of one to eleven FOLLOWER devices as shown in Figure 11. The EZSync protocol allows for simple synchronization of all devices due to loose timing constraints on the SYNC signal. When used in a JESD204B/C application, SYSREF requests may need to be retimed to a free running SYSREF output to assure all FOLLOWERS start and stop their SYSREF signals at the same time. It is recommended that LTC6953 be used as any FOLLOWER device. However, LTC6952 can be used as a FOLLOWER if necessary by disabling its PLL (PDPLL=1).

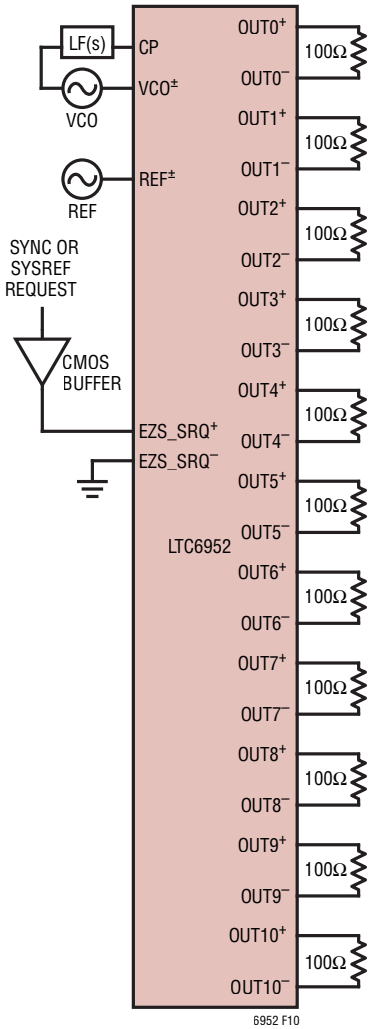



**Figure 10. EZSync Standalone**

To simplify both SYNC and/or SYSREF requests down to a simple software write to the CONTROLLER's SSRQ bit, the devices may be connected as shown in Figure 12, where an additional CONTROLLER output drives each FOLLOWER's EZS_SRQ pins (only available for LTC6952 or LTC6953 FOLLOWERs). This request passthrough configuration reduces the system complexity at the cost of fewer possible FOLLOWERs (a maximum of 5). Note that MPx for the CONTROLLER's passthrough output must be set greater than 0.

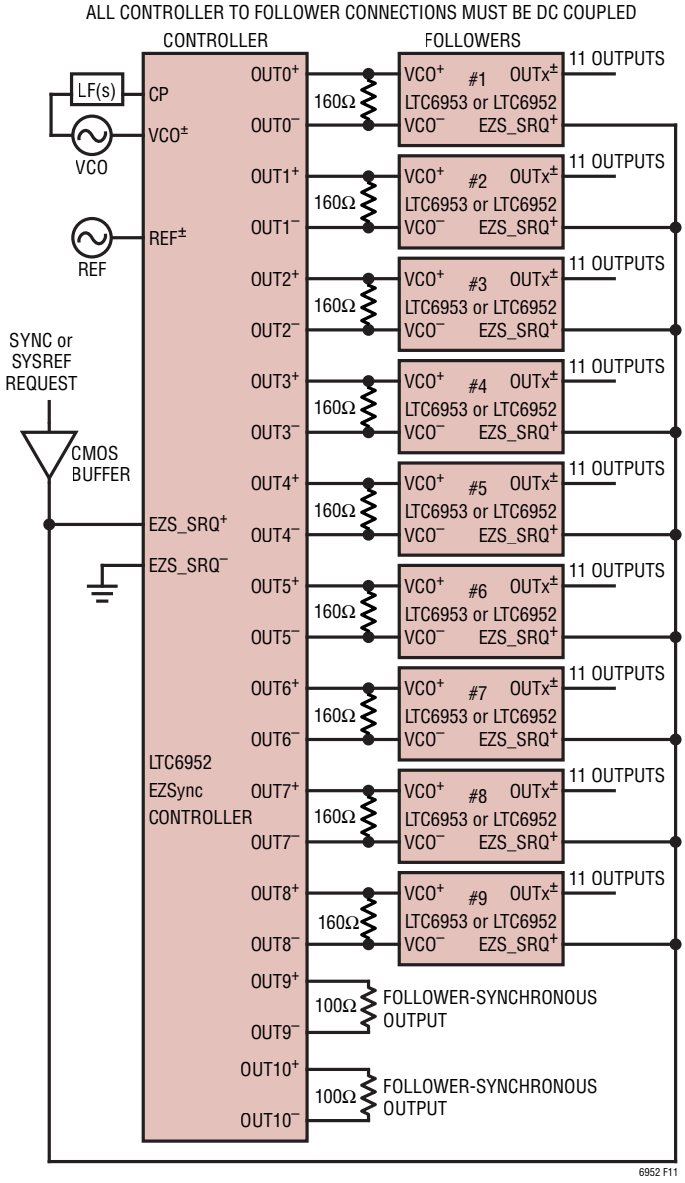



**Figure 11. EZSync Multi-Chip Synchronization (Nine Followers Shown) Max Eleven Possible**

## OPERATION

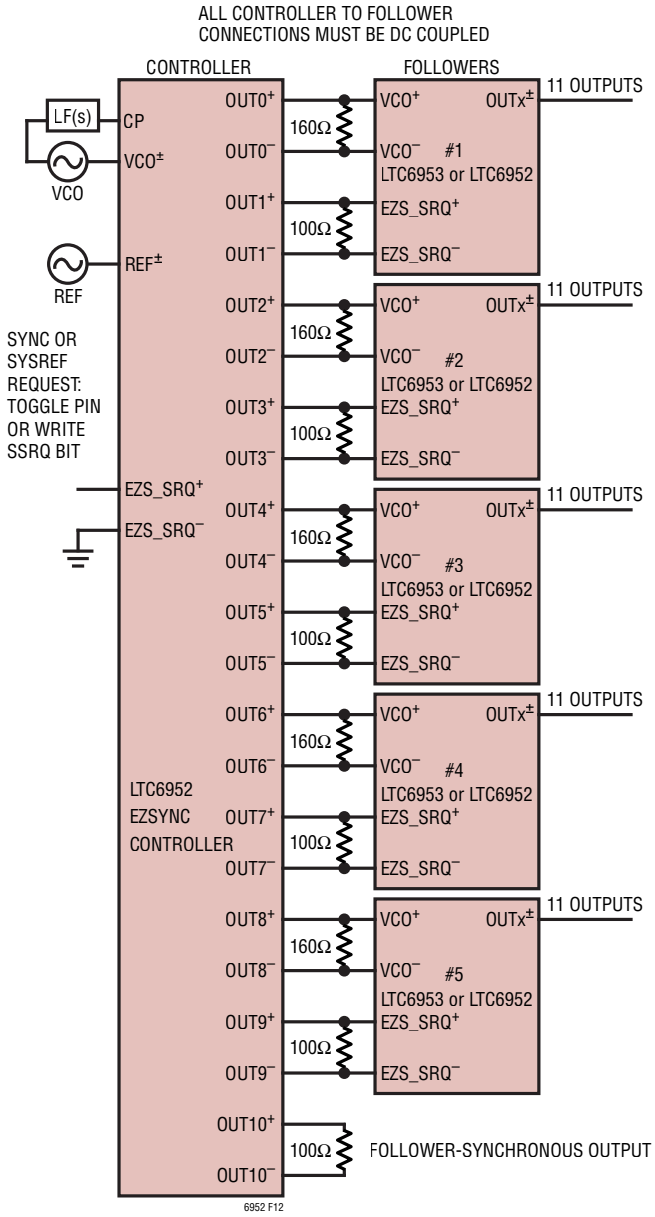



**Figure 12. EZSync Multi-Chip Synchronization with Request Passthrough**

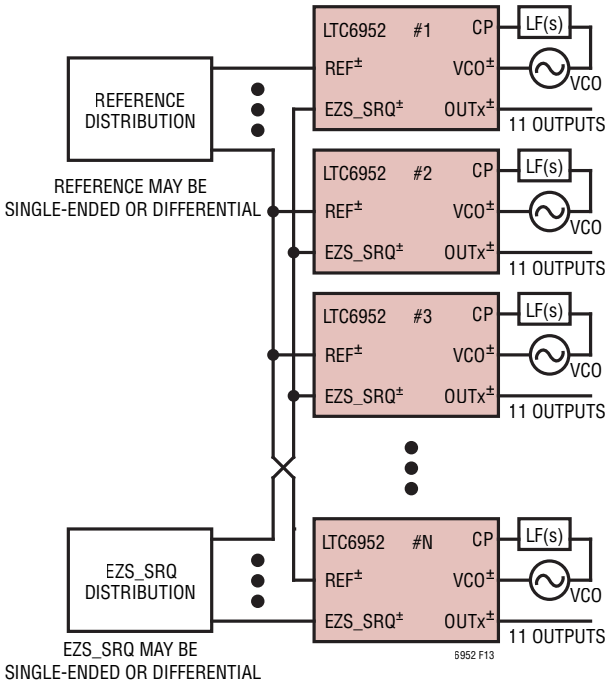



**Figure 13. ParallelSync Multi-Chip Synchronization**

For all cases of EZSync synchronization, the CONTROLLER must be programmed to output seven pre-pulses to each FOLLOWER before the FOLLOWER outputs or any follower-synchronous CONTROLLER outputs start clocking. Additionally, a CONTROLLER must have bit EZMD set to "0" and a FOLLOWER must have both EZMD and PDPLL set to "1". See the Applications Information section for a programming example. Additionally, the LTC6952Wizard provides programming guidance.

### ParallelSync

In a ParallelSync application, multiple ParallelSync compatible devices are connected in parallel with a shared distributed REF signal as shown in Figure 13. The advantage of parallel connection is improved jitter performance, as the clock signals do not propagate through two or more cascaded devices. However, synchronization requires tighter control of the SYNC and SYSREF request (SRQ) signals' timing because of the need to have the SYNC/SRQ edges fall within the same REF cycle for all connected devices. See Reference Signal and EZS_SRQ Timing for PARSYNC Mode in the Applications Information section for the timing requirements of EZS_SRQ to REF in this mode.

## OPERATION

The SYNC/SRQ timing for ParallelSync can be simplified to a single software bit write by using an LTC6953 (or an LTC6952 configured as a reference clean-up loop) as the reference and EZS_SRQ distribution block, as shown in Figure 14. In this application, the EZS_SRQ outputs of the reference distribution part should be set to transition on the falling edge of its corresponding reference clock output. To achieve this, first synchronize the reference distribution part using the settings given in Table 18, where $DDEL_{REF}$ can be any valid DDEL value.

Just before sending a SYNC or SYSREF request to the parallel parts, set the reference distribution part's SRQMD bit to "1". This will automatically retime the passed-through requests to the reference clocks. After the request is done, set the SRQMD bit back to "0" to save supply current from the reference distribution part. See the Applications Information section for a programming example.

**Table 18. Reference Distribution Divider and DDEL Settings for ParallelSync**

| REF CLK Divide | REF CLK DDEL | EZS_SRQ Divide | EZS_SRQ DDEL |
|---|---|---|---|
| 1 | $DDEL_{REF}$ | 2 | $DDEL_{REF}$+1 |
| 2 | $DDEL_{REF}$ | 2 | $DDEL_{REF}$+2 |
| 3 | $DDEL_{REF}$ | 3 | $DDEL_{REF}$+3 |
| 4 | $DDEL_{REF}$ | 4 | $DDEL_{REF}$+4 |
| REF Divide >4 | $DDEL_{REF}$ | =REF Divide | $DDEL_{REF}$ |

Part-to-part skew in a ParallelSync application can be minimized by setting the RAO bit in register h06 to "1". RAO stands for "reference aligned output", and it aligns internal delays such that the output rising edge will always occur at an exact integer number of VCO clock cycles from the incoming reference signal. The trade-off for using the RAO mode is slightly degraded PLL in-band noise (<1.0dB).

To determine the best configuration for a given application, the flowchart in Figure 15 can be used. This flowchart uses the parameters from Table 17 to guide the user to the most suitable configuration.

Depending on the user's system requirements, many simplifications or additions can be made for multiple chip synchronization. For example, the above applications only assume a maximum of two stages, even though more stages can be added to increase the number of outputs. However, these applications are beyond the scope of this data sheet. Please contact the factory.

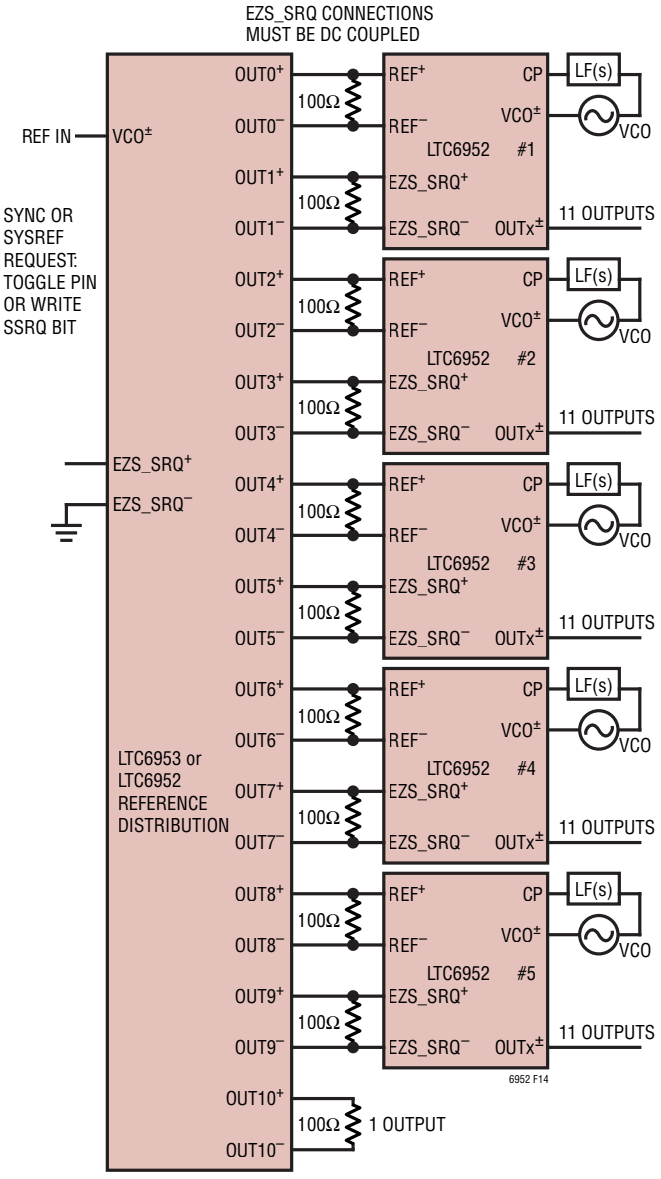



**Figure 14. ParallelSync Multi-Chip Synchronization with LTC6953 or LTC6952 Reference Distribution**

## OPERATION

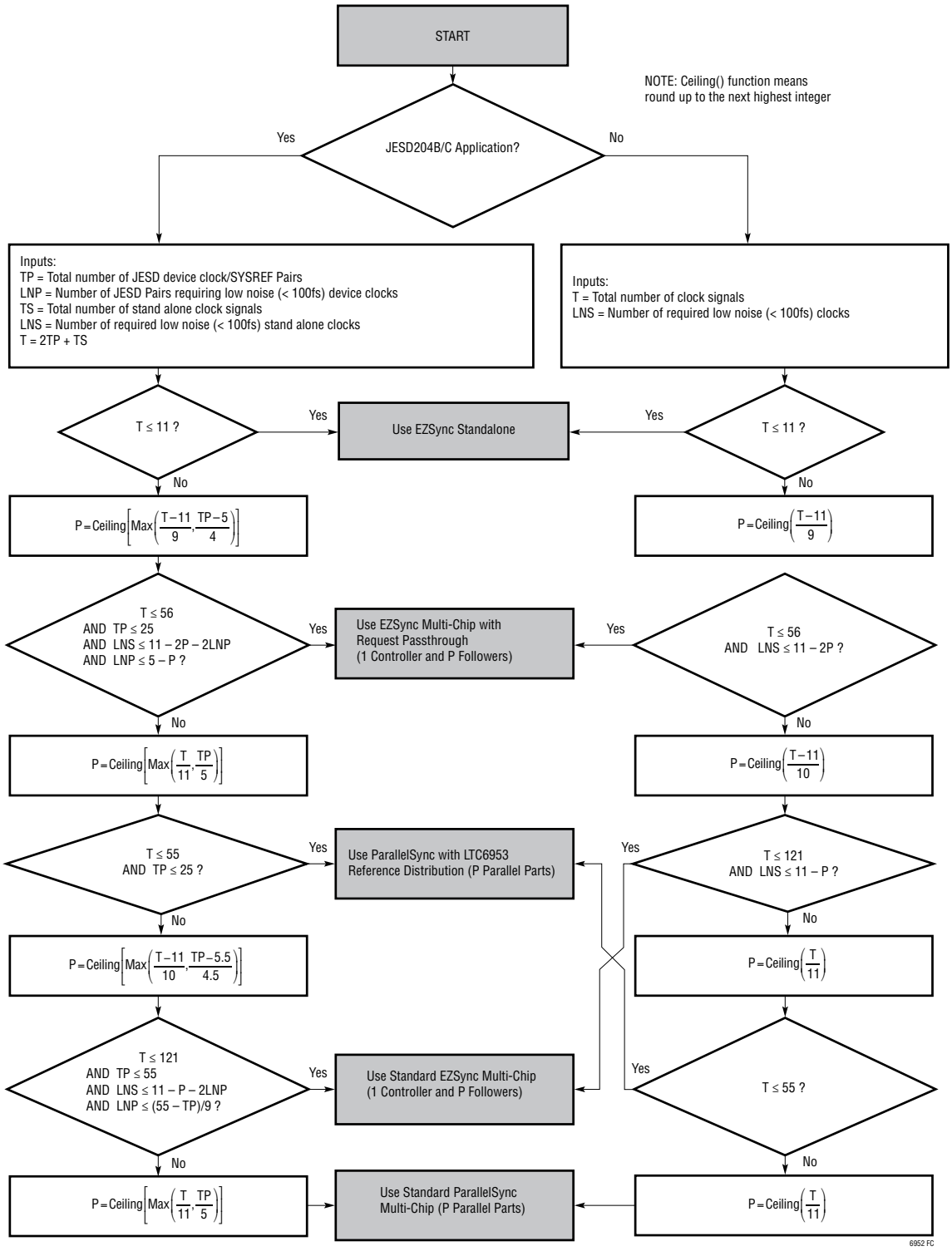



**Figure 15. Flowchart to Determine the Best Synchronization Protocol for a Given Application**

# OPERATION

### Power Savings in SYSREF Generation Mode

In most applications, SYSREF requests are a rare occurrence. The LTC6952 provides modes to shut down as much circuitry as possible while still maintaining the correct timing relationship between SYSREF outputs and clock outputs. Individual outputs may be put into a low power mode while leaving the internal dividers running by writing a “2” to the PDx bits, where x is the output of interest. Additionally, putting the LTC6952 into SYSREF generation mode (SRQMD=1) causes the part to draw a significantly higher current than SRQMD=0. Therefore, leave the SRQMD bit set to “0” until a SYSREF request is required. When a SYSREF signal is needed, set SRQMD to “1”, return the PDx bits to “0”, then wait at least 50µs before issuing a SYSREF request. Put the SYSREF outputs back into low power mode (PDx=2) and set SRQMD=0 when finished.

### SERIAL PORT

The SPI-compatible serial port provides control and monitoring functionality. A configurable status output STAT gives additional instant monitoring.

### Communication Sequence

The serial bus is composed of $\overline{CS}$, SCLK, SDI, and SDO. Data transfers to the part are accomplished by the serial bus master device first taking $\overline{CS}$ low to enable the LTC6952’s port. Input data applied on SDI is clocked on the rising edge of SCLK, with all transfers MSB first. The communication burst is terminated by the serial bus master returning $\overline{CS}$ high. See Figure 16 for details.

Data is read from the part during a communication burst using SDO. Readback may be multidrop (more than one LTC6952 connected in parallel on the serial bus), as SDO is three-stated (Hi-Z) when $\overline{CS}$ is high, or when data is not being read from the part. *If the LTC6952 is not used in a multidrop configuration, or if the serial port master is not capable of setting the SDO line level between read sequences, it is recommended to attach a high-value resistor of greater than 200k between SDO and GND to ensure the line returns to a known level during Hi-Z states.* See Figure 17 for details.

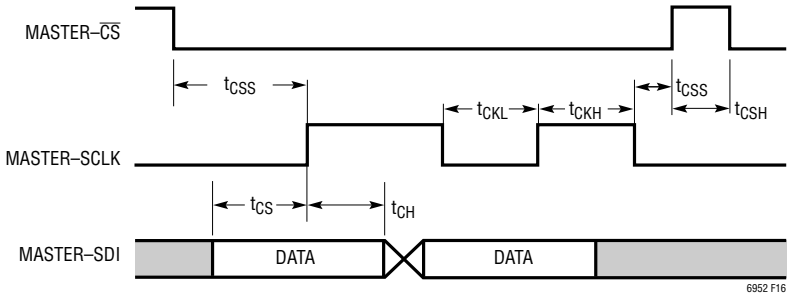



**Figure 16. Serial Port Write Timing Diagram**

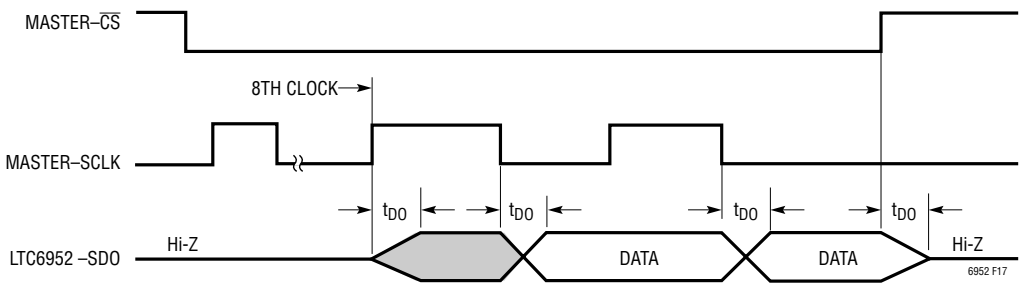



**Figure 17. Serial Port Read Timing Diagram**

# OPERATION

### Single Byte Transfers

The serial port is arranged as a simple memory map, with status and control available in 56, byte-wide registers. All data bursts are comprised of at least two bytes. The 7 most significant bits of the first byte are the register address, with an LSB of "1" indicating a read from the part, and LSB of "0" indicating a write to the part. The subsequent byte, or bytes, is data from/to the specified register address. See Figure 18 for an example of a detailed write sequence, and Figure 19 for a read sequence.

Figure 20 shows an example of two write communication bursts. The first byte of the first burst sent from the serial bus master on SDI contains the destination register address (ADDRX) and an LSB of "0" indicating a write. The next byte is the data intended for the register at address ADDRX. $\overline{CS}$ is then taken high to terminate the transfer. The first byte of the second burst contains the destination register address (ADDRY) and an LSB indicating a write. The next byte on SDI is the data intended for the register at address ADDRY. $\overline{CS}$ is then taken high to terminate the transfer.

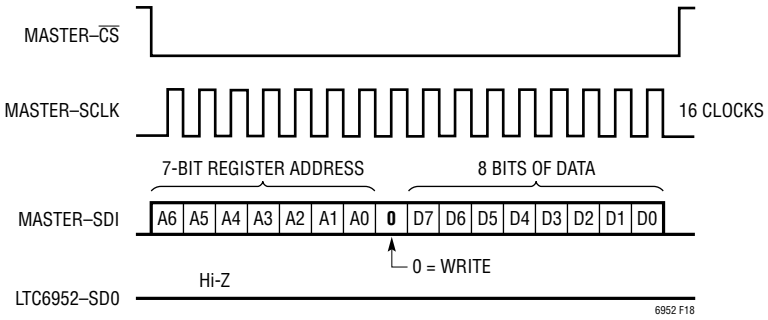



**Figure 18. Serial Port Write Sequence**

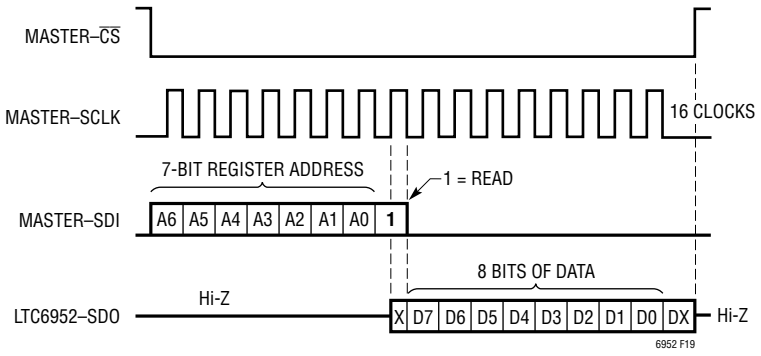



**Figure 19. Serial Port Read Sequence**

## OPERATION

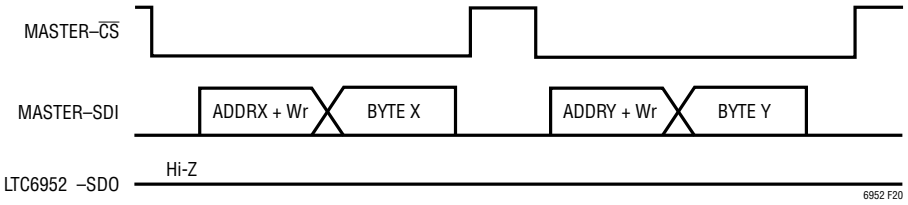



**Figure 20. Serial Port Single Byte Writes**

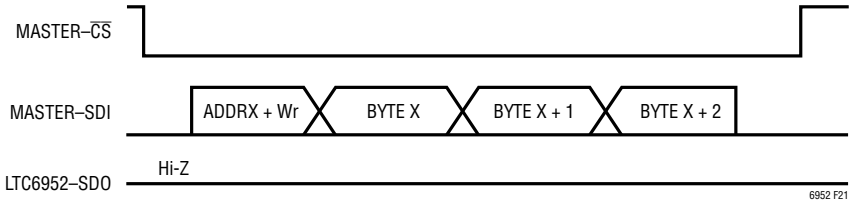



**Figure 21. Serial Port Auto-Increment Write**

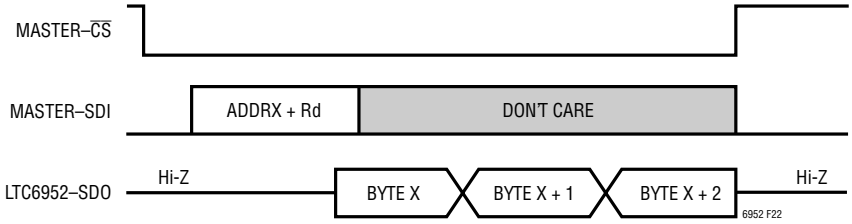



**Figure 22. Serial Port Auto-Increment Read**

### Multiple Byte Transfers

More efficient data transfer of multiple bytes is accomplished by using the LTC6952's register address auto-increment feature as shown in Figure 21. The serial port master sends the destination register address in the first byte and its data in the second byte as before, but continues sending bytes destined for subsequent registers. Byte 1's address is ADDRX+1, Byte 2's address is ADDRX+2, and so on. If the register address pointer attempts to increment past 56 (h38), it is automatically reset to 0.

An example of an auto-increment read from the part is shown in Figure 22. The first byte of the burst sent from the serial bus master on SDI contains the destination register address (ADDRX) and an LSB of "1" indicating a read. Once the LTC6952 detects a read burst, it takes SDO out of the Hi-Z condition and sends data bytes sequentially, beginning with data from register ADDRX. The part ignores all other data on SDI until the end of the burst.

# OPERATION

### Multidrop Configuration

Several LTC6952s may share the serial bus. In this multidrop configuration, SCLK, SDI, and SDO are common between all parts. The serial bus master must use a separate $\overline{CS}$ for each part and ensure that only one device has $\overline{CS}$ asserted at any time. It is recommended to attach a high-value resistor to SDO to ensure the line returns to a known level during Hi-Z states.

### Serial Port Registers

The memory map of the LTC6952 may be found in Table 19, with detailed bit descriptions found in Table 20. The register address shown in hexadecimal format under the "ADDR" column is used to specify each register. Each register is denoted as either read-only (R) or read-write (R/W). The register's default value on device power-up or after a reset is shown at the right.

