Ultralow Jitter, 4.5GHz Clock Distributor with 11 Outputs and JESD204B Support

FEATURES

- **JESD204B, Subclass 1 SYSREF Signal Generation**
- **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**
- 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
- 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

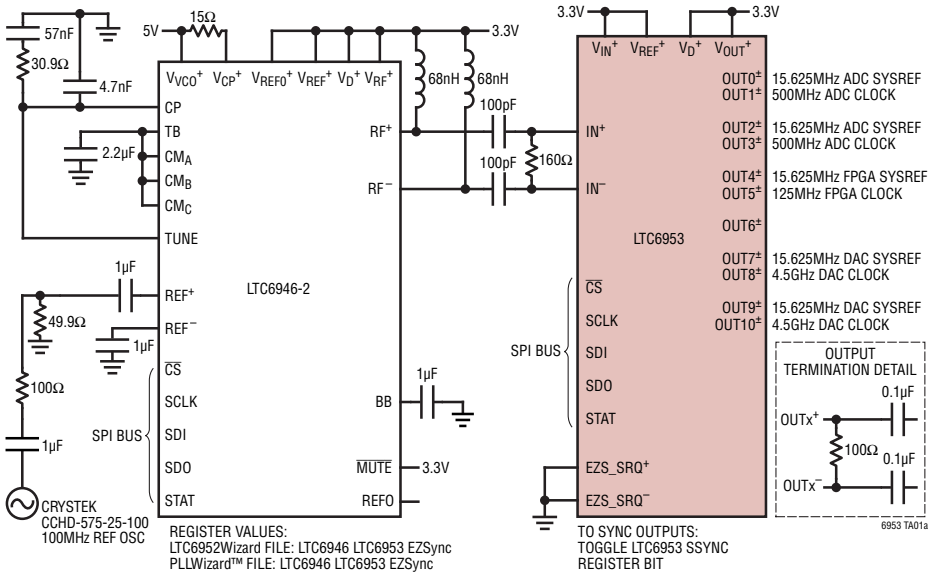
DESCRIPTION

The LTC®6953 is a high performance, ultralow jitter, JESD204B clock distribution IC. The LTC6953's eleven outputs can be configured as up to five JESD204B subclass 1 device clock/SYSREF pairs plus one general purpose output, or simply eleven general purpose clock outputs for non-JESD204B 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 LTC6953s can be connected together with LTC6952s and LTC6955s using the EZSync or ParallelSync synchronization protocols.

TYPICAL APPLICATION

Low Cost, Eleven Output JESD204B Solution



Cumulative Phase Noise, LTC6946 Driving LTC6953

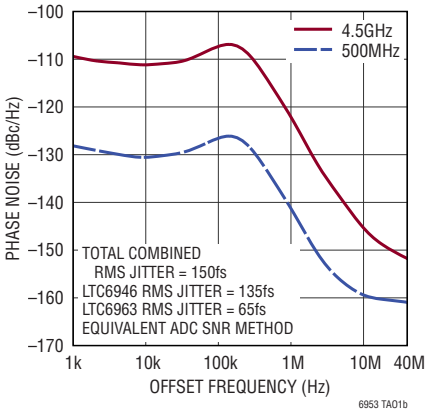


TABLE OF CONTENTS

TABLE OF CONTENTS

ABSOLUTE MAXIMUM RATINGS

(Note 1)

Supply Voltages

V^+ (V_{REF}^+, V_{IN}^+, V_D^+, V_{OUT}^+) to GND 3.6V

Voltage on All Pins.....GND – 0.3V to V^+ + 0.3V

Current Into $OUTx^+$, $OUTx^-$, (x = 0 to 10) .....±25mA

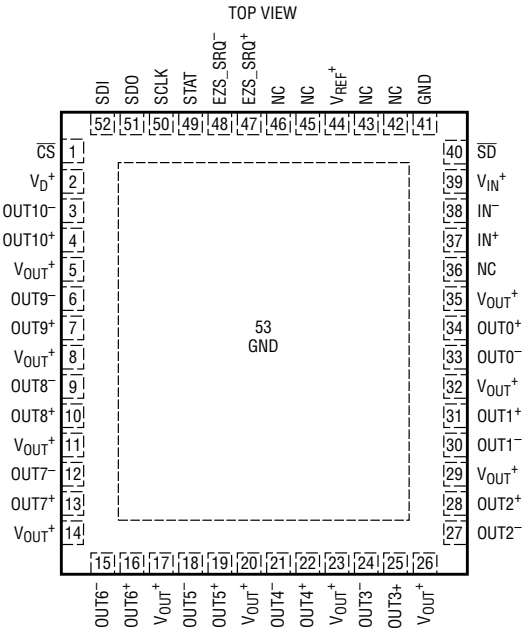
Operating Junction Temperature Range, T_J (Note 2)

LTC6953I ..... –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, θ_{JC} = 2°C/W
EXPOSED PAD (PIN 53) IS GND, MUST BE SOLDERED TO PCB

ELECTRICAL CHARACTERISTICS

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

SYMBOL	PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
Input (IN^+, IN^-)							
f_{IN}	Frequency Range		●			4500	MHz
	Input Signal Level	R_Z = 50Ω, Single-Ended	●	0.25	0.8	1.6	V_{P-P}
	Self-Bias Voltage				2.05		V
	Input Common Mode Voltage	800mV_{P-P} Differential Input	●	1.6		2.7	V
	Input Duty Cycle				50		%
	Minimum Input Slew Rate				100		V/µs
	Minimum Input Signal Detected (V_{COOK} = 1)	PDVCOPK = 0, f_{IN} = 10MHz, Single-Ended Sine Wave	●	350			mV_{P-P}
	Maximum Input Signal Not Detected (V_{COOK} = 0)	PDVCOPK = 0, f_{IN} = 10MHz, Single-Ended Sine Wave	●			100	mV_{P-P}
	Input Resistance	Differential			250		Ω

ELECTRICAL CHARACTERISTICS

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

SYMBOL	PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
	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		●	1.8	2.1	2.4	V
	Input Common Mode Voltage	800mV_{P-P} Differential Input		1.5		3.0	V
	Input Resistance	Differential			53		kΩ
	Input Capacitance	Differential			1		pF
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							
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
Output Dividers (M0, M1, M2, M3, M4, M5, M6, M7, M8, M9 and M10)							
Mx	Output Divider Range (x = 0 to 10)	$Mx = P \times 2^N$, Where P = 1 to 32 All Integers, N = 0 to 7	●	1		4096	counts
D_{DELx}	Output Digital Delay (x = 0 to 10)	½ Input Cycles (Note 3)	●	0		4095	½ cycles

ELECTRICAL CHARACTERISTICS

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

SYMBOL	PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
t_{ADELx}	Output Analog Delay (x = 0 to 10)	ADELx = 0			0		ps
		ADELx = 1, $f_{OUTx} \le$ 300MHz			90		ps
		ADELx = 63, $f_{OUTx} \le$ 300MHz			1100		ps
	Output Analog Delay (x = 0 to 10), Step Size	ADELx = 1 to 31, $f_{OUTx} \le$ 300MHz			11		ps
		ADELx = 32 to 63, $f_{OUTx} \le$ 300MHz			26		ps
	Output Analog Delay (x = 0 to 10)	300MHz $\le f_{OUTx} \le$ 2.25GHz			Note 4		ps
	Maximum Output Frequency for Analog Delay				2.25		GHz
	Temperature Coefficient of Analog Delay	ADELx = 1 to 31	●		0.06		%/°C
		ADELx = 32 to 63	●		0.06		%/°C
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	MHz
V_{OD}	Output Differential Voltage	Differential Termination = 100Ω	●	350	440	520	mV_{PK}
	Output Resistance	Differential			100		Ω
	Output Common Mode Voltage	Differential Termination = 100Ω			$V_{OUT}^+ - 1.0$		V
t_{RISE}	Output Rise Time, 20% to 80%	Differential Termination = 100Ω			50		ps
t_{FALL}	Output Fall Time, 80% to 20%	Differential Termination = 100Ω			50		ps
	Output Duty Cycle	Differential Termination = 100Ω	●	45	50	55	%
t_{PD}	Propagation Delay from IN± to OUT10	f_{IN} = 4500MHz, Mx = 16	●		335		ps
	Propagation Delay from IN± to OUT10, Temperature Variation	f_{IN} = 4500MHz, Mx = 16			0.35		ps/°C
t_{SKEW}	Skew, All Outputs (Note 13)	One Part, All Mx the Same, Even or 1			±10	±35	ps
		One Part, Any Mx			±30		ps
	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		ps
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_{IN}^+ Supply Range		●	3.15	3.3	3.45	V
Power Supply Currents							
I_{DDOUT}	Sum V_{OUT}^+ Supply Currents	All Outputs Enabled (Note 5)	●		732	**TBD**	mA
		Typical JESD204B Application (Note 6)	●		570		mA
		PDALL = 1			500		µA
I_{CC} – 3.3V	Sum V_D^+, V_{REF}+, V_{IN}^+ Supply Currents	Digital Inputs at Supply Levels	●		98	**TBD**	mA
		Digital Inputs at Supply Levels, PDALL = 1	●		500		µA

ELECTRICAL CHARACTERISTICS

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

SYMBOL	PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
	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
		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
Additive Phase Noise, Jitter and Spurious (Note 7)							
	Additive Phase Noise and RMS Jitter (f_{IN} = 4.5GHz, f_{OUTx} = 4.5GHz, Mx = 1)	Phase Noise Floor			–154.3		dBc/Hz
		RMS Jitter, 12kHz to 20MHz Integration BW			6		fs_{RMS}
		RMS Jitter, ADC SNR Method (Note 9)			65		fs_{RMS}
	Additive Phase Noise and RMS Jitter (f_{IN} = 4.5GHz, f_{OUTx} = 2.25GHz, Mx = 2)	Phase Noise Floor			–157.1		dBc/Hz
		RMS Jitter, 12kHz to 20MHz Integration BW			8		fs_{RMS}
		RMS Jitter, ADC SNR Method (Note 9)			66		fs_{RMS}
	Additive Phase Noise and RMS Jitter (f_{IN} = 4.5GHz, f_{OUTx} = 1.125GHz, Mx = 4)	Phase Noise Floor			–160.2		dBc/Hz
		RMS Jitter, 12kHz to 20MHz Integration BW			9		fs_{RMS}
		RMS Jitter, ADC SNR Method (Note 9)			65		fs_{RMS}
	Additive Phase Noise and RMS Jitter (f_{IN} = 3.2GHz, f_{OUTx} = 200MHz, Mx = 16)	Phase Noise Floor			–167.8		dBc/Hz
		RMS Jitter, 12kHz to 20MHz Integration BW			21		fs_{RMS}
		RMS Jitter, ADC SNR Method (Note 9)			65		fs_{RMS}
	Additive Phase Noise and RMS Jitter (f_{IN} = 3.2GHz, f_{OUTx} = 50MHz, Mx = 64)	Phase Noise Floor			–173.8		dBc/Hz
		RMS Jitter, 12kHz to 20MHz Integration BW			41		fs_{RMS}
		RMS Jitter, ADC SNR Method (Note 9)			65		fs_{RMS}

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 LTC6953 is guaranteed to meet specified performance limits over the full operating junction temperature range of –40°C to 125°C. Under maximum operation conditions, airflow 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 Operation section

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

Note 7: Additive phase noise and jitter from LTC6953 only. Incoming clock phase noise is not included.

Note 8: Measured using DC2610.

Note 9: Additive RMS jitter (ADC SNR method) is calculated by integrating the distribution section's measured phase noise floor out to f_{CLK}. Actual ADC SNR measurements show good agreement with this method.

Note 10: Measured with 36" cables from output of DC2610 to measurement instrument. Cable loss is NOT accounted for in this plot.

Note 11: Statistics calculated from 1200 total measured parts from two process lots.

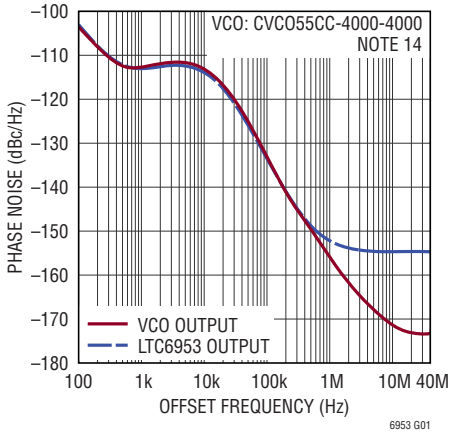
Note 12: Skew is defined as the difference between the zero-crossing time of a given output and the average zero-crossing time of all outputs.

Note 13: Measured input power is de-embedded to the IN± pins of the LTC6953.

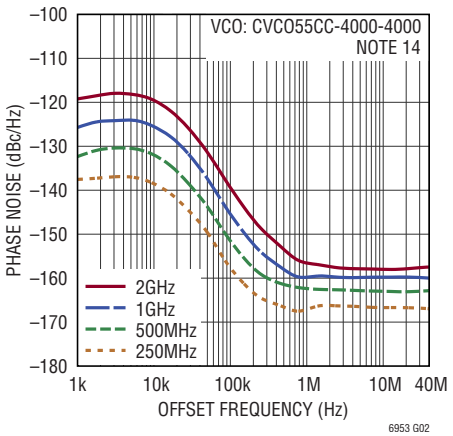
Note 14: The LTC6953 is driven from a VCO (CVCO55CC-4000-4000) through a splitter. The other side of the splitter drives the input of a LTC6952 to lock the VCO in a PLL. The reference for the LTC6952 PLL is a Pascal OCXO-E, f_{REF} = 100MHz, P_{REF} = 6dBm.

TYPICAL PERFORMANCE CHARACTERISTICS

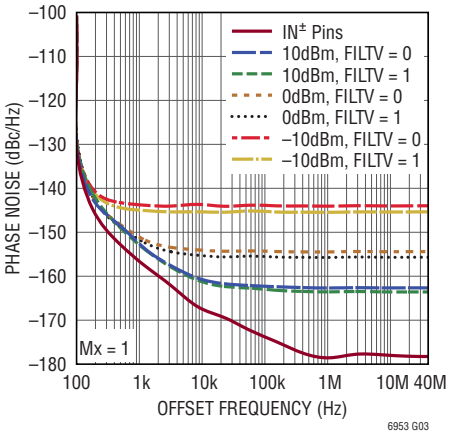
Total Phase Noise, Driven from VCO in a Locked PLL, f_{VCO} = 4GHz, Mx = 1



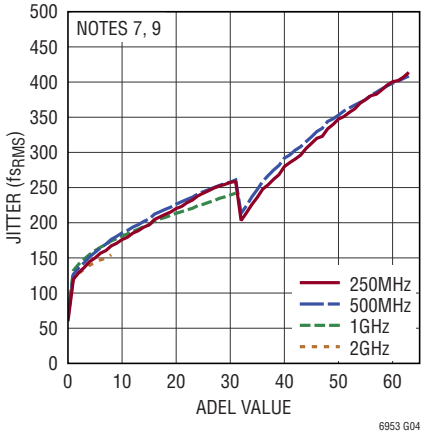
Total Phase Noise, Driven from VCO in a Locked PLL, f_{VCO} = 4GHz, Mx = 2, 4, 8 and 16



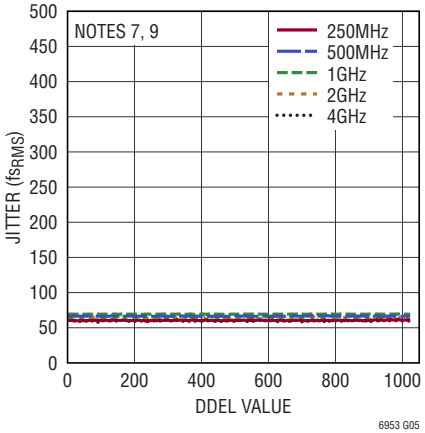
Total Phase Noise, 100MHz Sine Wave Input Signal



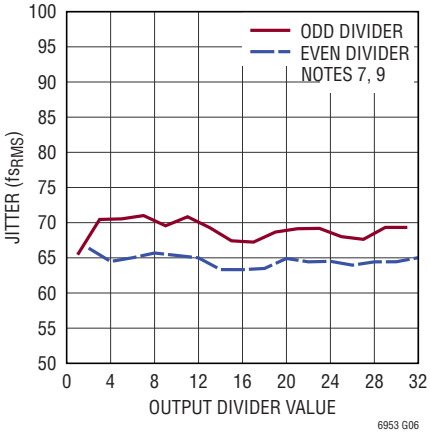
Additive Jitter vs ADEL Value, ADC SNR Method, f_{IN} = 4GHz, Mx = 2, 4, 8 and 16



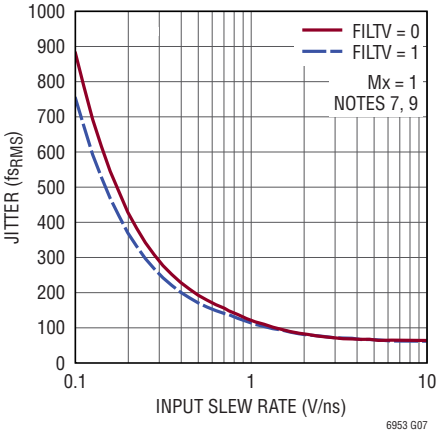
Additive Jitter vs DDEL Value, ADC SNR Method, f_{IN} = 4GHz, Mx = 1, 2, 4, 8 and 16



Additive Jitter vs Divider Setting, ADC SNR Method



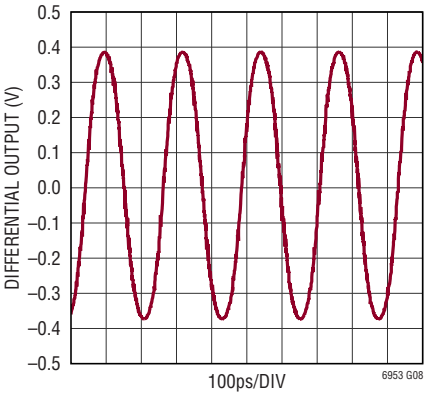
Additive Jitter vs Input Slew Rate, ADC SNR Method



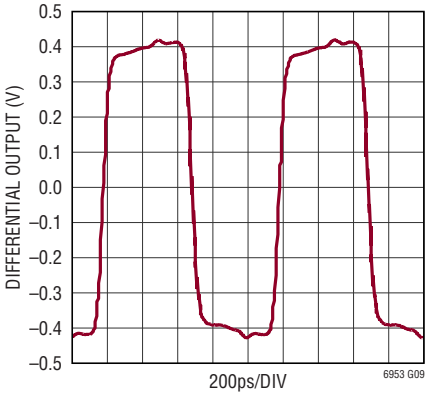
Rev PrA

TYPICAL PERFORMANCE CHARACTERISTICS

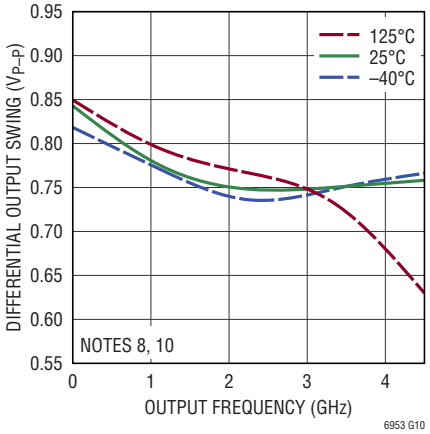
CML Differential Output at 4.5GHz



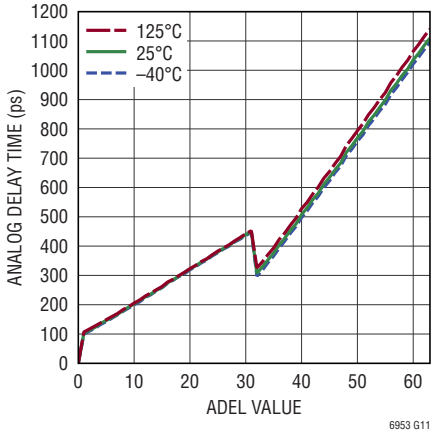
CML Differential Output at 1GHz



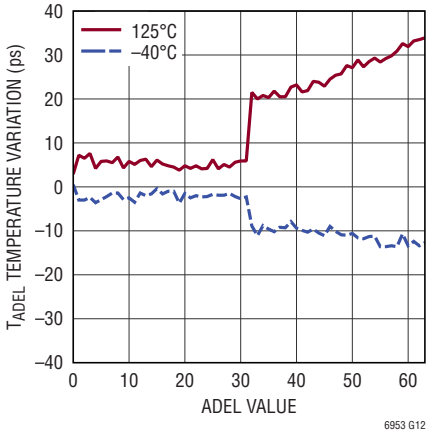
Differential Output Swing vs Frequency, Temperature



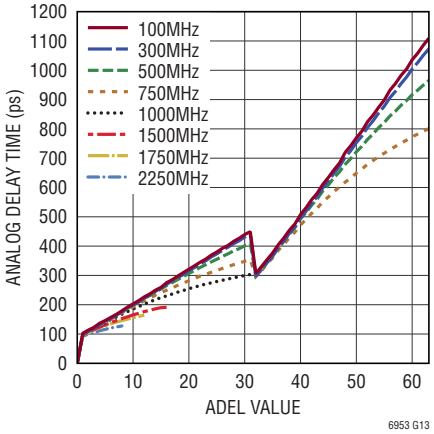
Analog Delay Time vs ADEL Value, Temperature



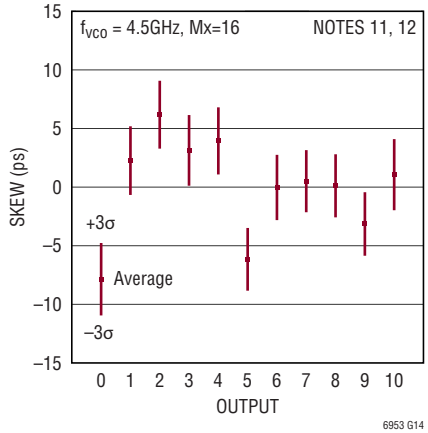
Analog Delay Time Temperature Variation from 25°C



Analog Delay Time vs ADEL Value Over Multiple Output Frequencies

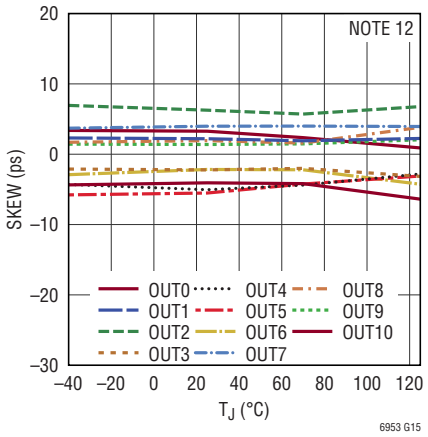


Expected Skew Variation for a Single Part

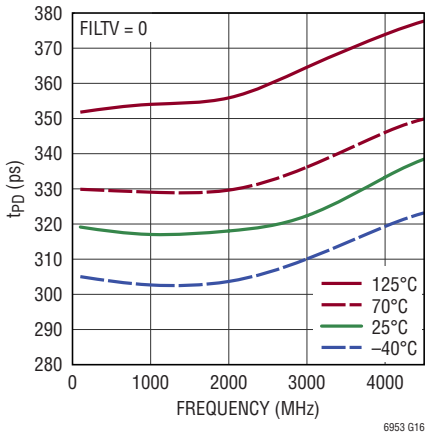


TYPICAL PERFORMANCE CHARACTERISTICS

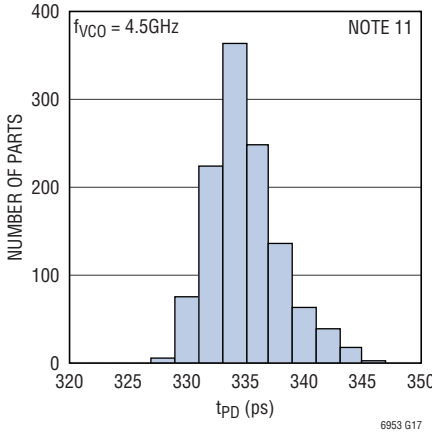
Skew Variation with Temperature for a Single Typical Part



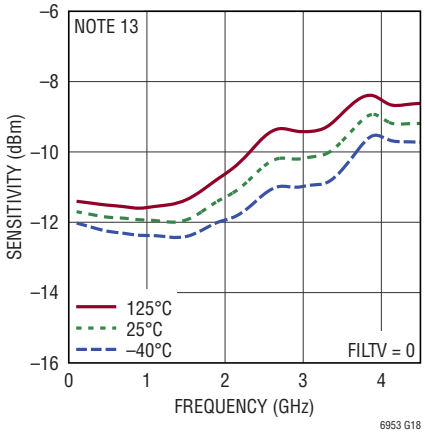
Propagation Delay vs Frequency, Temperature



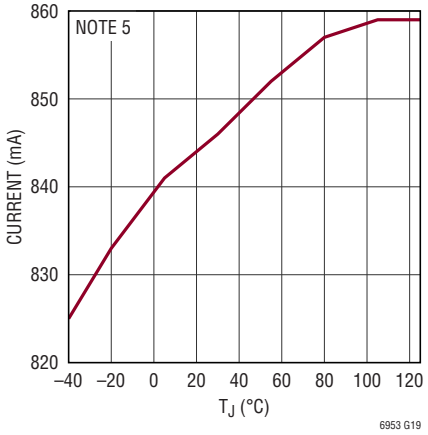
Propagation Delay Variation, Input to OUT4



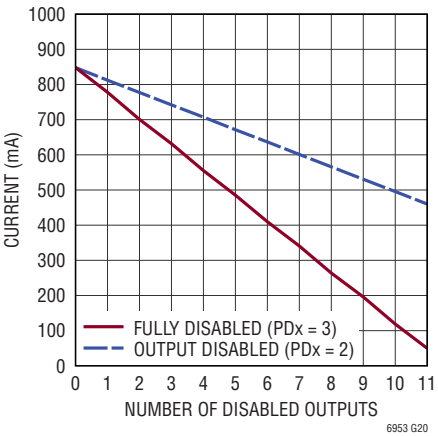
Input Signal Detected vs Frequency, Temperature



Supply Current vs Junction Temperature, All Outputs Enabled



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 (Pins 36, 42, 43, 45, 46): Not Connected Internally. It is recommended that these pins be connected to the ground pad (pin 53).

IN$^+$, IN$^-$ (Pins 37, 38): Input Signals. The differential signal placed on these pins is buffered with a low noise amplifier and fed to the internal distribution path. 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 Operation section.

V_{IN}^+ (Pins 39): 3.15V to 3.45V Positive Supply Pin for Input 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_{REF}^+ (Pin 44): 3.15V to 3.45V Positive Supply Pin for EZS_SRQ circuitry. This pin should be bypassed directly to the ground plane using a 0.1µF ceramic capacitor as close to the pin as possible.

EZS_SRQ$^+$, EZS_SRQ$^-$ (Pins 47, 48): Synchronization or SYSREF Request Input. Bit SRQMD defines this differential or single-ended 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 sections for more details.

STAT (Pin 49): Status Output. This signal is a configurable logical OR combination of the $\overline{VCOOK}$ and VCOOK 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.

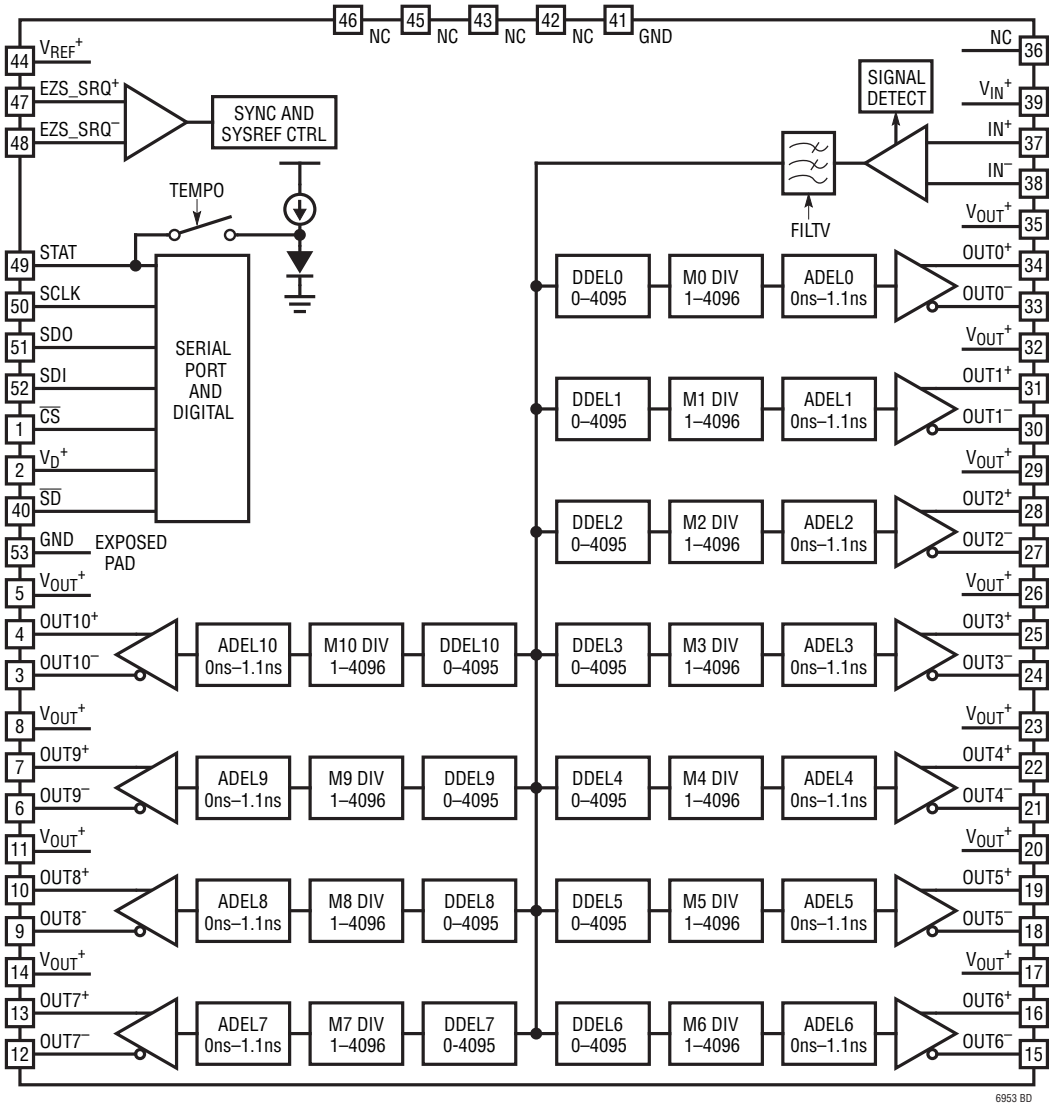
SCLK (Pin 50): Serial Port Clock. This CMOS input clocks serial port input data on its rising edge. See the Operation section for more details.

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.

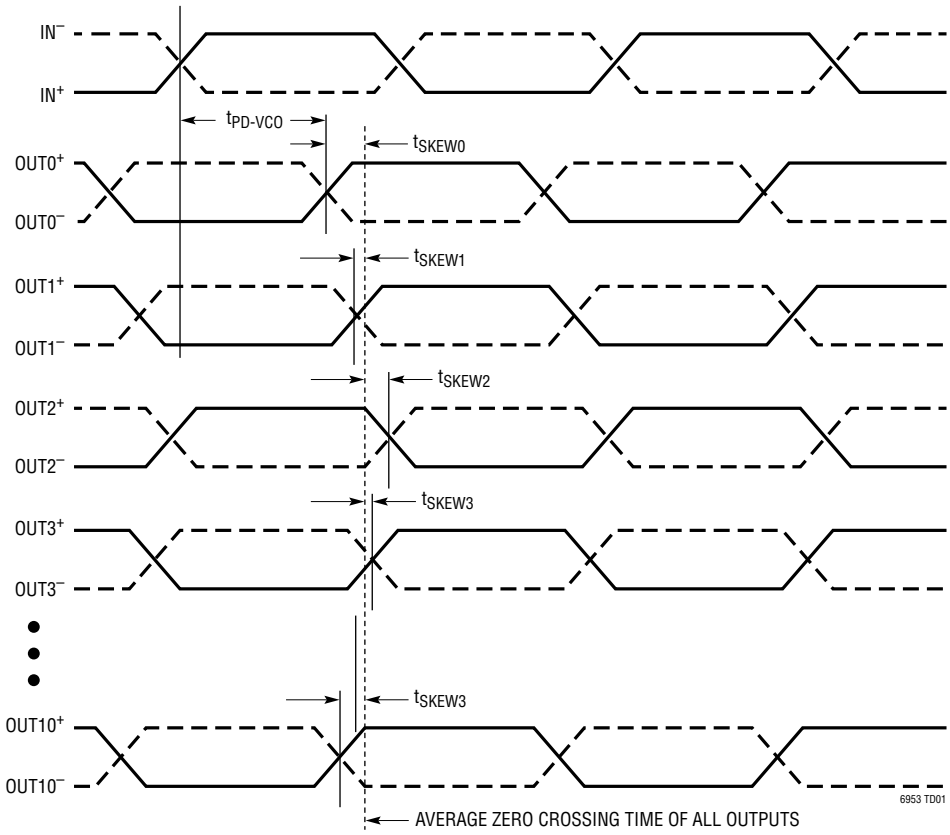
GND (Exposed Pad Pin 53): Negative Power Supply (Ground). The package exposed pad must be soldered directly to the PCB land. The PCB land pattern should have multiple thermal vias to the ground plane for both low ground inductance and also low thermal resistance.

BLOCK DIAGRAM

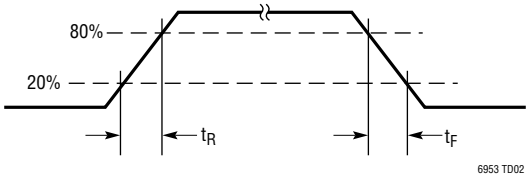


TIMING DIAGRAMS

Propagation Delay and Output Skew



Differential CML Rise/Fall Times



OPERATION

The LTC6953 is a high performance multioutput clock distributor that operates up to 4.5GHz. Utilizing Analog Devices's proprietary EZSync and ParallelSync standards, users of the LTC6953 and its companion part LTC6952 can synchronize clocks across multiple outputs and multiple chips. The device is able to achieve superior additive integrated jitter performance by way of its excellent output noise floor. For JESD204B subclass 1 applications, the LTC6953 also provides several convenient methods to generate SYSREF pulses.

INPUT BUFFER

The LTC6953's 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 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 input interface configurations, noting that the LTC6953's input buffer has an internal differential resistance of 250Ω as shown in Figure 1.

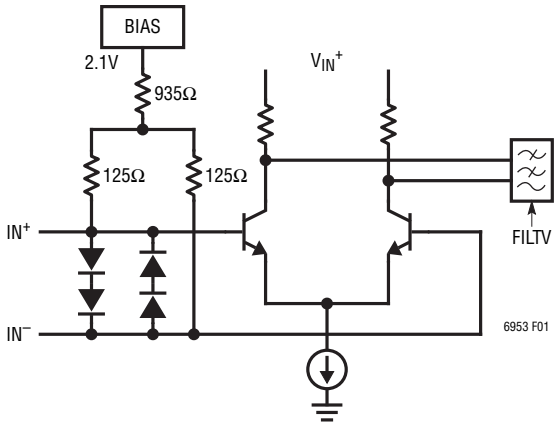


Figure 1. Simplified Input Interface Schematic

The maximum input frequency for the input buffer is 4.5GHz, and the maximum amplitude is $1.6V_{P-P}$. It is also important that the IN± 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 input slew rates less than 2V/ns, better phase noise performance will be achieved by enabling the internal broadband noise filtering circuit within the 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 input is greater than 2V/ns will degrade the overall PLL-phase noise performance. See Table 1 for recommended settings of FILTV.