The read-only register at address h00 is used to determine different status flags. These flags may be instantly output on the STAT pin by configuring register h01. See "STAT Output" section below for more information.

The register at address h38 is a read-only byte for device identification.

**Table 19. Serial Port Register Contents**

| ADDR | MSB | [6] | [5] | [4] | [3] | [2] | [1] | LSB | R/W | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|
| h00 | | UNLOCK | $\overline{LOCK}$ | LOCK | $\overline{VCOOK}$ | VCOOK | $\overline{REFOK}$ | REFOK | R | |
| h01 | INVSTAT | x[6] | x[5] | x[4] | x[3] | x[2] | x[1] | x[0] | R/W | hAA |
| h02 | PDALL | PDPLL | PDVCOPK | PDREFPK | BST | FILTR | FILTV | POR | R/W | h08 |
| h03 | PD3[1] | PD3[0] | PD2[1] | PD2[0] | PD1[1] | PD1[0] | PD0[1] | PD0[0] | R/W | h00 |
| h04 | PD7[1] | PD7[0] | PD6[1] | PD6[0] | PD5[1] | PD5[0] | PD4[1] | PD4[0] | R/W | h00 |
| h05 | TEMPO | | PD10[1] | PD10[0] | PD9[1] | PD9[0] | PD8[1] | PD8[0] | R/W | h00 |
| h06 | RAO | PARSYNC | | LKWIN | LKCT[1] | LKCT[0] | RD[9] | RD[8] | R/W | h0C |
| h07 | RD[7] | RD[6] | RD[5] | RD[4] | RD[3] | RD[2] | RD[1] | RD[0] | R/W | h01 |
| h08 | ND[15] | ND[14] | ND[13] | ND[12] | ND[11] | ND[10] | ND[9] | ND[8] | R/W | h00 |
| h09 | ND[7] | ND[6] | ND[5] | ND[4] | ND[3] | ND[2] | ND[1] | ND[0] | R/W | h2D |
| h0A | CPRST | CPUP | CPDN | CP[4] | CP[3] | CP[2] | CP[1] | CP[0] | R/W | h93 |
| h0B | CPMID | CPWIDE | CPINV | EZMD | SRQMD | SYSCT[1] | SYSCT[0] | SSRQ | R/W | h86 |
| h0C | MP0[4] | MP0[3] | MP0[2] | MP0[1] | MP0[0] | MD0[2] | MD0[1] | MD0[0] | R/W | h00 |
| h0D | SRQEN0 | MODE0[1] | MODE0[0] | OINV0 | DDEL0[11] | DDEL0[10] | DDEL0[9] | DDEL0[8] | R/W | h00 |
| h0E | DDEL0[7] | DDEL0[6] | DDEL0[5] | DDEL0[4] | DDEL0[3] | DDEL0[2] | DDEL0[1] | DDEL0[0] | R/W | h00 |
| h0F | | | ADEL0[5] | ADEL0[4] | ADEL0[3] | ADEL0[2] | ADEL0[1] | ADEL0[0] | R/W | h00 |
| h10 | MP1[4] | MP1[3] | MP1[2] | MP1[1] | MP1[0] | MD1[2] | MD1[1] | MD1[0] | R/W | h00 |
| h11 | SRQEN1 | MODE1[1] | MODE1[0] | OINV1 | DDEL1[11] | DDEL1[10] | DDEL1[9] | DDEL1[8] | R/W | h00 |
| h12 | DDEL1[7] | DDEL1[6] | DDEL1[5] | DDEL1[4] | DDEL1[3] | DDEL1[2] | DDEL1[1] | DDEL1[0] | R/W | h00 |
| h13 | | | ADEL1[5] | ADEL1[4] | ADEL1[3] | ADEL1[2] | ADEL1[1] | ADEL1[0] | R/W | h00 |
| h14 | MP2[4] | MP2[3] | MP2[2] | MP2[1] | MP2[0] | MD2[2] | MD2[1] | MD2[0] | R/W | h00 |
| h15 | SRQEN2 | MODE2[1] | MODE2[0] | OINV2 | DDEL2[11] | DDEL2[10] | DDEL2[9] | DDEL2[8] | R/W | h00 |
| h16 | DDEL2[7] | DDEL2[6] | DDEL2[5] | DDEL2[4] | DDEL2[3] | DDEL2[2] | DDEL2[1] | DDEL2[0] | R/W | h00 |
| h17 | | | ADEL2[5] | ADEL2[4] | ADEL2[3] | ADEL2[2] | ADEL2[1] | ADEL2[0] | R/W | h00 |
| h18 | MP3[4] | MP3[3] | MP3[2] | MP3[1] | MP3[0] | MD3[2] | MD3[1] | MD3[0] | R/W | h00 |
| h19 | SRQEN3 | MODE3[1] | MODE3[0] | OINV3 | DDEL3[11] | DDEL3[10] | DDEL3[9] | DDEL3[8] | R/W | h00 |

## OPERATION

**Table 19. Serial Port Register Contents (Continued)**

| ADDR | MSB | [6] | [5] | [4] | [3] | [2] | [1] | LSB | R/W | DEFAULT |
|---|---|---|---|---|---|---|---|---|---|---|
| h1A | DDEL3[7] | DDEL3[6] | DDEL3[5] | DDEL3[4] | DDEL3[3] | DDEL3[2] | DDEL3[1] | DDEL3[0] | R/W | h00 |
| h1B | | | ADEL3[5] | ADEL3[4] | ADEL3[3] | ADEL3[2] | ADEL3[1] | ADEL3[0] | R/W | h00 |
| h1C | MP4[4] | MP4[3] | MP4[2] | MP4[1] | MP4[0] | MD4[2] | MD4[1] | MD4[0] | R/W | h00 |
| h1D | SRQEN4 | MODE4[1] | MODE4[0] | OINV4 | DDEL4[11] | DDEL4[10] | DDEL4[9] | DDEL4[8] | R/W | h00 |
| h1E | DDEL4[7] | DDEL4[6] | DDEL4[5] | DDEL4[4] | DDEL4[3] | DDEL4[2] | DDEL4[1] | DDEL4[0] | R/W | h00 |
| h1F | | | ADEL4[5] | ADEL4[4] | ADEL4[3] | ADEL4[2] | ADEL4[1] | ADEL4[0] | R/W | h00 |
| h20 | MP5[4] | MP5[3] | MP5[2] | MP5[1] | MP5[0] | MD5[2] | MD5[1] | MD5[0] | R/W | h00 |
| h21 | SRQEN5 | MODE5[1] | MODE5[0] | OINV5 | DDEL5[11] | DDEL5[10] | DDEL5[9] | DDEL5[8] | R/W | h00 |
| h22 | DDEL5[7] | DDEL5[6] | DDEL5[5] | DDEL5[4] | DDEL5[3] | DDEL5[2] | DDEL5[1] | DDEL5[0] | R/W | h00 |
| h23 | | | ADEL5[5] | ADEL5[4] | ADEL5[3] | ADEL5[2] | ADEL5[1] | ADEL5[0] | R/W | h00 |
| h24 | MP6[4] | MP6[3] | MP6[2] | MP6[1] | MP6[0] | MD6[2] | MD6[1] | MD6[0] | R/W | h00 |
| h25 | SRQEN6 | MODE6[1] | MODE6[0] | OINV6 | DDEL6[11] | DDEL6[10] | DDEL6[9] | DDEL6[8] | R/W | h00 |
| h26 | DDEL6[7] | DDEL6[6] | DDEL6[5] | DDEL6[4] | DDEL6[3] | DDEL6[2] | DDEL6[1] | DDEL6[0] | R/W | h00 |
| h27 | | | ADEL6[5] | ADEL6[4] | ADEL6[3] | ADEL6[2] | ADEL6[1] | ADEL6[0] | R/W | h00 |
| h28 | MP7[4] | MP7[3] | MP7[2] | MP7[1] | MP7[0] | MD7[2] | MD7[1] | MD7[0] | R/W | h00 |
| h29 | SRQEN7 | MODE7[1] | MODE7[0] | OINV7 | DDEL7[11] | DDEL7[10] | DDEL7[9] | DDEL7[8] | R/W | h00 |
| h2A | DDEL7[7] | DDEL7[6] | DDEL7[5] | DDEL7[4] | DDEL7[3] | DDEL7[2] | DDEL7[1] | DDEL7[0] | R/W | h00 |
| h2B | | | ADEL7[5] | ADEL7[4] | ADEL7[3] | ADEL7[2] | ADEL7[1] | ADEL7[0] | R/W | h00 |
| h2C | MP8[4] | MP8[3] | MP8[2] | MP8[1] | MP8[0] | MD8[2] | MD8[1] | MD8[0] | R/W | h00 |
| h2D | SRQEN8 | MODE8[1] | MODE8[0] | OINV8 | DDEL8[11] | DDEL8[10] | DDEL8[9] | DDEL8[8] | R/W | h00 |
| h2E | DDEL8[7] | DDEL8[6] | DDEL8[5] | DDEL8[4] | DDEL8[3] | DDEL8[2] | DDEL8[1] | DDEL8[0] | R/W | h00 |
| h2F | | | ADEL8[5] | ADEL8[4] | ADEL8[3] | ADEL8[2] | ADEL8[1] | ADEL8[0] | R/W | h00 |
| h30 | MP9[4] | MP9[3] | MP9[2] | MP9[1] | MP9[0] | MD9[2] | MD9[1] | MD9[0] | R/W | h00 |
| h31 | SRQEN9 | MODE9[1] | MODE9[0] | OINV9 | DDEL9[11] | DDEL9[10] | DDEL9[9] | DDEL9[8] | R/W | h00 |
| h32 | DDEL9[7] | DDEL9[6] | DDEL9[5] | DDEL9[4] | DDEL9[3] | DDEL9[2] | DDEL9[1] | DDEL9[0] | R/W | h00 |
| h33 | | | ADEL9[5] | ADEL9[4] | ADEL9[3] | ADEL9[2] | ADEL9[1] | ADEL9[0] | R/W | h00 |
| h34 | MP10[4] | MP10[3] | MP10[2] | MP10[1] | MP10[0] | MD10[2] | MD10[1] | MD10[0] | R/W | h00 |
| h35 | SRQEN10 | MODE10[1] | MODE10[0] | OINV10 | DDEL10[11] | DDEL10[10] | DDEL10[9] | DDEL10[8] | R/W | h00 |
| h36 | DDEL10[7] | DDEL10[6] | DDEL10[5] | DDEL10[4] | DDEL10[3] | DDEL10[2] | DDEL10[1] | DDEL10[0] | R/W | h00 |
| h37 | | | ADEL10[5] | ADEL10[4] | ADEL10[3] | ADEL10[2] | ADEL10[1] | ADEL10[0] | R/W | h00 |
| h38 | REV[3] | REV[2] | REV[1] | REV[0] | PART[3] | PART[2] | PART[1] | PART[0] | R | hX2* |

* Varies depending on revision

# OPERATION

**Table 20. Serial Port Register Bit Field Summary**

| BITS | DESCRIPTION | DEFAULT | ADDR |
|---|---|---|---|
| ADELx[5:0] | OUTx Analog Delay Setting | 0 | Various |
| BST | REF Buffer Boost Current | 1 | h02 |
| CP[4:0] | CP Output Current | h13 | h0A |
| CPDN | Force CP Down | 0 | h0A |
| CPINV | Invert PFD Phase | 0 | h0B |
| CPMID | CP Bias to Mid-Rail | 1 | h0B |
| CPRST | CP Hi-Z | 1 | h0A |
| CPUP | Force CP Up | 0 | h0A |
| CPWIDE | Extend CP Pulse Width | 0 | h0B |
| DDELx[11:0] | OUTx Delay in ½ VCO Cycles | h000 | Various |
| EZMD | EZSync Mode | 0 | h0B |
| FILTR | REF Input Buffer Filter | 0 | h02 |
| FILTV | VCO Input Buffer Filter | 0 | h02 |
| INVSTAT | Invert STAT Output | 1 | h01 |
| LKCT[1:0] | PLL Lock Cycle Count:<br>0 = Disabled, 1 = 32 Counts,<br>2 = 256 Counts, 3 = 2048 Counts | h3 | h06 |
| LKWIN | PLL Lock Indicator Window:<br>0 = 3ns, 1 = 10ns | 0 | h06 |
| LOCK | PLL Lock Indicator Flag | | h00 |
| $\overline{\text{LOCK}}$ | PLL Lock Indicator Flag Inverted | | h00 |
| MDx[2:0] | OUTx 2N Value | 0 | Various |
| MODEx[1:0] | SYSREF Mode (SRQENx=1):<br>0 = Free Run<br>1 = Gated Pulses<br>2 = Request Passthrough<br>3 = $2^{SYSCNT}$ Pulses | 0 | Various |
| MPx[4:0] | OUTx Prescaler Value | h0 | Various |
| ND[15:0] | N Divider Value | h002D | h08, h09 |
| OINVx | OUTx Inversion | 0 | Various |
| PARSYNC | ParallelSync Timing Enable | 0 | h06 |
| PART[3:0] | Part Code (h2 = 6952, h3=6953) | | h38 |
| PDALL | Full Chip Power-Down | 0 | h02 |
| PDPLL | Powers Down REF, R DIV, PFD, CP, N DIV | 0 | h02 |
| PDREFPK | Powers Down REF Input Signal Detector | 0 | h02 |
| PDVCOPK | Powers Down VCO Input Signal Detector | 0 | h02 |
| PDx[1:0] | OUTx Power-Down Mode:<br>0 = Normal Operation<br>1 = Output Muted to Logic “0”<br>2 = Power Down Output Driver<br>3 = Power Down Full Divider | h0 | Various |
| POR | Force Power-on-Reset | 0 | h02 |
| RAO | Enable REF Alignment to Output Mode | 0 | h06 |
| RD[9:0] | R Divider Value (RD[9:0] > 0) | 1 | h06,h07 |
| REFOK | Reference Valid Flag | | h00 |
| $\overline{\text{REFOK}}$ | Reference Not Valid Flag | | h00 |
| REV[3:0] | Revision Code | | h38 |
| SRQENx | Enable SYNC or SYSREF on OUTx | 0 | Various |
| SRQMD | 0 - Synchronization Mode<br>1 - SYSREF Request Mode | 0 | h0B |
| SSRQ | Software SYNC or SYSREF Request | 0 | h0B |
| SYSCT[1:0] | SYSREF pulse count for MODEx=3:<br>0=One Pulse, 1=Two Pulses,<br>2=Four Pulses, 3=Eight Pulses | h3 | h0B |
| TEMPO | Enable Temperature Measurement Diode on STAT | | h05 |
| UNLOCK | PLL Unlock Flag | | h00 |
| VCOOK | VCO Valid Flag | | h00 |
| $\overline{\text{VCOOK}}$ | VCO Not Valid Flag | | h00 |
| x[6:0] | STAT Output or Mask | h2A | h01 |

# OPERATION

### STAT Output

The STAT output pin is configured with the x[6:0] bits and INVSTAT of register h01. These bits are used to bit-wise mask, or enable, the corresponding status flags of status register h00, according to Equation 2 and shown schematically in Figure 23. The result of this bit-wise Boolean operation is then output on the STAT pin if TEMPO is set to "0".

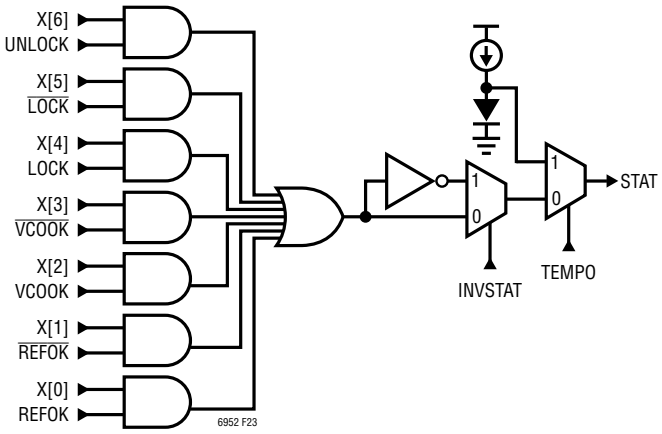



**Figure 23. STAT Simplified Schematic**

$$STAT = (OR\ (Reg00[6:0]\ AND\ Reg01[6:0])) \text{ XOR } INVSTAT \quad (2)$$

For example, if the application requires STAT to go high whenever the LOCK, VCOOK, or REFOK flags are set, then x[4], x[2], and x[0] should be set to "1" and INVSTAT should be set to "0", giving a register value of h15.

The STAT pin may be transformed to a temperature measurement diode with internal 300µA bias current by setting bit TEMPO in register h05 to "1". To get an approximate die temperature, a single calibration point is needed first. Measure the STAT pin voltage ($V_{TEMPC}$) with the LTC6952 powered down (PDALL = 1) at a known temperature ($T_{CAL}$). Then the operating temperature may be calculated in a desired application by measuring the STAT voltage again ($V_{TEMP}$) and using the following equation:

$$T = 665 \times (V_{TEMPC} - V_{TEMP}) + T_{CAL}$$

where T and $T_{CAL}$ are in °C. Note that no external bias current is required. Allow 50µs settling time after setting TEMPO to "1".

### BLOCK POWER-DOWN CONTROL

The LTC6952's power-down control bits are located in register h02, described in Table 20. Different portions of the device may be powered down independently. To power down individual outputs, see Table 10. *Care must be taken with the LSB of this register, the POR (power-on-reset) bit. When written to "1", this bit forces a full reset of the part's digital circuitry to its power-up default state.*

# APPLICATIONS INFORMATION

## INTRODUCTION

A PLL is a complex feedback system that may conceptually be considered a frequency multiplier. The system multiplies the frequency input at REF± up to the VCO frequency. The PFD, charge pump, N divider, VCO, and loop filter form a feedback loop to accurately control the VCO frequency (see Figure 24). The R divider, output dividers (Mx), and input frequency $f_{REF}$ are used to set the output frequency value and resolution, and the external loop filter is used to set the PLL's loop bandwidth, BW.

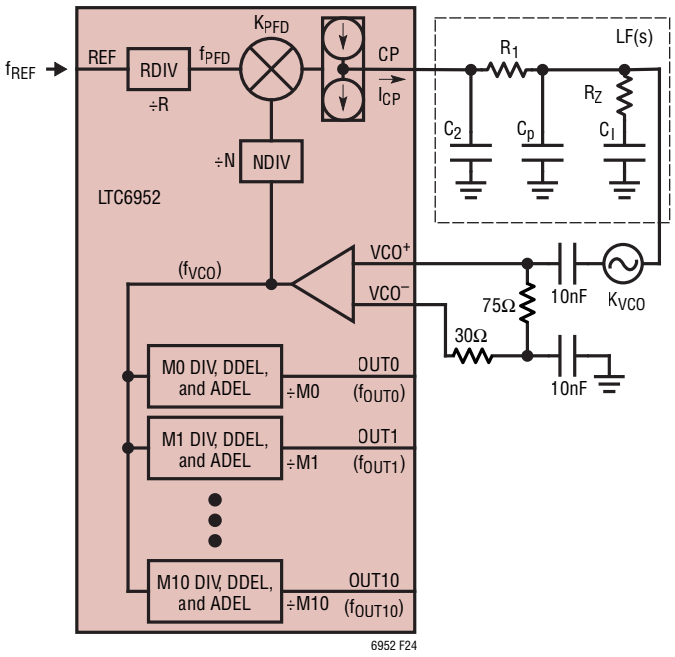



**Figure 24. PLL Loop Diagram**

## OUTPUT FREQUENCY

When the loop is locked, the frequency $f_{VCO}$ (in Hz) produced at the output of the VCO is determined by the reference frequency $f_{REF}$, and the R and N divider values, given by Equation 3:

$$f_{VCO} = f_{REF} \bullet N/R \quad (3)$$

The PFD frequency $f_{PFD}$ is given by the following equation:

$$f_{PFD} = f_{REF}/R \quad (4)$$

and $f_{VCO}$ may be alternatively expressed as:

$$f_{VCO} = f_{PFD} \bullet N \quad (5)$$

or

$$N = f_{VCO}/f_{PFD} \quad (6)$$

The output frequency $f_{OUTx}$ produced at the output of the Mx dividers is given by Equation 7:

$$f_{OUTx} = f_{VCO}/Mx \quad (7)$$

Using Equation 3 and Equation 7, the output frequency resolution $f_{STEPx}$ produced by a unit change in N is given by Equation 8:

$$f_{STEPx} = f_{REF}/(R \bullet Mx) \quad (8)$$

## LOOP FILTER DESIGN

A stable PLL system requires care in designing the external loop filter. The Analog Devices LTC6952Wizard application, available from designtools, aids in design and simulation of the complete system. Optimum phase noise and spurious performance can be obtained by using the third-order loop filter shown in Figure 24.

The loop design should use the following algorithm:

1. *Determine the output frequencies $f_{OUTx}$ based on application requirements.* Using Equation 3, Equation 4, and Equation 7, change $f_{REF}$, N, R and Mx until the application frequency constraints are met. Use the minimum R value that still satisfies the constraints.
2. *Select the open loop bandwidth BW constrained by $f_{PFD}$.* A stable loop requires that BW is less than $f_{PFD}$ by at least a factor of 10.
3. *Select loop filter component $R_Z$ and charge pump current $I_{CP}$ based on BW and the VCO gain factor, $K_{VCO}$.* BW (in Hz) is approximated by the following equation:

$$BW \cong I_{CP} \bullet R_Z \bullet K_{VCO}/(2 \bullet \pi \bullet N) \quad (9)$$

or

$$R_Z = (2 \bullet \pi \bullet BW \bullet N)/(I_{CP} \bullet K_{VCO})$$

## APPLICATIONS INFORMATION

where $K_{VCO}$ is in Hz/V, $I_{CP}$ is in Amps, and $R_Z$ is in Ohms. $K_{VCO}$ depends on the external VCO chosen for the application. If the chosen VCO has a negative $K_{VCO}$, the CPINV bit must be set to "1" for correct operation. Use $I_{CP}$ = 11.2mA to lower in-band noise unless component values force a lower setting.

4. *Select loop filter components $C_I$, $C_P$, $C_2$, and $R_1$ based on BW and $R_Z$.* Use the following equations to calculate the remaining loop filter components.

$$C_I = 4/(\pi \bullet BW \bullet R_Z) \quad (10)$$

$$C_P = 1/(12 \bullet \pi \bullet BW \bullet R_Z) \quad (11)$$

$$C_2 = 1/(18 \bullet \pi \bullet BW \bullet R_Z) \quad (12)$$

$$R_1 = R_Z \quad (13)$$

### DIGITAL AND ANALOG OUTPUT DELAYS

Synchronization allows the start times of each output divider to be delayed by the value programmed into the digital delay bits (DDELx), expressed in ½ VCO cycles. Applications needing to calculate the delay in terms of time can use Equation 14 where DDELx is DDEL0 to DDEL10:

$$t_{DDELx} = DDELx/(2 \bullet f_{VCO}) \quad (14)$$

The analog delay blocks (ADELx) are useful in trimming signal timing differences caused by non-ideal PCB routing. This is effective for optimizing set-up and hold times for SYSREFs versus device clocks in JESD204B/C applications. Unlike digital delay, adding analog delay will adversely affect the jitter performance. Add analog delay to the SYSREF path whenever possible to minimize the impact on the device clocks. For example, if the SYSREF signal in a SYSREF/Clock pair is arriving at the destination device too late, it is better to add one digital delay code to the device clock and then add analog delay to the SYSREF, if necessary, to bring it closer to the device clock.

The *approximated* analog delay time can be calculated in picoseconds (ps) by Equation 15 (for ADELx < 32) while adhering to the frequency limitations described in Table 11.

ADELx = 1 to 31

$$t_{ADELx} = [(11.25 \bullet ADELx + 93.8)^{-2.5} + (0.00285 \bullet f_{OUTx})^{2.5}]^{-0.4} \quad (15)$$

ADELx = 32 to 63

$$t_{ADELx} = [(26 \bullet ADELx - 517)^{-2.5} + (0.00125 \bullet f_{OUTx})^{2.5}]^{-0.4} \quad (16)$$

where $f_{OUT}$ is the output frequency in GHz. The LTC6952Wizard may be used for analog delay calculation and visualization.

### REFERENCE INPUT

The LTC6952's reference input buffer, shown in Figure 1, provides a flexible interface to either differential or single-ended frequency sources. The frequency range for the reference input is from 1MHz to 500MHz. As discussed in the Operation section, a high quality signal must be applied to the REF± inputs as they provide the frequency reference to the entire PLL. To achieve the part's in-band phase noise performance, apply a sine wave signal of at least 6dBm into 50Ω, or a square wave of at least $0.5V_{P-P}$ with slew rate of at least 20V/µs. Figure 25 shows recommended interfaces for different reference signal types.

### VCO INPUT

The LTC6952's VCO input buffer, shown in Figure 6, has a frequency range of DC to 4.5GHz. The buffer has a partial on-chip differential input termination of 250Ω, allowing some flexibility for an external matching network if desired. Figure 26 shows recommended interfaces for different VCO input signal types.

## APPLICATIONS INFORMATION

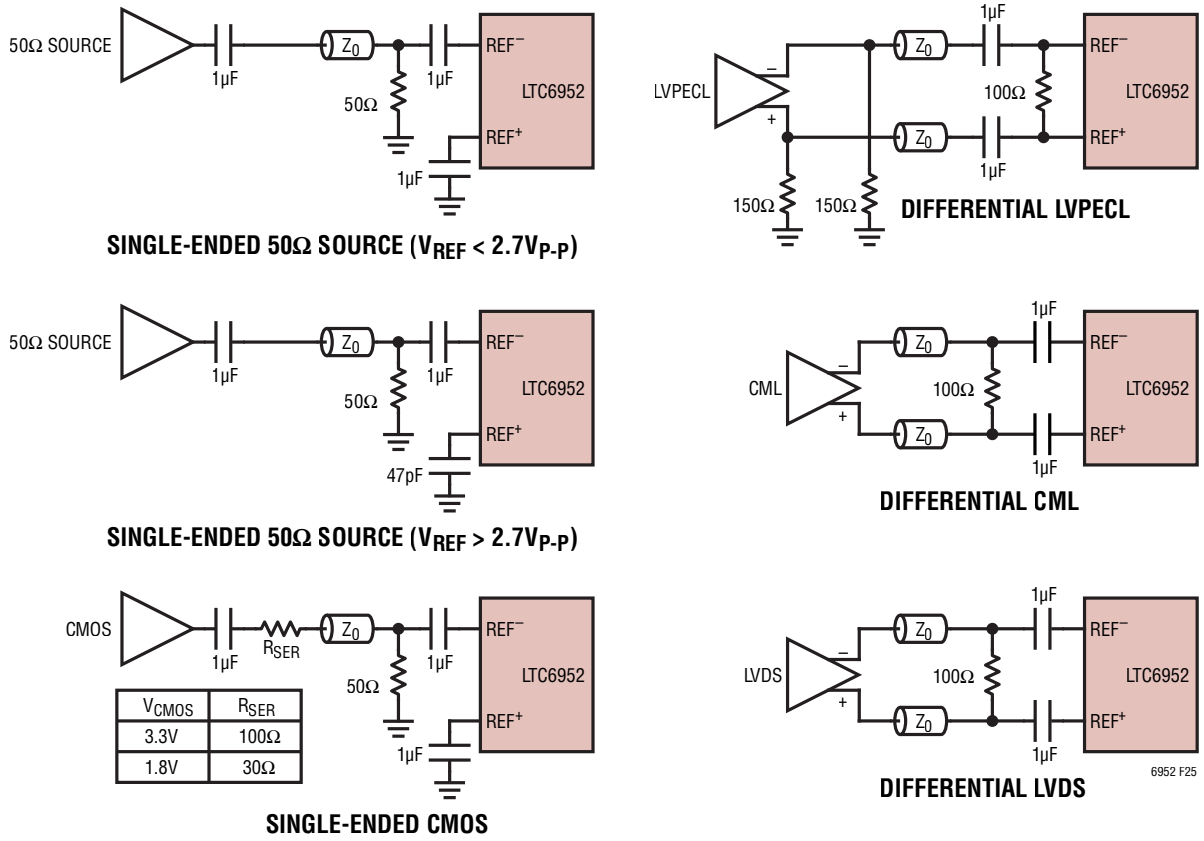
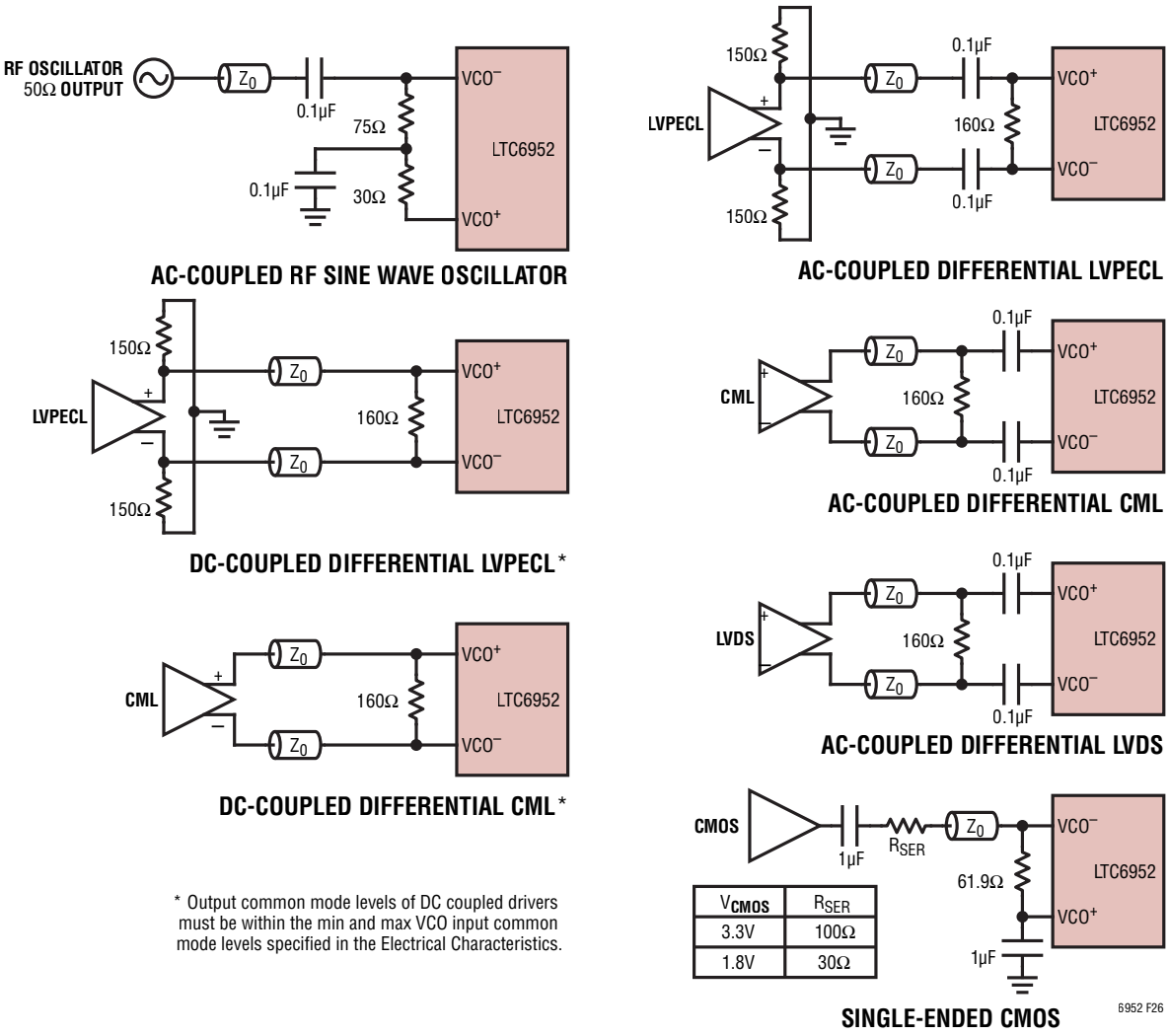



**Figure 25. Common Reference Input Interface Configurations. All $Z_0$ Signal Traces Are 50Ω Transmission Lines**

## APPLICATIONS INFORMATION

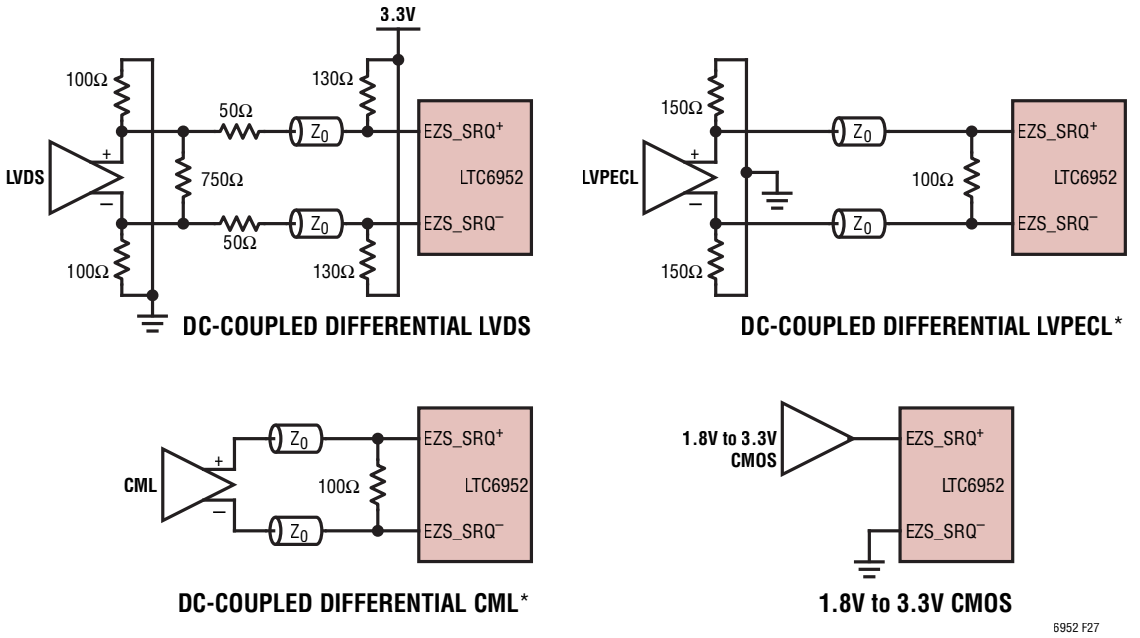



| $V_{CMOS}$ | $R_{SER}$ |
|---|---|
| 3.3V | 100Ω |
| 1.8V | 30Ω |

* Output common mode levels of DC coupled drivers must be within the min and max VCO input common mode levels specified in the Electrical Characteristics.

**Figure 26. Common VCO Input Interface Configurations. All $Z_0$ Signal Traces are 50Ω Transmisson Lines**

# APPLICATIONS INFORMATION




* Output common mode levels of differential drivers must be within the min and max EZS_SRQ input common mode levels specified in the Electrical Characteristics.

**Figure 27. Common EZS_SRQ Input Interface Configurations. All $Z_0$ Signal Traces are 50Ω Transmisson Lines**

## EZS_SRQ INPUT

The LTC6952's EZS_SRQ input buffer, shown in Figure 9, controls synchronization requests and SYSREF requests. All connections must be DC-coupled and can be either differential CML or LVPECL, differential LVDS with a level-shifting network, or single-ended 1.8V to 3.3V CMOS into the EZS_SRQ⁺ input pin (EZS_SRQ⁻ must be grounded for CMOS drive). Figure 27 shows the recommended interface types.

## JESD204B/C DESIGN EXAMPLE USING EZSync STANDALONE

This design example consists of a system of two JESD204B/C analog-to-digital converters (ADCs), two JESD204B/C digital-to-analog converters (DACs), and a JESD204B/C compatible FPGA. All of the data converters (ADCs and DACs) and the FPGA require JESD204B/C subclass 1 device clocks and SYSREFs, and the FPGA requires an extra management clock. Additionally, the ADCs require a low noise clock of less than 100fs total RMS jitter. This leads to a total of 11 separate signals to generate, with frequencies listed below. For this example, the SYSREF frequencies for all devices are the same and should output four pulses upon a SYSREF request rising edge:

$f_{ADC\text{-}CLK}$ = 500MHz

$f_{DAC\text{-}CLK}$ = 4000MHz

$f_{FPGA\text{-}CLK}$ = 125MHz

$f_{FPGA\text{-}MGMT}$ = 100MHz

$f_{SYSREF}$ = 12.5MHz

Since the total number of outputs is 11, a single LTC6952 can be used to generate all of the outputs needed as shown in Figure 28. Note that termination resistors and AC coupling caps are not shown for clarity.

## APPLICATIONS INFORMATION

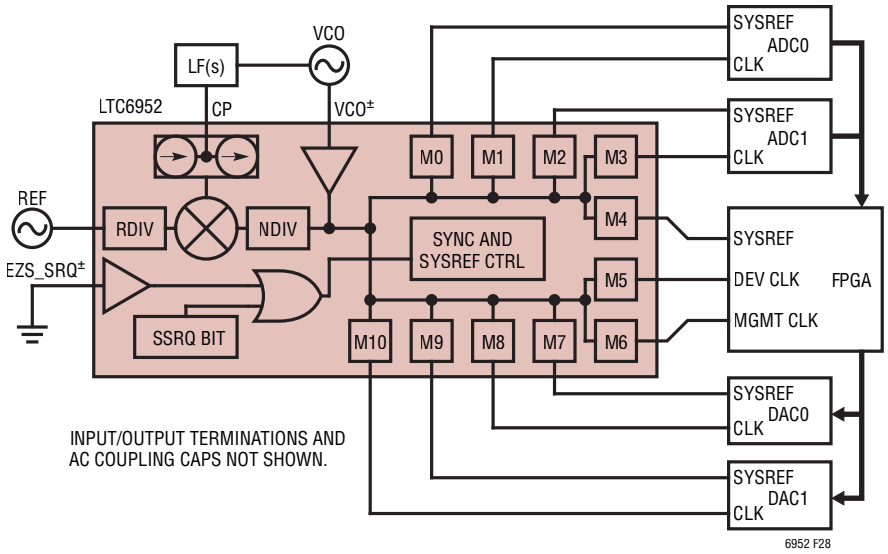



**Figure 28. Block Diagram for JESD204B/C EZSync Standalone Design Example**

### Reference and VCO Assumptions

For this example, assume the available reference is a 100MHz sine wave oscillator with 8dBm output power, and the VCO is a 4000MHz oscillator with a $K_{VCO}$ of 5MHz/V, output power of 7dBm, and phase noise of –115dBc/Hz at 10kHz.

$f_{REF}$ = 100MHz

$f_{VCO}$ = 4000MHz

$K_{VCO}$ = 5MHz/V

### Design Procedure

Designing and enabling this clock generation solution consists of the following steps:

1. Determine R and N divider values
2. Determine the optimum loop bandwidth
3. Select loop filter component values
4. Determine all output modes
5. Determine all M divider values
6. Determine all digital delay values
7. Program the IC with the correct divider values, output delays, and other settings
8. Synchronize the outputs
9. Place the SYSREF outputs in a lower power mode until the next SYSREF request (optional, see Operations section)
10. Place IC into SYSREF request mode (SRQMD=1) and send a SYSREF request when needed
11. Return IC into SYNC mode (SRQMD=0) and place the SYSREF outputs into a lower power mode for power savings (optional).

*Note that synchronization MUST be performed before a SYSREF request. The synchronization must be repeated only if the divider setting is changed, or if the divider is powered down.*

### Determining R and N Divider Values

Following the “Loop Filter Design” algorithm, first determine all the divider values. From the Electrical Characteristics, the maximum $f_{PFD}$ is 167MHz, which is larger than the $f_{REF}$ of 100MHz. Therefore R should be “1” noting that maximizing $f_{PFD}$ in a data converter application will minimize integrated jitter. Use Equation 4 to determine $f_{PFD}$ and Equation 6 to determine N:

R = 1

$f_{PFD} = f_{REF}/R$ = 100MHz

$N = f_{VCO}/f_{PFD}$ = 40

# APPLICATIONS INFORMATION

### Selecting Loop Bandwidth

The next step in the algorithm is choosing the open loop bandwidth (BW). The maximum BW should be at least 10x smaller than $f_{PFD}$. Most data converter applications will place the bandwidth at the optimal intersection of VCO noise and in-band noise. Narrower bandwidths or higher order loop filters can be used to lower spurious power. For this example, the third-order loop filter shown in Figure 24 will be used.

After inputting the external VCO phase noise and $K_{VCO}$ characteristics, the LTC6952Wizard reports the thermal noise optimized loop bandwidth is approximately 16kHz.

### Loop Filter Component Selection

For the lowest in-band phase noise, $I_{CP}$ should be set to the highest value possible that results in practical loop filter component values. Therefore, $I_{CP}$ for our example is chosen to be 11.2mA. The LTC6952Wizard uses Equation 9 to determine $R_Z$:

$$R_Z = 2 \bullet \pi \bullet 16k \bullet 40/(11.2m \bullet 5M)$$

$$R_Z = 71.8\Omega \approx 71.5\Omega$$

The LTC6952Wizard uses Equation 10 through 13 to calculate $C_I$, $C_P$, $C_2$, and $R_1$:

$$C_I = 4/(\pi \bullet 16k \bullet 71.5) = 1.11\mu F \approx 1.2\mu F$$

$$C_P = 1/(12 \bullet \pi \bullet 16k \bullet 71.5) = 23.2nF \approx 22nF$$

$$C_2 = 1/(18 \bullet \pi \bullet 16k \bullet 71.5) = 15.5nF \approx 15nF$$

$$R_1 = 71.5\Omega$$

Note that resistors are rounded to standard ±1% values, and capacitors are rounded to standard ±10% values.

### Determining Output Modes

All outputs can be programmed as clocks (MODEx = 0), SYSREFs (MODEx = 1 or 3), or SYNC/SRQ passthrough outputs (MODEx = 2) using each output's individual MODEx bits as described in Table 13, Table 14, and Table 15. Any output can also be programmed to ignore SYNC and SYSREF requests by setting that output's corresponding SRQENx bit to "0". Noting that this design example calls for pulsed SYSREFs (MODEx = 3) and that the FPGA management clock should always be free running (SRQENx = 0), Table 21 summarizes each output's mode settings.

**Table 21. Output Mode Settings for EZSync Standalone Design Example**

| OUTPUT | PURPOSE | SRQENx | MODEx | PDx |
|---|---|---|---|---|
| OUT0 | ADC0 SYSREF | 1 | 3 | 0 |
| OUT1 | ADC0 CLK | 1 | 0 | 0 |
| OUT2 | ADC1 SYSREF | 1 | 3 | 0 |
| OUT3 | ADC1 CLK | 1 | 0 | 0 |
| OUT4 | FPGA SYSREF | 1 | 3 | 0 |
| OUT5 | FPGA DEV CLK | 1 | 0 | 0 |
| OUT6 | FPGA MGMT CLK | 0 | 0 | 0 |
| OUT7 | DAC0 SYSREF | 1 | 3 | 0 |
| OUT8 | DAC0 CLK | 1 | 0 | 0 |
| OUT9 | DAC1 SYSREF | 1 | 3 | 0 |
| OUT10 | DAC1 CLK | 1 | 0 | 0 |

### Determining Output Divider Values

Since the desired frequencies of each output are already determined, the output divider values can be calculated using Equation 7. The results are shown in Table 22.

**Table 22. Output Divide Settings for EZSync Standalone Design Example**

| OUTPUT | PURPOSE | Frequency (MHz) | Divide Value (Mx) |
|---|---|---|---|
| OUT0 | ADC0 SYSREF | 12.5 | 320 |
| OUT1 | ADC0 CLK | 500 | 8 |
| OUT2 | ADC1 SYSREF | 12.5 | 320 |
| OUT3 | ADC1 CLK | 500 | 8 |
| OUT4 | FPGA SYSREF | 12.5 | 320 |
| OUT5 | FPGA DEV CLK | 125 | 32 |
| OUT6 | FPGA MGMT CLK | 100 | 40 |
| OUT7 | DAC0 SYSREF | 12.5 | 320 |
| OUT8 | DAC0 CLK | 4000 | 1 |
| OUT9 | DAC1 SYSREF | 12.5 | 320 |
| OUT10 | DAC1 CLK | 4000 | 1 |

# APPLICATIONS INFORMATION

### Determining Output Digital Delay Values

The output digital delay is used to control phase relationships between outputs. The minimum delay step is ½ of a period of the incoming VCO signal. For this design example, the digital delay is used to place each device's SYSREF signal edges into a known phase relationship to its corresponding device clock, optimized for the set-up ($t_S$) and hold time ($t_h$) requirements for that device. Assume that the optimum SYSREF edge location for each device occurs on the first falling clock edge before the desired SYSREF valid rising clock edge. In other words, SYSREF should change states ½ of a corresponding device clock period before the SYSREF valid device clock edge. Refer to Figure 29 for a visual example.

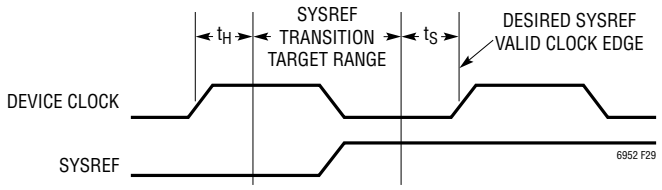



**Figure 29. SYSREF Edge Timing Example**

In order to calculate each output's digital delay value for this design example, use the following procedure:

1. Delay all of the JESD204B/C device clocks by half of a period of the *slowest* JESD204B/C device clock. This delay setting is equal to the divide value of the slowest device clock because a one code digital delay equals half of a VCO cycle. Non-JESD204B/C clocks (such as the FPGA management clock) are not included in this calculation. This delay value defines the desired SYSREF valid clock edge. In this example, the slowest JESD204B/C clock is the FPGA device clock:

   $DDEL_{SYSvalid} = M_{FPGACLK} = 32$

   $DDEL_{ADC\text{-}CLK} = DDEL_{SYSvalid} = 32$

   $DDEL_{DAC\text{-}CLK} = DDEL_{SYSvalid} = 32$

   $DDEL_{FPGA\text{-}CLK} = DDEL_{SYSvalid} = 32$

2. For each device clock/SYSREF pair, subtract half a device clock period from $DDEL_{SYSvalid}$ to find the SYSREF delay. This is equivalent to subtracting the corresponding divide value of the device clock, i.e.

   $DDEL_{ADC\text{-}SYS} = DDEL_{SYSvalid} - M_{ADC\text{-}CLK}$

   $DDEL_{ADC\text{-}SYS} = 32 - 8 = 24$

   $DDEL_{DAC\text{-}SYS} = DDEL_{SYSvalid} - M_{DAC\text{-}CLK}$

   $DDEL_{DAC\text{-}SYS} = 32 - 1 = 31$

   $DDEL_{FPGA\text{-}SYS} = DDEL_{SYSvalid} - M_{FPGA\text{-}CLK}$

   $DDEL_{FPGA\text{-}SYS} = 32 - 32 = 0$

Table 23 summarizes the DDEL settings for all outputs.

**Table 23. Output DDELx Settings for EZSync Standalone Design Example**

| OUTPUT | PURPOSE | DDELx |
|---|---|---|
| OUT0 | ADC0 SYSREF | 24 |
| OUT1 | ADC0 CLK | 32 |
| OUT2 | ADC1 SYSREF | 24 |
| OUT3 | ADC1 CLK | 32 |
| OUT4 | FPGA SYSREF | 0 |
| OUT5 | FPGA DEV CLK | 32 |
| OUT6 | FPGA MGMT CLK | 0 |
| OUT7 | DAC0 SYSREF | 31 |
| OUT8 | DAC0 CLK | 32 |
| OUT9 | DAC1 SYSREF | 31 |
| OUT10 | DAC1 CLK | 32 |

Now that the output divider and delays have been determined, the LTC6952 can be programmed.

### Status Register Programming

This example will use the STAT pin to alert the system whenever the LTC6952 generates a fault condition. Program x[5], x[3], x[1] = 1 to force the STAT pin high whenever any of the $\overline{LOCK}$, $\overline{VCOOK}$, or $\overline{REFOK}$ flags asserts:

Reg01 = h2A

## APPLICATIONS INFORMATION

### Power and FILT Register Programming

For correct PLL operation, all internal blocks should be enabled. Additionally, the REF and VCO input signals have a sufficient slew rate and power to not need the FILT or BST bits:

Reg02 = h00

### Output Power-Down Programming

During initial setup and synchronization, all used outputs and the SRQ circuitry should be set to full power. These bits will be used later to place the IC in a lower power mode while waiting for SYSREF requests:

Reg03 = h00

Reg04 = h00

Reg05 = h00

### RAO and PARSYNC Programming

PARSYNC should be set to "0" since this example is not a ParallelSync application. RAO should also be set to "0" unless a highly accurate reference input to clock output timing is required.

### Lock Detect Programming

Next, determine the lock indicator window from $f_{PFD}$. From Table 4 we see that LKWIN = 0 with a $t_{LWW}$ of 3ns. The LTC6952 will consider the loop "locked" as long as the phase coincidence at the PFD is within ±3ns, or 108° as calculated below.

phase = 360° • $t_{LWW}$ • $f_{PFD}$

= 360° • 3n • 100M

≈ 108°

Larger values of COUNTS lead to more accurate and stable lock indications at the expense of longer lock indication times. A COUNTS value of 2048 will work for this application. From Table 5, LKCT[1:0] = 3 for 2048 counts.

### R and N Divider Programming

The previously determined R and N divider values are 1 and 40, respectively. Using these values and the PARSYNC and lock detector values, registers 6 through 9 can be programmed:

Reg06 = h0C

Reg07 = h01

Reg08 = h00

Reg09 = h28

### Charge Pump Function and Current Programming

Disable all the charge pump functions (CPMID, CPWIDE, CPRST, CPUP, CPDN, and CPINV), allowing the loop to lock. Using Table 6 with the previously selected $I_{CP}$ of 11.2mA gives CP[4:0] = h13.

### SYNC and SYSREF Global Modes Programming

Bit EZMD controls whether the IC is an EZSync standalone/controller ("0") or follower ("1"). Since this example is an EZSync standalone application, set bit EZMD to "0". Bit SRQMD determines if the part is in synchronization mode ("0") or SYSREF request mode ("1"). SYSCT programs the number of pulses for any output in pulsed SYSREF mode (# pulses = $2^{SYSCT}$, so SYSCT = 2 to achieve four pulses for this example). Combining this information with the previously determined charge pump controls, registers h0A and h0B can be programmed:

Reg0A = h13

Reg0B = h04

Note that the SSRQ bit will remain "0" for now, but will be used later during the synchronization and SYSREF request procedures. Also, the EZS_SRQ± pins should be grounded because the synchronization and SYSREF requests will be accomplished through software control of the SSRQ bit.

### Output Divider, Delay and Function Programming

Four registers for each output allow the outputs to be configured independently of each other. The first register controls the output divide ratio through two control words, MPx and MDx, as described in Equation 1.

## APPLICATIONS INFORMATION

The second register contains the control modes and the most significant bits of the digital delay control word. The third register contains the remainder of the digital delay control word, and the fourth register is the analog delay control.

Both the analog delay and the output invert (OINVx) bits can be used to correct PC board layout issues such as mismatched trace lengths and differential signal crossovers, respectively. *Note that the use of analog delay on clock signals will degrade jitter performance.* For this example, assume the PC board is laid out in an ideal manner and no output inversions or analog delays are needed. With this information, all of registers h0C through h37 can be programmed to the values in Table 24, calculated using the information in Table 20, Table 21 (with Equation 1), and Table 22.

**Table 24. Output Register Settings for EZSync Standalone Design Example**

| ADDR | Value | ADDR | Value | ADDR | Value |
|---|---|---|---|---|---|
| **h0C** | h9C | **h1C** | h9C | **h2C** | h00 |
| **h0D** | hE0 | **h1D** | hE0 | **h2D** | h80 |
| **h0E** | h18 | **h1E** | h00 | **h2E** | h20 |
| **h0F** | h00 | **h1F** | h00 | **h2F** | h00 |
| **h10** | h38 | **h20** | hF8 | **h30** | h9C |
| **h11** | h80 | **h21** | h80 | **h31** | hE0 |
| **h12** | h20 | **h22** | h20 | **h32** | h1F |
| **h13** | h00 | **h23** | h00 | **h33** | h00 |
| **h14** | h9C | **h24** | h99 | **h34** | h00 |
| **h15** | hE0 | **h25** | h00 | **h35** | h80 |
| **h16** | h18 | **h26** | h00 | **h36** | h20 |
| **h17** | h00 | **h27** | h00 | **h37** | h00 |
| **h18** | h38 | **h28** | h9C | | |
| **h19** | h80 | **h29** | hE0 | | |
| **h1A** | h20 | **h2A** | h1F | | |
| **h1B** | h00 | **h2B** | h00 | | |

### Synchronization

The outputs in this example are now running at the desired frequency, but have random phase relationships with each other. Synchronization forces the outputs to run at known and repeatable phases and can be achieved in this example either externally, by driving the EZS_SRQ± pins, or internally, with the SSRQ bit in Reg0B. Since the part was just programmed, set the SSRQ bit to "1" and hold the EZS_SRQ± pins low:

Reg0B = h05

After waiting a minimum of 1ms, set SSRQ to "0":

Reg0B = h04

Once the internal synchronization process completes, the outputs will be aligned as shown in Figure 30. Note that the internal divider behavior for the muted SYSREF outputs is shown as well as the actual outputs to demonstrate the phase alignment following synchronization.

### Putting the IC into a Lower Power Mode (Optional)

If desired, the LTC6952 can be placed into a lower power mode while awaiting a SYSREF request. This is achieved by setting PDx = 2 for all SYSREF-defined outputs. This powers down the output driver circuitry but leaves the internal divider running and in the correct phase relationship to the clocks.

### Performing a SYSREF Request

To produce SYSREF pulses, write a "1" to SRQMD and take the LTC6952 out of low power mode (if used) by writing all the SYSREF output PDx bits to "0". Wait 50µs to allow circuitry to power up. Send the SYSREF request by writing a "1" to the SSRQ bit in Reg0B:

Reg0B = h05

After waiting a minimum of 1ms, set SSRQ to "0":

Reg0B = h04

Place the IC back into low power mode if desired by writing a "0" to SRQMD and setting PDx = 2 for all SYSREF defined outputs. After the rising edge of the SYSREF request, the SYSREF outputs will pulse four times and then return to a "0" state as shown in Figure 31.