Table 1. FILTV Programming

FILTV	SLEW RATE OF INPUT
1	< 2V/ns
0	≥ 2V/ns

Input Peak Detector

An input peak detection circuit on the IN± inputs detects the presence of an input 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 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 2 for VCOOK detection values.

Table 2. VCOOK, $\overline{\text{VCOOK}}$ Status Output vs Input Amplitude

VCOOK	$\overline{\text{VCOOK}}$	SINE WAVE f_{IN}	SQUARE WAVE f_{IN}
1	0	≥ $350mV_{P-P}$	≥ $275mV_{P-P}$
0	1	< $100mV_{P-P}$	< $75mV_{P-P}$

OUTPUT DIVIDERS (M0 TO M10)

The eleven independent, identical output dividers are driven directly from the input buffer. They divide the input frequency f_{IN} 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 Equation 1:

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

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

OPERATION

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 3.

Table 3. 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 input 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 frequency dependent, and the valid useful range of ADELx is reduced according to Table 4.

Table 4. Maximum ADELx vs Output Frequency Range

f_{OUT} RANGE	MAXIMUM ADELx
$f_{OUTx} \leq 750MHz$	63
$750MHz < f_{OUTx} \leq 1GHz$	31
$1GHz < f_{OUTx} \leq 1.5GHz$	16
$1.5GHz < f_{OUTx} \leq 1.75GHz$	12
$1.75GHz < f_{OUTx} \leq 2.25GHz$	8
$f_{OUTx} > 2.25GHz$	0

Figure 2 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.

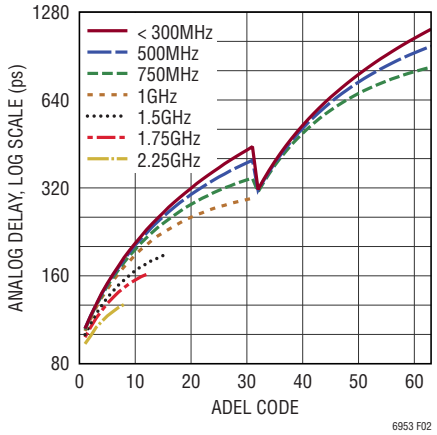


Figure 2. Analog Delay Time vs ADEL Code and Output Frequency

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 an 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

OPERATION

matched-length traces. OINVx also determines the state of the output in a muted condition (PDx = 1) as shown in Table 3. See Figure 3 for circuit details.

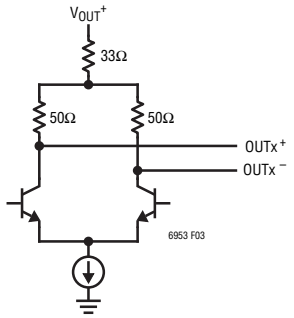


Figure 3. Simplified CML Interface Schematic (All OUTx)

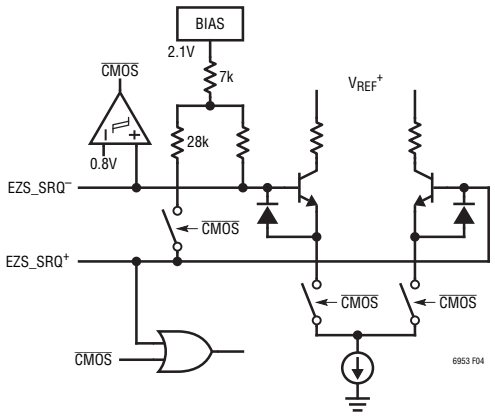


Figure 4. Simplified EZS_SRQ Interface Schematic

OUTPUT SYNCHRONIZATION AND SYSREF GENERATION

The LTC6953 has circuitry to allow all outputs to be synchronized into known phase alignments in multiple ways to suit different applications using the EZSync Multichip Clock Edge Synchronization protocol. Synchronization can be between any combination of outputs on the same chip (EZSync Standalone) or across multiple cascaded FOLLOWER chips (EZSync Multichip). Once the outputs are at the correct frequency and synchronized, the LTC6953 also has the ability to produce free running, gated or finitely pulsed SYSREF signals as indicated by the JESD204B 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 4. 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”. Bit SRQMD determines the type of request issued by the EZS_SRQ± pins or SSRQ bit. When SRQMD = 0, the request is for synchronization of the outputs. When SRQMD = 1, it is a request to output SYSREF pulses.

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.

Synchronization Overview

The goal of synchronization is to align all output dividers on single or multiple LTC6953s (or other compatible Analog Devices clocking 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 LTC6953, 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. 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.*

OPERATION

When the EZS_SRQ input is driven back low or "0" is written to the SSRQ bit, the synchronized internal dividers will start after an initial latency. Outputs with DDELx ≠ 0 will be delayed by an extra DDELx/2 input cycles. The behavior of each output will be defined individually by the output's corresponding SRQENx and MODEx bits as shown in Table 5. 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 input *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.

Table 5. Synchronization (SRQMD = 0) Output Behavior vs Device Settings

SRQENx	MODEx	INTERNAL DIVIDER SYNC TO OTHER DIVIDERS	INTERNAL DIVIDER START LATENCY FROM SYNC SIGNAL FALLING EDGE (DDELx = 0)	OUTPUT BEHAVIOR
0	0	No	N/A	Free Running
	1, 2 or 3			Muted
1	0	Yes	~45µs + 7/ f_{IN}	Mute on SYNC High, Run on Sync Low
	1 or 3			Muted
	2			Sync Signal Pass-Through

SYSREF Generation Overview

The JESD204B 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.

The LTC6953 supports three different methods of SYSREF generation as described in the JESD204B 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" 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". Table 6 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.

Table 6. 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

Multichip Synchronization and SYSREF Generation

Using one LTC6953 in EZSync Standalone configuration (Figure 5), up to eleven clock signals or SYSREFs can be generated and synchronized. For applications requiring more than eleven clock outputs, the LTC6953 and its companion chip, the LTC6952, support two methods of multichip synchronization and SYSREF generation: EZSync Multichip and ParallelSync. The synchronization configuration is determined by bits EZMD and PARSYNC (on the LTC6952 only), and their required settings are shown in Table 7. Table 8 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.

OPERATION

Table 7. Settings of EZMD and PARSYNC for Different Synchronization Topologies

CONTROL BIT	EZSync STANDALONE (SEE FIGURE 5)	EZSync MULTICHIP (SEE FIGURES 6 AND 7)		ParallelSync MULTICHIP (SEE FIGURES 8 AND 9)
		CONTROLLER	FOLLOWER	
PARSYNC (LTC6952 Only)	0	0	0	1
EZMD	0	0	1	0

Table 8. Parameters and Limitations of EZSync and ParallelSync (Two Stages Only)

	EZSync STANDALONE (SEE FIGURE 5)	EZSync MULTICHIP				ParallelSync MULTICHIP	
		PIN CONTROLLED REQUESTS (SEE FIGURE 6)		REQUEST PASS-THROUGH (SEE FIGURE 7)		GENERAL REFERENCE DISTRIBUTION (SEE FIGURE 8)	LTC6953 REFERENCE DISTRIBUTION SEE FIGURE 9)
		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 – 2 Nfol	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

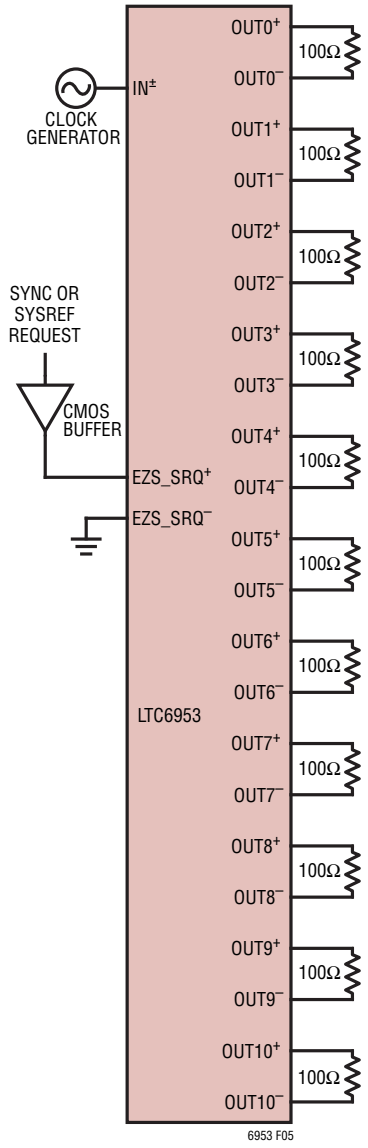


Figure 5. EZSync Standalone

EZSync Multichip

When using EZSync multichip, compatible devices are cascaded together, with the clock output of a CONTROLLER device driving the inputs of one to eleven FOLLOWER devices as shown in Figure 6. The EZSync protocol allows for simple synchronization of all devices due to loose timing constraints on the SYNC signal. When used in a JESD204B 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

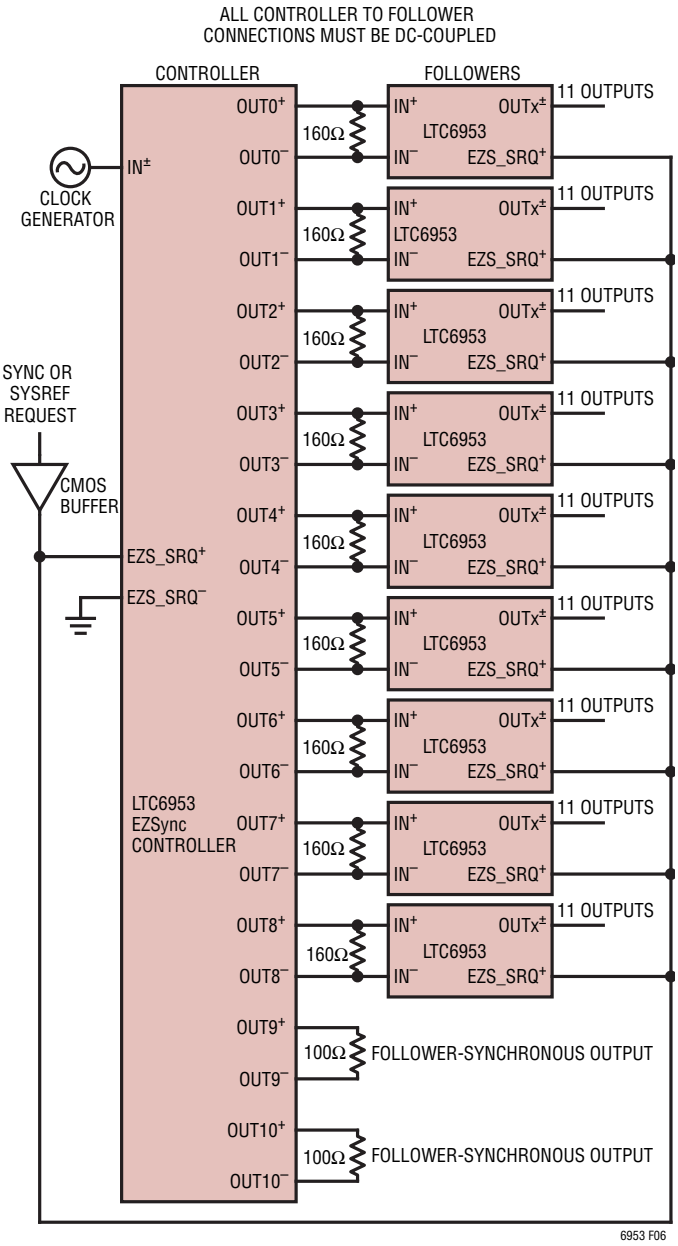


Figure 6. EZSync Multichip Synchronization (Nine Followers Shown, Max Eleven Possible

as any FOLLOWER devices. However, LTC6952 can be used as a FOLLOWER if necessary by disabling its PLL (PDPLL = 1).

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 7, where an additional CONTROLLER output drives each FOLLOWER's EZS_SRQ pins (only available for LTC6953

OPERATION

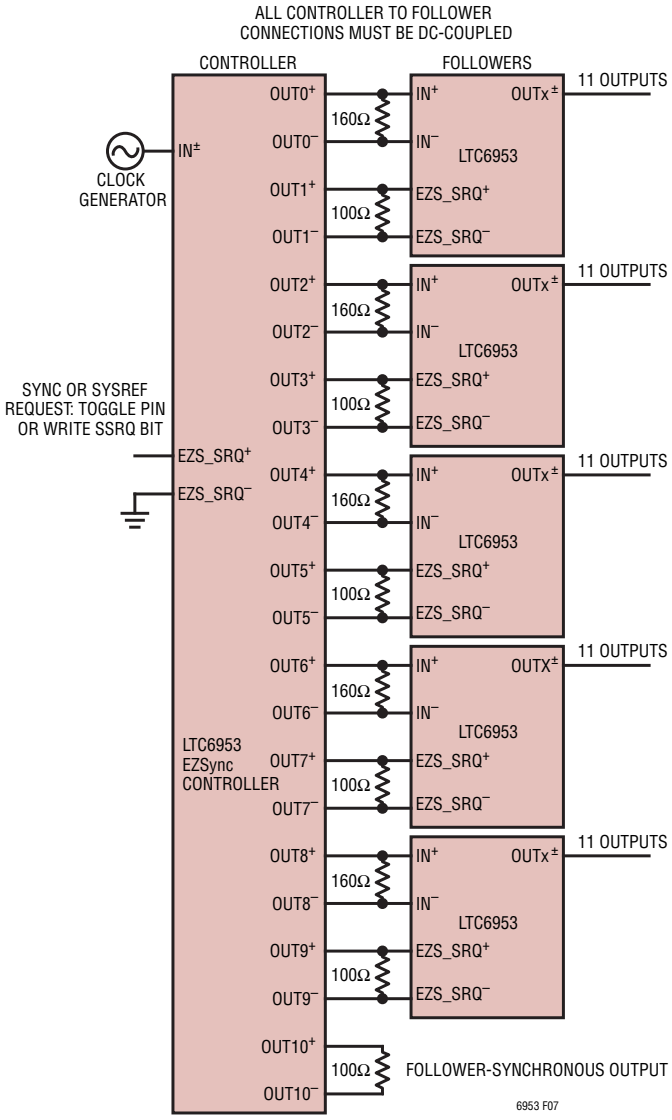


Figure 7. EZSync Multichip Synchronization with Request Pass-Through

or LTC6952 FOLLOWERS). This request pass-through configuration reduces the system complexity at the cost of fewer possible FOLLOWERS (a maximum of 5). Note that MPx for the CONTROLLER's pass-through output must be set greater than 0.

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 8. 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. The LTC6953 is not compatible as a ParallelSync device except as part of the reference and EZS_SRQ distribution

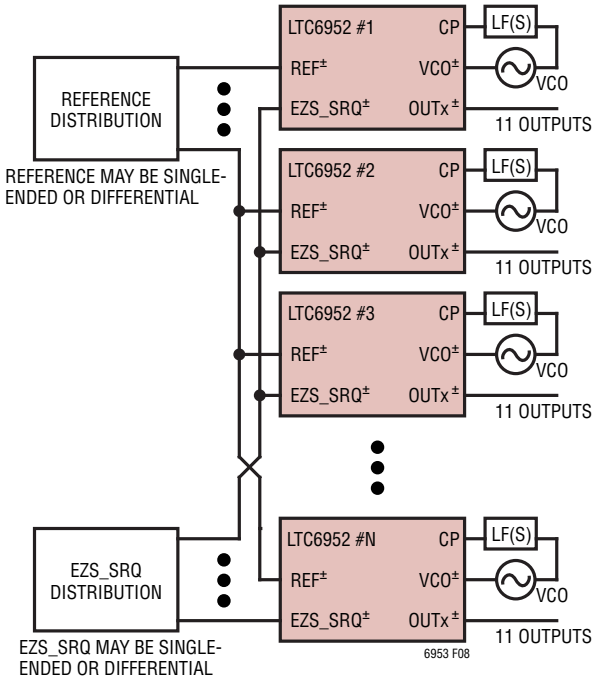


Figure 8. ParallelSync Multichip Synchronization

OPERATION

as described below. See the LTC6952 data sheet for more information on ParallelSync.

The SYNC/SRQ timing for ParallelSync can be simplified to a single software bit write by using an LTC6953 (or LTC6952 with its PLL disabled) as the reference and EZS_SRQ distribution block, as shown in Figure 9. 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

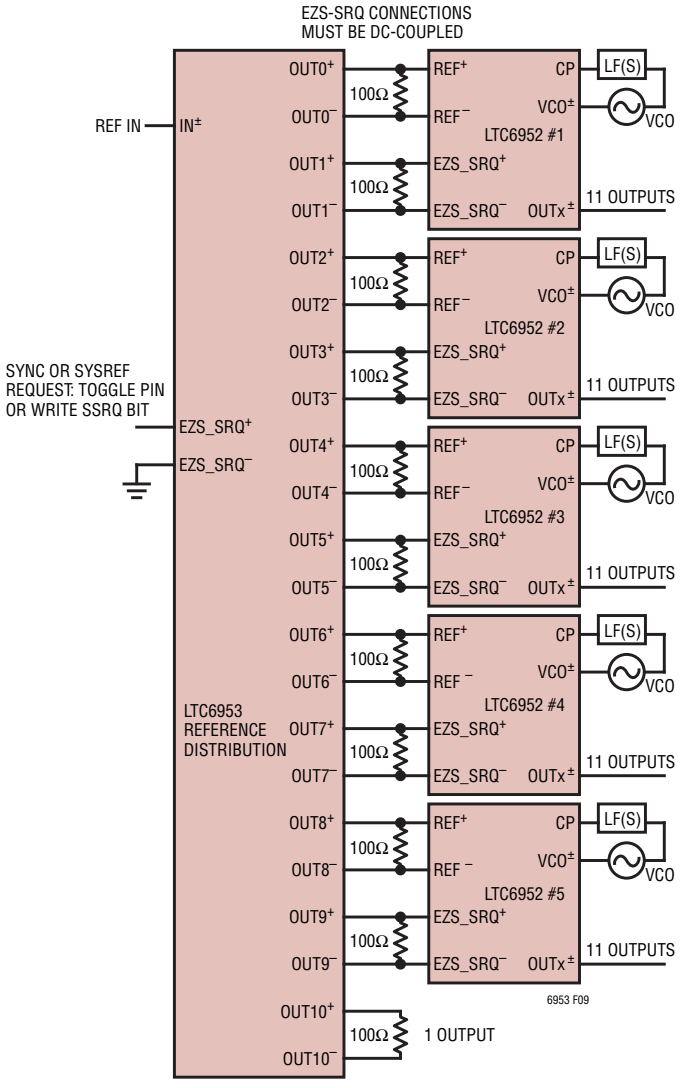


Figure 9. ParallelSync Multichip Synchronization with LTC6953 or LTC6952 Reference Distribution

the settings given in Table 9, where $DDEL_{REF}$ can be any valid DDEL value.