# APPLICATIONS INFORMATION

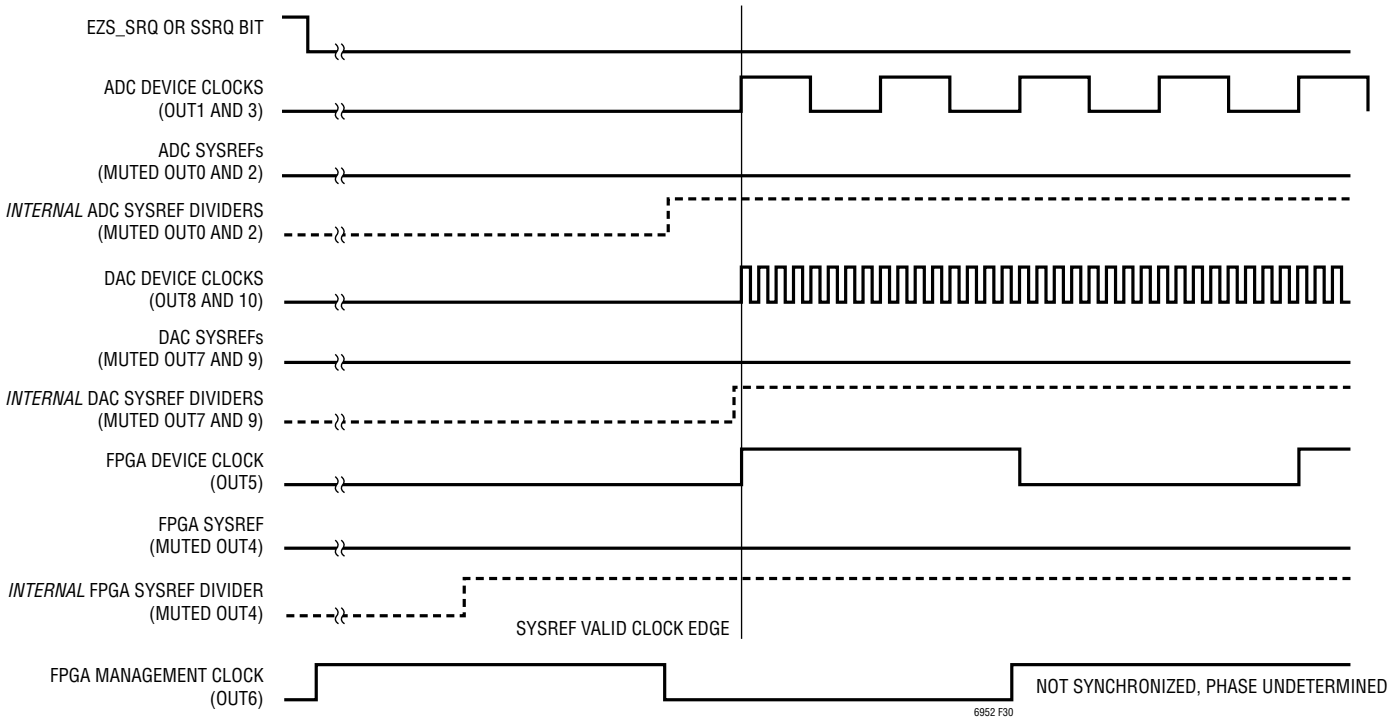



**Figure 30. Outputs after Synchronization for the EZSync Standalone Design Example (SRQMD=0)**

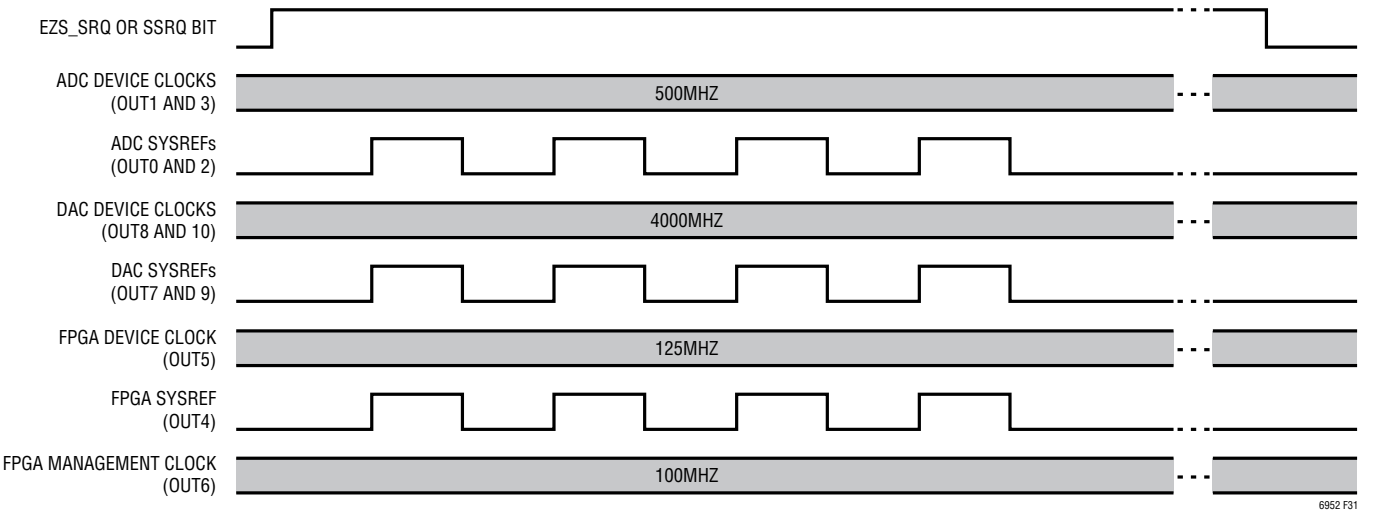



**Figure 31. Outputs after SYSREF Request for the EZSync Standalone Design Example (SRQMD = 1)**

# APPLICATIONS INFORMATION

### JESD204B/C DESIGN EXAMPLE USING EZSync MULTI-CHIP

This design example consists of a system of four JESD204B/C analog-to-digital converters (ADCs), four JESD204B/C digital-to-analog converters (DACs), and a JESD204B/C compatible FPGA. All of the data converters (ADCs and DACs) and the FPGA require JESD204B/C subclass 1 defined device clocks and SYSREFs, and the FPGA requires an extra management clock. Additionally, the ADCs require a low noise clock of less than 100fs total RMS jitter. This leads to a total of 19 separate signals to generate, with frequencies listed below. For this example, the SYSREF frequencies for all devices are the same and should output four pulses upon a SYSREF request rising edge:

$f_{ADC-CLK}$ = 500MHz

$f_{DAC-CLK}$ = 4000MHz

$f_{FPGA-CLK}$ = 125MHz

$f_{FPGA-MGMT}$ = 100MHz

$f_{SYSREF}$ = 12.5MHz

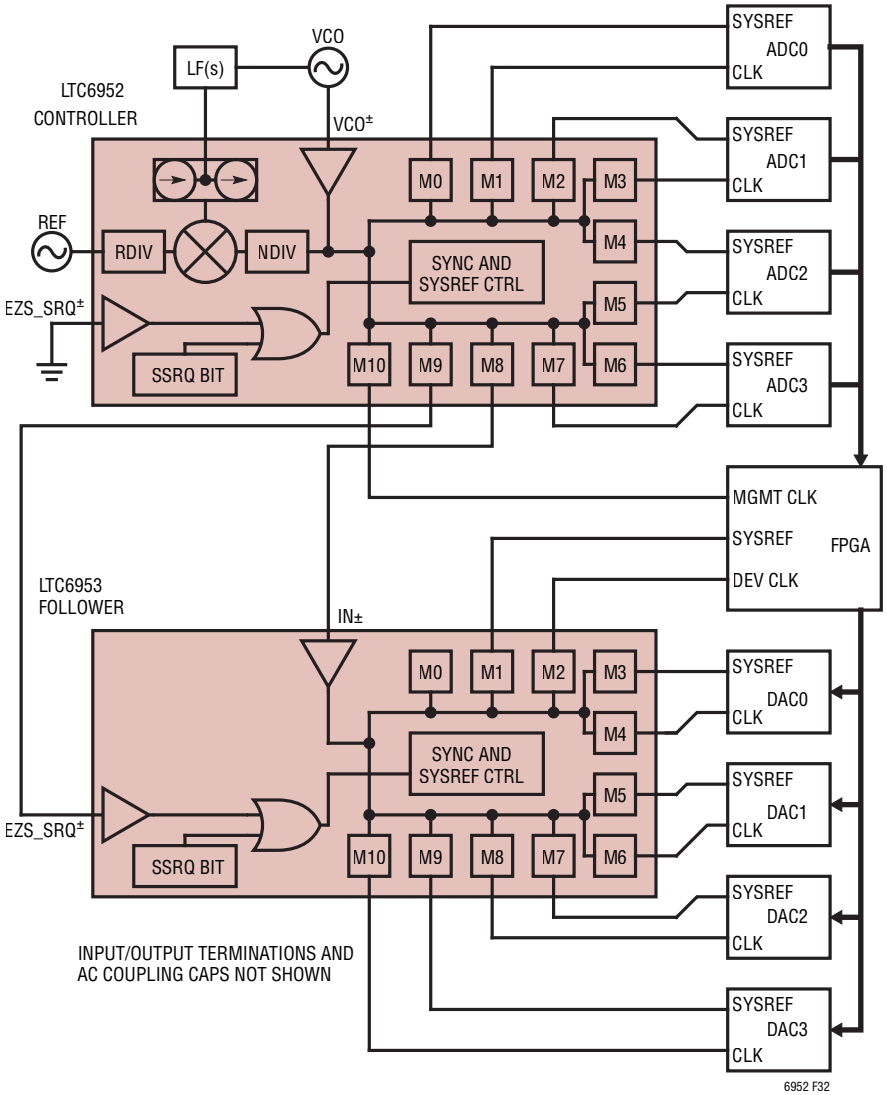



**Figure 32. Block Diagram for JESD204B/C EZSync Multi-Chip Design Example**

## APPLICATIONS INFORMATION

To determine which multi-chip configuration to use, we utilize the flowchart in Figure 15. This example has nine total JESD204B/C device clock/SYSREF pairs, four of which need to be less than 100fs total jitter. We also need one additional non-low noise standalone clock for the FPGA. Therefore:

TP = 9

LNP = 4

TS = 1

LNS = 0

Based on these inputs, Figure 15 suggests using the EZSync Multi-Chip protocol with request passthrough topology shown in Figure 12, using one CONTROLLER and one FOLLOWER chip. Noting that the use of LTC6953 for any FOLLOWER chips is recommended, Figure 32 shows a block diagram of the full system. OUT8 of the CONTROLLER LTC6952 is driving the IN± inputs of the FOLLOWER LTC6953. This output is referred to as the "follower-driver" output. OUT9 of the CONTROLLER is driving the EZS_SRQ± pins of the FOLLOWER, and is therefore the SYNC/SRQ passthrough output. Also notice that the CONTROLLER clock outputs are the lowest jitter clocks, and should therefore be used to drive the ADCs.

### Reference and VCO Assumptions

For this example, assume the available reference is a 100MHz sine wave oscillator with 8dBm output power, and the VCO is a 4000MHz oscillator with a $K_{VCO}$ of 5MHz/V, output power of 7dBm, and phase noise of –115dBc/Hz at 10kHz.

$f_{REF}$ = 100MHz

$f_{VCO}$ = 4000MHz

$K_{VCO}$ = 5MHz/V

### Design Procedure

Designing and enabling this clock generation solution consists of the following steps:

1. Determine CONTROLLER R and N divider values
2. Determine the optimum loop bandwidth
3. Select loop filter component values
4. Determine all output modes for CONTROLLER and FOLLOWER
5. Determine all M divider values
6. Determine all digital delay values
7. Program the ICs with the correct divider values, output delays, and other settings
8. Synchronize the outputs
9. Place the SYSREF outputs in a lower power mode until the next SYSREF request (optional, see Operations section)
10. Place ICs into SYSREF request mode (SRQMD=1) and send a SYSREF request when needed
11. Return IC into SYNC mode (SRQMD=0) and place the SYSREF outputs into a lower power mode for power savings (optional).

*Note that synchronization MUST be performed before a SYSREF request. The synchronization must be repeated only if the divider setting is changed, or if the divider is powered down.*

### Determining CONTROLLER R and N Divider Values

Following the "Loop Filter Design" algorithm, first determine all the divider values. From the Electrical Characteristics, the maximum $f_{PFD}$ is 167MHz, which is larger than the $f_{REF}$ of 100MHz. Therefore R should be 1 noting that maximizing $f_{PFD}$ in a data converter application will minimize integrated jitter. Use Equation 4 to determine $f_{PFD}$ and Equation 6 to determine N:

R = 1

$f_{PFD} = f_{REF}/R$ = 100MHz

$N = f_{VCO}/f_{PFD}$ = 40

## APPLICATIONS INFORMATION

### Selecting Loop Bandwidth

The next step in the algorithm is choosing the open loop bandwidth. The maximum BW should be at least 10x smaller than $f_{PFD}$. Most data converter applications will place the bandwidth at the optimal intersection of VCO noise and in-band noise. Narrower bandwidths or higher order loop filters can be used to lower spurious power. For this example, the third-order loop filter shown in Figure 24 will be used.

After inputting the external VCO phase noise and $K_{VCO}$ characteristics, the LTC6952Wizard reports the thermal noise optimized loop bandwidth is approximately 16kHz.

### Loop Filter Component Selection

For the lowest in-band phase noise, $I_{CP}$ should be set to the highest value possible that results in practical loop filter component values. Therefore, $I_{CP}$ for our example is chosen to be 11.2mA. LTC6952Wizard uses Equation 9 to determine $R_Z$:

$$R_Z = 2 \bullet \pi \bullet 16k \bullet 40/(11.2m \bullet 5M)$$

$$R_Z = 71.8\Omega \approx 71.5\Omega$$

The LTC6952Wizard uses Equation 10 through 13 to calculate $C_I$, $C_P$, $C_2$, and $R_1$:

$$C_I = 4/(\pi \bullet 16k \bullet 71.5) = 1.11\mu F \approx 1.2\mu F$$

$$C_P = 1/(12 \bullet \pi \bullet 16k \bullet 71.5) = 23.2nF \approx 22nF$$

$$C_2 = 1/(18 \bullet \pi \bullet 16k \bullet 71.5) = 15.5nF \approx 15nF$$

$$R_1 = 71.5\Omega$$

Note that resistors are rounded to standard ±1% values, and capacitors are rounded to standard ±10% values.

### Determining Output Modes

All outputs can be programmed as clocks (MODEx=0), SYSREFs (MODEx = 1 or 3), or SYNC/SRQ passthrough outputs (MODEx = 2) using each output's individual MODEx bits as described in Table 13, Table 14, and Table 15. Any output can also be programmed to ignore SYNC and SYSREF requests by setting that output's corresponding SRQENx bit to "0". Noting that this design example calls for pulsed SYSREFs (MODEx = 3) and that the FPGA management clock should always be free running (CONTROLLER SRQEN10 = 0), Table 25 summarizes each output's mode settings.

**Table 25. Output Mode Settings for EZSync Multi-Chip Design Example**

| IC | OUTPUT | PURPOSE | SRQENx | MODEx | PDx |
|---|---|---|---|---|---|
| CONTROLLER | OUT0 | ADC0 SYSREF | 1 | 3 | 0 |
| | OUT1 | ADC0 CLK | 1 | 0 | 0 |
| | OUT2 | ADC1 SYSREF | 1 | 3 | 0 |
| | OUT3 | ADC1 CLK | 1 | 0 | 0 |
| | OUT4 | ADC2 SYSREF | 1 | 3 | 0 |
| | OUT5 | ADC2 CLK | 1 | 0 | 0 |
| | OUT6 | ADC3 SYSREF | 1 | 3 | 0 |
| | OUT7 | ADC3 CLK | 1 | 0 | 0 |
| | OUT8 | FOLLOWER VCO | 1 | 0 | 0 |
| | OUT9 | FOLLOWER EZS_SRQ | 1 | 2 | 0 |
| | OUT10 | FPGA MGMT CLK | 0 | 0 | 0 |
| FOLLOWER | OUT0 | Unused | 0 | 0 | 3 |
| | OUT1 | FPGA SYSREF | 1 | 3 | 0 |
| | OUT2 | FPGA DEV CLK | 1 | 0 | 0 |
| | OUT3 | DAC0 SYSREF | 1 | 3 | 0 |
| | OUT4 | DAC0 CLK | 1 | 0 | 0 |
| | OUT5 | DAC1 SYSREF | 1 | 3 | 0 |
| | OUT6 | DAC1 CLK | 1 | 0 | 0 |
| | OUT7 | DAC2 SYSREF | 1 | 3 | 0 |
| | OUT8 | DAC2 CLK | 1 | 0 | 0 |
| | OUT9 | DAC3 SYSREF | 1 | 3 | 0 |
| | OUT10 | DAC3 CLK | 1 | 0 | 0 |

### Determining Output Divider Values

Once the desired frequencies of the outputs are determined, the output divider values can be calculated. The ADC, DAC, and FPGA clock frequencies are already known, which leaves the two CONTROLLER outputs that drive the FOLLOWER to be described. Since OUT8 of the CONTROLLER drives the FOLLOWER VCO input, its frequency must be equal to, or larger than, the highest FOLLOWER frequency. Therefore:

$$f_{CONT\text{-}OUT8} = 4000MHz$$

Additionally, when using the software-controlled EZSync configuration in a JESD204B/C application, the CONTROLLER output which drives the FOLLOWER's EZS_SRQ inputs should be set to the same frequency as the SYSREF frequency (or the slowest SYSREF frequency if multiple SYSREF periods are used).

$$f_{CONT\text{-}OUT9} = 12.5MHz$$

## APPLICATIONS INFORMATION

Now that all frequencies are known, use Equation 7 to determine the output divider values. The results are shown in Table 26.

**Table 26. Output Divide Settings for EZSync Multi-Chip Design Example**

| IC | OUTPUT | PURPOSE | Frequency (MHz) | Divide Value (Mx) |
|---|---|---|---|---|
| CONTROLLER | OUT0 | ADC0 SYSREF | 12.5 | 320 |
| | OUT1 | ADC0 CLK | 500 | 8 |
| | OUT2 | ADC1 SYSREF | 12.5 | 320 |
| | OUT3 | ADC1 CLK | 500 | 8 |
| | OUT4 | ADC2 SYSREF | 12.5 | 320 |
| | OUT5 | ADC2 CLK | 500 | 8 |
| | OUT6 | ADC3 SYSREF | 12.5 | 320 |
| | OUT7 | ADC3 CLK | 500 | 8 |
| | OUT8 | FOLLOWER VCO | 4000 | 1 |
| | OUT9 | FOLLOWER EZS_SRQ | 12.5 | 320 |
| | OUT10 | FPGA MGMT CLK | 100 | 40 |
| FOLLOWER | OUT0 | Unused | N/A | N/A |
| | OUT1 | FPGA SYSREF | 12.5 | 320 |
| | OUT2 | FPGA DEV CLK | 125 | 32 |
| | OUT3 | DAC0 SYSREF | 12.5 | 320 |
| | OUT4 | DAC0 CLK | 4000 | 1 |
| | OUT5 | DAC1 SYSREF | 12.5 | 320 |
| | OUT6 | DAC1 CLK | 4000 | 1 |
| | OUT7 | DAC2 SYSREF | 12.5 | 320 |
| | OUT8 | DAC2 CLK | 4000 | 1 |
| | OUT9 | DAC3 SYSREF | 12.5 | 320 |
| | OUT10 | DAC3 CLK | 4000 | 1 |

### Determining Output Digital Delay Values

The output digital delay is used to control phase relationships between outputs. The minimum delay step is ½ of a period of the incoming VCO signal. For this design example, the digital delay is used to place each device's SYSREF signal edges into a known phase relationship to its corresponding device clock, optimized for the set-up ($t_S$) and hold time ($t_H$) requirements for that device. Assume that the optimum SYSREF edge location for each device occurs on the first falling clock edge before the desired SYSREF valid rising clock edge. In other words, SYSREF should change states ½ of a corresponding device clock period before the SYSREF valid device clock edge. Refer to Figure 29 for a visual example.

For EZSync multi-chip synchronization, the CONTROLLER output which drives the FOLLOWER VCO input (follower-driver) must output seven pulses before the FOLLOWER outputs begin. This means that any CONTROLLER outputs which should be aligned with the FOLLOWER outputs (follower-synchronous) must be delayed by the same amount of time as the seven pulses, leading to a delay offset for each of these follower-synchronous outputs ($DDEL_{FS-OS}$):

$$DDEL_{FS-OS} = 14 \bullet M_{FD} + DDEL_{FD} \quad (17)$$

where $M_{FD}$ and $DDEL_{FD}$ are the divider value and digital delay value, respectively, of the follower-driver. In most applications, $DDEL_{FD}$ will be set to 0.

In order to calculate each output's delay value for this design example, use the following procedure:

1. Delay all of the JESD204B/C device clocks by half of a period of the slowest JESD204B/C device clock. This delay setting is equivalent to the divide value of the slowest device clock since a one code digital delay equals half of a VCO cycle. Non-JESD204B/C clocks (such as the FPGA management clock) are not included in this calculation. This delay value defines the desired SYSREF valid clock edge. In this example, the slowest JESD204B/C clock is the FPGA device clock:

   $DDEL_{SYSvalid} = M_{FPGA-CLK} = 32$

   $DDEL_{ADC-CLK´} = DDEL_{SYSvalid} = 32$

   $DDEL_{DAC-CLK´} = DDEL_{SYSvalid} = 32$

   $DDEL_{FPGA-CLK´} = DDEL_{SYSvalid} = 32$

2. For each device clock/SYSREF pair, subtract half a device clock period from DDEL to find the SYSREF delay. This is equivalent to subtracting the corresponding divide value of the device clock, i.e.

   $DDEL_{ADC-SYS´} = DDEL_{SYSvalid} – M_{ADC-CLK}$

   $DDEL_{ADC-SYS´} = 32 – 8 = 24$

   $DDEL_{DAC-SYS´} = DDEL_{SYSvalid} – M_{DAC-CLK}$

   $DDEL_{DAC-SYS´} = 32 – 1 = 31$

   $DDEL_{FPGA-SYS´} = DDEL_{SYSvalid} – M_{FPGA-CLK}$

   $DDEL_{FPGA-SYS´} = 32 – 32 = 0$

## APPLICATIONS INFORMATION

3. Adjust for CONTROLLER vs FOLLOWER outputs. Any CONTROLLER output that is synchronous with the FOLLOWER must have the delay offset from Equation 17 added to its DDEL value. FOLLOWER outputs need no adjustment. For this example, the ADC CLKs and SYSREFs come from the CONTROLLER:

$DDEL_{ADC\text{-}CLK} = DDEL_{ADC\text{-}CLK}' + DDEL_{FS\text{-}OS}$

$DDEL_{ADC\text{-}CLK} = 32 + 14 = 46$

$DDEL_{ADC\text{-}SYS} = DDEL_{ADC\text{-}SYS}' + DDEL_{FS\text{-}OS}$

$DDEL_{ADC\text{-}SYS} = 24 + 14 = 38$

$DDEL_{DAC\text{-}CLK} = DDEL_{DAC\text{-}CLK}' + 0 = 32$

$DDEL_{DAC\text{-}SYS} = DDEL_{DAC\text{-}SYS}' + 0 = 31$

$DDEL_{FPGA\text{-}CLK} = DDEL_{FPGA\text{-}CLK}' + 0 = 32$

$DDEL_{FPGA\text{-}SYS} = DDEL_{FPGA\text{-}SYS}' + 0 = 0$

Even though CONTROLLER OUT9 is only passing through the SYNC/SRQ pulse, its digital delay should be set so that its edges occur at the same time as the latest occurring SYSREF in the overall system. This is to ensure that future SYSREF requests are aligned properly. Note that the delay offset from Equation 17 will need to be added as well since it is located on the CONTROLLER part. The latest occurring SYSREF is the DAC SYSREF, therefore:

$DDEL_{OUT9} = DDEL_{DAC\text{-}SYS} + DDEL_{FS\text{-}OS}$

$DDEL_{OUT9} = 31 + 14 = 45$

The follower-driver digital delay is set to 0 in this example:

$DDEL_{FD} = DDEL_{OUT8} = 0$

Table 27 summarizes the DDEL settings for all outputs.

**Table 27. Output DDELx Settings for EZSync Multi-Chip Design Example**

| IC | OUTPUT | PURPOSE | DDELx |
|---|---|---|---|
| CONTROLLER | OUT0 | ADC0 SYSREF | 38 |
| | OUT1 | ADC0 CLK | 46 |
| | OUT2 | ADC1 SYSREF | 38 |
| | OUT3 | ADC1 CLK | 46 |
| | OUT4 | ADC2 SYSREF | 38 |
| | OUT5 | ADC2 CLK | 46 |
| | OUT6 | ADC3 SYSREF | 38 |
| | OUT7 | ADC3 CLK | 46 |
| | OUT8 | FOLLOWER VCO | 0 |
| | OUT9 | FOLLOWER EZS_SRQ | 45 |
| | OUT10 | FPGA MGMT CLK | 0 |
| FOLLOWER | OUT0 | Unused | N/A |
| | OUT1 | FPGA SYSREF | 0 |
| | OUT2 | FPGA DEV CLK | 32 |
| | OUT3 | DAC0 SYSREF | 31 |
| | OUT4 | DAC0 CLK | 32 |
| | OUT5 | DAC1 SYSREF | 31 |
| | OUT6 | DAC1 CLK | 32 |
| | OUT7 | DAC2 SYSREF | 31 |
| | OUT8 | DAC2 CLK | 32 |
| | OUT9 | DAC3 SYSREF | 31 |
| | OUT10 | DAC3 CLK | 32 |

Now that the output divider and delays have been determined, the ICs can be programmed.

### Status Register Programming

This example will use the STAT pin to alert the system whenever the LTC6952 generates a fault condition. For the CONTROLLER, program x[5], x[3], x[1] = 1 to force the STAT pin high whenever any of the $\overline{LOCK}$, $\overline{VCOOK}$, or $\overline{REFOK}$ flags asserts:

CONTROLLER Reg01 = h2A

For the FOLLOWER, only the $\overline{VCOOK}$ flag is valid:

FOLLOWER Reg01 = h08

## APPLICATIONS INFORMATION

### Power and FILT Register Programming

For correct PLL operation on the CONTROLLER, all internal blocks should be enabled. Additionally, the REF and VCO input signals have a sufficient slew rate and power to not need the FILT or BST bits:

CONTROLLER Reg02 = h00

The FOLLOWER doesn't need the FILTV bit either:

FOLLOWER Reg02 = h00

### Output Power-Down Programming

During initial setup and synchronization, all used outputs should be set to full power. These bits will be used later to place the ICs in a lower power mode while waiting for SYSREF requests:

CONTROLLER Reg03 = h00

CONTROLLER Reg04 = h00

CONTROLLER Reg05 = h00

FOLLOWER Reg03 = h03

FOLLOWER Reg04 = h00

FOLLOWER Reg05 = h00

### RAO and PARSYNC Programming

PARSYNC should be set to "0" for the CONTROLLER since this example is not a ParallelSync application. RAO should also be set to "0" unless a highly accurate reference input-to-clock output timing is required. Note that RAO for the LTC6953 FOLLOWER is not applicable.

### Lock Detect Programming (CONTROLLER Only)

Next, determine the lock indicator window from $f_{PFD}$. From Table 4 we see that LKWIN = 0 with a $t_{LWW}$ of 3ns. The LTC6952 will consider the loop "locked" as long as the phase coincidence at the PFD is within ±3ns, or 108° as calculated below.

phase = 360° • $t_{LWW}$ • $f_{PFD}$

= 360° • 3n • 100M

≈ 108°

Larger values of COUNTS lead to more accurate and stable lock indications at the expense of longer lock indication times. A COUNTS value of 2048 will work for this application. From Table 5, LKCT[1:0] = 3 for 2048 counts.

### R and N Divider Programming

The previously determined R and N divider values for the CONTROLLER are 1 and 40, respectively. R and N for the FOLLOWER are not applicable and can be set to 0. Using these values and the PARSYNC and lock detector values, registers 6 through 9 can be programmed for the CONTROLLER:

CONTROLLER Reg06 = h0C

CONTROLLER Reg07 = h01

CONTROLLER Reg08 = h00

CONTROLLER Reg09 = h28

Note that registers 6 through 9 for the LTC6953 FOLLOWER are not applicable.

### Charge Pump Function and Current Programming

Disable all the charge pump functions (CPMID, CPWIDE, CPRST, CPUP, CPDN, and CPINV), allowing the loop to lock. Using Table 6 with the previously selected $I_{CP}$ of 11.2mA gives CP[4:0] = h13. The FOLLOWER charge pump settings are not applicable.

### SYNC and SYSREF Global Modes Programming

Bit EZMD controls whether the IC is an EZSync standalone/controller ("0") or follower ("1"). Bit SRQMD determines if the part is in synchronization mode ("0") or SYSREF request mode ("1"). SYSCT programs the number of pulses for any output in pulsed SYSREF mode (# pulses = $2^{SYSCT}$, so SYSCT = 2 to achieve four pulses for this example). Combining this information with the previously determined charge pump controls, registers h0A and h0B can be programmed:

CONTROLLER Reg0A = h13

CONTROLLER Reg0B = h04

FOLLOWER Reg0A = N/A

FOLLOWER Reg0B = h14

## APPLICATIONS INFORMATION

Note that the SSRQ bits will remain "0" for now, but will be used later during the synchronization and SYSREF request procedures.

### Output Divider, Delay and Function Programming

Four registers for each output allow the outputs to be configured independently of each other. The first register controls the output divide ratio through two control words, MPx and MDx, as described in Equation 1.

The second register contains the control modes and the most significant bits of the digital delay control word. The third register contains the remainder of the digital delay control word, and the fourth register is the analog delay control.

Both the analog delay and the output invert (OINVx) bits can be used to correct PC board layout issues such as mismatched trace lengths and differential signal crossovers, respectively. *Note that the use of analog delay on clock signals will degrade jitter performance.* For this example, assume the PC board is laid out in an ideal manner and no output inversions or analog delays are needed. With this information, all of registers h0C through h37 for both CONTROLLER and FOLLOWER can be programmed to the values in Table 28 and Table 29, calculated using the information in Table 25, Table 26 (with Equation 1), and Table 27.

**Table 28. CONTROLLER Output Register Settings for EZSync Multi-Chip Design Example**

| ADDR | Value | ADDR | Value | ADDR | Value |
|---|---|---|---|---|---|
| **h0C** | h9C | **h1C** | h9C | **h2C** | h00 |
| **h0D** | hE0 | **h1D** | hE0 | **h2D** | h80 |
| **h0E** | h26 | **h1E** | h26 | **h2E** | h00 |
| **h0F** | h00 | **h1F** | h00 | **h2F** | h00 |
| **h10** | h38 | **h20** | h38 | **h30** | h9C |
| **h11** | h80 | **h21** | h80 | **h31** | hC0 |
| **h12** | h2E | **h22** | h2E | **h32** | h2D |
| **h13** | h00 | **h23** | h00 | **h33** | h00 |
| **h14** | h9C | **h24** | h9C | **h34** | h99 |
| **h15** | hE0 | **h25** | hE0 | **h35** | h00 |
| **h16** | h26 | **h26** | h26 | **h36** | h00 |
| **h17** | h00 | **h27** | h00 | **h37** | h00 |
| **h18** | h38 | **h28** | h38 | | |
| **h19** | h80 | **h29** | h80 | | |
| **h1A** | h2E | **h2A** | h2E | | |
| **h1B** | h00 | **h2B** | h00 | | |

**Table 29. FOLLOWER Output Register Settings for EZSync Multi-Chip Design Example**

| ADDR | Value | ADDR | Value | ADDR | Value |
|---|---|---|---|---|---|
| **h0C** | h00 | **h1C** | h00 | **h2C** | h00 |
| **h0D** | h00 | **h1D** | h80 | **h2D** | h80 |
| **h0E** | h00 | **h1E** | h20 | **h2E** | h20 |
| **h0F** | h00 | **h1F** | h00 | **h2F** | h00 |
| **h10** | h9C | **h20** | h9C | **h30** | h9C |
| **h11** | hE0 | **h21** | hE0 | **h31** | hE0 |
| **h12** | h00 | **h22** | h1F | **h32** | h1F |
| **h13** | h00 | **h23** | h00 | **h33** | h00 |
| **h14** | hF8 | **h24** | h00 | **h34** | h00 |
| **h15** | h80 | **h25** | h80 | **h35** | h80 |
| **h16** | h20 | **h26** | h20 | **h36** | h20 |
| **h17** | h00 | **h27** | h00 | **h37** | h00 |
| **h18** | h9C | **h28** | h9C | | |
| **h19** | hE0 | **h29** | hE0 | | |
| **h1A** | h1F | **h2A** | h1F | | |
| **h1B** | h00 | **h2B** | h00 | | |

### Synchronization

The outputs in this example are now running at the desired frequency, but have random phase relationships with each other. Synchronization forces the outputs to run at known and repeatable phases and can be achieved in this example either externally, by driving the CONTROLLER's EZS_SRQ± pins, or internally, with the CONTROLLER's SSRQ bit in Reg0B. Since the part was just programmed, set the SSRQ bit to "1" and hold the EZS_SRQ± pins low:

CONTROLLER Reg0B = h05

After waiting a minimum of 1ms, set SSRQ back to "0":

CONTROLLER Reg0B = h04

Once the internal synchronization process completes, the outputs will be aligned as shown in Figure 33. Note that the internal divider behavior for the muted SYSREF outputs is shown as well as the actual outputs to demonstrate the phase alignment following synchronization. Also notice that all FOLLOWER outputs will have additional delay from the CONTROLLER outputs equal to the FOLLOWER's $t_{PD}$ as described in the Electrical Characteristics.

# APPLICATIONS INFORMATION

### Putting the ICs into a Lower Power Mode (Optional)

If desired, both ICs can be placed into lower power modes while awaiting a SYSREF request. This is achieved by setting PDx = 2 for all SYSREF-defined outputs. This powers down the output driver circuitry but leaves the internal divider running and in the correct phase relationship to the clocks.

### Performing a SYSREF Request

To produce SYSREF pulses, write a "1" to SRQMD and take the parts out of low power mode (if used) by writing all the SYSREF output PDx bits to "0". Wait 50µs to allow circuitry to power up. Send the SYSREF request by writing a "1" to the CONTROLLER SSRQ bit in Reg0B:

CONTROLLER Reg0B = h05

After waiting a minimum of 1ms, write Reg0B again:

CONTROLLER Reg0B = h04

Place the ICs back into low power mode if desired by writing a "0" to SRQMD and setting PDx = 2 for all SYSREF-defined outputs. After the rising edge of the SYSREF request, the SYSREF outputs will pulse four times and then return to a "0" state as shown in Figure 34. Note that the FOLLOWER SYSREF pulses may not start and stop at exactly the same time as the CONTROLLER's. This is not an issue, since the SYSREF edges will still be aligned correctly.

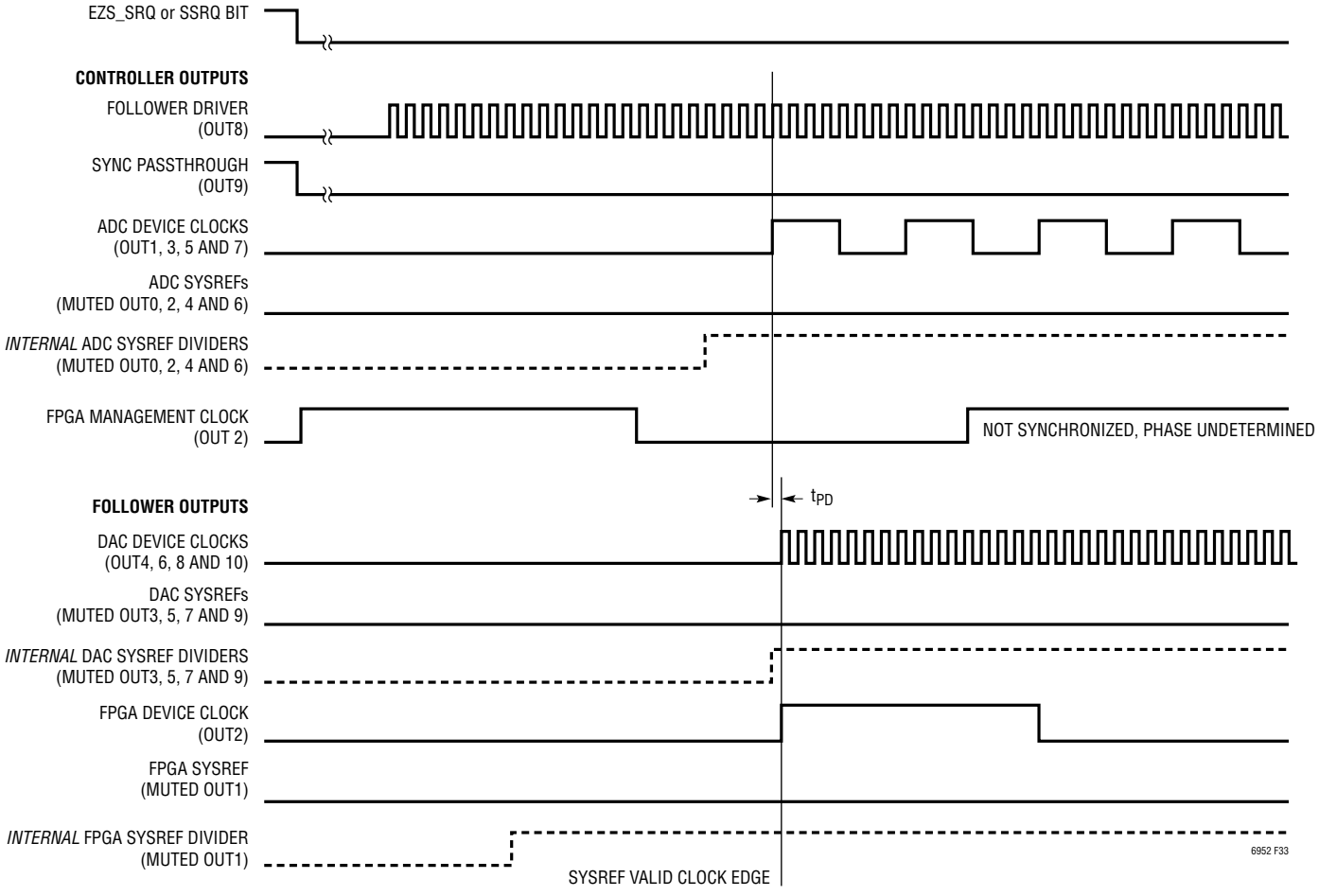



**Figure 33. Outputs after Synchronization for the EZSync Multi-Chip Design Example (SRQMD=0)**

## APPLICATIONS INFORMATION

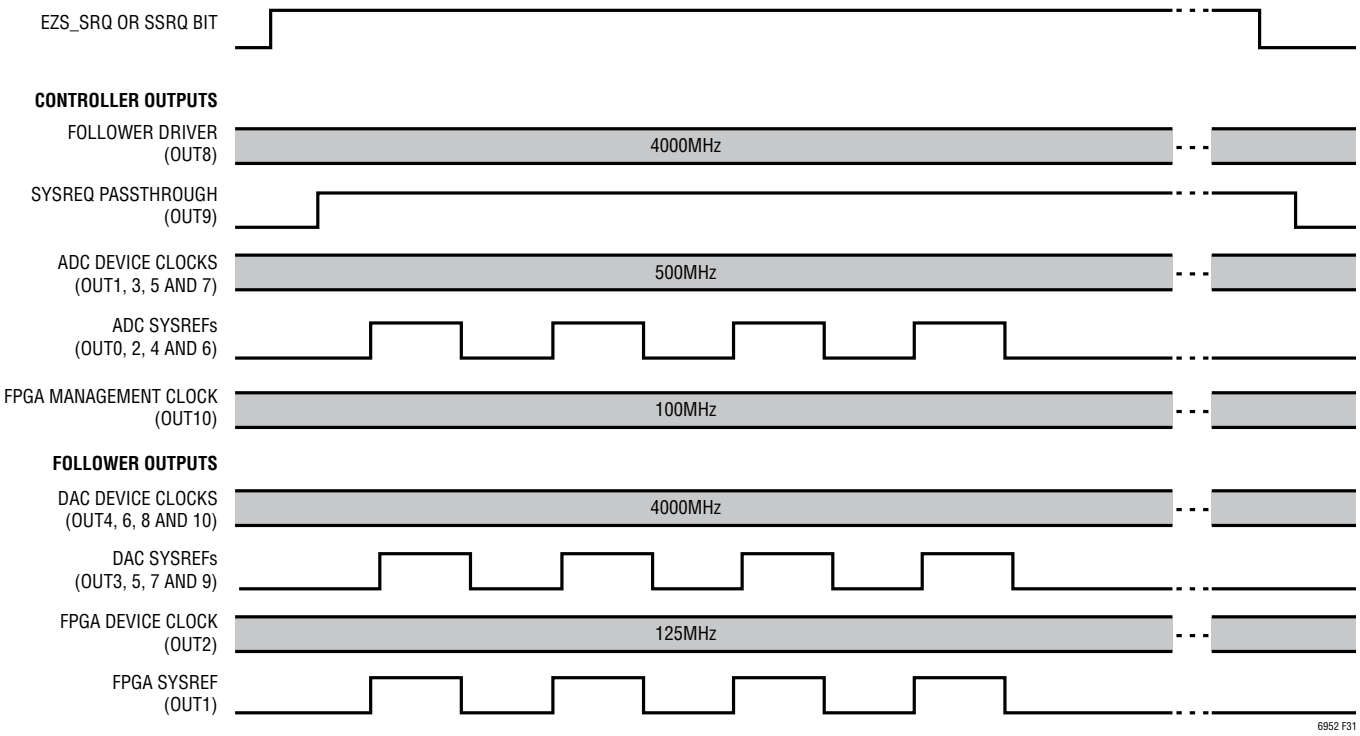



**Figure 34. Outputs after SYSREF Request for the EZSync Multi-Chip Design Example (SRQMD=1)**

### JESD204B/C DESIGN EXAMPLE USING ParallelSync

This design example consists of a system of eight JESD204B/C analog-to-digital converters (ADCs) and a JESD204B/C compatible FPGA. All of the ADCs and the FPGA require JESD204B/C subclass 1 device clocks and SYSREFs, and the FPGA requires an extra management clock. Additionally, the ADCs require low noise clocks of less than 100fs total RMS jitter. This leads to a total of 19 separate signals to generate, with frequencies listed below. For this example, the SYSREF frequencies for all devices are the same and should output four pulses upon a SYSREF request rising edge:

$f_{ADC-CLK}$ = 294.912MHz

$f_{FPGA-CLK}$ = 147.456MHz

$f_{FPGA-MGMT}$ = 98.304MHz

$f_{SYSREF}$ = 9.216MHz

To determine which multi-chip configuration to use, we utilize the flowchart in Figure 15. This example has nine total JESD204B/C device clock/SYSREF pairs, eight of which need to be less than 100fs total jitter. We also need one additional non-low noise standalone clock for the FPGA. Therefore:

TP = 9

LNP = 8

TS = 1

LNS = 0

Based on these inputs, Figure 15 suggests using the ParallelSync Multi-Chip protocol with LTC6953 Reference Distribution topology shown in Figure 14, using one LTC6953 as the reference distribution chip (REF LTC6953) and two LTC6952s in parallel to generate the clocks (LTC6952 #1

## APPLICATIONS INFORMATION

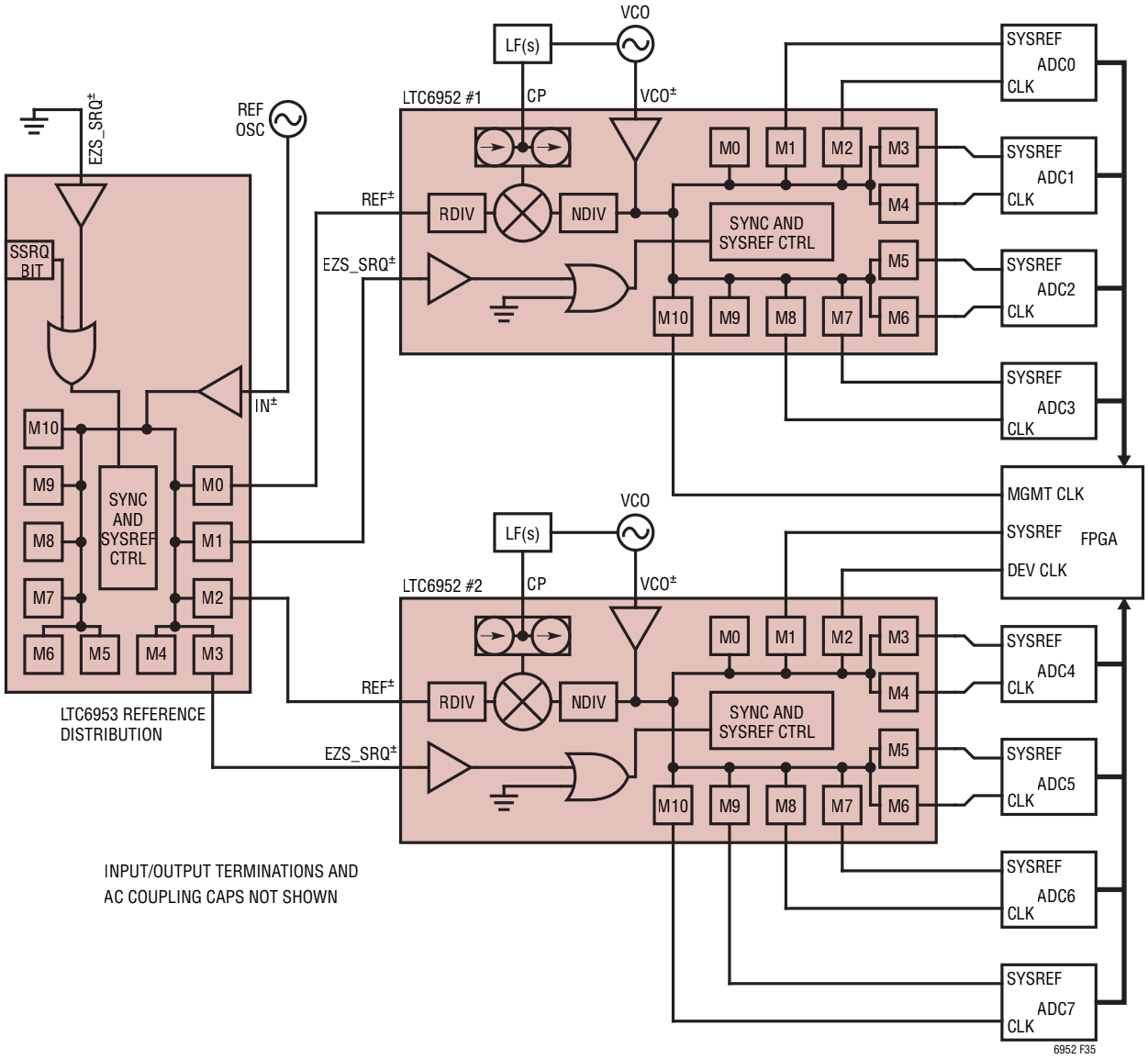



**Figure 35. Block Diagram for JESD204B/C ParallelSync with LTC6953 Reference Distribution Design Example**

and LTC6952 #2). Figure 35 shows a block diagram of the full system. Note that OUT0 of the reference LTC6953 is driving the REF± inputs of LTC6952 #1 and OUT1 is driving the EZS_SRQ± pins of LTC6952 #1. Likewise, OUT2 of the reference LTC6953 is driving the REF± inputs of LTC6952 #2 and OUT3 is driving the EZS_SRQ± pins of LTC6952 #2. All outputs in this configuration are low RMS jitter (~75fs ADC SNR Method).

### Reference and VCO Assumptions

For this example, assume the available reference is a 245.76MHz sine wave oscillator with 8dBm output power, and the VCOs are 2949.12MHz oscillators with a $K_{VCO}$ of 15MHz/V, output power of 6dBm, and phase noise of –112dBc/Hz at 10kHz.

$f_{REF}$ = 245.76MHz

$f_{VCO}$ = 2949.12MHz

$K_{VCO}$ = 15MHz/V

## APPLICATIONS INFORMATION

### Design Procedure

The design procedure for ParallelSync is similar to EZSync:

1. Determine R and N divider values for all ICs
2. Determine the optimum loop bandwidth(s)
3. Select loop filter component values
4. Determine output modes for all ICs
5. Determine all M divider values
6. Determine all digital delay values
7. Program the ICs with the correct divider values, output delays, and other settings
8. Synchronize the reference distribution part (if necessary)
9. Synchronize the parallel LTC6952 outputs
10. Place the SYSREF outputs in a lower power mode until the next SYSREF request (optional, see Operations section)
11. Place ICs into SYSREF request mode (SRQMD=1) and send a SYSREF request when needed
12. Return IC into SYNC mode (SRQMD=0) and place the SYSREF outputs into a lower power mode for power savings (optional).

*Note that synchronization MUST be performed before a SYSREF request. The synchronization must be repeated only if the divider setting is changed, or if the divider is powered down.*

### Determining R and N Divider Values

The reference LTC6953 does not have a PLL, so R and N do not apply. However, it can divide the incoming reference so that it falls below the 167MHz upper limit for the parallel parts' $f_{PFD}$. In this example, the output dividers of the reference clock outputs of the reference LTC6953 can be set to 2, leading to an effective reference frequency of 122.88MHz for the parallel parts. Following the "Loop Filter Design" algorithm for the two parallel LTC6952s, R should be 1 noting that maximizing $f_{PFD}$ in a data converter application will minimize integrated jitter. Use Equation 4 to determine $f_{PFD}$ and Equation 6 to determine N:

$R = 1$

$f_{PFD} = f_{REF}/R = 122.88MHz$

$N = f_{VCO}/f_{PFD} = 24$

### Selecting Loop Bandwidth

The next step in the algorithm is choosing the open loop bandwidth for the parallel LTC6952s. The maximum BW should be at least 10x smaller than $f_{PFD}$. Most data converter applications will place the bandwidth at the optimal intersection of VCO noise and in-band noise. Narrower bandwidths or higher order loop filters can be used to lower spurious power. For this example, the third-order loop filter shown in Figure 24 will be used.

After inputting the external VCO phase noise and $K_{VCO}$ characteristics, the LTC6952Wizard reports the thermal noise optimized loop bandwidth is approximately 38kHz.

### Loop Filter Component Selection

For the lowest in-band phase noise, $I_{CP}$ should be set to the highest value possible that results in practical loop filter component values. Therefore, $I_{CP}$ for our example is chosen to be 11.2mA. LTC6952Wizard uses Equation 9 to determine $R_Z$:

$R_Z = 2 \bullet \pi \bullet 38k \bullet 24/(11.2m \bullet 15M)$

$R_Z = 34.1\ \Omega \approx 34$

The LTC6952Wizard uses Equation 10 through 13 to calculate $C_I$, $C_P$, $C_2$, and $R_1$:

$C_I = 4/(\pi \bullet 38k \bullet 34) = 985nF \approx 1.0\mu F$

$C_P = 1/(12 \bullet \pi \bullet 38k \bullet 34) = 20.5nF \approx 22nF$

$C_2 = 1/(18 \bullet \pi \bullet 38k \bullet 34) = 13.7nF \approx 15nF$

$R_1 = 34\Omega$

Note that resistors are rounded to standard ±1% values, and capacitors are rounded to standard ±10% values.

# APPLICATIONS INFORMATION

## Determining Output Modes

All outputs can be programmed as clocks (MODEx = 0), SYSREFs (MODEx = 1 or 3), or SYNC/SRQ passthrough outputs (MODEx = 2) using each output's individual MODEx bits as described in Table 13, Table 14, and Table 15. Any output can also be programmed to ignore SYNC and SYSREF requests by setting that output's SRQENx bit to "0". Noting that this design example calls for pulsed SYSREFs (MODEx = 3) and that the FPGA management clock should always be free running (SRQENx = 0), Table 30 summarizes each output's mode settings.

**Table 30. Output Mode Settings for ParallelSync Example**

| IC | OUTPUT | PURPOSE | SRQENx | MODEx | PDx |
|---|---|---|---|---|---|
| REF LTC6953 | OUT0 | LTC6952 #1 REF | 1 | 0 | 0 |
| | OUT1 | LTC6952 #1 EZS-SRQ | 1 | 2 | 0 |
| | OUT2 | LTC6952 #2 REF | 1 | 0 | 0 |
| | OUT3 | LTC6952 #2 EZS-SRQ | 1 | 2 | 0 |
| | OUT4-10 | Unused | 0 | 0 | 3 |
| LTC6952 #1 | OUT0 | Unused | 0 | 0 | 3 |
| | OUT1 | ADC0 SYSREF | 1 | 3 | 0 |
| | OUT2 | ADC0 CLK | 1 | 0 | 0 |
| | OUT3 | ADC1 SYSREF | 1 | 3 | 0 |
| | OUT4 | ADC1 CLK | 1 | 0 | 0 |
| | OUT5 | ADC2 SYSREF | 1 | 3 | 0 |
| | OUT6 | ADC2 CLK | 1 | 0 | 0 |
| | OUT7 | ADC3 SYSREF | 1 | 3 | 0 |
| | OUT8 | ADC3 CLK | 1 | 0 | 0 |
| | OUT9 | Unused | 0 | 0 | 3 |
| | OUT10 | FPGA MGMT CLK | 0 | 0 | 0 |
| LTC6952 #2 | OUT0 | Unused | 0 | 0 | 3 |
| | OUT1 | FPGA SYSREF | 1 | 3 | 0 |
| | OUT2 | FPGA DEV CLK | 1 | 0 | 0 |
| | OUT3 | ADC4 SYSREF | 1 | 3 | 0 |
| | OUT4 | ADC4 CLK | 1 | 0 | 0 |
| | OUT5 | ADC5 SYSREF | 1 | 3 | 0 |
| | OUT6 | ADC5 CLK | 1 | 0 | 0 |
| | OUT7 | ADC6 SYSREF | 1 | 3 | 0 |
| | OUT8 | ADC6 CLK | 1 | 0 | 0 |
| | OUT9 | ADC7 SYSREF | 1 | 3 | 0 |
| | OUT10 | ADC7 CLK | 1 | 0 | 0 |

## Determining Output Divider Values

Since the desired frequencies of each output are already known, the output divider values can be calculated using Equation 7. Note that the reference LTC6953 EZS_SRQ passthrough outputs' internal dividers should be set as shown in Table 18. The results are shown in Table 31.

**Table 31. Output Divide Settings for ParallelSync Example**

| IC | OUTPUT | PURPOSE | Frequency (MHz) | Divide Value (Mx) |
|---|---|---|---|---|
| REF LTC6953 | OUT0 | LTC6952 #1 REF | 122.88 | 2 |
| | OUT1 | LTC6952 #1 EZS-SRQ | 122.88 | 2 |
| | OUT2 | LTC6952 #2 REF | 122.88 | 2 |
| | OUT3 | LTC6952 #2 EZS-SRQ | 122.88 | 2 |
| | OUT4-10 | Unused | N/A | N/A |
| LTC6952 #1 | OUT0 | Unused | N/A | N/A |
| | OUT1 | ADC0 SYSREF | 9.216 | 320 |
| | OUT2 | ADC0 CLK | 294.912 | 10 |
| | OUT3 | ADC1 SYSREF | 9.216 | 320 |
| | OUT4 | ADC1 CLK | 294.912 | 10 |
| | OUT5 | ADC2 SYSREF | 9.216 | 320 |
| | OUT6 | ADC2 CLK | 294.912 | 10 |
| | OUT7 | ADC3 SYSREF | 9.216 | 320 |
| | OUT8 | ADC3 CLK | 294.912 | 10 |
| | OUT9 | Unused | N/A | N/A |
| | OUT10 | FPGA MGMT CLK | 98.304 | 30 |
| LTC6952 #2 | OUT0 | Unused | N/A | N/A |
| | OUT1 | FPGA SYSREF | 9.216 | 320 |
| | OUT2 | FPGA DEV CLK | 147.456 | 20 |
| | OUT3 | ADC4 SYSREF | 9.216 | 320 |
| | OUT4 | ADC4 CLK | 294.912 | 10 |
| | OUT5 | ADC5 SYSREF | 9.216 | 320 |
| | OUT6 | ADC5 CLK | 294.912 | 10 |
| | OUT7 | ADC6 SYSREF | 9.216 | 320 |
| | OUT8 | ADC6 CLK | 294.912 | 10 |
| | OUT9 | ADC7 SYSREF | 9.216 | 320 |
| | OUT10 | ADC7 CLK | 294.912 | 10 |

## APPLICATIONS INFORMATION

### Determining Output Digital Delay Values

In order to calculate each parallel LTC6952 output's delay value for this design example, use the following procedure:

1. Delay all of the JESD204B/C device clocks by half of a period of the *slowest* JESD204B/C device clock. This delay setting is equal to the divide value of the slowest device clock because a one code digital delay equals half of a VCO cycle. Non-JESD204B/C clocks (such as the FPGA management clock) are not included in this calculation. This delay value defines the desired SYSREF valid clock edge. In this example, the slowest JESD204B/C clock is the FPGA device clock:

   $DDEL_{SYSvalid} = M_{FPGACLK} = 20$

   $DDEL_{ADC\text{-}CLK}' = DDEL_{SYSvalid} = 20$

   $DDEL_{FPGA\text{-}CLK}' = DDEL_{SYSvalid} = 20$

2. For each device clock/SYSREF pair, subtract half a device clock period from DDEL to find the SYSREF delay. This is equivalent to subtracting the corresponding divide value of the device clock, i.e.

   $DDEL_{ADC\text{-}SYS}' = DDEL_{SYSvalid} - M_{ADC\text{-}CLK}$

   $DDEL_{ADC\text{-}SYS}' = 20 - 10 = 10$

   $DDEL_{FPGA\text{-}SYS}' = DDEL_{SYSvalid} - M_{FPGA\text{-}CLK}$

   $DDEL_{FPGA\text{-}SYS}' = 20 - 20 = 0$

3. Although not usually required, the outputs can be aligned to the incoming reference. To achieve this, additional digital delay is added to each parallel LTC6952 output to align it with the next reference edge. This delay can be calculated by Equation 18:

   $$DDEL_{REF\text{-}OS} = 2 \bullet N \bullet CEILING(16/(2 \bullet N)) - 16 \quad (18)$$

where CEILING() means round up to the nearest integer. Remembering that the N value was previously calculated to be 24, the result of this equation is 32. Add 32 to all of the parallel LTC6952s DDEL values if reference alignment is desired. For this example, reference alignment is not necessary.

Even though OUT1 and OUT3 on the reference LTC6953 are only passing through the SYNC/SRQ pulse, their digital delays should be set so that its edges occur on the falling edge of their corresponding reference clocks. This is to ensure that the EZS_SRQ signals into the parallel ICs have a proper setup and hold time relationship to the reference clock. Since the reference LTC6953 has its reference clock output dividers set to 2 and DDEL set to 0, the passthrough output DDELs should be set to 2 based on Table 18.

All of the calculated digital delay values are shown in Table 32.