Table 9. 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	$DDEL_{REF}$

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.

To determine the best configuration for a given application, the flowchart in Figure 10 can be used. This flowchart uses the parameters from Table 8 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.

Power Savings in SYSREF Generation Mode

In most applications, SYSREF requests are a rare occurrence. The LTC6953 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 LTC6953 into SYSREF generation mode (SRQMD = 1) causes the part to draw a significantly higher current than SRQMD = 0. Therefore,

OPERATION

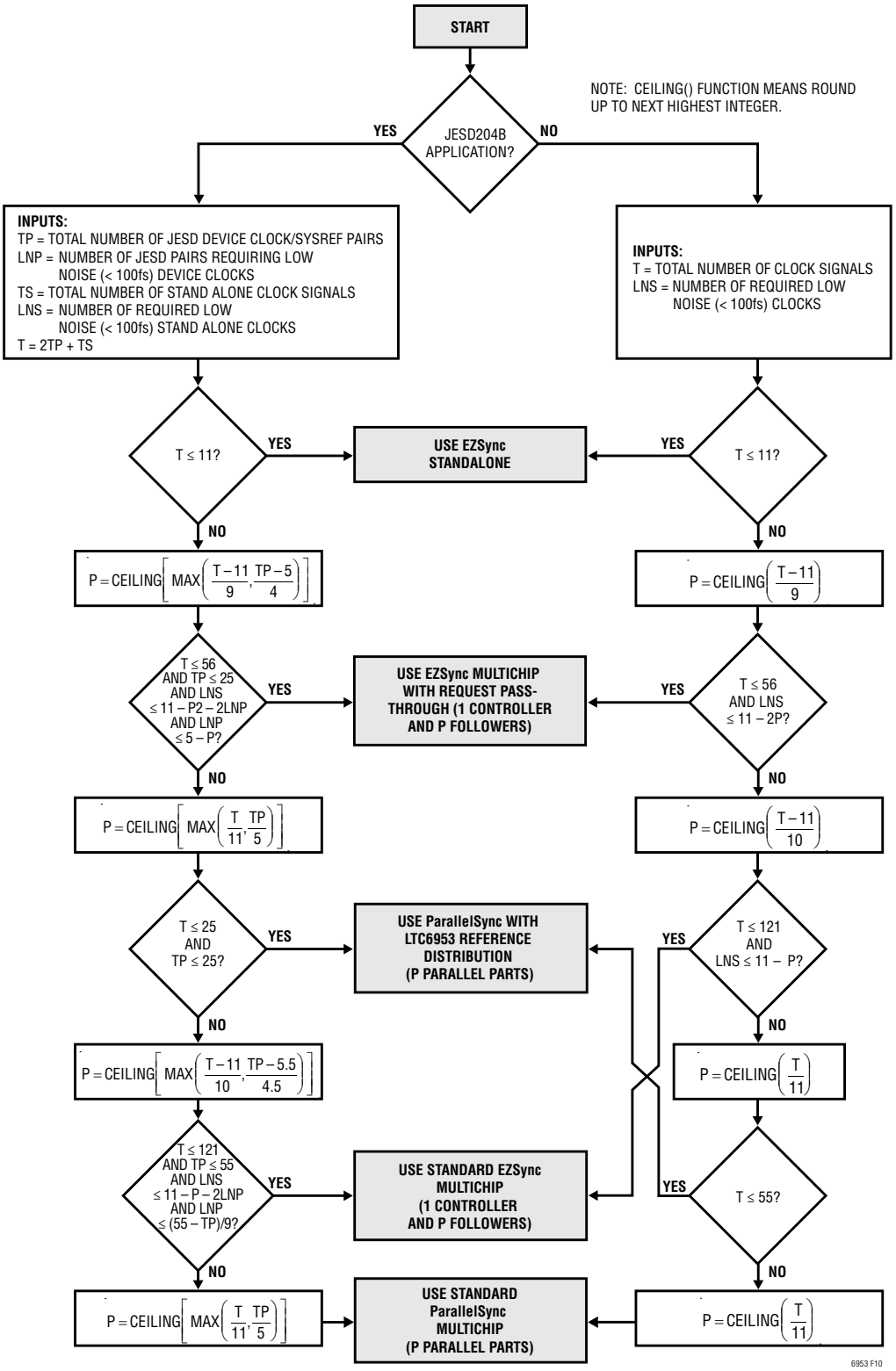


Figure 10. Flowchart to Determine the Best Synchronization Protocol for a Given Application

OPERATION

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 LTC6953’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 11 for details.

Data is read from the part during a communication burst using SDO. Readback may be multidrop (more than one LTC6953 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 LTC6953 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 12 for details.

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.

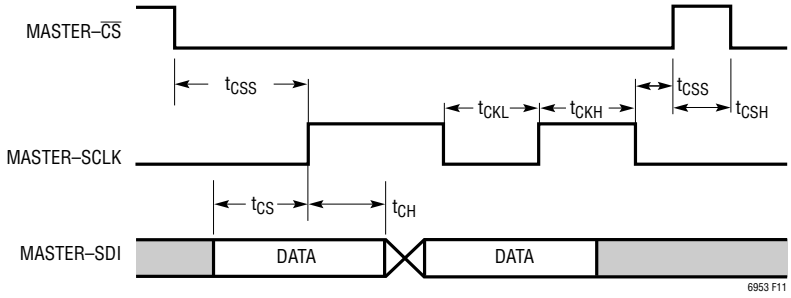


Figure 11. Serial Port Write Timing Diagram

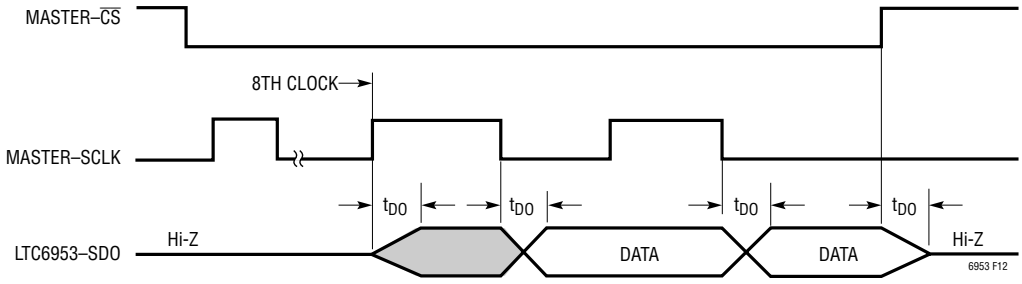


Figure 12. Serial Port Read Timing Diagram

OPERATION

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 13 for an example of a detailed write sequence and Figure 14 for a read sequence.

Figure 15 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.

Multiple Byte Transfers

More efficient data transfer of multiple bytes is accomplished by using the LTC6953's register address autoincrement feature as shown in Figure 16. The serial port master sends the destination register address in the first byte and its data in the second byte as before, but

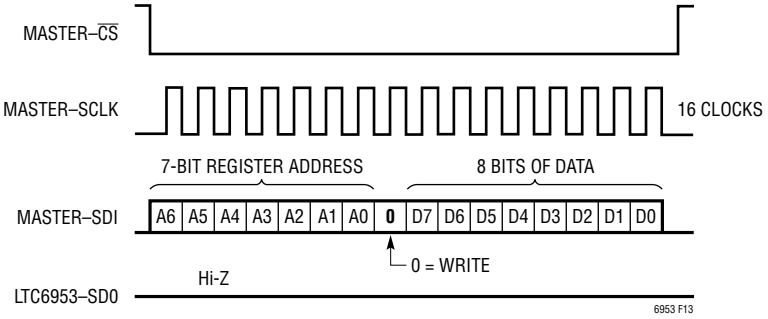


Figure 13. Serial Port Write Sequence

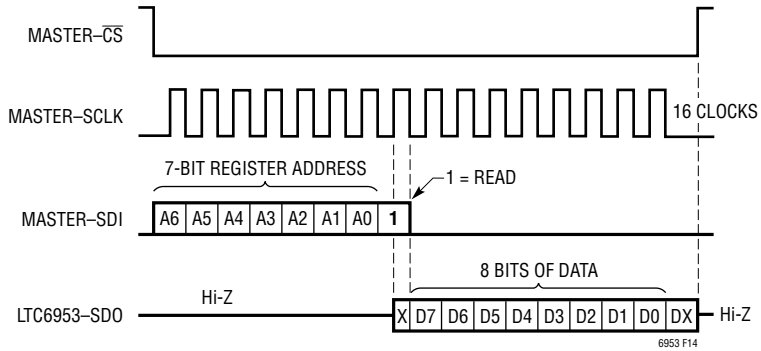


Figure 14. Serial Port Read Sequence

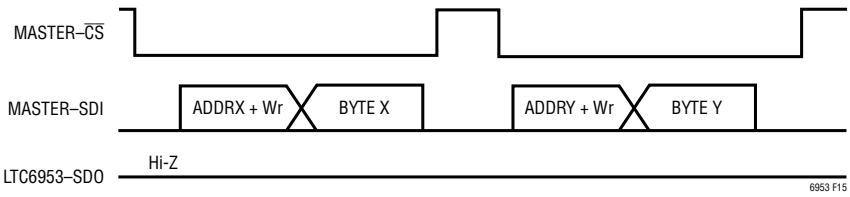


Figure 15. Serial Port Single Byte Writes

OPERATION

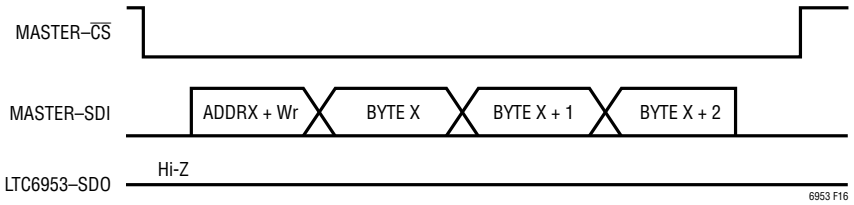


Figure 16. Serial Port Auto-Increment Write

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 17. 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 LTC6953 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.

Multidrop Configuration

Several LTC6953s 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 LTC6953 is found in Table 10, with detailed bit descriptions found in Table 11. 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 for more information.

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

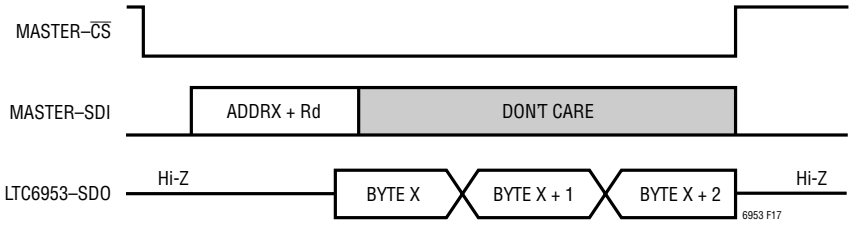


Figure 17. Serial Port Auto-Increment Read

OPERATION

Table 10. Serial Port Register Contents

ADDR	MSB	[6]	[5]	[4]	[3]	[2]	[1]	LSB	R/W	DEFAULT
h00	0	0	1	0	$\overline{\text{VCOOK}}$	VCOOK	0	1	R	
h01	INVSTAT	x[6]	x[5]	x[4]	x[3]	x[2]	x[1]	x[0]	R/W	h04
h02	PDALL	*	PDVCOPK	*	*	*	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	*	*	*	*	*	*	*	*	R/W	h0C
h07	*	*	*	*	*	*	*	*	R/W	h01
h08	*	*	*	*	*	*	*	*	R/W	h00
h09	*	*	*	*	*	*	*	*	R/W	h2D
h0A	*	*	*	*	*	*	*	*	R/W	h93
h0B	*	*	*	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
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

OPERATION

ADDR	MSB	[6]	[5]	[4]	[3]	[2]	[1]	LSB	R/W	DEFAULT
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	hx3†

* Unused

† Varies depending on revision

Table 11. Serial Port Register Bit Field Summary

BITS	DESCRIPTION	DEFAULT	ADDR
ADELx[5:0]	OUT0 Analog Delay Setting	0	Various
DDELx[11:0]	OUTx Delay in ½ Input Cycles	h000	Various
EZMD	EZSync Mode	0	h0B
FILTV	Input Buffer Filter	0	h02
INVSTAT	Invert STAT Output	0	h01
MDx[2:0]	OUTx 2^N Value	0	Various
MODEx[1:0]	SYSREF Mode (SRQENx = 1): 0 = Free Run 1 = Gated Pulses 2 = Request Pass-Through 3 = 2^{SYSCT} Pulses	0	Various
MPx[4:0]	OUTx Prescaler Value	h0	Various
OINVx	OUTx Inversion	0	Various
PART[3:0]	Part Code (h2 = 6952, h3 = 6953)		h38
PDALL	Full Chip Power-Down	0	h02
PDVCOPK	Powers Down Input Signal Detector	0	h02
PDx[1:0]	OUTx Power-Down Mode: 0 = Normal Operation 1 = Output Muted to Logic “0” 2 = Power-Down Output Driver 3 = Power-Down Full Divider	h0	Various
POR	Force Power-On-Reset	0	h02

BITS	DESCRIPTION	DEFAULT	ADDR
REV[3:0]	Revision Code		h38
SRQENx	Enable SYNC or SYSREF on OUTx	0	Various
SRQMD	0 = Synchronization Mode 1 = SYSREF Request Mode	0	h0B
SSRQ	Software SYNC or SYSREF Request	0	h0B
SYSCT[1:0]	SYSREF Pulse Count for MODEx = 3: 0 = One Pulse 1 = Two Pulses 2 = Four Pulses 3 = Eight Pulses	h3	h0B
TEMPO	Enable temperature Measurement Diode on STAT		h05
VCOOK	Input Valid Flag		h00
$\overline{\text{VCOOK}}$	Input Not Valid Flag		h00
x[6:0]	STAT Output OR Mask	h04	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 18. The result of this bit-wise boolean operation is then output on the STAT pin if TEMP0 is "0".

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

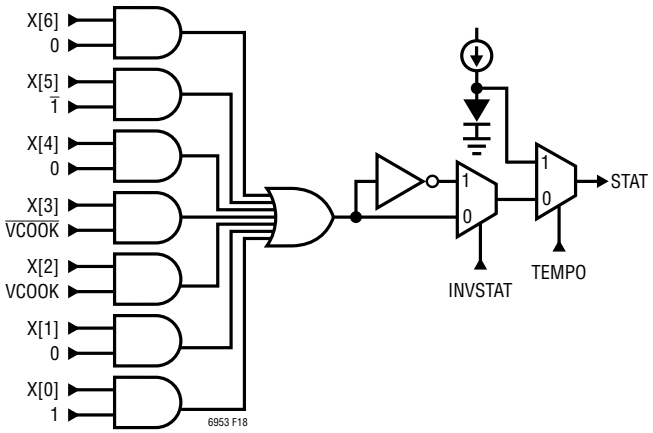


Figure 18. STAT Simplified Schematic

For example, if the application requires STAT to go high whenever the VCOOK flag is set, then x[2] should be set to "1" and INVSTAT should be set to "0", giving a register value of h04. Since only VCOOK and $\overline{VCOOK}$ are used for the LTC6953, only bits x[3:2] should be used. Bits x[6:4] and x[1:0] should always be set to 0.

The STAT pin may be transformed to a temperature measurement 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 LTC6953 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 (V_{TEMP}) and using Equation 3:

$$T = 665 \bullet (V_{TEMPC} - V_{TEMP}) + T_{CAL} \quad (3)$$

where T and T_{CAL} are in ºC, and V_{TEMPC} and V_{TEMP} are in V. Note that no external bias current is required. Allow 50µs settling time after setting TEMPO to "1".

Block Power-Down Control

The LTC6953's power-down control bits are located in register h02, described in Table 11. Different portions of the device may be powered down independently. To power down individual outputs, see Table 3. *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

The purpose of a clock distributor is to take an incoming clock signal of frequency f_{IN} and produce multiple new clock signals at the same frequency or some other frequency value divided down from the input frequency. Each output clock can have its phase individually adjusted relative to the other clocks through the synchronization process. Figure 19 shows a typical application of the LTC6953.

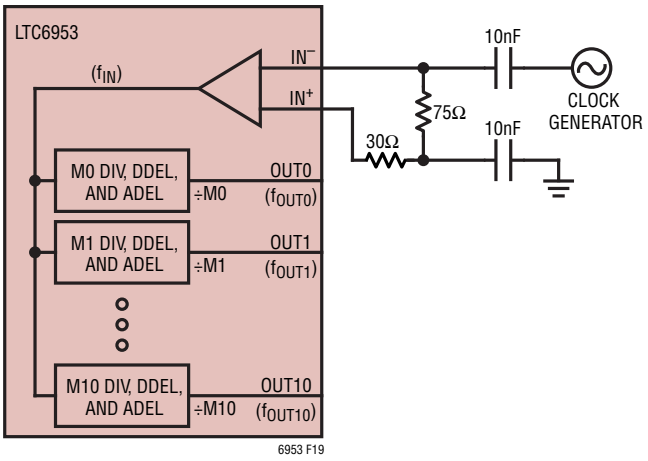


Figure 19. LTC6953 Typical Application

OUTPUT FREQUENCY

For a given input frequency f_{IN}, the output frequency f_{OUTx} produced at the output of the Mx dividers is given by Equation 4:

$$f_{OUTx} = \frac{f_{IN}}{Mx} \quad (4)$$

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 input cycles. Applications needing to calculate the delay in terms of time can use Equation 5 where DDELx is DDEL0 to DDEL10:

$$t_{DDELx} = \frac{D_{DELx}}{(2 \bullet f_{IN})} \quad (5)$$

The analog delay blocks (ADELx) are useful in trimming signal timing differences caused by non-ideal PCB routing. This is effective for optimizing setup and hold times for SYSREFs versus device clocks in JESD204B 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 Equations 6 (for ADELx < 32) and Equation 7 (for ADELx ≥ 32) while adhering to the frequency limitations described in Table 4.

ADELx = 1 to 31:

$$t_{ADEL} = [(11.25 \bullet ADELx + 93.8)^{-2.5} + (0.00285 \bullet f_{OUT})^{2.5}]^{-0.4} \quad (6)$$

ADELx = 32 to 63:

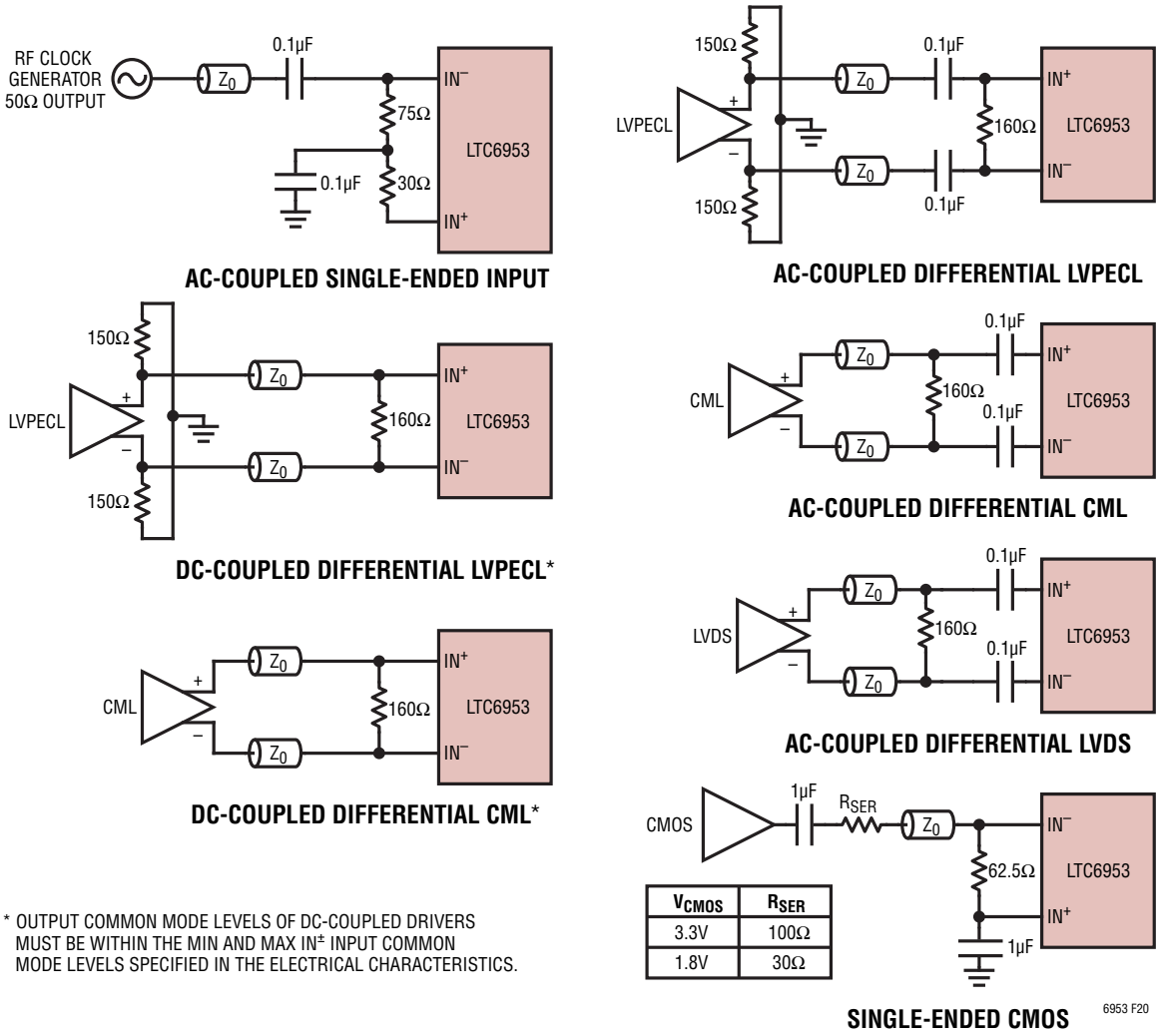
$$t_{ADEL} = [(26 \bullet ADELx - 517)^{-2.5} + (0.00125 \bullet f_{OUT})^{2.5}]^{-0.4} \quad (7)$$

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

APPLICATIONS INFORMATION

INPUT BUFFER

The LTC6953's input buffer, shown in Figure 1, 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 20 shows recommended interfaces for different input signal types.



V_{CMOS}	R_{SER}
3.3V	100Ω
1.8V	30Ω

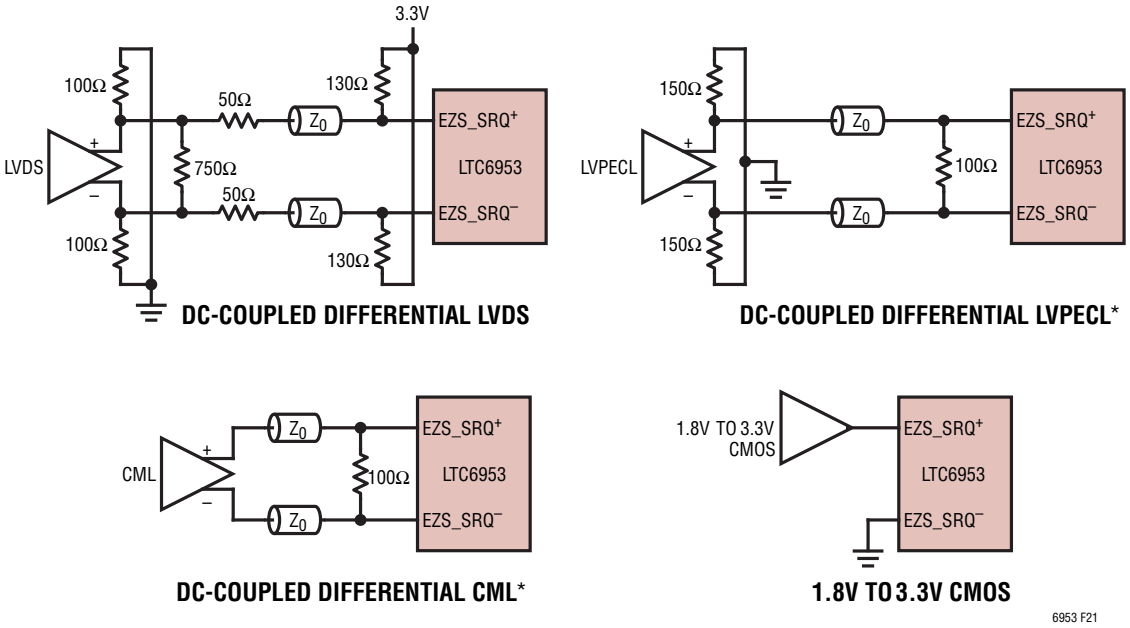
* OUTPUT COMMON MODE LEVELS OF DC-COUPLED DRIVERS MUST BE WITHIN THE MIN AND MAX IN± INPUT COMMON MODE LEVELS SPECIFIED IN THE ELECTRICAL CHARACTERISTICS.

Figure 20. Common Input Interface Configurations. All Z_0 Signal Traces Are 50Ω Transmission Lines

APPLICATIONS INFORMATION

EZS_SRQ INPUT

The LTC6953's EZS_SRQ input buffer, shown in Figure 4, 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 21 shows the recommended interface types.



* 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 21. Common EZS_SRQ Input Interface Configurations. All Z_0 Signal Traces Are 50Ω Transmission Lines

APPLICATIONS INFORMATION

JESD204B DESIGN EXAMPLE USING EZSync STANDALONE

This design example consists of a system of two JESD204B analog-to-digital converters (ADCs), two JESD204B digital-to-analog converters (DACs) and a JESD204B compatible FPGA. All of the data converters (ADCs and DACs) and the FPGA require JESD204B 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:

$f_{ADC-CLK}$ = 500MHz

$f_{DAC-CLK}$ = 4000MHz

$f_{FPGA-CLK}$ = 125MHz

$f_{FPGA-MGMT}$ = 100MHz

f_{SYSREF} = 12.5MHz

Since the total number of outputs is 11, a single LTC6953 can be used to generate all of the outputs needed as shown in Figure 22. Note that termination resistors and AC-coupling caps are not shown for clarity.

Input Assumptions

For this example, assume the input is driven by an external clock generator with a frequency of 4000MHz.

f_{IN} = 4000MHz

Design Procedure

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

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

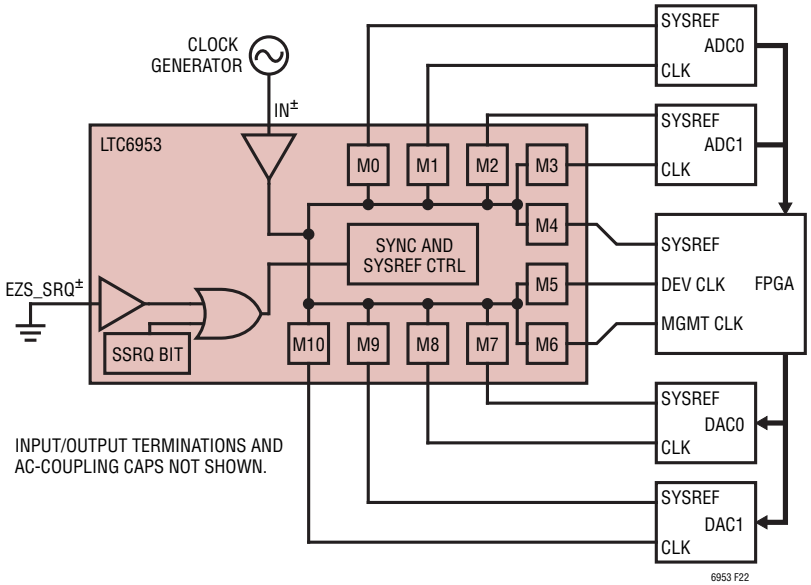


Figure 22. Block Diagram for JESD204B EZSync Standalone Design Example

APPLICATIONS INFORMATION

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 Output Modes

All outputs can be programmed as clocks (MODEx = 0), SYSREFs (MODEx = 1 or 3) or SYNC/SRQ pass-through outputs (MODEx = 2) using each output's individual MODEx bits as described in Table 5 and Table 6. 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 12 summarizes each output's mode settings.

Table 12. 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 4. The results are shown in Table 13.

Table 13. 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

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 input 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 23 for an example.

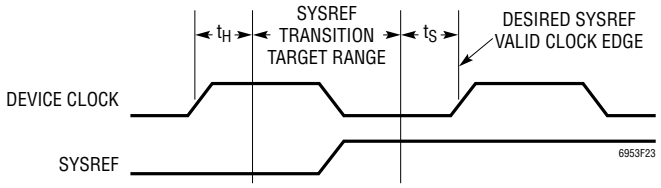


Figure 23. SYSREF Edge Timing Example

APPLICATIONS INFORMATION

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

1. Delay all of the JESD204B device clocks by half of a period of the *slowest* JESD204B 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 an input cycle. Non-JESD204B 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 clock is the FPGA device clock:

   $DDEL_{SYSvalid} = MFPG_{ACLK} = 32$

   $DDEL_{ADC–CLK} = DDEL_{SYSvalid} = 32$

   $DDEL_{DAC–CLK} = DDEL_{SYSvalid} = 32$

   $DDEL_{FPGA–CLK} = DDEL_{SYSvalid} = 32$

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

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

   $DDELA_{DC–SYS} = 32 – 8 = 24$

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

   $DDEL_{DAC–SYS} = 32 – 1 = 31$

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

   $DDEL_{FPGA–SYS} = 32 – 32 = 0$

Table 14 summarizes the DDEL settings for all outputs.

Table 14. 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 LTC6953 can be programmed.

Status Register Programming

This example will use the STAT pin to alert the system whenever the LTC6953 generates a fault condition. Program x[3] = 1 to force the STAT pin high whenever the $\overline{VCOOK}$ flag asserts:

Reg01 = h08

Power and FILT Register Programming

For correct operation, all internal blocks should be enabled. Additionally, the input signal has a sufficient slew rate and power to not need the FILTV bit:

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

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). Using this information, register h0B can be programmed:

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

APPLICATIONS INFORMATION

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.

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 15, calculated using the information in Tables 20, 21 (with Equation 1) and 22.

Table 15. 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 24. 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 LTC6953 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 LTC6953 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 25.

APPLICATIONS INFORMATION

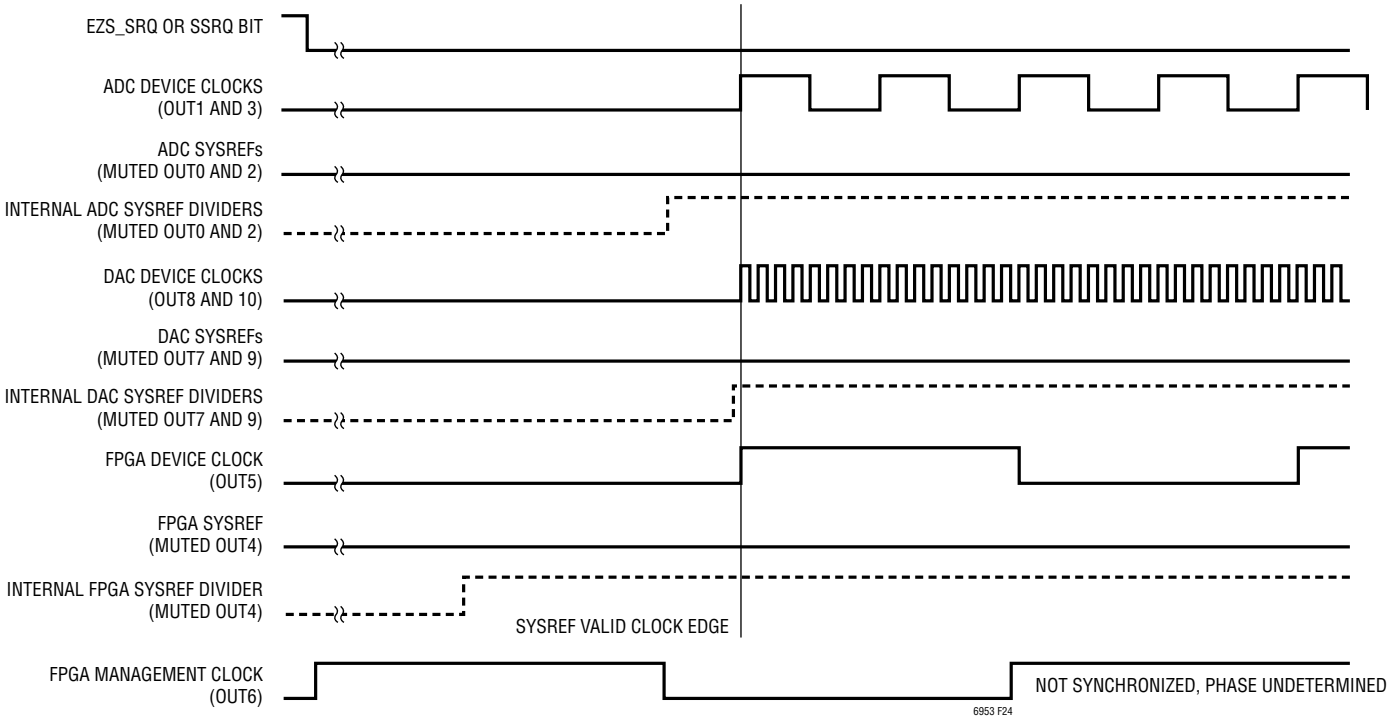


Figure 24. Outputs after Synchronization for the EZSync Standalone Design Example (SRQMD = 0)