**Table 32. Output DDELx Settings for ParallelSync Example**

| IC | OUTPUT | PURPOSE | DDELx |
|---|---|---|---|
| REF LTC6953 | OUT0 | LTC6952 #1 REF | 0 |
| | OUT1 | LTC6952 #1 EZS-SRQ | 2 |
| | OUT2 | LTC6952 #2 REF | 0 |
| | OUT3 | LTC6952 #2 EZS-SRQ | 2 |
| | OUT4-10 | Unused | N/A |
| LTC6952 #1 | OUT0 | Unused | N/A |
| | OUT1 | ADC0 SYSREF | 10 |
| | OUT2 | ADC0 CLK | 20 |
| | OUT3 | ADC1 SYSREF | 10 |
| | OUT4 | ADC1 CLK | 20 |
| | OUT5 | ADC2 SYSREF | 10 |
| | OUT6 | ADC2 CLK | 20 |
| | OUT7 | ADC3 SYSREF | 10 |
| | OUT8 | ADC3 CLK | 20 |
| | OUT9 | Unused | N/A |
| | OUT10 | FPGA MGMT CLK | 0 |
| LTC6952 #2 | OUT0 | Unused | N/A |
| | OUT1 | FPGA SYSREF | 0 |
| | OUT2 | FPGA DEV CLK | 20 |
| | OUT3 | ADC4 SYSREF | 10 |
| | OUT4 | ADC4 CLK | 20 |
| | OUT5 | ADC5 SYSREF | 10 |
| | OUT6 | ADC5 CLK | 20 |
| | OUT7 | ADC6 SYSREF | 10 |
| | OUT8 | ADC6 CLK | 20 |
| | OUT9 | ADC7 SYSREF | 10 |
| | OUT10 | ADC7 CLK | 20 |

Now that the output divider and delays have been determined, the ICs can be programmed.

## APPLICATIONS INFORMATION

### Status Register Programming

This example will use the STAT pin to alert the system whenever the ICs generate a fault condition. For the two parallel LTC6952s, program x[5], x[3], x[1] = 1 to force the STAT pin high whenever any of the $\overline{\text{LOCK}}$, $\overline{\text{VCOOK}}$, or $\overline{\text{REFOK}}$ flags asserts. The REF LTC6953 only needs x[3] = 1 because it does not have a PLL:

REF LTC6953 Reg01 = h08

LTC6952 #1 Reg01 = h2A

LTC6952 #2 Reg01 = h2A

### Power and FILT Register Programming

For correct PLL operation on the three ICs, all internal blocks should be enabled. Additionally, the REF and VCO input signals have sufficient slew rate and power to not need the FILT or BST bits:

REF LTC6953 Reg02 = h00

LTC6952 #1 Reg02 = h00

LTC6952 #2 Reg02 = h00

### Output Power-Down Programming

During initial setup and synchronization, all used outputs and the SRQ circuitry should be set to full power. These bits will be used later to place the ICs in a lower power mode while waiting for SYSREF requests:

REF LTC6953 Reg03 = h00

REF LTC6953 Reg04 = hFF

REF LTC6953 Reg05 = h3F

LTC6952 #1 Reg03 = h03

LTC6952 #1 Reg04 = h00

LTC6952 #1 Reg05 = h0C

LTC6952 #2 Reg03 = h03

LTC6952 #2 Reg04 = h00

LTC6952 #2 Reg05 = h00

### RAO and PARSYNC Programming

PARSYNC should be set to "1" for the two parallel LTC6952s. Also, set RAO to "1" for the two parallel LTC6952s as this will minimize chip-to-chip skew. A slight degradation of in-band PLL noise will result from setting RAO to "1".

### Lock Detect Programming

Next, determine the lock indicator window from $f_{PFD}$. From Table 4 we see that LKWIN = 0 with a $t_{LWW}$ of 3ns. The LTC6952 will consider the loop "locked" as long as the phase coincidence at the PFD is within ±3ns, or 133° as calculated below.

phase = 360° • $t_{LWW}$ • $f_{PFD}$

= 360° • 3n • 122.88M

≈ 133°

Larger values of COUNTS lead to more accurate and stable lock indications at the expense of longer lock indication times. A COUNTS value of 2048 will work for this application. From Table 5, LKCT[1:0] = 3 for 2048 counts.

### R and N Divider Programming

For the reference LTC6953, the R and N values are irrelevant as there is no PLL. The previously determined R and N divider values for the two parallel LTC6952s are 1 and 24, respectively. Using these values and the RAO, PARSYNC, and lock detector values, registers 6 through 9 can be programmed for the parallel LTC6952s:

LTC6952 #1 Reg06 = hCC

LTC6952 #1 Reg07 = h01

LTC6952 #1 Reg08 = h00

LTC6952 #1 Reg09 = h18

LTC6952 #2 Reg06 = hCC

LTC6952 #2 Reg07 = h01

LTC6952 #2 Reg08 = h00

LTC6952 #2 Reg09 = h18

Note that registers 6 through 9 for the REF LTC6953 are not applicable.

## APPLICATIONS INFORMATION

### Charge Pump Function and Current Programming

For the parallel LTC6952s, disable all the charge pump functions (CPMID, CPWIDE, CPRST, CPUP, CPDN, and CPINV), allowing the loop to lock. Using Table 6 with the previously selected $I_{CP}$ of 11.2mA gives CP[4:0] = h13.

### SYNC and SYSREF Global Modes Programming

Bit EZMD controls whether the IC is an EZSync standalone/controller ("0") or follower ("1"). Since this example is a ParallelSync application, set bit EZMD to "0". Bit SRQMD determines if the part is in synchronization mode ("0") or SYSREF request mode ("1"). SYSCT programs the number of pulses for any output in pulsed SYSREF mode (# pulses = $2^{SYSCT}$, so SYSCT = 2 to achieve four pulses for this example). Combining this information with the previously determined charge pump controls, registers h0A and h0B can be programmed:

REF LTC6953 Reg0A = N/A

REF LTC6953 Reg0B = h04

LTC6952 #1 Reg0A = h13

LTC6952 #1 Reg0B = h04

LTC6952 #2 Reg0A = h13

LTC6952 #2 Reg0B = h04

Note that the SSRQ bits will remain "0".

### Output Divider, Delay and Function Programming

Four registers for each output allow the outputs to be configured independently of each other. The first register controls the output divide ratio through two control words, MPx and MDx, as described in Equation 1.

The second register contains the control modes and the most significant bits of the digital delay control word. The third register contains the remainder of the digital delay control word, and the fourth register is the analog delay.

Both the analog delay and the output invert (OINVx) bits can be used to correct PC board layout issues such as mismatched trace lengths and differential signal crossovers, respectively. *Note that the use of analog delay on clock signals will degrade jitter performance*. For this example, assume the PC board is laid out in an ideal manner and no output inversions or analog delays are needed. With this information all of registers h0C through h37 for all ICs can be programmed using the information in Table 30, Table 31 (with Equation 1), and Table 32.

### Reference Synchronization

In applications where the output divider values of the reference clock outputs are greater than one, it is necessary to synchronize the distributed reference signals before synchronizing the parallel LTC6952s. To do this, the SRQEN bits of the valid outputs on the reference distribution LTC6953 are set to "1", and SRQMD is set to "0". Set the SSRQ bit in Reg0B of the REF LTC6952 to perform the synchronization (or drive the EZS_SRQ input to the high state):

REF LTC6953 Reg0B = h05

After waiting a minimum of 1ms, write Reg0B again (or drive the EZS_SRQ input to the low state):

REF LTC6953 Reg0B = h04

*Note that synchronizing the REF LTC6953 will cause the parallel LTC6952s to lose lock temporarily. Allow sufficient lock time for the parallel LTC6952s before synchronizing them.*

At this point, the reference outputs are synchronized and must continue to run during the synchronization of the parallel parts. To ensure this, set the SRQEN bits of the two reference clock outputs to "0". Also set the SRQMD bit to "1" to allow passthrough retiming to the reference clocks.

REF LTC6953 Reg0D = h00

REF LTC6953 Reg15 = h00

REF LTC6953 Reg0B = h0C

# APPLICATIONS INFORMATION

### Parallel LTC6952 Synchronization

The outputs of the parallel LTC6952s in this example are now running at the desired frequency, but have random phase relationships with each other. Synchronization forces the outputs to run at known and repeatable phases and can be achieved in this example either externally, by driving the REF LTC6953's EZS_SRQ± pins, or internally, with the REF LTC6953's SSRQ bit in Reg0B. For this example, use the SSRQ bit and hold the EZS_SRQ± pins low:

REF LTC6953 Reg0B = h05

After waiting a minimum of 1ms, write Reg0B again:

REF LTC6953 Reg0B = h04

Once the internal synchronization process completes, the outputs will be aligned as shown in Figure 36. Note that the internal divider behavior for the muted SYSREF outputs is shown as well as the actual outputs to demonstrate the phase alignment following synchronization.

### Putting the ICs into a Low Power Mode (Optional)

If desired, the ICs can be placed into lower power modes while awaiting a SYSREF request. This is achieved by setting PDx = 2 for all SYSREF-defined outputs. This powers down the output driver circuitry but leaves the internal divider running and in the correct phase relationship to the clocks.

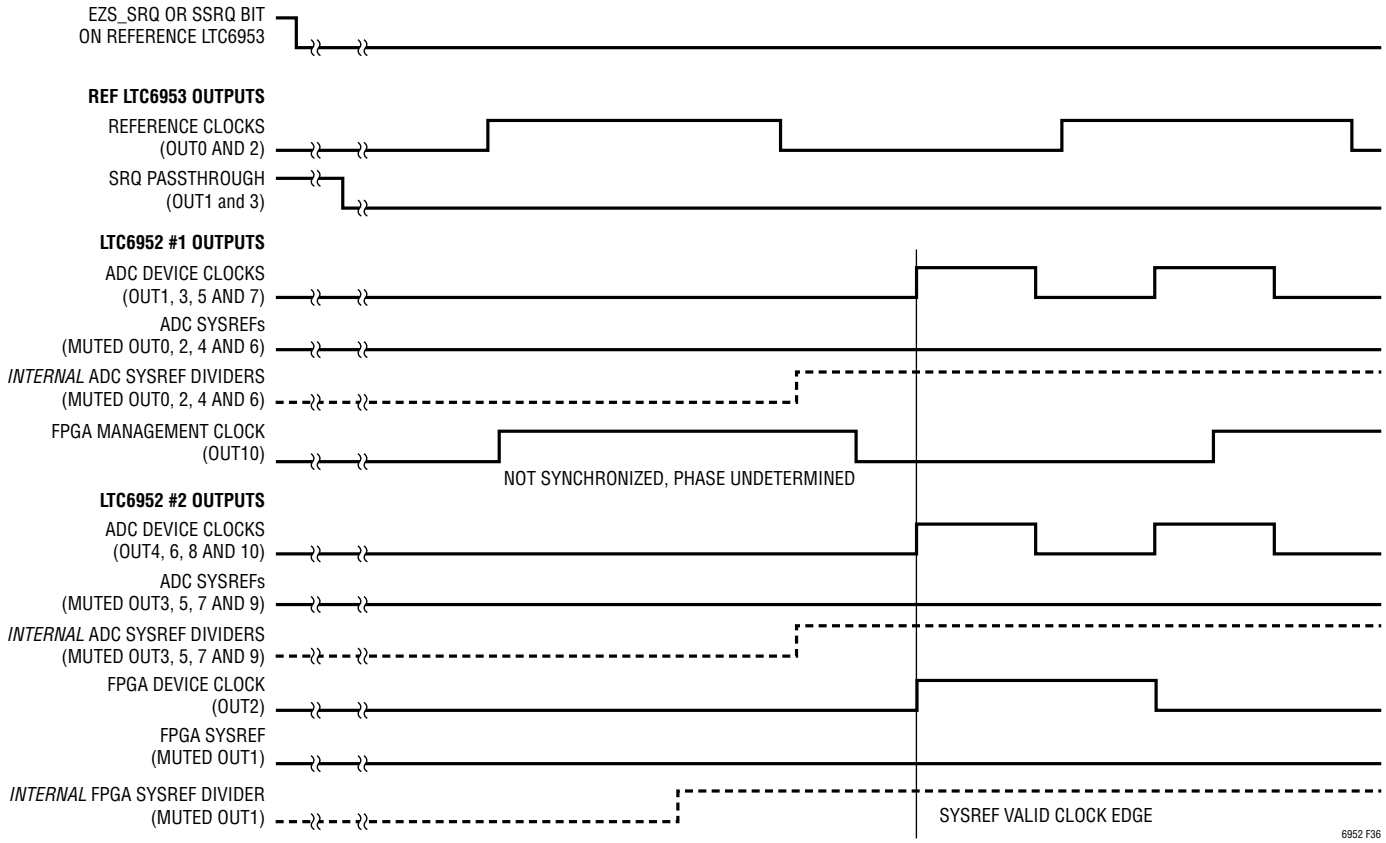



**Figure 36. Outputs after Synchronization for the ParallelSync Design Example (SRQMD=0)**

# APPLICATIONS INFORMATION

### Performing a SYSREF Request

To produce SYSREF pulses, write a “1” to SRQMD in both of the parallel LTC6952s and take the parts out of low power mode (if used) by writing all the SYSREF output PDx bits to “0”. Wait 50µs to allow the circuitry to power up. Send the SYSREF request by writing a “1” to the REF LTC6952’s SSRQ bit in Reg0B:

REF LTC6952 Reg0B = h05

After waiting a minimum of 1ms, write Reg0B again:

REF LTC6952 Reg0B = h04

If desired, the EZS_SRQ± pins of the reference LTC6952 can be used instead of the SSRQ bit to request a SYSREF.

After the rising edge of the SYSREF request, the SYSREF outputs will pulse four times and then return to a “0” state as shown in Figure 37. Place the ICs back into low power mode if desired by writing a “0” to SRQMD in the parallel LTC6952s and setting PDx = 2 for all SYSREF defined outputs.

### REFERENCE SOURCE CONSIDERATIONS

A high quality signal must be applied to the REF± inputs as they provide the frequency reference to the entire PLL. As mentioned previously, to achieve the part’s in-band phase noise performance, apply a sine wave of at least 6dBm into 50Ω, or a square wave of at least $0.5V_{P-P}$, with slew rate of at least 20V/µs.

The LTC6952 may be driven single-ended from CMOS levels (greater than $2.7V_{P-P}$). Apply the reference signal at $REF^-$, and bypass $REF^+$ to GND with a 47pF capacitor. The BST bit must also be set to “0”, according to guidelines given in Table 2. Setting FILT to a “0” is recommended as the input is a square wave.

The LTC6952 achieves an in-band normalized phase noise floor $L_{NORM}$ = –229dBc/Hz typical. To calculate its equivalent input phase noise floor $L_{IN}$, use Equation 19.

$$L_{IN} = L_{NORM} + 10 \bullet \log_{10}(f_{REF}) \quad (19)$$

For example, using a 10MHz reference frequency gives an input phase noise floor of –159dBc/Hz. The reference frequency source’s phase noise must be at least 3dB better than this to prevent limiting the overall system performance.

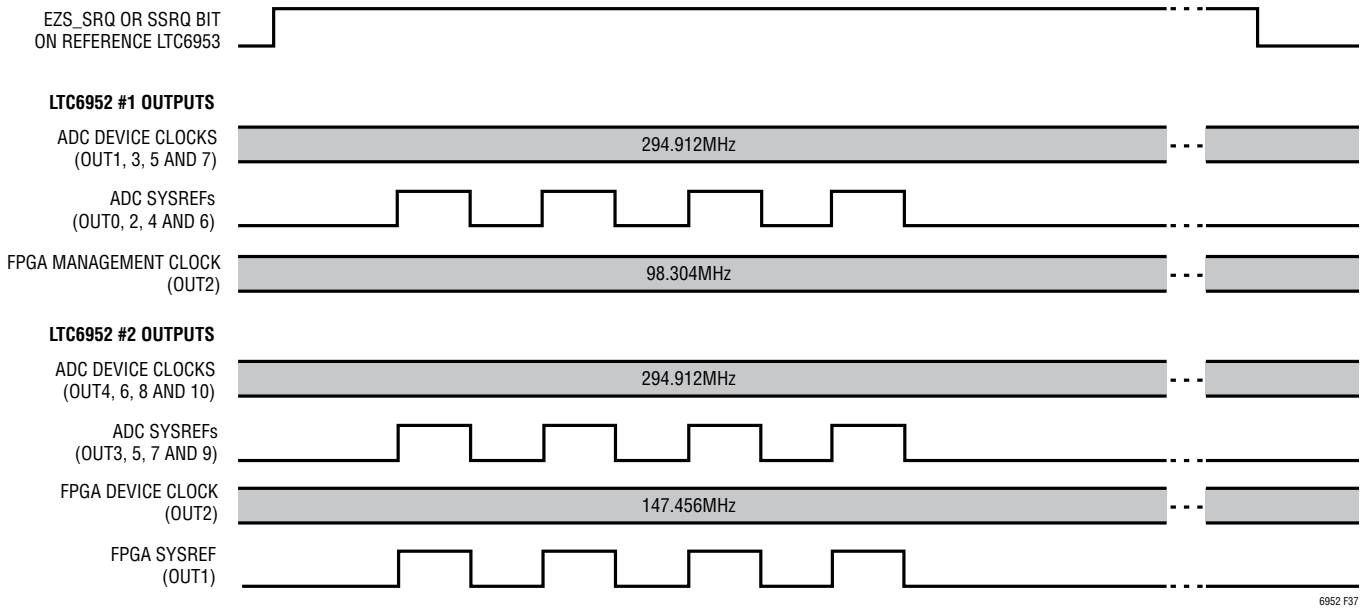



**Figure 37. Outputs after SYSREF Request for the ParallelSync Design Example (SRQMD=1)**

# APPLICATIONS INFORMATION

## IN-BAND OUTPUT PHASE NOISE

The in-band phase noise floor $L_{OUT}$ produced at $f_{OUTx}$ may be calculated by using Equation 20.

$$L_{OUT} = L_{NORM} + 10 \bullet \log_{10}(f_{PFD}) + 20 \cdot \log_{10}(f_{OUTx}/f_{PFD})$$

or

$$L_{OUT} = L_{NORM} + 10 \bullet \log_{10}(f_{PFD}) + 20 \bullet \log_{10}(N/Mx) \quad (20)$$

where $L_{NORM}$ is –229dBc/Hz.

As can be seen, for a given PFD frequency $f_{PFD}$, the output in-band phase noise increases at a 20 dB-per-decade rate with the N divider count. So, for a given output frequency $f_{OUTx}$, $f_{PFD}$ should be as large as possible (or N should be as small as possible) while still satisfying the application's frequency step size requirements.

## OUTPUT PHASE NOISE DUE TO 1/f NOISE

In-band phase noise at very low offset frequencies may be influenced by the LTC6952's 1/f noise, depending upon $f_{PFD}$. Use the normalized in-band 1/f noise $L_{1/f}$ of –277dBc/Hz with Equation 21 to approximate the output 1/f phase noise at a given frequency offset $f_{OFFSET}$:

$$L_{OUT(1/f)}(f_{OFFSET}) = L_{1/f} + 20 \bullet \log_{10}(f_{OUTx}) - 10 \bullet \log_{10}(f_{OFFSET}) \quad (21)$$

Unlike the in-band noise floor $L_{OUT}$, the 1/f noise $L_{OUT(1/f)}$ does not change with $f_{PFD}$, and is not constant over offset frequency. See Figure 38 for an example of in-band phase noise for $f_{PFD}$ equal to 5MHz and 100MHz. The total phase noise will be the summation of $L_{OUT}$ and $L_{OUT(1/f)}$.

## REFERENCE SIGNAL ROUTING, SPURIOUS, AND PHASE NOISE

The charge pump operates at the PFD's comparison frequency $f_{PFD}$. The resultant output spurious energy is small and is further reduced by the loop filter before it modulates the VCO frequency.

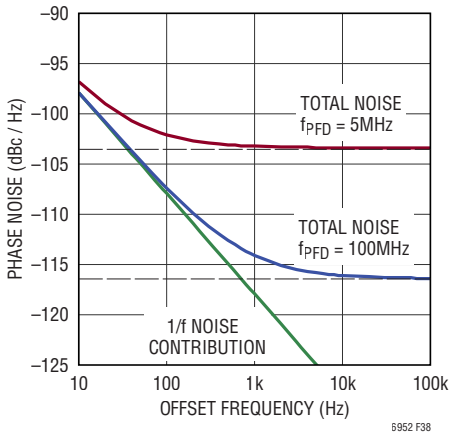



**Figure 38. Theoretical In-Band Phase Noise, $f_{OUTx}$ = 4500MHz**

However, improper PCB layout can degrade the LTC6952's inherent spurious performance. Care must be taken to prevent the reference signal $f_{REF}$ from coupling onto the VCO's tune line, or into other loop filter signals. Example suggestions are the following.

1. Do not share power supply decoupling capacitors between same-voltage power supply pins.
2. Use separate ground vias for each power supply decoupling capacitor, especially those connected to $V_{REF}^+$, $V_D^+$, $V_{OUT}^+$, $V_{CP}^+$, and $V_{VCO}^+$.
3. Physically separate the reference frequency signal from the loop filter and VCO.

## REFERENCE SIGNAL AND EZS_SRQ TIMING FOR ParallelSync MODE

Setting PARSYNC to "1" requires tighter timing between the REF± inputs and the EZS_SRQ input. The LTC6952 is designed to allow sine wave or square wave reference inputs at various levels and all settings of BST or FILT, and have consistent performance with respect to setup and hold times of the EZS_SRQ input pulse. The parameters $t_{SS}$ and $t_{SH}$ are tested and specified for both CMOS and differential input levels applied to REF± and EZS_SRQ±, having the performance characteristics shown in Figure 39 for EZS_SRQ± rising and Figure 40 for EZS_SRQ± falling. For CMOS EZS_SRQ signals, $V_{IH}$ = 1.3V and $V_{IL}$ = 0.6V. For differential EZS_SRQ signals, $V_{IH}$ = $V_{IL}$ = 50% of the signal swing. The trip point for any type of REF± input is always 50%.

## APPLICATIONS INFORMATION

Even if the reference input is a CMOS signal, it is still required to be a high quality signal and is best routed on 50Ω transmission lines. Because CMOS drivers typically are not capable of driving 50Ω, it is recommended to put a resistor in series with the reference output before being applied to the transmission line, and loaded with 50Ω to GND as close to the LTC6952 as possible as shown in Figure 25. See "REFERENCE SOURCE CONSIDERATIONS" above. For production testing, the rise and fall times of the EZS_SRQ and REF± signals are 1ns.

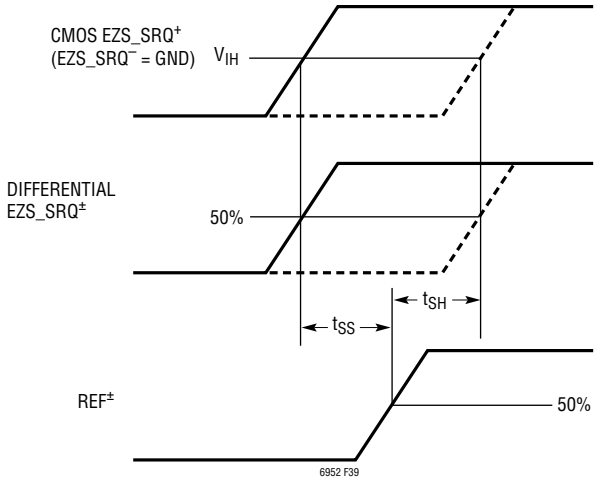



**Figure 39. Rising EZS_SRQ to REF Timing Detail**

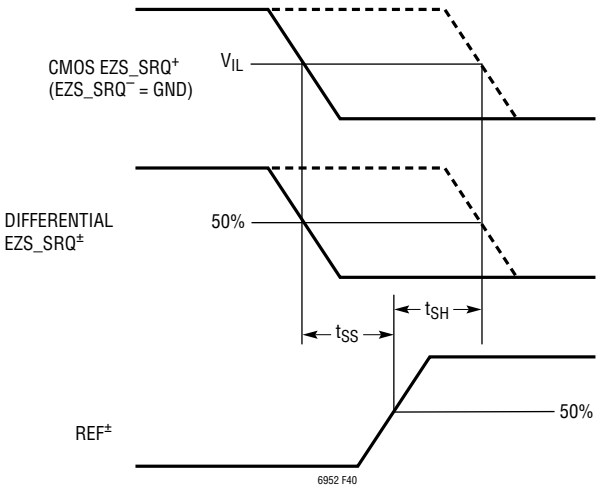



**Figure 40. Falling EZS_SRQ to REF Timing Detail**

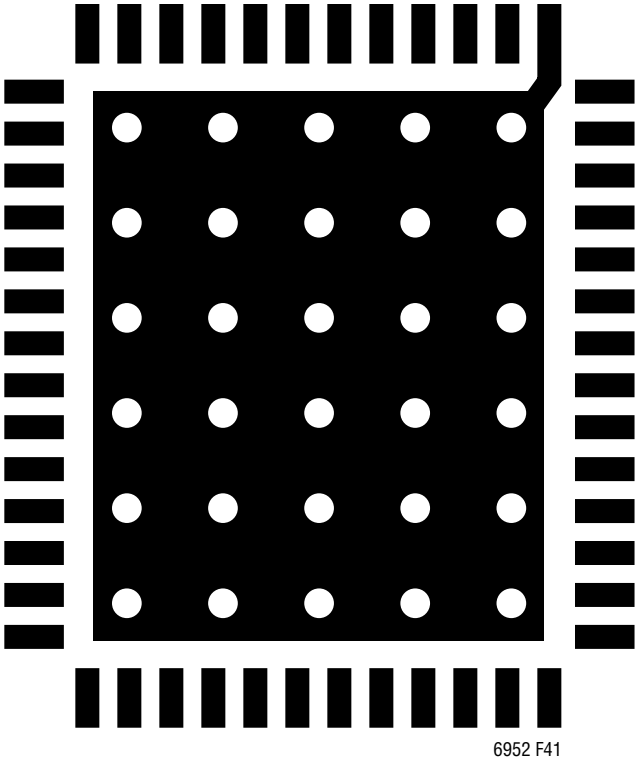



**Figure 41. PCB Top Metal Layer Pin and Exposed Ground Pad Design. Pin 41 is Signal Ground and Connected Directly to the Exposed Pad Metal**

### SUPPLY BYPASSING AND PCB LAYOUT GUIDELINES

Care must be taken when creating a PCB layout to minimize power supply decoupling and ground inductances. All power supply $V^+$ pins should be bypassed directly to the ground plane using either a 0.01µF or a 0.1µF ceramic capacitor as called out in the Pin Functions section as close to the pin as possible. Multiple vias to the ground plane should be used for all ground connections, including to the power supply decoupling capacitors.

The package's exposed pad is a ground connection, and must be soldered directly to the PCB land. The PCB land pattern should have multiple thermal vias to the ground plane for both low ground inductance and also low thermal resistance (see Figure 41 for an example). An example of grounding for electrical and thermal performance can be found on the DC2609 layout.

# APPLICATIONS INFORMATION

## ADC CLOCKING AND JITTER REQUIREMENTS

Adding noise directly to a clean signal clearly reduces its signal to noise ratio (SNR). In data acquisition applications, digitizing a clean signal with a noisy clock signal also degrades the SNR. This issue is best explained in the time domain using jitter instead of phase noise. For this discussion, assume that the jitter is white (flat with frequency) and of Gaussian distribution.

Figure 42 shows a sine wave signal entering a typical data acquisition circuit composed of an ADC, an input signal amplifier and a sampling clock. Also shown are three signal sampling scenarios for sampling the sine wave at its zero crossing.

In the first scenario, a perfect sine wave input is buffered by a noiseless amplifier to drive the ADC. Sampling is performed by a perfect, zero jitter clock. Without any added noise or sampling clock jitter, the ADC's digitized output value is very clearly determined and perfectly repeatable from cycle to cycle.

In the second scenario, a perfect sine wave input is buffered by a noisy amplifier to drive the ADC. Sampling is performed by a perfect, zero jitter clock. The added noise results in an uncertainty in the digitized value, causing an error term which degrades the SNR. The degraded SNR in this scenario, from adding noise to the signal, is expected.

In the third scenario, a perfect sine wave input is buffered by a noiseless amplifier to drive the ADC. Sampling is performed by a clock signal with added jitter. Note that as the signal is slewing, the jitter of the clock signal leads to an uncertainty in the digitized value and an error term just as in the previous scenario. Again, this error term degrades the SNR.

A real-world system will have both additive amplifier noise and sample clock jitter. Once the signal is digitized, determining the root cause of any SNR degradation – amplifier noise or sampling clock jitter – is essentially impossible.