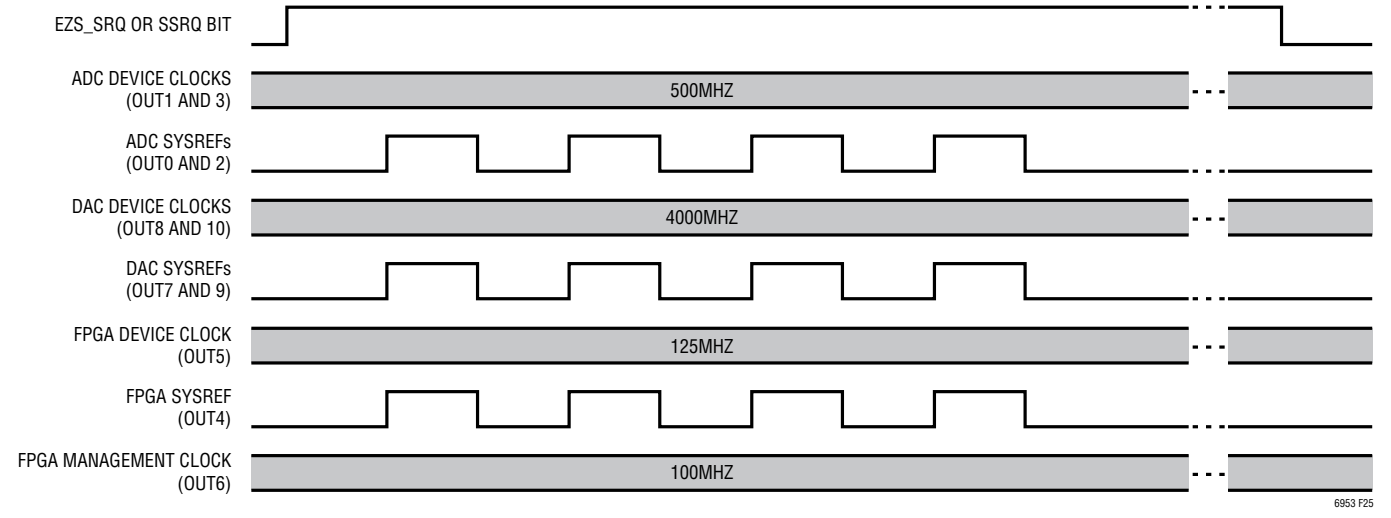


Figure 25. Outputs after SYSREF Request for the EZSync Standalone Design Example (SRQMD = 1)

APPLICATIONS INFORMATION

JESD204B DESIGN EXAMPLE USING EZSync MULTICHIP

This design example consists of a system of four JESD204B analog-to-digital converters (ADCs), four JESD204B digital-to-analog converters (DACs), and a JESD204B compatible FPGA. All of the data converters (ADCs and DACs) and the FPGA require JESD204B 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 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

To determine which multichip configuration to use, we utilize the flowchart in Figure 10. This example has nine total JESD204B 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 10 suggests using the EZSync multichip protocol with request pass-through topology shown in Figure 7, using one CONTROLLER and one FOLLOWER chip. Noting that the use of LTC6953 for any FOLLOWER chips is recommended, Figure 26

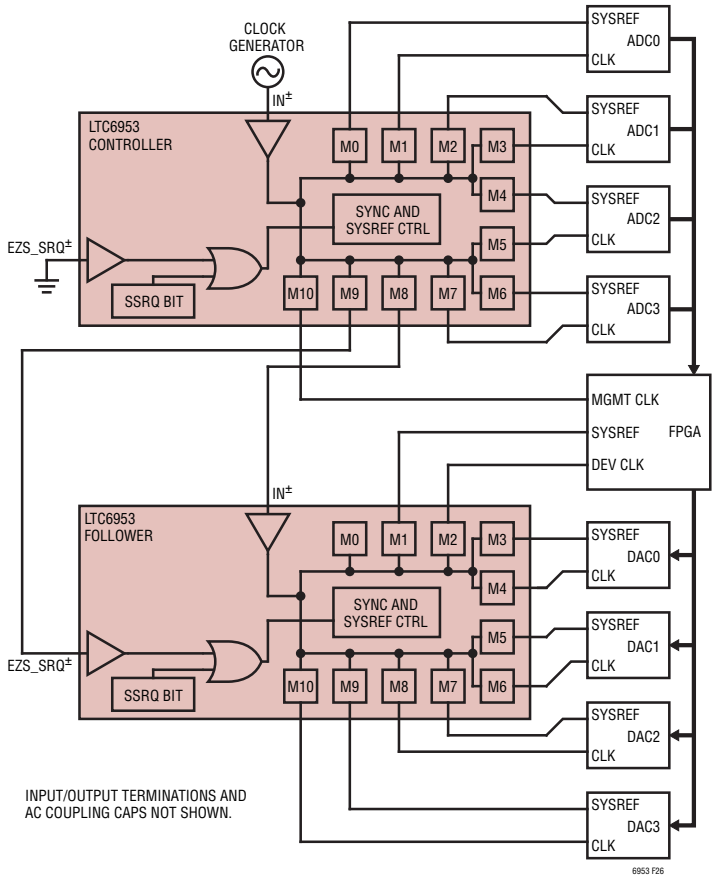


Figure 26. Block Diagram for JESD204B EZSync Multichip Design Example

APPLICATIONS INFORMATION

shows a block diagram of the full system. OUT8 of the CONTROLLER LTC6953 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 pass-through output. Also notice that the CONTROLLER clock outputs are the lowest jitter clocks, and should therefore be used to drive the ADCs.

Input Assumptions

For this example, assume the input to the CONTROLLER LTC6953 is driven by an external clock generator with a frequency of 4000MHz.

f_{IN} = 4000MHz

Design Procedure

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

1. Determine all output modes for CONTROLLER and FOLLOWER.
2. Determine all M divider values.
3. Determine all digital delay values.
4. Program the ICs with the correct divider values, output delays, and other settings.
5. Synchronize the outputs.
6. Place the SYSREF outputs in a lower power mode until the next SYSREF request (optional, see Operations section).
7. Place ICs into SYSREF request mode and send a SYSREF request when needed.
8. Return the 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 Output Modes

All outputs can be programmed as clocks (MODEx = 0), SYSREFs (MODEx = 1 or 3) or SYNC/SRQ pass-through outputs (MODEx = 2) using each output's individual MODEx bits as described in Table 5 and Table 6. 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 16 summarizes each output's mode settings.

Table 16. Output Mode Settings for EZSync Multichip 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	To FOLLOWER IN±	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

APPLICATIONS INFORMATION

known, which leaves the two CONTROLLER outputs that drive the FOLLOWER to be described. Since OUT8 of the CONTROLLER drives the FOLLOWER 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 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).

$f_{CONT–OUT9}$ = 12.5MHz

Now that all frequencies are known, use Equation 4 to determine the output divider values. The results are shown in Table 17.

Table 17. Output Divide Settings for EZSync Multichip 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	To FOLLOWER IN±	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 input 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. Refer to Figure 23 for an example.

For EZSync multichip synchronization, the CONTROLLER output which drives the FOLLOWER 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 ($DDEL_{FS\text{-}OS}$):

$$DDEL_{FS–OS} = 14 \bullet M_{FD} + DDEL_{FD} \quad (8)$$

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 device clocks by half of a period of the *slowest* JESD204B 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 an input clock cycle. Non-JESD204B 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 clock is the FPGA device clock:

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

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

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

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

APPLICATIONS INFORMATION

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:

   $DDEL_{ADC–SYS}' = DDEL_{SYSvalid} – M_{ADC–CLK}$

   $DDEL_{ADC–SYS}' = 32 – 8 = 24$

   $DDEL_{DAC–SYS}' = DDEL_{SYSvalid} – M_{DAC–CLK}$

   $DDEL_{DAC–SYS}' = 32 – 1 = 31$

   $DDEL_{FPGA–SYS}' = DDEL_{SYSvalid} – M_{FPGA–CLK}$

   $DDEL_{FPGA–SYS}' = 32 – 32 = 0$

3. Adjust for CONTROLLER vs FOLLOWER outputs. Any CONTROLLER output that is synchronous with the FOLLOWER must have the delay offset from Equation 8 added to its DDEL value. FOLLOWER outputs need no adjustment. For this example, the ADC CLKs and SYSREFs come from the CONTROLLER:

   $DDEL_{ADC–CLK} = DDEL_{ADC–CLK}' + DDEL_{FS–OS}$

   $DDEL_{ADC–CLK} = 32 + 14 = 46$

   $DDEL_{ADC–SYS} = DDEL_{ADC–SYS}' + DDEL_{FS–OS}$

   $DDEL_{ADC–SYS} = 24 + 14 = 38$

   $DDEL_{DAC–CLK} = DDEL_{DAC–CLK}' + 0 = 32$

   $DDEL_{DAC–SYS} = DDEL_{DAC–SYS}' + 0 = 31$

   $DDEL_{FPGA–CLK} = DDEL_{FPGA–CLK}' + 0 = 32$

   $DDEL_{FPGA–SYS} = DDEL_{FPGA–SYS}' + 0 = 0$

   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 8 will need to be added as well since it is located on the CONTROLLER part. The latest occurring SYSREF is the DAC SYSREF, therefore:

   $DDEL_{OUT9} = DDEL_{DAC–SYS} + DDEL_{FS–OS}$

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

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

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

Table 18 summarizes the DDEL settings for all outputs.

Table 18. O]utput DDELx Settings for EZSync Multichip 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 Input	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 LTC6953 generates a fault condition. For the CONTROLLER and FOLLOWER, program x[3] = 1 to force the STAT pin high whenever the $\overline{VCOOK}$ flag asserts:

CONTROLLER Reg01 = h08

FOLLOWER Reg01 = h08

Power and FILT Register Programming

For correct operation on both CONTROLLER and FOLLOWER, all internal blocks should be enabled.

APPLICATIONS INFORMATION

Additionally, the input signal has a sufficient slew rate and power to not need the FILTV bit:

CONTROLLER Reg02 = h00

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

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). Using this information, register h0B for CONTROLLER and FOLLOWER can be programmed:

CONTROLLER Reg0B = h04

FOLLOWER Reg0B = h14

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 cross-overs, 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 19 and Table 20, calculated using the information in Table 24, Table 25 (with Equation 1), and Table 26.

Table 19. CONTROLLER Output Register Settings for EZSync Multichip 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		

APPLICATIONS INFORMATION

Table 20. FOLLOWER Output Register Settings for EZSync Multichip 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 27. 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.

Putting the ICs Into a Lower Power Mode

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 28. 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.

APPLICATIONS INFORMATION

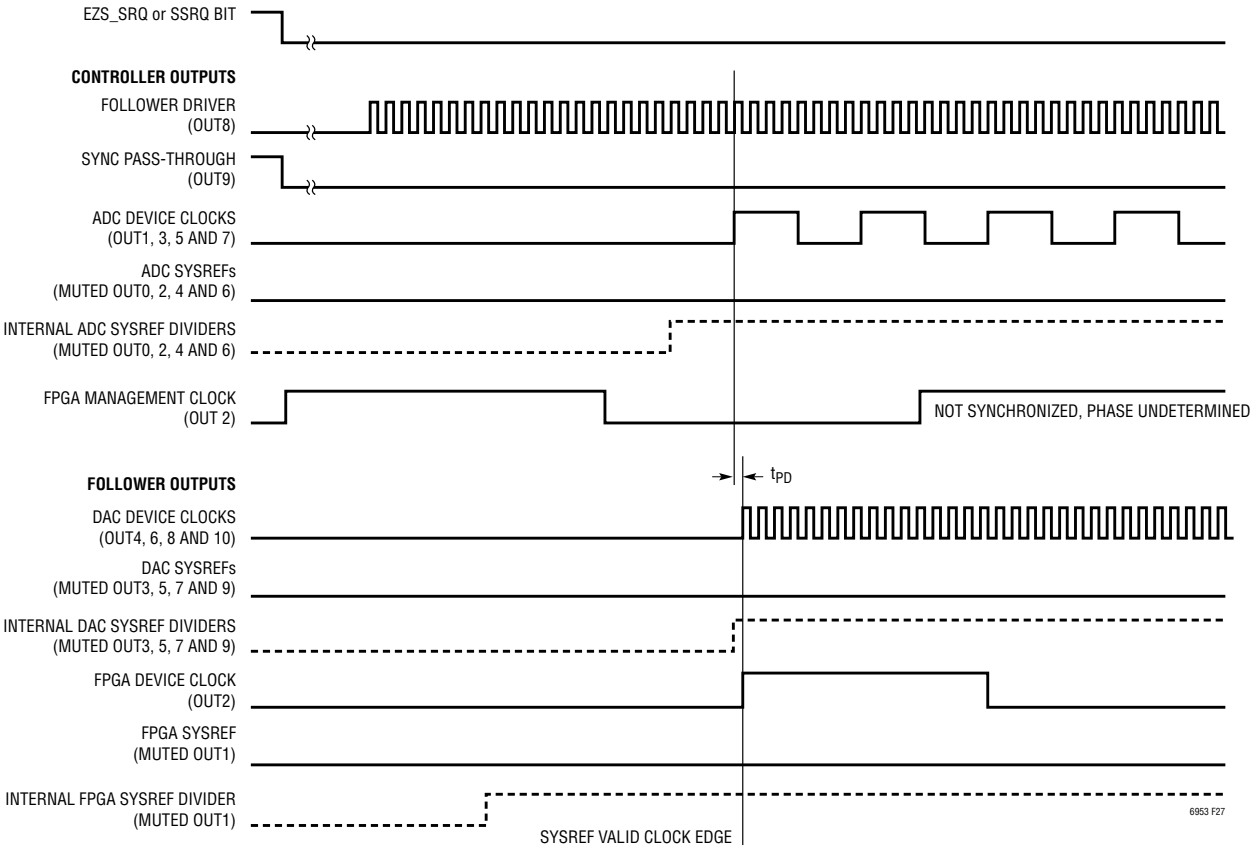


Figure 27. Outputs After Synchronization for the EZSync Multichip Design Example (SRQMD = 0)

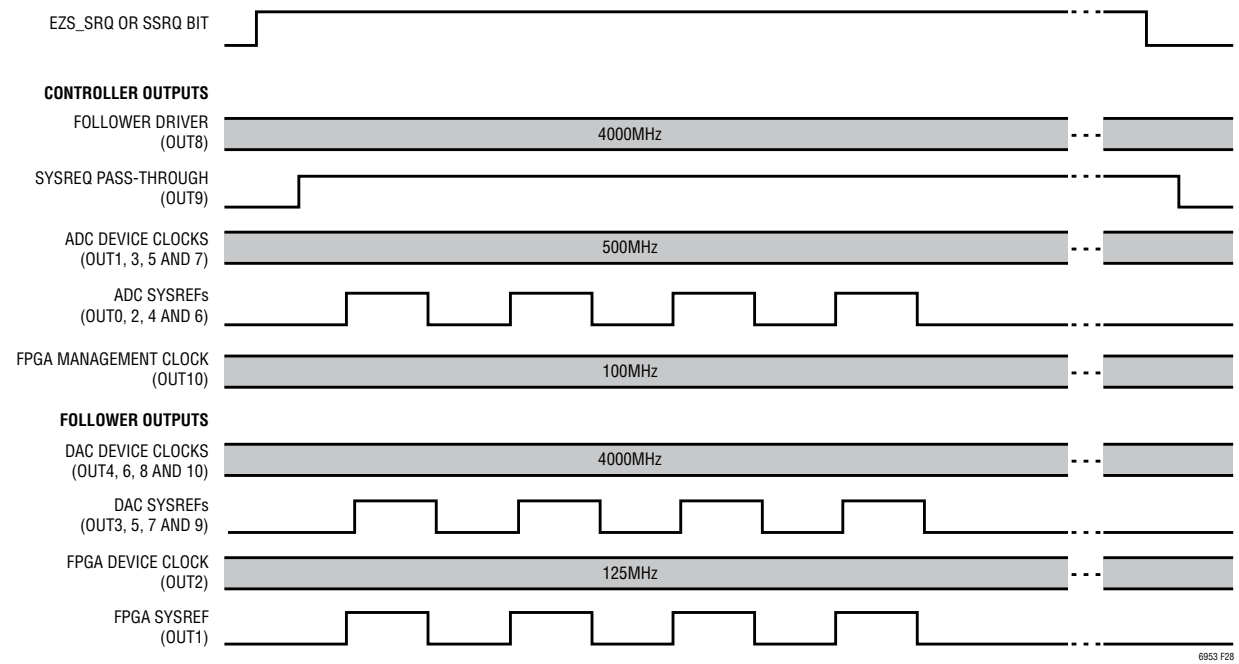


Figure 28. Outputs After SYSREF Request for the EZSync Multichip Design Example (SRQMD = 1)

APPLICATIONS INFORMATION

JESD204B DESIGN EXAMPLE USING ParallelSync

This design example consists of a system of eight JESD204B analog-to-digital converters (ADCs) and a JESD204B compatible FPGA. All of the ADCs and the FPGA require JESD204B 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 multichip configuration to use, we utilize the flowchart in Figure 10. This example has nine total JESD204B 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 10 suggests using the ParallelSync multichip protocol with LTC6953 reference distribution topology shown in Figure 9, using one LTC6953 as the reference distribution chip (REF LTC6953) and two LTC6952s in parallel to generate the clocks (LTC6952 #1 and LTC6952 #2). Figure 29 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).

Although the ParallelSync design example has been shown here for reference, most of the design work for it involves the LTC6952. Please refer to the LTC6952 data sheet for detailed instructions on programming the ICs for this example.

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 30 for an example). An example of grounding for electrical and thermal performance can be found on the DC2610 layout.

APPLICATIONS INFORMATION

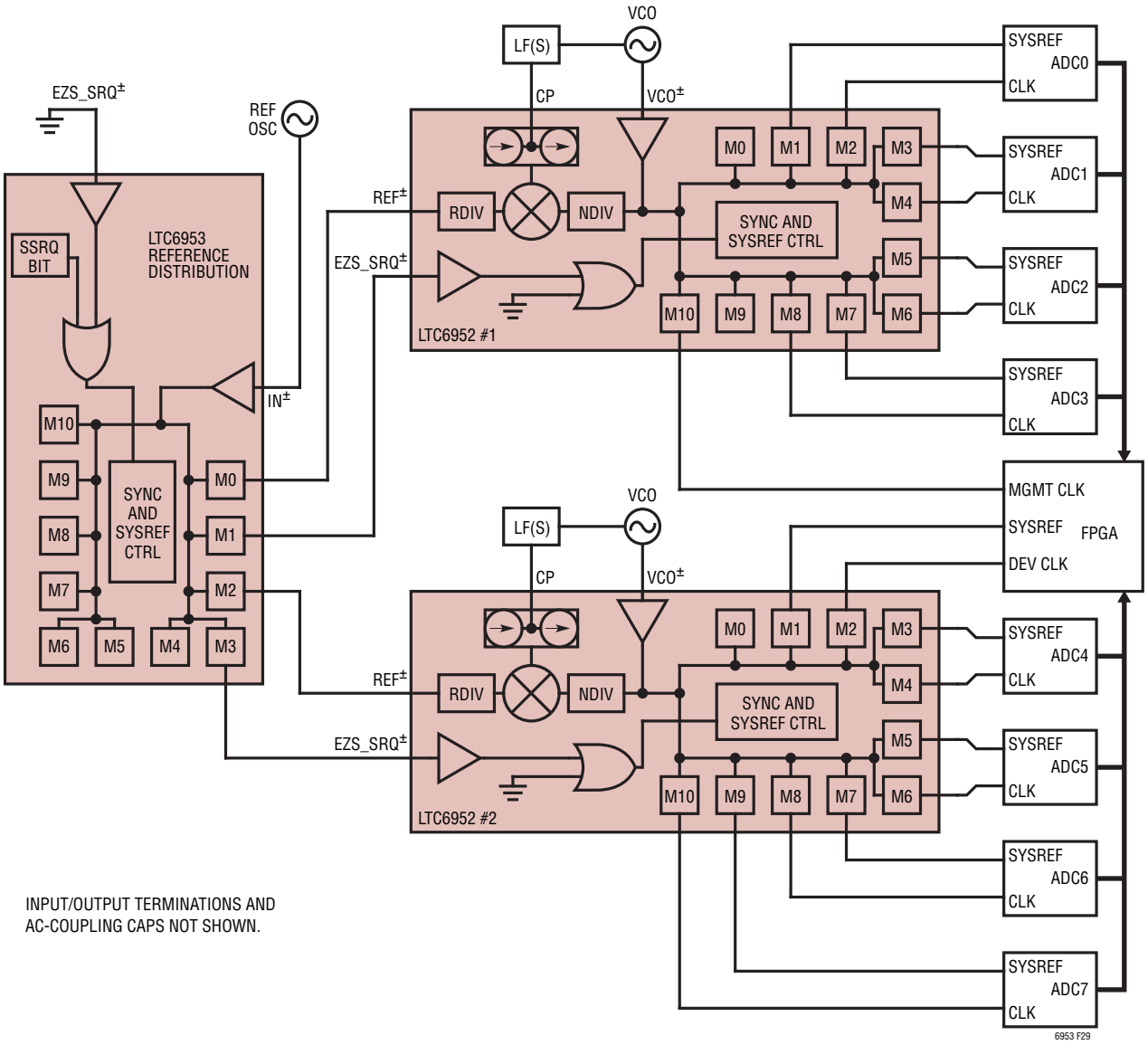


Figure 29. Block Diagram for JESD204B ParallelSync with LTC6953 Reference Distribution Design Example

APPLICATIONS INFORMATION

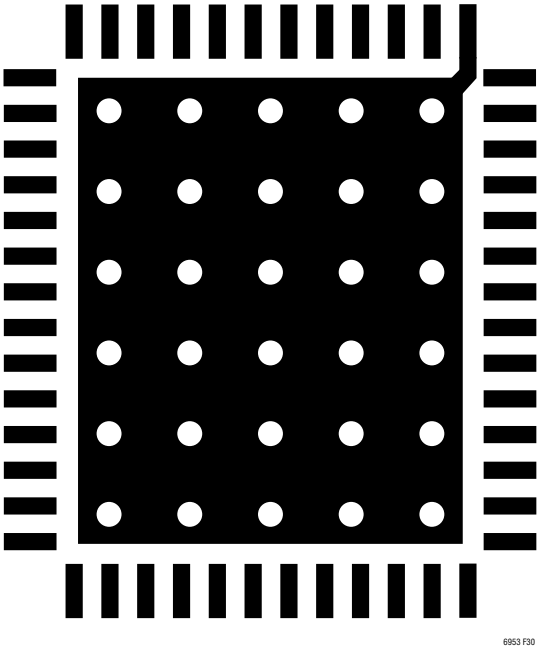


Figure 30. PCB Top Metal Layer Pin and Exposed Ground Pad Design. Pin 41 is Signal Ground and Connected Directly to the Exposed Pad Metal

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 31 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.

APPLICATIONS INFORMATION

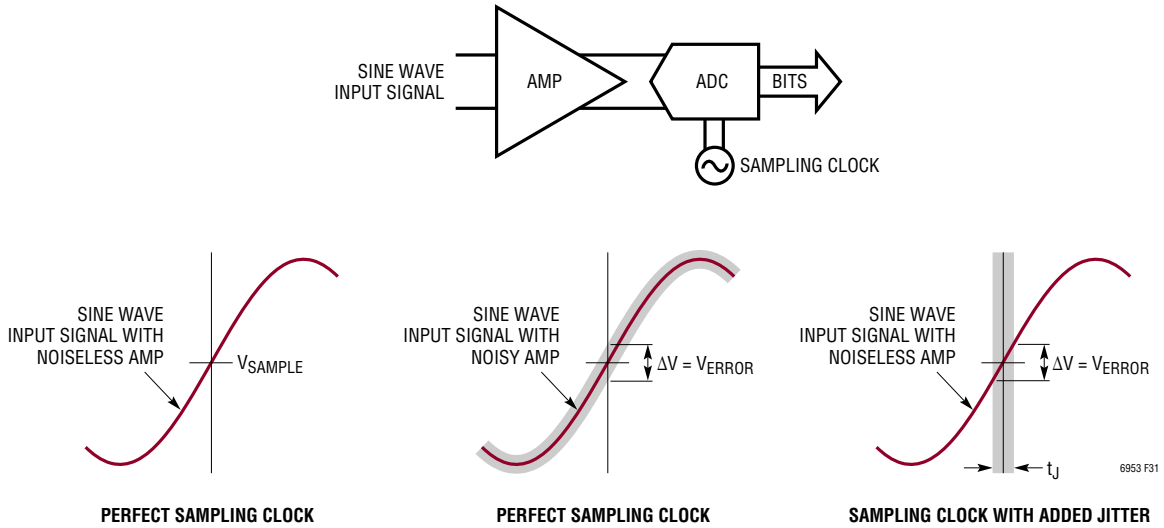


Figure 31. A Typical Data Acquisition Circuit Showing the Sampling Error Effects of a Noisy Amplifier and a Jittery Sampling Clock

Figure 32 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.

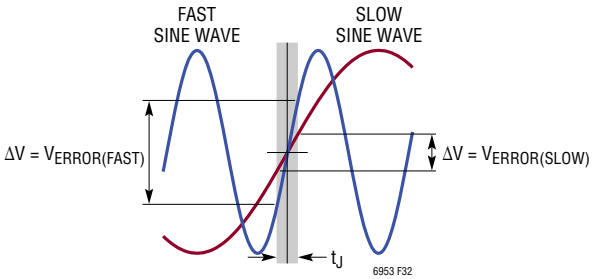


Figure 32. Fast and Slow Sine Wave Signals Sampled with a Jittery Clock

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:

$$t_{J(TOTAL)} = \frac{10^{\frac{-SNR_{dB}}{20}}}{2 \bullet \pi \bullet f_{SIG}} \quad (9)$$

Where fSIG is the highest frequency signal to be digitized expressed in Hz, SNRdB 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:

$$t_{J(TOTAL)} = \sqrt{t_{J(CLK)}^2 + t_{J(ADC)}^2} \quad (10)$$

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

$$SNR_{dB} = -20\log_{10}(2 \bullet \pi \bullet f_{SIG} \bullet t_{J(TOTAL)}) \quad (11)$$

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.

APPLICATIONS INFORMATION

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 33 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.

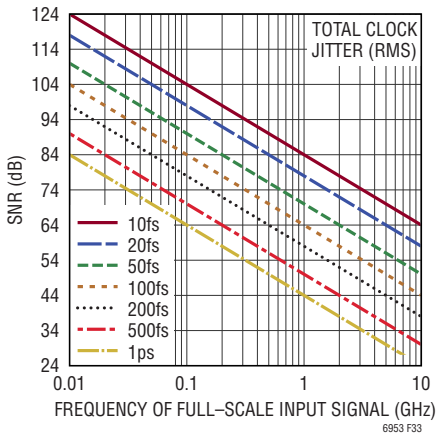


Figure 33. SNR vs Input Signal Frequency vs Sample Clock Jitter

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 12.

$$t_{J(TOTAL)} = \frac{10 - \frac{1}{2}\log_{10}\left[10^{-}\left(\frac{SNR_{jitter}}{-10}\right) - 10^{-}\left(\frac{SNR_{base}}{10}\right)\right]}{2\pi f_{IN}} \quad (12)$$

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 9.

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 34 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.

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.

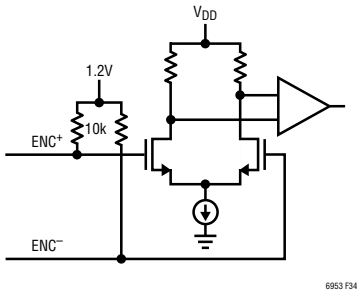


Figure 34. Simplified Sample Clock Input Circuit

APPLICATIONS INFORMATION

As shown in Figure 34, the ADC's sample clock input is typically differential, with a differential sampling clock delivering the best performance. Figure 34 also shows the sample clock input having a different common mode input voltage than the LTC6953'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 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 35 shows the preferred method of far-end termination of the transmission line.

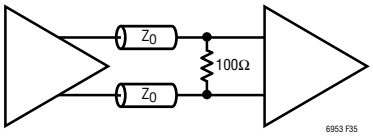


Figure 35. Far-End Transmission Line Termination (Z0 = 50Ω)

USING THE LTC6953 TO DRIVE DEVICE CLOCK INPUTS

The LTC6953's CML outputs are designed to interface with standard CML or LVPECL devices while driving transmission lines with far-end termination. Figure 36 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.

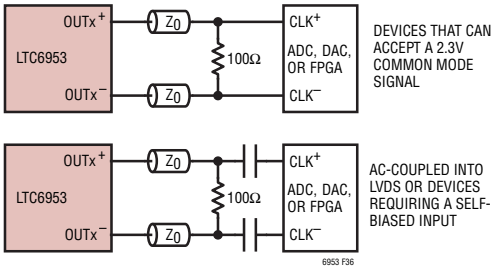


Figure 36. OUTx CML Connections to Device Clock Inputs (Z0 = 50Ω)

USING THE LTC6953 TO DRIVE DC-COUPLED SYSREF INPUTS

For JESD204B applications, the SYSREF signal would ideally be DC-coupled from the LTC6953 to the data converter or FPGA as shown in Figure 37. 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.

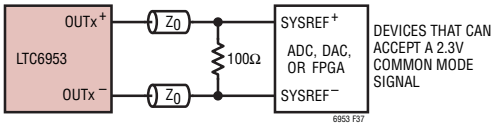


Figure 37. OUTx CML DC-Coupled Connections to SYSREF Inputs

Use the following procedure to achieve correct JESD204B 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 LTC6953 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".

APPLICATIONS INFORMATION

4. After the SYSREFs have been accepted by the receiver device, set the device to stop accepting SYSREFs.
5. Disable the LTC6953 SYSREF output drivers by setting PDx = 2 and set SRQMD = 0.

USING THE LTC6953 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 38 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.

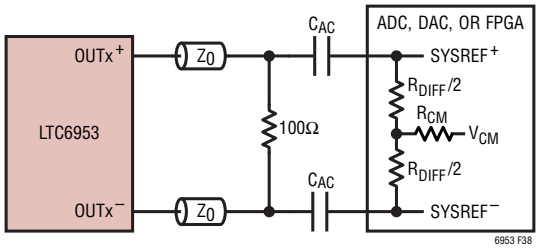


Figure 38. OUTx CML AC Coupled Connections to SYSREF Inputs for Continuous or Gated Mode Operation

Settling time for continuous or gated SYSREF connections is determined by the AC-coupling capacitors (CAC), and both the differential and common mode input resistances of the receiver device (RDIFF and RCM):

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

Use the following procedure to achieve correct JESD204B 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 LTC6953 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 LTC6953 SYSREF output drivers by setting PDx = 2 and set SRQMD = 0.

USING THE LTC6953 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 39 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.*

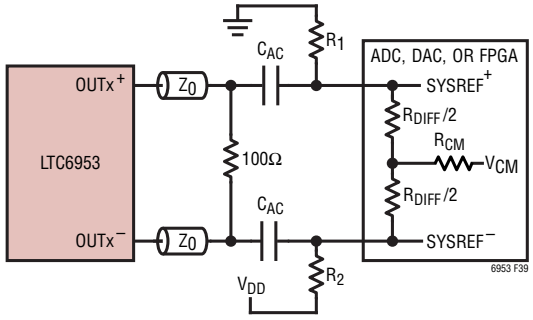


Figure 39. OUTx CML AC Coupled Connections to SYSREF Inputs for Pulsed Mode Operation

The purpose of R_1 and R_2 in Figure 39 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 (V_{DD}) and the receiver device’s input common mode voltage (VCM)

APPLICATIONS INFORMATION

and differential input resistance (R_{DIFF}). Use Equation 14 to calculate R_1 and Equation 15 to calculate R_2.

$$R_1 = R_{DIFF} \bullet \left[\frac{V_{CM}}{0.44 - 0.5}\right] \quad (14)$$

$$R_2 = R_{DIFF} \bullet \left[\frac{V_{DD} - V_{CM}}{0.44 - 0.5}\right] \quad (15)$$

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 39) is *approximately* determined by the AC-coupling capacitors (CAC), both the differential and common mode input resistance of the receiver device (R_{DIFF} and RCM), and resistors R_1 and R_2:

$$t_{settleP} \cong 10 \bullet \left(\frac{R_{DEV} \bullet R_{OS}}{R_{DEV} + R_{OS}}\right) \bullet C_{AC} \quad (16)$$

where:

$$R_{DEV} = \frac{2R_{CM} + R_{DIFF}}{2}$$
$$R_{OS} = MINIMUM(R_1, R_2) \quad (17)$$

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 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 LTC6953 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 LTC6953 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 LTC6953 as the clock generator and an LTC6955 as the limiter is shown in Figure 40.

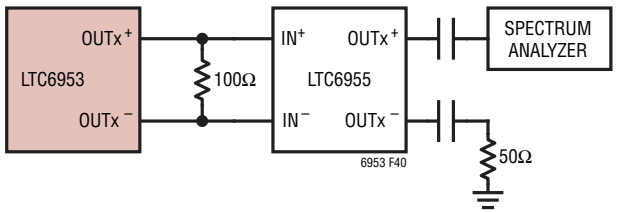
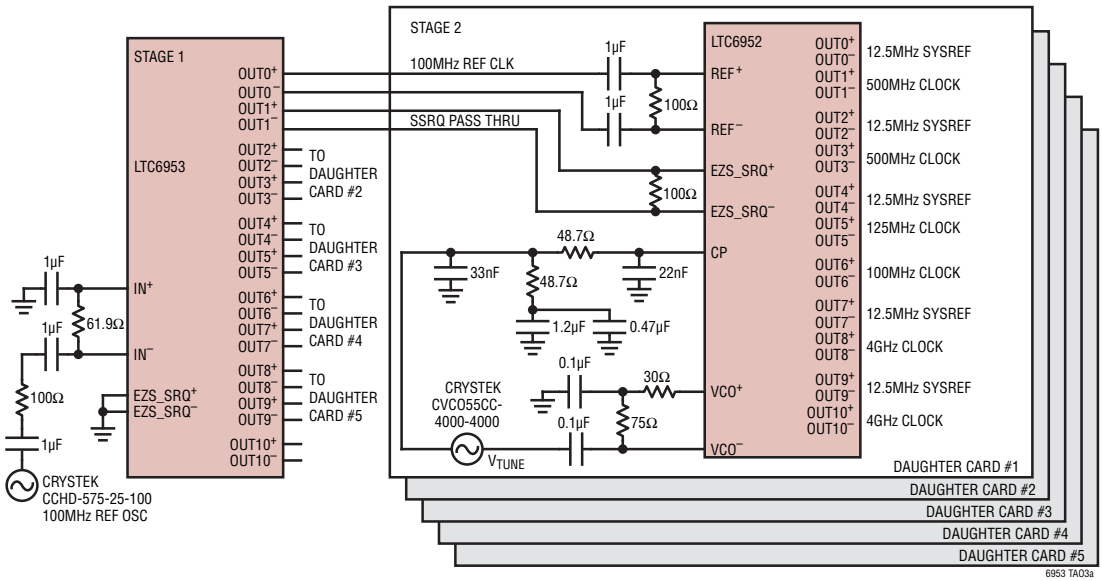


Figure 40. Example of Spurious Measurement Technique

TYPICAL APPLICATIONS

ParallelSync Multichip Synchronization with Request Pass-Through



INITIAL SETUP: PROGRAM LTC6952 AND LTC6953 REGISTERS SETTINGS CREATED FROM THE LTC6952Wizard.