Degradation of the SNR due to sample clock jitter only occurs if the analog input signal is slewing. If the analog input signal is stationary (DC) then it does not matter when in time the sampling occurs. Additionally, a faster slewing input signal yields a greater error (more noise) than a slower slewing input signal.

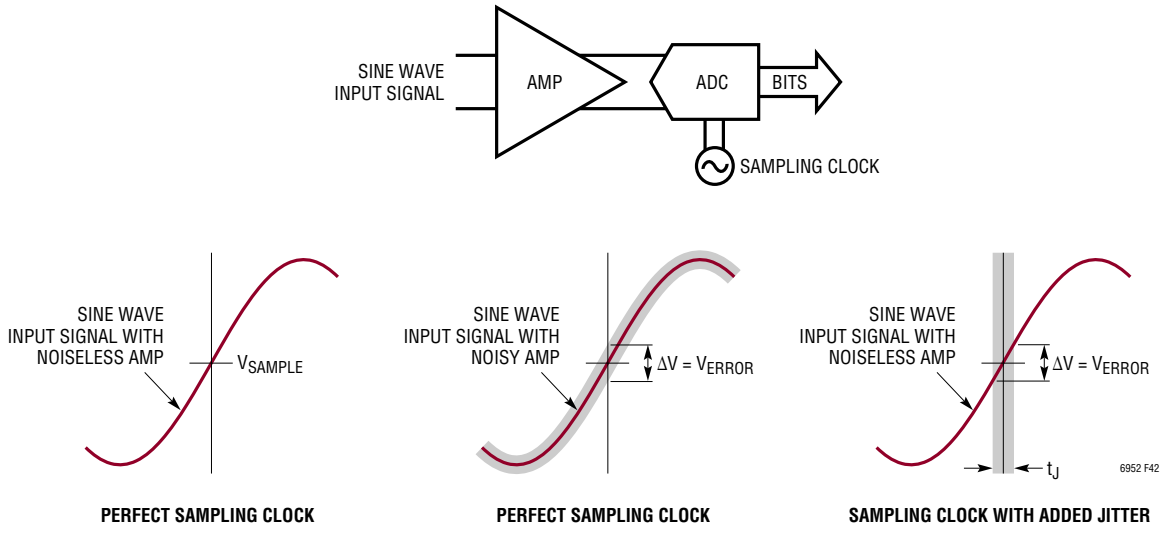



**Figure 42. A Typical Data Acquisition Circuit Showing the Sampling Error Effects of a Noisy Amplifier and a Jittery Sampling Clock**

## APPLICATIONS INFORMATION

Figure 43 demonstrates this effect. Note how much larger the error term is with the fast slewing signal than with the slow slewing signal. To maintain the data converter's SNR performance, digitization of high input frequency signals requires a clock with much less jitter than applications with lower frequency input signals.

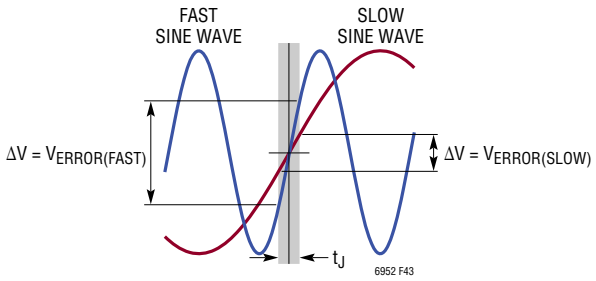



**Figure 43. Fast and Slow Sine Wave Signals Sampled with a Jittery Clock**

It is important to note that the frequency of the analog input signal determines the sample clock's jitter requirement. The actual sample clock frequency does not matter. Many ADC applications that undersample high frequency signals have especially challenging sample clock jitter requirements.

The previous discussion was useful for gaining an intuitive feel for the SNR degradation due to sampling clock jitter.

Quantitatively, the actual sample clock jitter requirement for a given application is calculated as follows:

$$t_{J(TOTAL)} = \frac{10^{\frac{-SNR_{dB}}{20}}}{2 \bullet \pi \bullet f_{SIG}} \quad (22)$$

Where $f_{SIG}$ is the highest frequency signal to be digitized expressed in Hz, $SNR_{dB}$ is the SNR requirement in decibels and $t_{J(TOTAL)}$ is the total RMS jitter in seconds. The total jitter is the RMS sum of the ADC's aperture jitter and the sample clock jitter calculated as follows:

$$t_{J(TOTAL)} = \sqrt{t_{J(CLK)}^2 + t_{J(ADC)}^2} \quad (23)$$

Alternatively, for a given total jitter, the attainable SNR is calculated as follows:

$$SNR_{dB} = -20\log_{10}\left(2 \bullet \pi \bullet f_{SIG} \bullet t_{J(TOTAL)}\right) \quad (24)$$

These calculations assume a full-scale sine wave input signal. If the input signal is a complex, modulated signal with a moderate crest factor, the peak slew rate of the signal may be lower and the sample clock jitter requirement may be relaxed.

These calculations are also theoretical. They assume a noiseless ADC with infinite resolution. All realistic ADCs have both added noise and a resolution limit. The limitations of the ADC must be accounted for to prevent overspecifying the sampling clock.

Figure 44 plots the previous equations and provides a simple, quick way to estimate the sampling clock jitter requirement for a given input signal or the expected SNR performance for a given sample clock jitter.

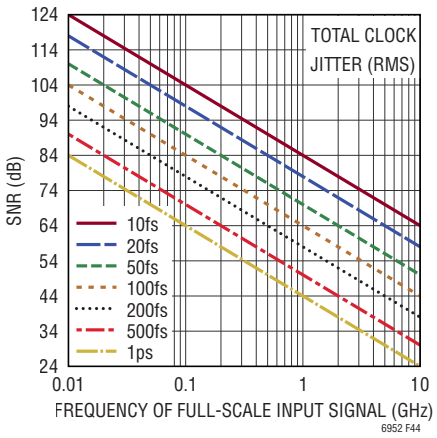



**Figure 44. SNR vs Input Signal Frequency vs Sample Clock Jitter**

# APPLICATIONS INFORMATION

## MEASURING CLOCK JITTER INDIRECTLY USING ADC SNR

For some applications, integrating a clock generator's phase noise within a defined offset frequency range (i.e. 12kHz to 20MHz) is sufficient to calculate the clock's impact on the overall system performance. In these situations, the RMS jitter can be calculated from a phase noise measurement.

However, other applications require knowledge of the clock's phase noise at frequency offsets that exceed the capabilities of today's phase noise analyzers. This limitation makes it difficult to calculate jitter from a phase noise measurement.

The RMS jitter of an ADC clock source can be indirectly measured by comparing a jitter dominated SNR measurement to a non-jitter dominated SNR measurement. A jitter dominated SNR measurement ($SNR_{JITTER}$) is created by applying a low jitter, high frequency full-scale sinewave to the ADC analog input. A non-jitter dominated SNR measurement ($SNR_{BASE}$) is created by applying a very low amplitude (or low frequency) sinewave to the ADC analog input. The total clock jitter ($t_{J(TOTAL)}$) can be calculated using Equation 25.

$$T_{J(TOTAL)} = \frac{10^{\frac{1}{2}\log_{10}\left[10^{-\left(\frac{SNR_{JITTER}}{10}\right)} - 10^{-\left(\frac{SNR_{BASE}}{10}\right)}\right]}}{2\pi f_{IN}} \quad (25)$$

Assuming the inherent aperture jitter of the ADC ($t_{J(ADC)}$) is known, the jitter of the clock generator ($t_{J(CLK)}$) is obtained using Equation 23.

## ADC SAMPLE CLOCK INPUT DRIVE REQUIREMENTS

Modern high speed, high resolution ADCs are incredibly sensitive components able to match or exceed laboratory instrument performance in many regards. Noise or interfering signals on the analog signal input, the voltage reference or the sampling clock input can easily appear in the digitized data. To deliver the full performance of any ADC, the sampling clock input must be driven with a clean, low jitter signal.

Figure 45 shows a simplified version of a typical ADC sample clock input. In this case the input pins are labeled ENC± for Encode while some ADCs label the inputs CLK± for Clock. The input is composed of a differential limiting amplifier stage followed by a buffer that directly controls the ADC's track and hold stage.

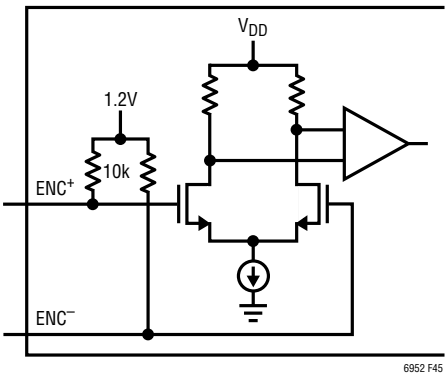



**Figure 45. Simplified Sample Clock Input Circuit**

The sample clock input amplifier also benefits from a fast slewing input signal as the amplifier has noise of its own. By slewing through the crossover region quickly, the amplifier noise creates less jitter than if the transition were slow.

As shown in Figure 45, the ADC's sample clock input is typically differential, with a differential sampling clock delivering the best performance. Figure 45 also shows the sample clock input having a different common mode input voltage than the LTC6952's CML outputs. Most ADC applications will require AC coupling to convert between the two common mode voltages.

## TRANSMISSION LINES AND TERMINATION

Interconnection of high speed signaling with fast rise and fall times requires the use of transmission lines with properly matched termination. The transmission lines may be stripline, microstrip or any other design topology. A detailed discussion of transmission line design is beyond the scope of this data sheet. Any mismatch between the transmission line's characteristic impedance and the

## APPLICATIONS INFORMATION

terminating impedance results in a portion of the signal reflecting back toward the other end of the transmission line. In the extreme case of an open or short circuit termination, all of the signal is reflected back. This signal reflection leads to overshoot and ringing on the waveform. Figure 46 shows the preferred method of far-end termination of the transmission line.

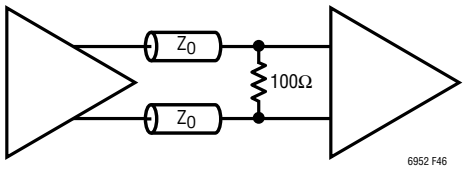



**Figure 46. Far-End Transmission Line Termination ($Z_0$ = 50Ω)**

### USING THE LTC6952 TO DRIVE DEVICE CLOCK INPUTS

The LTC6952's CML outputs are designed to interface with standard CML or LVPECL devices while driving transmission lines with far-end termination. Figure 47 shows DC coupled and AC coupled output configurations for the CML outputs. Note that some receiver devices have the 100Ω termination resistor internal to the part, in which case the external 100Ω resistor is unnecessary.

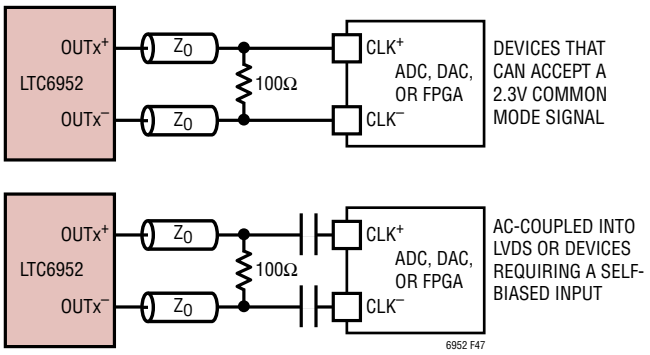



**Figure 47. OUTx CML Connections to Device Clock Inputs ($Z_0$ = 50Ω)**

### USING THE LTC6952 TO DRIVE DC COUPLED SYSREF INPUTS

For JESD204B/C applications, the SYSREF signal would ideally be DC coupled from the LTC6952 to the data converter or FPGA as shown in Figure 48. This is possible for receiver devices that can accept a 2.3V common mode input signal. Note that some receiver devices have the 100Ω termination resistor internal to the part, in which case the external 100Ω resistor is unnecessary.

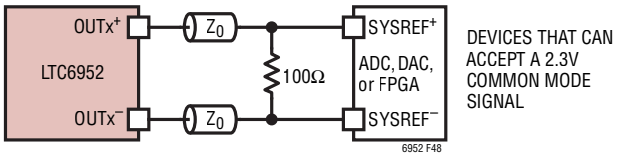



**Figure 48. OUTx CML DC Coupled Connections to SYSREF Inputs**

Use the following procedure to achieve correct JESD204B/C SYSREF behavior for DC coupled SYSREFs in any mode.

*These methods assume that the SYSREF outputs have already been synchronized and that the SYSREF output drivers have been disabled for power savings (PDx = 2).*

#### DC Coupled SYSREFs (MODEx = 0, 1, or 3)

1. Enable the LTC6952 SYSREF output drivers by setting PDx = 0 and set SRQMD = 1.
2. Set the receiver device to accept SYSREFs.
3. Set SSRQ or the EZS_SRQ inputs to "1" for at least 1ms, then set back to "0".
4. After the SYSREFs have been accepted by the receiver device, set the device to stop accepting SYSREFs.
5. Disable the LTC6952 SYSREF output drivers by setting PDx = 2 and set SRQMD = 0.

# APPLICATIONS INFORMATION

## USING THE LTC6952 TO DRIVE AC COUPLED SYSREF INPUTS IN CONTINUOUS OR GATED MODE

Some converters cannot accept a 2.3V common mode CML signal. In this situation, the SYSREF must be AC coupled. AC coupling complicates the usage of SYSREF since it is generally not continuously operational, leading to long settling time requirements before a SYSREF is requested. However, AC coupling on SYSREF can be accomplished by using the connections shown in Figure 49 for continuous or gated SYSREF pulses (MODEx = 0 or 1). Note that some receiver devices have the 100Ω termination resistor internal to the part, in which case the external 100Ω resistor is unnecessary for continuous or gated SYSREFs.

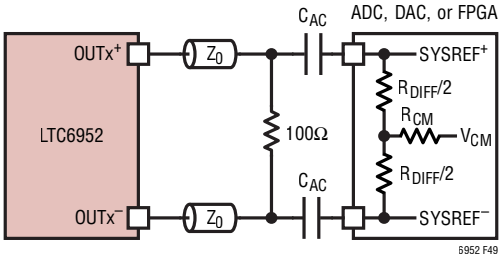



**Figure 49. OUTx CML AC Coupled Connections to SYSREF Inputs for Continuous or Gated Mode Operation**

Settling time for continuous or gated SYSREF connections is determined by the AC coupling capacitors ($C_{AC}$), and both the differential and common mode input resistances of the receiver device ($R_{DIFF}$ and $R_{CM}$):

$$t_{settleC} \cong 10 \bullet (2R_{CM} + R_{DIFF}/2) \bullet C_{AC}$$

Use the following procedure to achieve correct JESD204B/C SYSREF behavior for AC coupled continuous or gated SYSREFs.

*These methods assume that the SYSREF outputs have already been synchronized and that the SYSREF output drivers have been disabled for power savings (PDx = 2).*

### Continuous or Gated SYSREFs (MODEx = 0 or 1)

1. Enable the LTC6952 SYSREF output drivers by setting PDx = 0 and set SRQMD = 1.
2. If gated SYSREFs (MODEx = 1) are being used, set SSRQ or the EZS_SRQ inputs to “1”.
3. Wait for a settling period of at least $t_{settleC}$.
4. Set the receiver device to accept SYSREFs.
5. After the SYSREFs have been accepted by the receiver device, set the device to stop accepting SYSREFs.
6. If gated SYSREFs (MODEx = 1) are being used, set SSRQ or the EZS_SRQ inputs to “0”.
7. Disable the LTC6952 SYSREF output drivers by setting PDx = 2 and set SRQMD = 0.

## USING THE LTC6952 TO DRIVE AC COUPLED SYSREF INPUTS IN PULSED MODE

If AC coupling is required for pulsed SYSREF applications (MODEx = 3), the connections shown in Figure 50 can be used. *Note that some receiver devices have the 100Ω termination resistor internal to the part. In this situation, the use of AC coupled, pulsed SYSREFs is not recommended.*

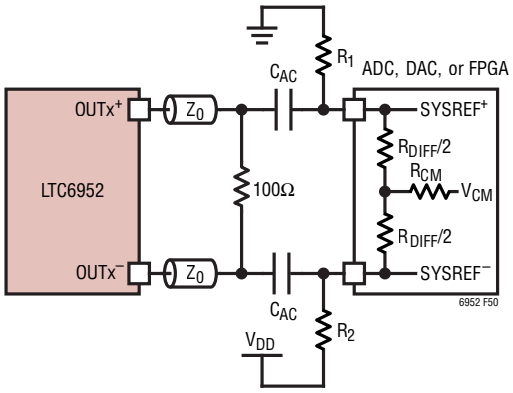



**Figure 50. OUTx CML AC Coupled Connections to SYSREF Inputs for Pulsed Mode Operation**

## APPLICATIONS INFORMATION

The purpose of $R_1$ and $R_2$ in Figure 50 is to force an offset at the SYSREF inputs equivalent to a CML logic "0" when the SYSREF output is not active. The resistors' values are determined by the supply voltage (VDD) and the receiver device's input common mode voltage ($V_{CM}$) and differential input resistance ($R_{DIFF}$). Use Equation 26 to calculate R1 and Equation 27 to calculate R2.

$$R_1 = R_{DIFF} \bullet [V_{CM}/0.44 - 0.5] \quad (26)$$

$$R_2 = R_{DIFF} \bullet [(VDD - V_{CM})/0.44 - 0.5] \quad (27)$$

*For receiver devices with internal 100Ω terminations, the values of $R_1$ and $R_2$ can be very small and will affect the overall termination impedance, leading to undesirable impedance mismatch. For this reason, the use of pulsed SYSREFs (MODEx = 3) AC coupled into receiver parts with internal 100Ω terminations is not recommended.*

Settling time for pulsed SYSREF connections (Figure 50) is *approximately* determined by the AC coupling capacitors ($C_{AC}$), both the differential and common mode input resistance of the receiver device ($R_{DIFF}$ and $R_{CM}$), and resistors $R_1$ and $R_2$:

$$t_{settleP} \cong 10 \bullet [R_{DEV} \bullet R_{OS}/(R_{DEV} + R_{OS})] \bullet C_{AC}$$

where:

$$R_{DEV} = 2R_{CM} + R_{DIFF}/2$$

$$R_{OS} = \text{minimum}(R_1, R_2)$$

For pulsed mode SYSREFs to work correctly with AC coupling, $t_{settleP}$ must be greater than $1000/f_{SYSREF}$, where $f_{SYSREF}$ is the frequency of the SYSREF pulses.

Use the following procedure to achieve correct JESD204B/C SYSREF behavior for AC coupled pulsed SYSREFs.

*These methods assume that the SYSREF outputs have already been synchronized and that the SYSREF output drivers have been disabled for power savings (PDx = 2).*

### Pulsed SYSREFs (MODEx = 3)

1. Enable the LTC6952 SYSREF output drivers by setting PDx = 0 and set SRQMD = 1.
2. Wait for a settling period of at least $t_{settleP}$.
3. Set the receiver device to accept SYSREFs.
4. Set SSRQ or the EZS_SRQ inputs to "1" for at least 1ms, then set back to "0".
5. Set the receiver device to stop accepting SYSREFs.
6. Disable the LTC6952 SYSREF output drivers by setting PDx = 2 and set SRQMD = 0

### MEASURING DIFFERENTIAL SPURIOUS SIGNALS USING SINGLE-ENDED TEST EQUIPMENT

Using a spectrum analyzer to measure spurious signals on the single-ended output of a clock generation chip will give pessimistic results, particularly for outputs that approximate square waves. There are two reasons for this.

First, since the spurious energy is often an AC signal superimposed on the power supply, a differential output will reject the spurs to within the matching of the positive and negative outputs. Observing only one side of the differential output will provide no rejection.

Second, and most importantly, the spectrum analyzer will display all of the energy at its input, including amplitude modulation that occurs at the top and bottom pedestal voltage of the square wave. However, only amplitude modulation near a zero crossing will affect the clock.

The best way to remove this measurement error is to drive the clock generator output differentially into a limiting buffer on a separate clean power supply. One of the differential outputs of the limiting buffer can then connect to a spectrum analyzer to correctly measure the spurious energy. An example of this technique using the LTC6952 as the clock generator and an LTC6955 as the limiter is shown in Figure 51.

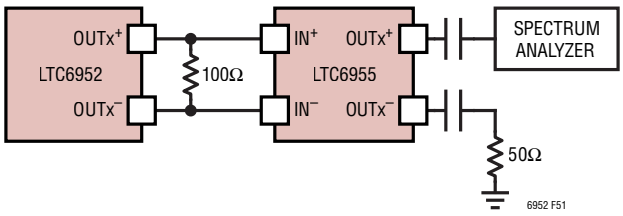
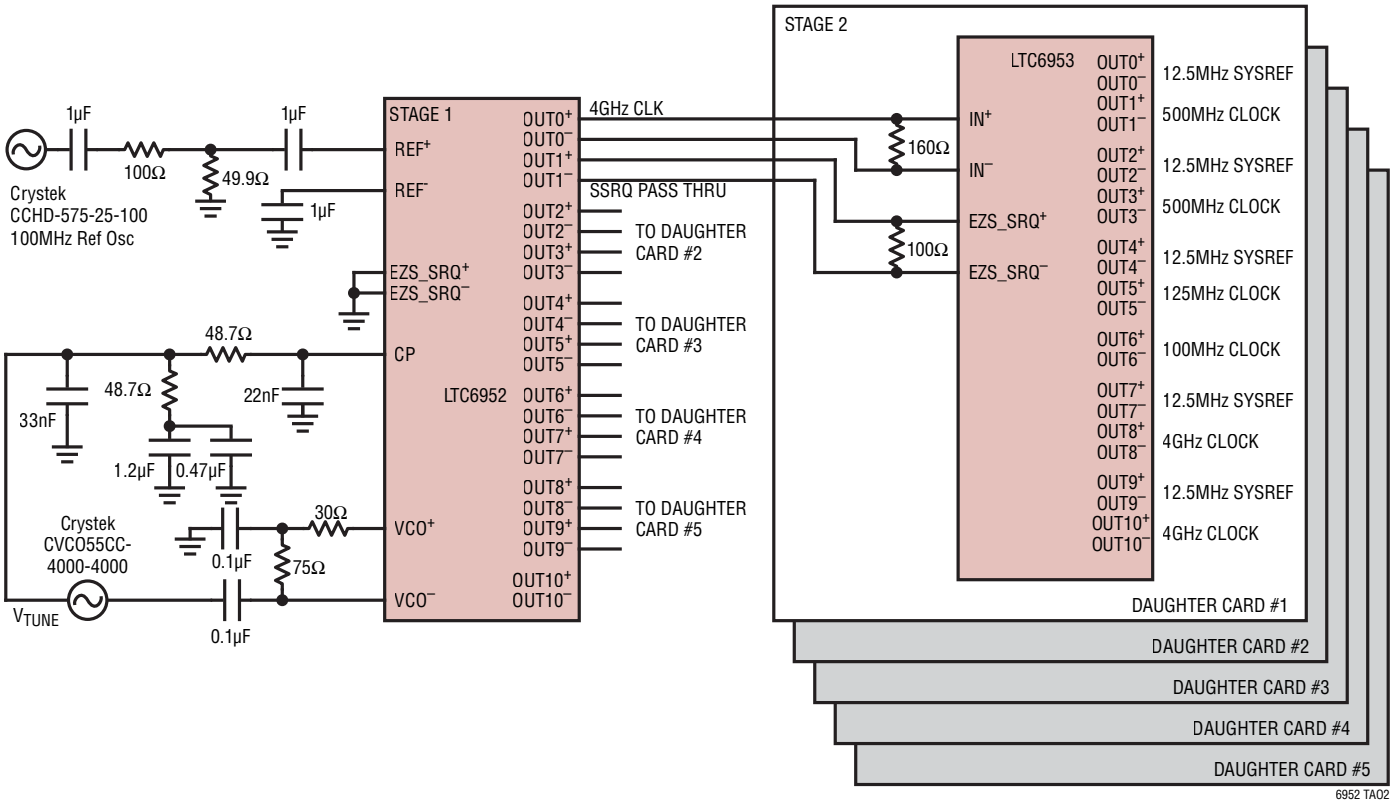


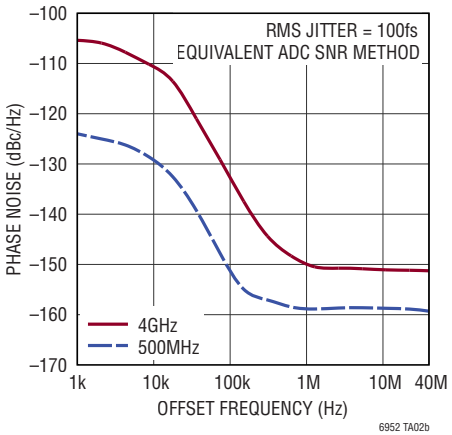

**Figure 51. Example of Spurious Measurement Technique**

# TYPICAL APPLICATIONS

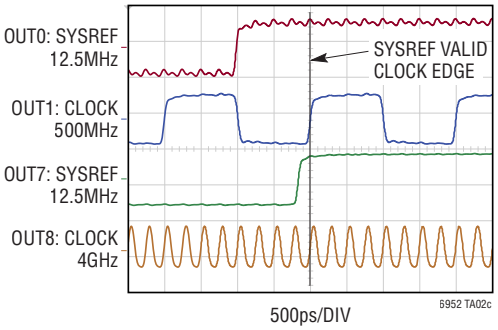
### EZSync Multi-Chip Synchronization with Request Passthrough




### LTC6952 and LTC6953 EZSync Cascaded Phase Noise

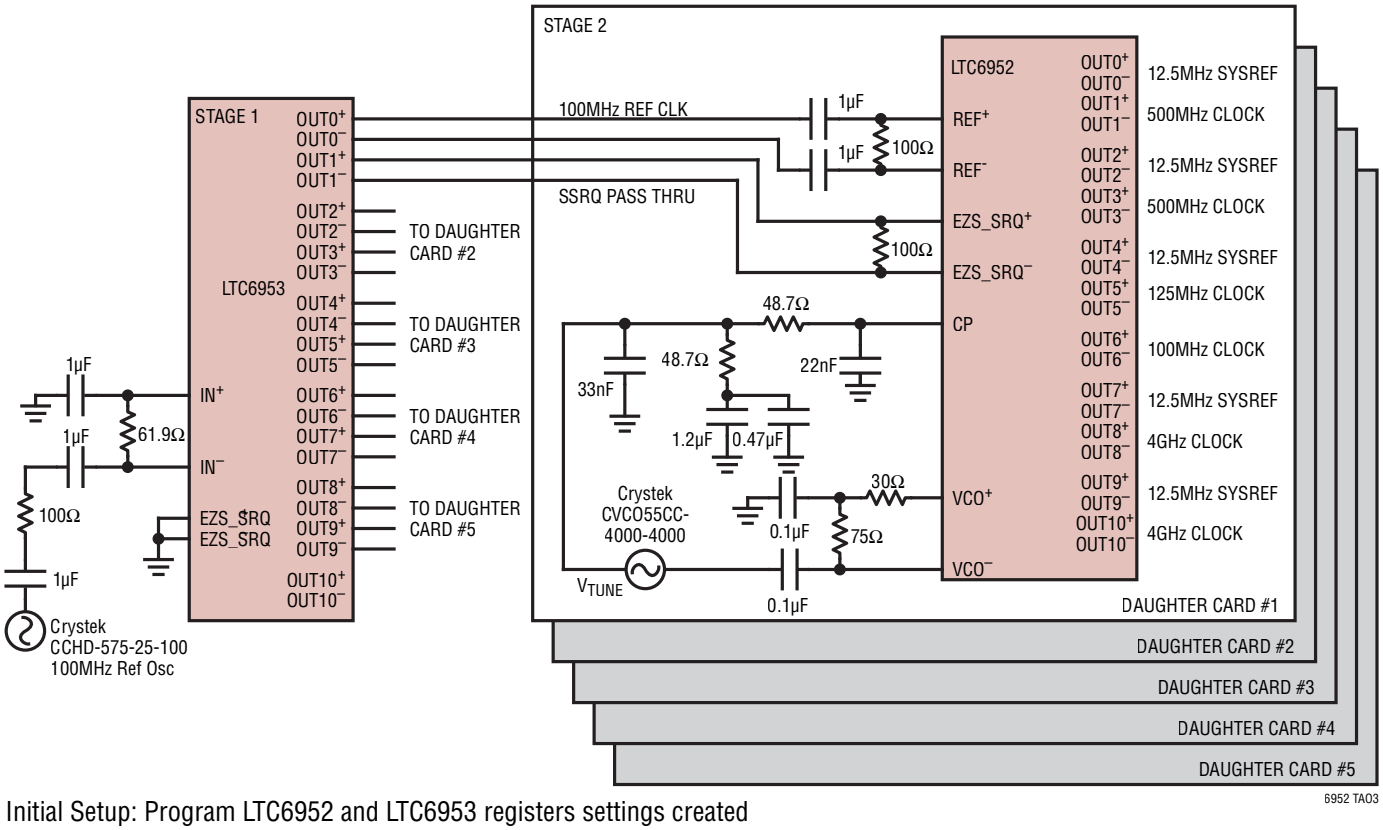



### SYSREF Alignment



# TYPICAL APPLICATIONS

**ParallelSync Multi-Chip Synchronization with Request Passthrough: Schematics and Synchronization Procedure**




Initial Setup: Program LTC6952 and LTC6953 registers settings created from the LTC6952Wizard.

Step 1: Synchronize Stage 1 Reference Signals
- A) EZSync: toggle Stage 1 LTC6953 SSRQ bit
- B) OPT: Fine Alignment, adjust Stage 1 ADEL bits

Step 2: Synchronize Stage 2 Output Signals
- A) Set Stage 1 LTC6953 SRQMD=1
- B) ParallelSync, Toggle Stage 1 LTC6953 SSRQ bit

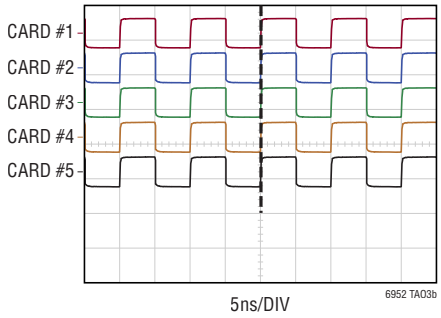
Step 3: Send SYSREF Request
- A) Power Up LTC6952 SYSREF outputs
- B) Set LTC6952 SRQMD=1
- C) SEND SYSREF, Toggle Stage 1 LTC6953 SSRQ bit

Step 4: Optional Reduce Power
- A) Power Down LTC6952 SYSREF outputs
- B) Set Stage 1 LTC6953 & LTC6952 SRQMD=0

# TYPICAL APPLICATIONS

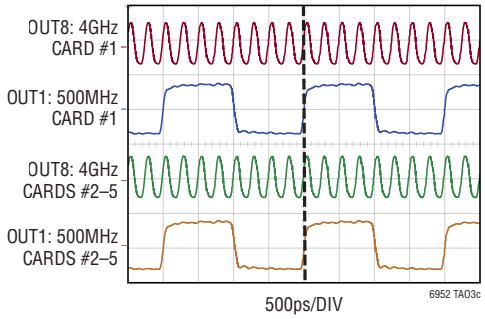
**ParallelSync Multi-Chip Synchronization with Request Passthrough: Measurement Results**

**Step 1: Reference Alignment at Daughter Card Inputs**

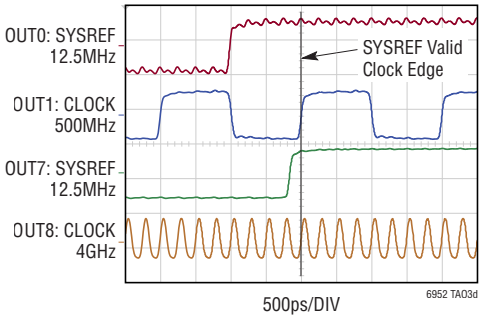



STAGE 1 ADEL ADJUSTMENTS
REF EDGE ALIGNED < ±5.5ps
(ADJUSTS FOR PART, CABLE & PCB MISMATCH)

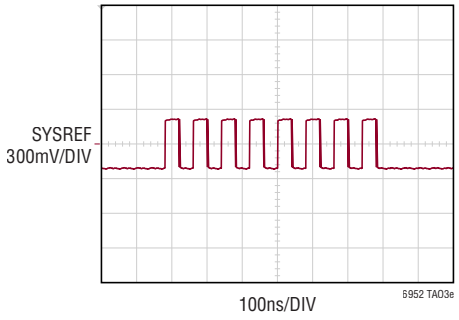
**Step 2: ParallelSYNC Multichip Clock Alignment**



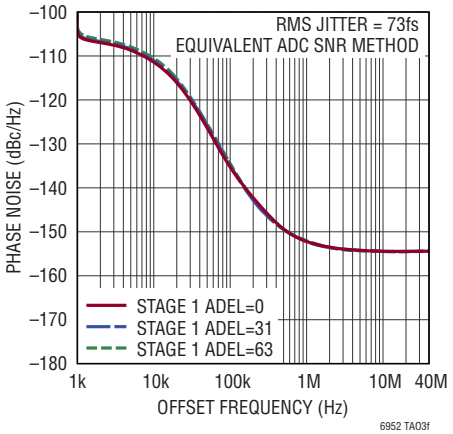

**Step 3: SYSREF Alignment**




**Step 3: SYSREF PULSES**

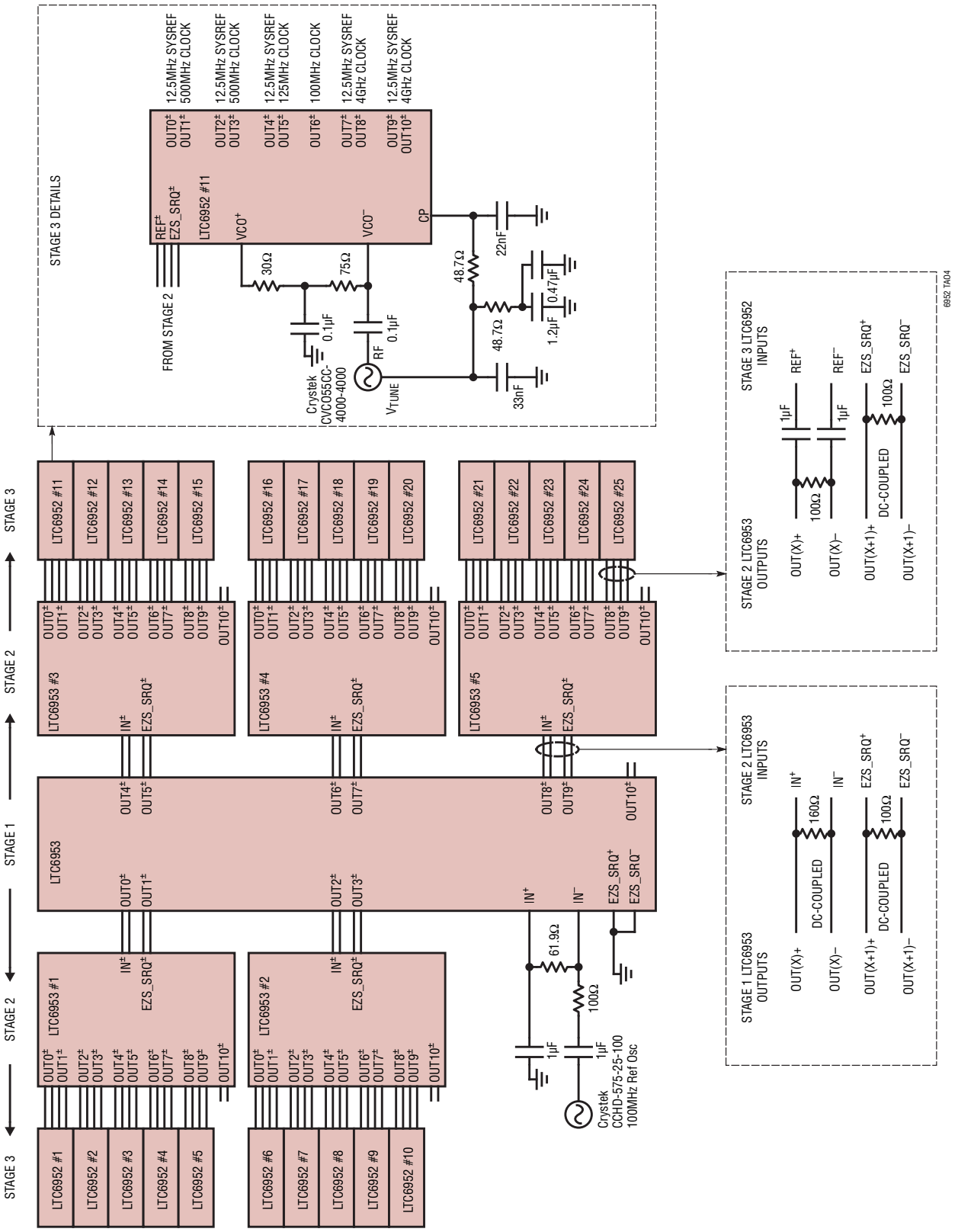



**Stage 2 LTC6952 Phase Noise vs Stage 1 LTC6953 ADEL Setting $f_{OUT}$ = 4GHz, Mx = 1**



# TYPICAL APPLICATIONS

**Generation of up to 125 ADC Clock/SYSREF Pairs Using a Three Stage Synchronization Architecture: Schematics**



# TYPICAL APPLICATIONS

**Generation of up to 125 ADC Clock/SYSREF Pairs Using a Three Stage Synchronization Architecture: Synchronization Procedure and Measurement Results**

Initial Setup: Program LTC6952 and LTC6953 Registers Settings Created from the LTC6952Wizard.

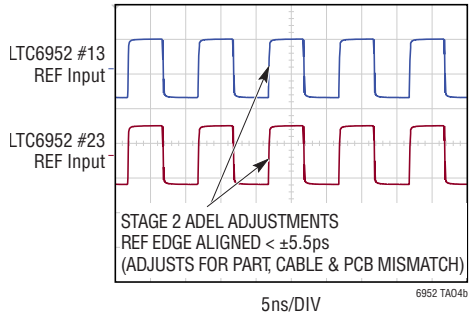
Step 1: Synchronize Stage 1&2 Reference Signals
- A) EZSync: Toggle Stage 1 LTC6953 SSRQ Bit
- B) OPT: Fine Alignment, Adjust Stage 2 ADEL Bits

Step 2: Synchronize Stage 3 Output Signals
- A) Set Stage 1&2 LTC6953 SRQMD=1
- B) ParallelSync: Toggle Stage 1 LTC6953 SSRQ Bit

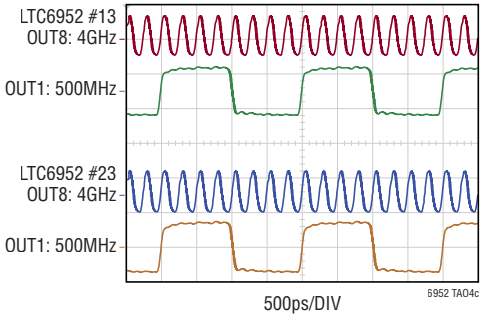
Step 3: Send SYSREF Request
- A) Power Up LTC6952 SYSREF Outputs
- B) Set LTC6952 SRQMD=1
- C) Send SYSREF: Toggle Stage 1 LTC6953 SSRQ Bit

Step 4: Optional Reduce Power
- A) Power Down LTC6952 SYSREF Outputs
- B) Set Stage 1&2 LTC6953 & LTC6952 SRQMD = 0

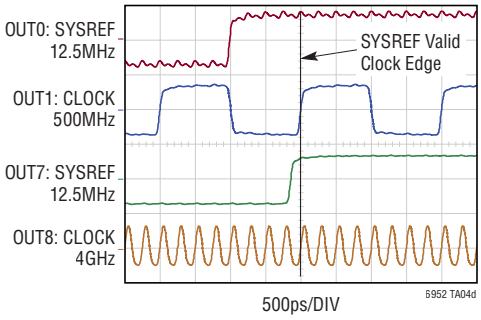
**Step 1: Reference Alignment**



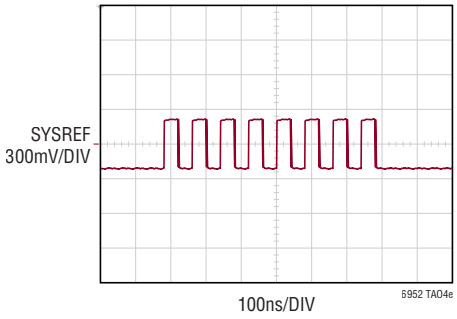

**Step 2: ParallelSYNC Multichip Clock Alignment**



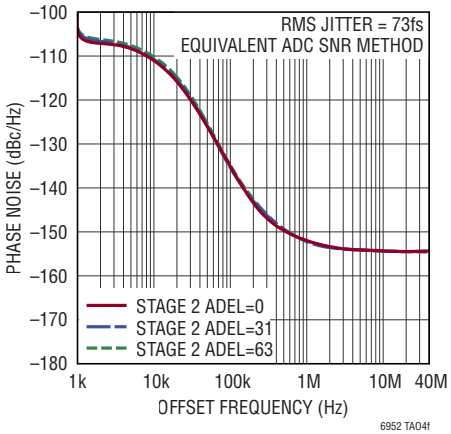

**Step 3: SYSREF Alignment**




**Step 3: SYSREF Pulses**

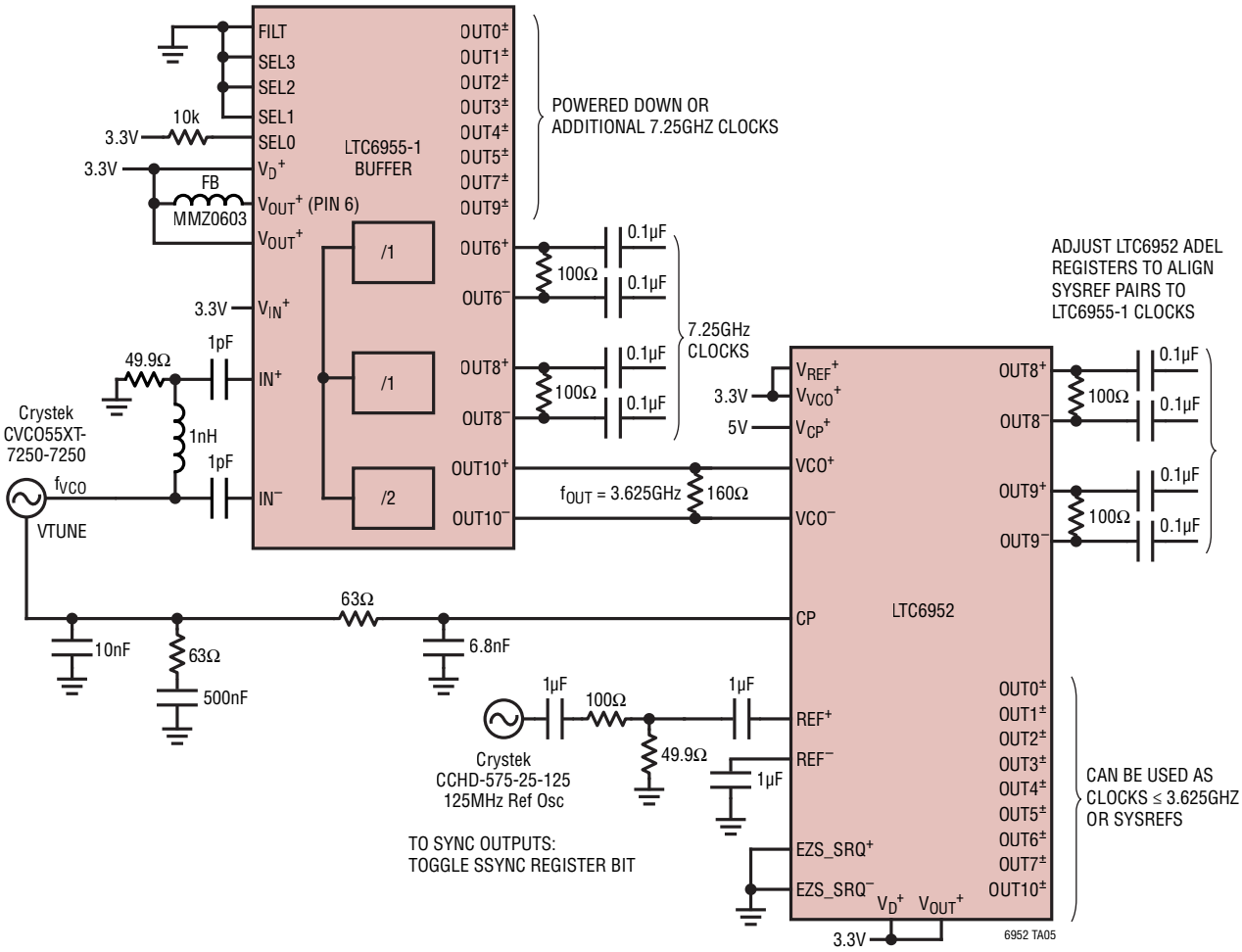



**Stage 3 LTC6952 Phase Noise vs Stage 2 LTC6953 ADEL Setting**
**$f_{OUT}$ = 4GHz, Mx = 1**

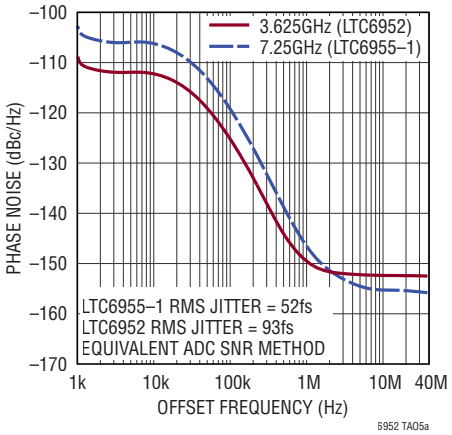


# TYPICAL APPLICATIONS

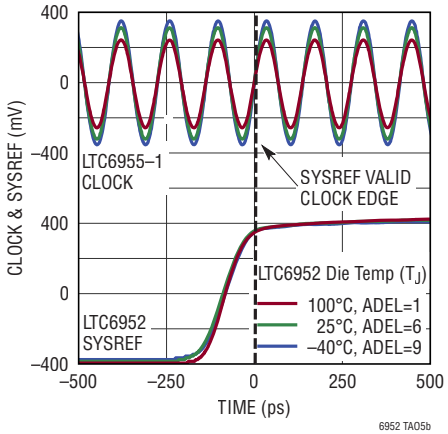
**Generation of 7.25GHz, 52fs ADC SNR Jitter Clocks Using LTC6952 and LTC6955-1**




**LTC6955-1 & LTC6952 Phase Noise $f_{VCO}$ = 7.25GHz**

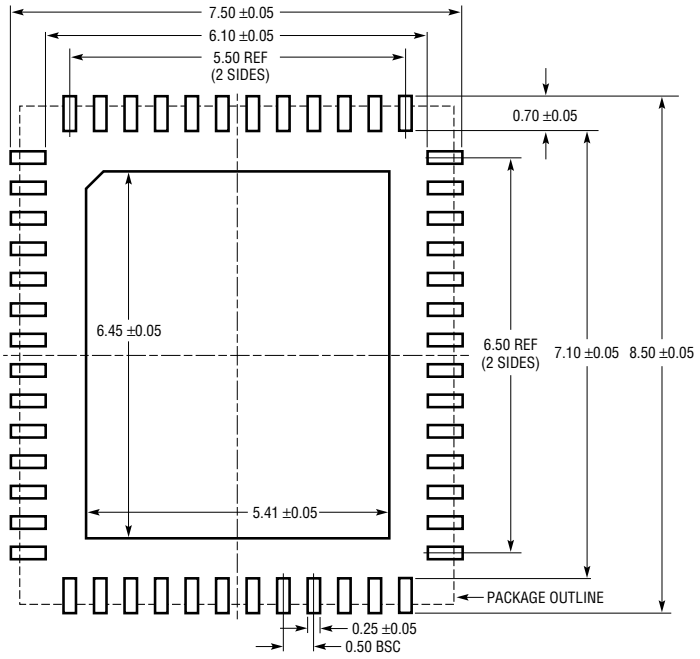


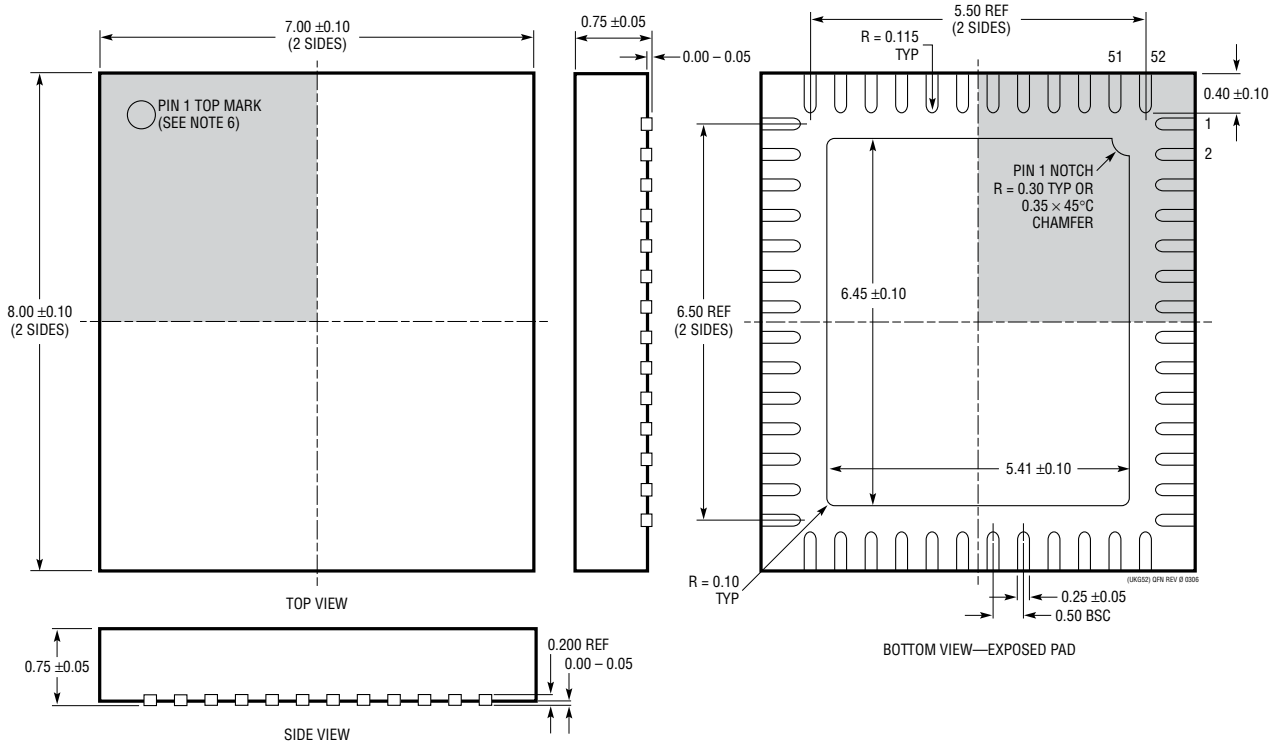

**7.25GHz JESD204B/C CLK to SYSREF Alignment Calibration Over Temperature**



# PACKAGE DESCRIPTION

**UKG Package**
**52-Lead Plastic QFN (7mm × 8mm)**
(Reference LTC DWG # 05-08-1729 Rev Ø)

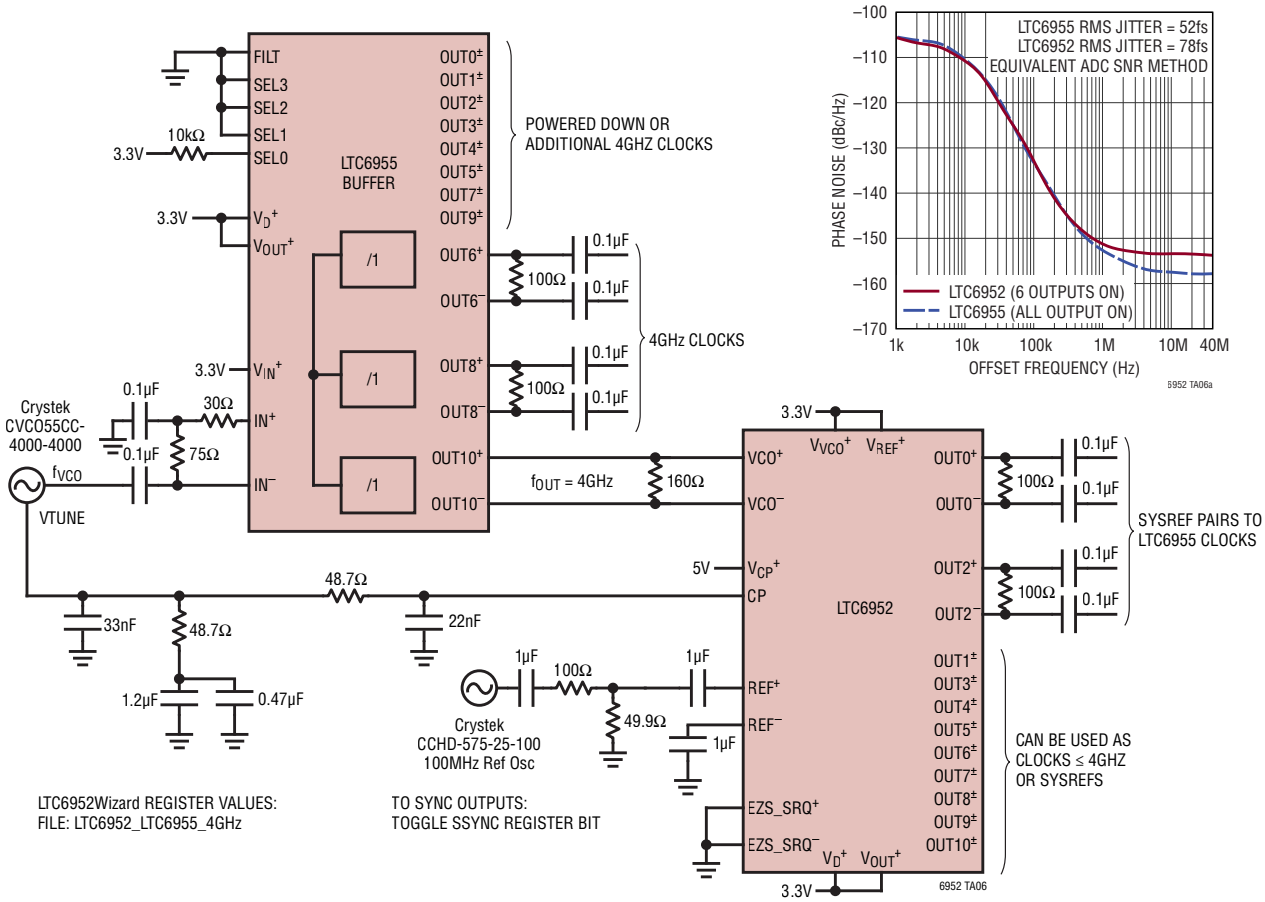



NOTE:
1. DRAWING IS NOT A JEDEC PACKAGE OUTLINE
2. DRAWING NOT TO SCALE
3. ALL DIMENSIONS ARE IN MILLIMETERS
4. DIMENSIONS OF EXPOSED PAD ON BOTTOM OF PACKAGE DO NOT INCLUDE MOLD FLASH. MOLD FLASH, IF PRESENT, SHALL NOT EXCEED 0.20mm ON ANY SIDE, IF PRESENT
5. EXPOSED PAD SHALL BE SOLDER PLATED
6. SHADED AREA IS ONLY A REFERENCE FOR PIN 1 LOCATION ON THE TOP AND BOTTOM OF PACKAGE

# TYPICAL APPLICATION

**Generation of 4GHz, 52fs ADC SNR Jitter Clocks Using LTC6952 and LTC6955**




**LTC6955 vs LTC6952 4GHz Phase Noise**




# RELATED PARTS

| PART NUMBER | DESCRIPTION | COMMENTS |
|---|---|---|
| LTC6953 | Ultralow Jitter, 4.5GHz Clock Distributor with 11 Outputs and JESD204B/C Support | Eleven Independent CML Outputs with Dividers and Delays, 65fs Additive ADC SNR Jitter |
| LTC6955/LTC6955-1 | Ultralow Jitter, 7.5GHz, 11 Output Fanout Buffer Family | Eleven CML Outputs, 45fs Additive ADC SNR Jitter |
| HMC7043 | High Performance, 3.2GHz, 14-Output Fanout Buffer | |
| HMC7044 | High Performance, 3.2GHz, 14-Output Jitter Attenuator with JESD204B/C | |
| HMC987 | 3.3V Low Noise 1:9 Fanout Buffer, DC - 8GHz | |
| LTC6951 | Ultralow Jitter Multioutput Clock Synthesizer with Integrated VCO | Four Independent CML Outputs and One LVDS Output, Integrated VCO, 110fs ADC SNR Jitter |