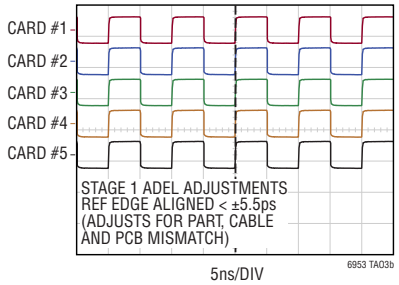
STEP 1: SYNCHRONIZE STAGE 1 REFERENCE SIGNALS
A) EZSync: TOGGLE STAGE 1 LTC6953 SSRQ BIT
B) OPT: FINE ALIGNMENT, ADJUST STAGE 1 ADEL BITS

STEP 2: SYNCHRONIZE STAGE 2 OUTPUT SIGNALS
A) SET STAGE 1 LTC6953 SRQMD = 1
B) ParallelSync, TOGGLE STAGE 1 LTC6953 SSRQ BIT

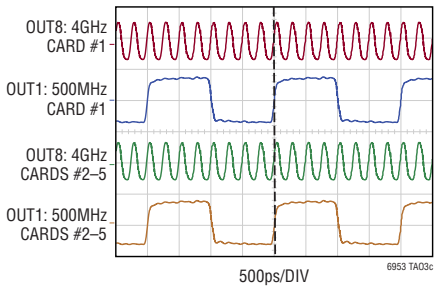
STEP 3: SEND SYSREF REQUEST
A) POWER UP LTC6952 SYSREF OUTPUTS
B) SET LTC6952 SRQMD = 1
C) SEND SYSREF, TOGGLE STAGE 1 LTC6953 SSRQ BIT

STEP 4: OPTIONAL REDUCE POWER
A) POWER DOWN LTC6952 SYSREF OUTPUTS
B) SET STAGE 1 LTC6953 AND LTC6952 SRQMD = 0

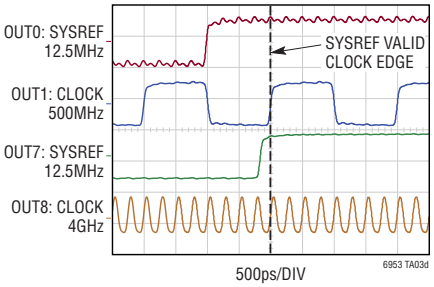
Step 1: Reference Alignment at Daughter Card Inputs



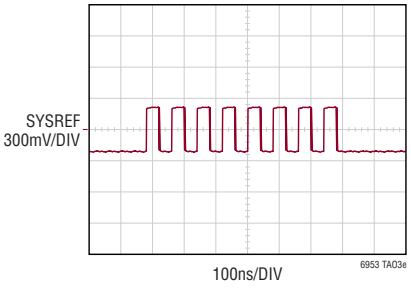
Step 2: ParallelSync Multichip Clock Alignment



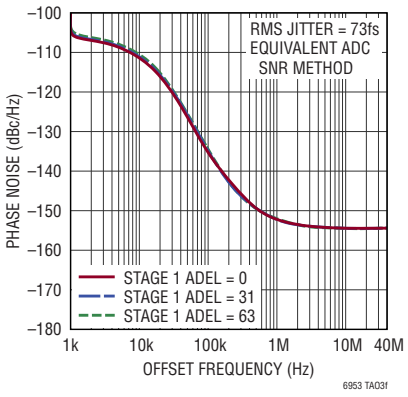
Step 3: SYSREF Alignment



Step 3: SYSREF Pulses



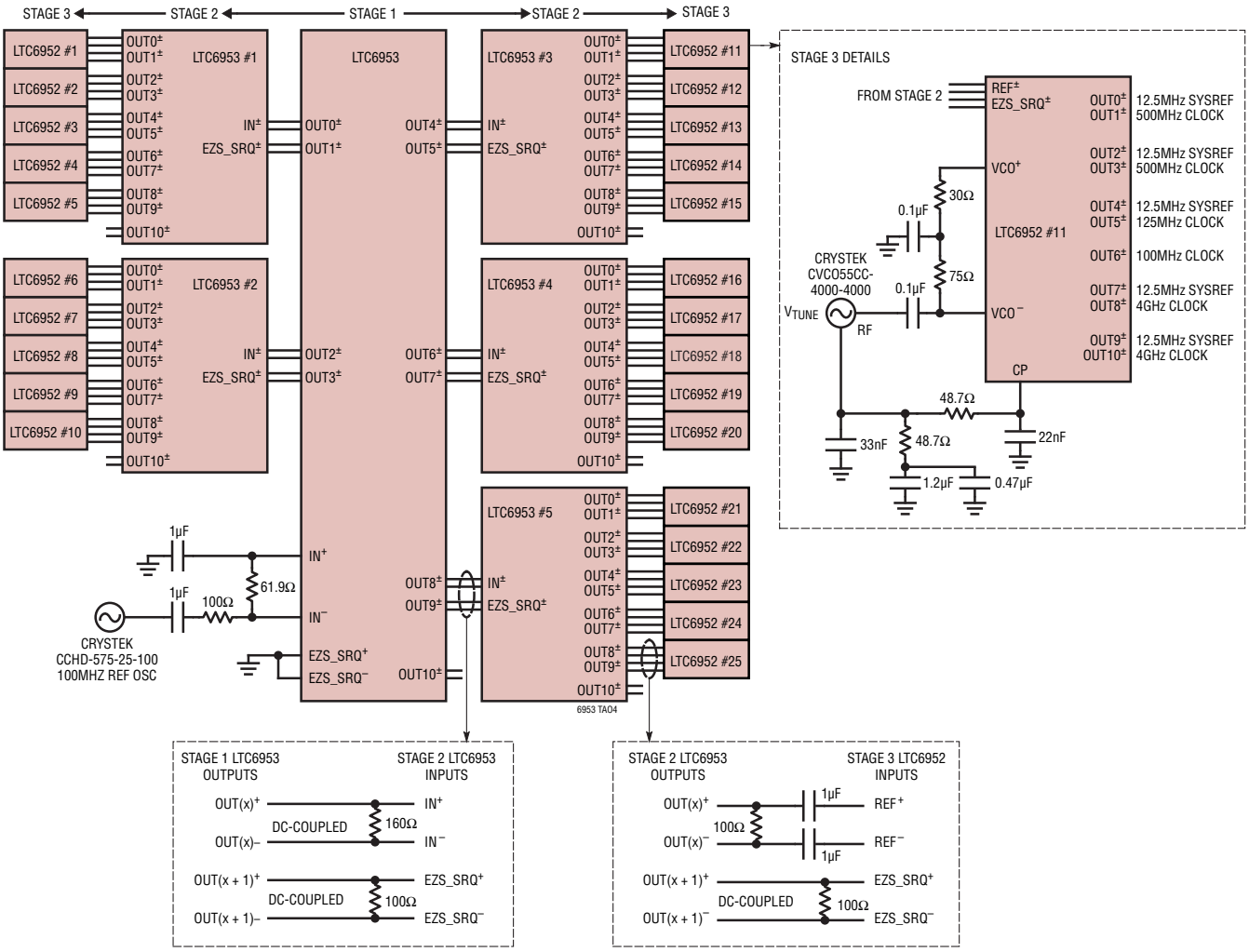
Stage 2 LTC6952 Phase Noise vs Stage 1 LTC6953 ADEL Setting f_{OUT} = 4GHz, Mx = 1



Rev PrA

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 and 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 and 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

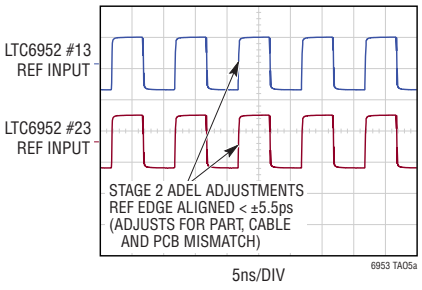
C) Send SYSREF: Toggle Stage 1 LTC6953 SSRQ Bit

Step 4: Optional Reduce Power

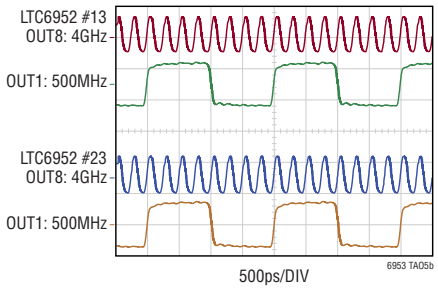
A) Power-Down LTC6952 SYSREF Outputs

B) Set Stage 1 and 2 LTC6953 and LTC6952 SRQMD = 0

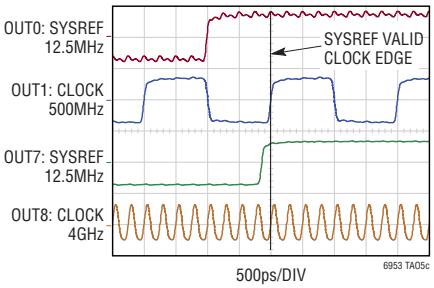
Step 1: Reference Alignment



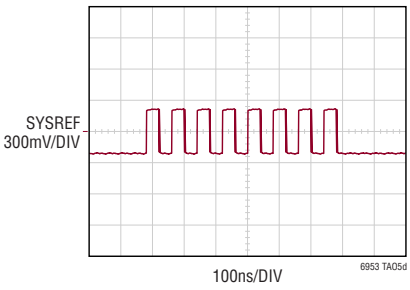
Step 2: ParallelSync Multichip Clock Alignment



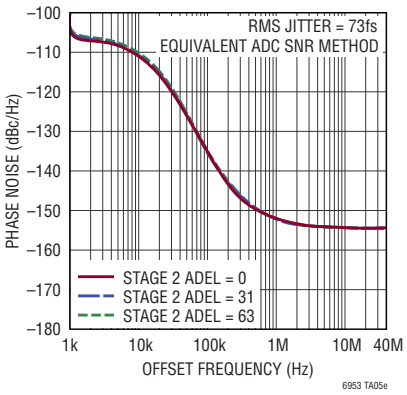
Step 3: SYSREF Alignment



Step 3: SYSREF Pulses



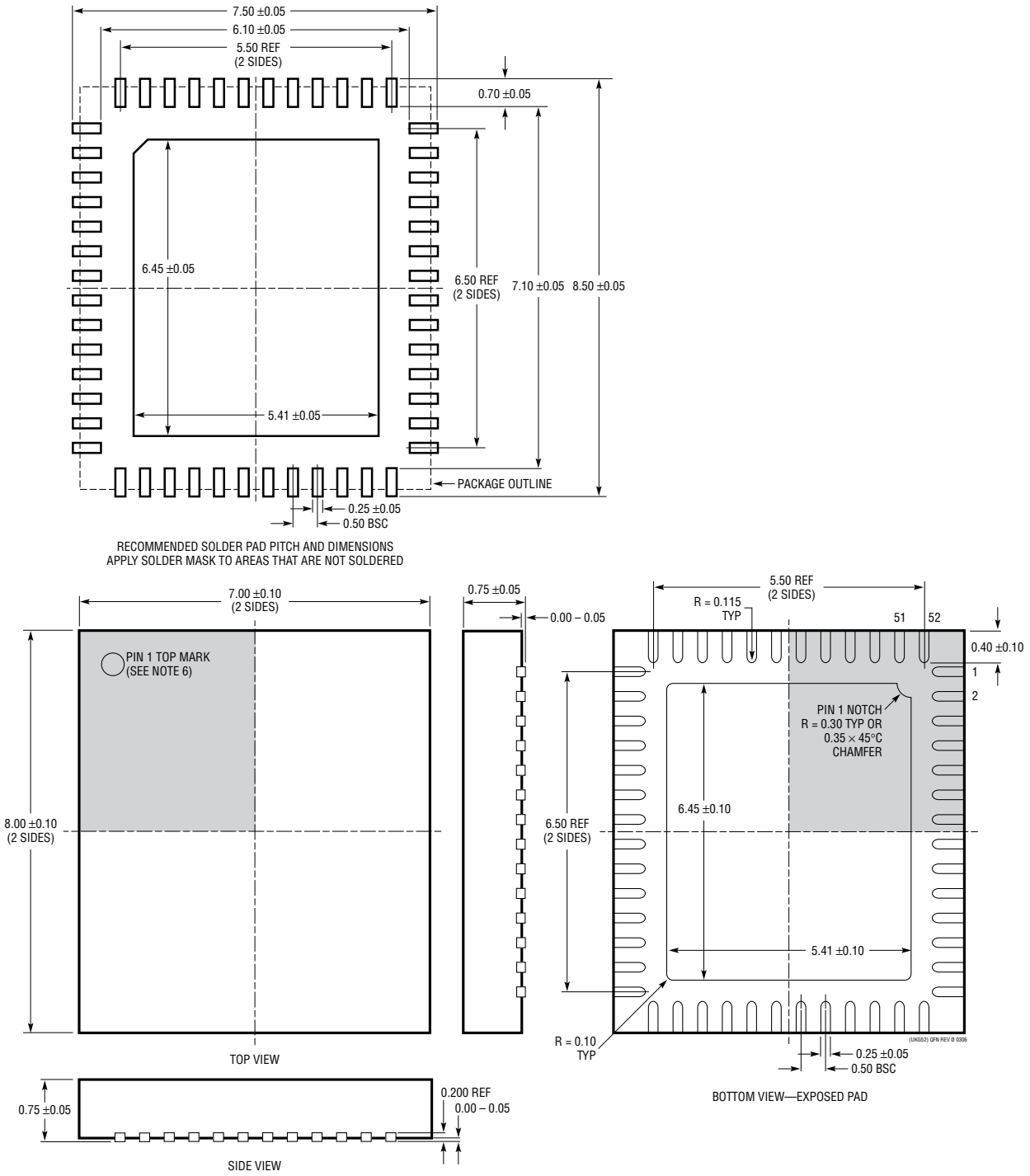
Stage 2 LTC6952 Phase Noise vs Stage 1 LTC6953 ADEL Setting f_{OUT} = 4GHz, Mx = 1



Rev PrA

PACKAGE DESCRIPTION

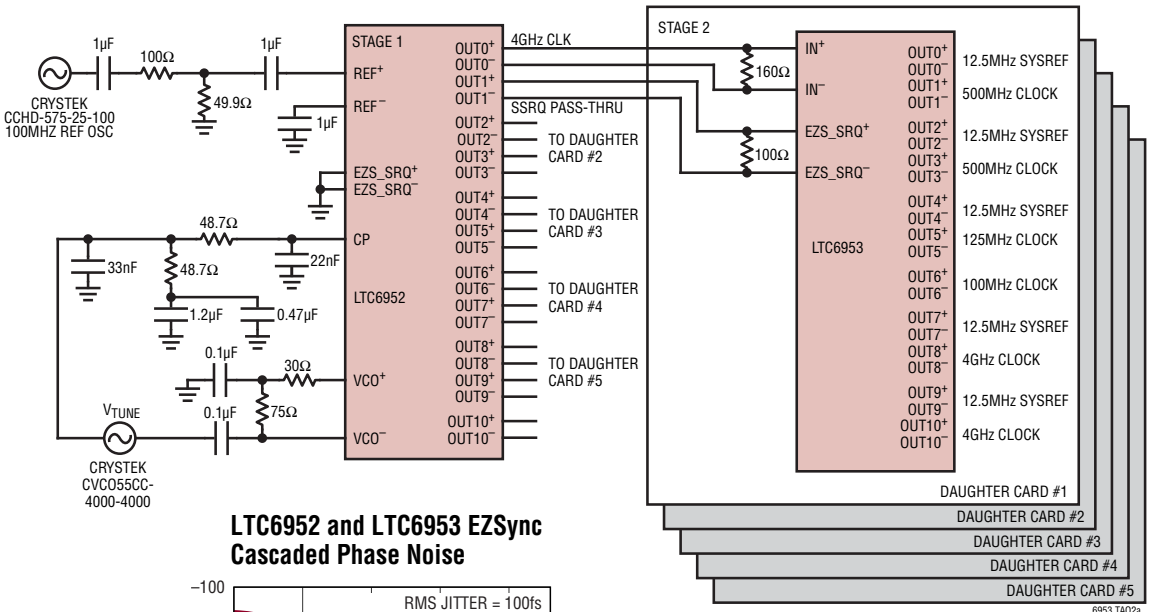
UKG Package
52-Lead Plastic QFN (7mm × 8mm)
(Reference LTC DWG # 05-08-1729 Rev Ø)



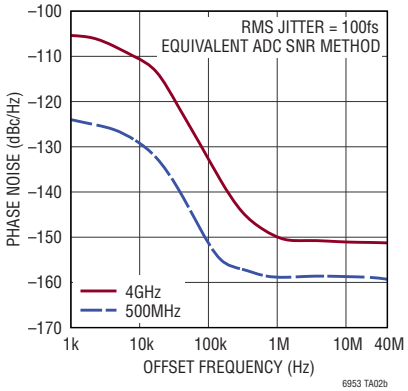
NOTE:
1. DRAWING IS NOT A JEDEC PACKAGE OUTLINE
2. DRAWING NOT TO SCALE
3. ALL DIMENSIONS ARE IN MILLIMETERS
4. DIMENSIONS OF EXPOSED PAD ON BOTTOM OF PACKAGE DO NOT INCLUDE MOLD FLASH. MOLD FLASH, IF PRESENT, SHALL NOT EXCEED 0.20mm ON ANY SIDE, IF PRESENT
5. EXPOSED PAD SHALL BE SOLDER PLATED
6. SHADED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

TYPICAL APPLICATION

EZSync Multichip Synchronization with Request Pass-Through



LTC6952 and LTC6953 EZSync Cascaded Phase Noise



RELATED PARTS

PART NUMBER	DESCRIPTION	COMMENTS
LTC6952	Ultralow Jitter, 4.5GHz PLL with 11 Outputs and JESD204B 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
LTC6945/LTC6946	Ultralow Noise and Spurious Integer-N Synthesizers	370MHz to 6.39GHz, –226dBc/Hz Normalized In-Band Phase Noise Floor, –157dBc/Hz Wideband Output Phase Noise Floor
LTC6947/LTC6948	Ultralow Noise and Spurious Frac-N Synthesizers	350MHz to 6.39GHz, –226dBc/Hz Normalized In-Band Phase Noise Floor, –157dBc/Hz Wideband Output Phase Noise Floor
LTC6950	1.4GHz Low Phase Noise, Low Jitter PLL with Clock Distribution	Four Independent LVPECL Outputs with 18fs$_{RMS}$ Additive Jitter (12kHz to 20MHz)
LTC6951	Ultralow Jitter Multioutput Clock Synthesizer with Integrated VCO	Four Independent CML Outputs and One LVDS Output, Integrated VCO, 110fs ADC SNR Jitter
LTC6954	Low Phase Noise, Triple Output Clock Distribution Divider/Driver	LVPECL, LVDS and CMOS Outputs with < 20fs$_{RMS}$ Additive Jitter (12kHz to 20MHz)