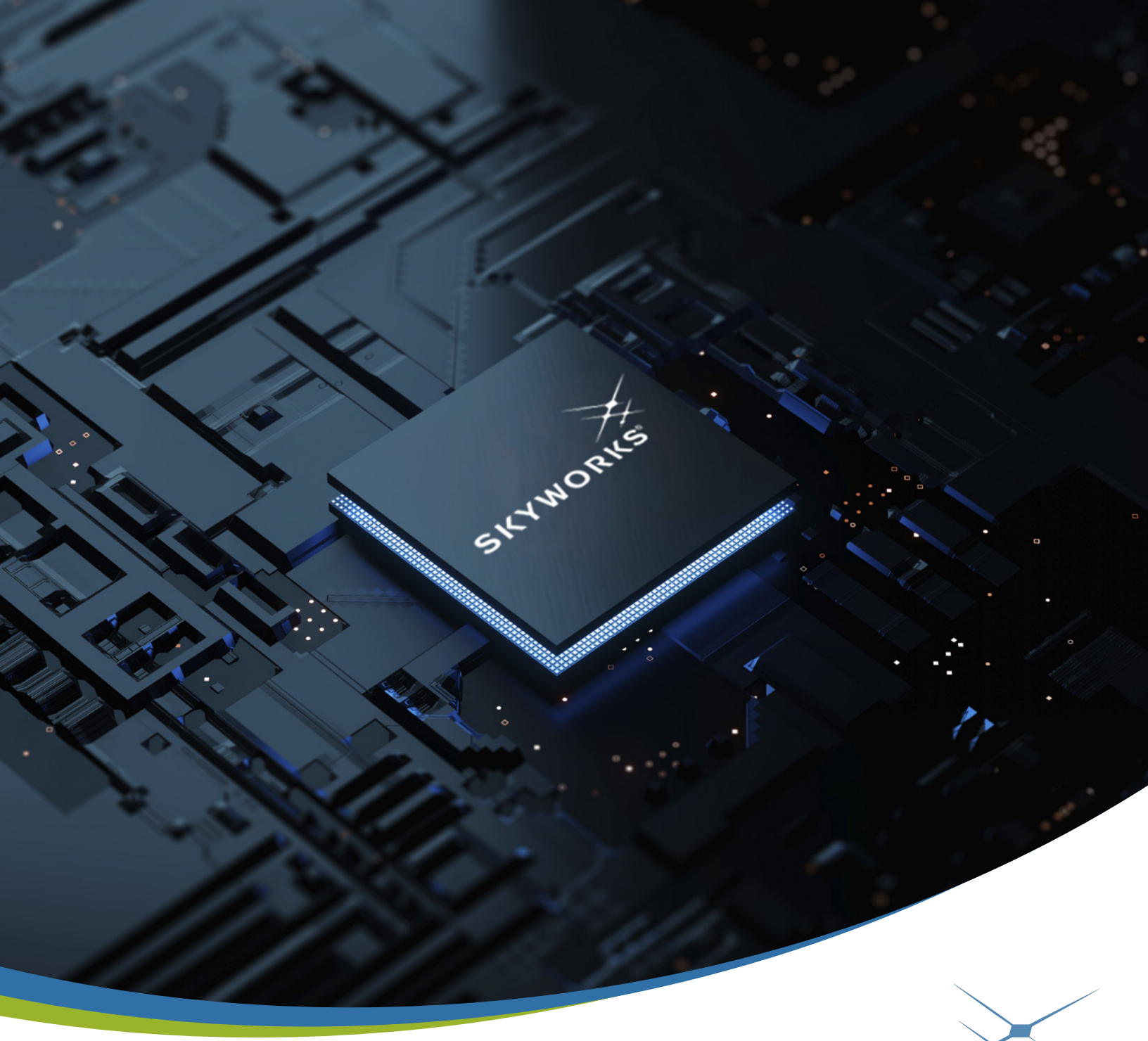


SKYWORKS®

Automotive Electronics Clock Tree Design Considerations

skyworksinc.com

Introduction

Hardware design in automotive electronics is becoming increasingly complex as systems integrate more functionality, adopt higher bandwidth processors and FPGAs, Ethernet, the PCI-Express data bus, and additional peripheral functions.

- Evolution of Automotive Clock Trees
- Reference Source: When to Use a Crystal vs a Clock
- Synchronous vs Free-Running Design
- Reference Clock Jitter
- Calculating Clock Tree Jitter
- Defining the Clock Tree
- Design Considerations and Selecting the Optimal Timing Solution
- Clock Tree Solution Example

Evolution Of Automotive Clock Trees

Automotive electronics are evolving faster than ever, with manufacturers bringing feature-rich infotainment systems and advanced driver assistance systems (ADAS) to market while simultaneously developing fully autonomous vehicles. Older generations of gateway, driver assistance, infotainment, and camera systems were based on simple microcontrollers or lowbandwidth FPGAs, often only needing a few single-ended reference clocks, which could be easily satisfied by using quartz crystal components. As automotive electronics adopt higher bandwidth processors, SoCs, and FPGAs, these designs and their associated timing requirements are becoming more complex, needing a higher number of reference clocks, in singleended and differential formats, at varying frequencies. Additionally, RMS phase jitter performance requirements have become more stringent to meet the needs of higher speed Ethernet and PCI-Express. Satisfying this growing, diverse set of reference clock requirements can be simplified using frequency-flexible, AEC-Q100 qualified clock generators instead of legacy quartzbased timing components. Silicon-based timing solutions provide reliability, cost, and board space advantages over legacy quartz-based crystals and oscillators, while at the same time provide numerous additional features and benefits, such as spread spectrum clock generation, EMI suppression frequency tuning, redundancy, fault detection, and other health monitoring functions to help meet system safety goals.

Reference Source – When to Use a Crystal vs. a Clock

In legacy automotive electronics designs, quartz crystals have traditionally been used as a reference source to most endpoint devices that did not feature high speed SerDes. Reference clocks did not need to be precise, jitter was not a concern, and only a handful of single-ended frequencies were needed. A quartz crystal is a mechanical component that uses the piezoelectric effect to produce a sine wave output when an electrical signal is applied to the device. Crystals are low cost components, with most suppliers offering AEC-Q200 options for the automotive market.

Though simple and easy to implement, quartz crystals do have limitations. One drawback is that frequency can vary significantly over temperature, exceeding the parts-per-million (ppm) stability requirements of many SerDes applications. Quartz crystals are also highly susceptible to shock and vibration failure because the piece of quartz is connected to the package through very thin pieces of metal, called tines. If one of the tines breaks, the quartz crystal fails, and the reference frequency is lost. As a result, quartz crystals have high failure in time (FIT) rates and are often the system components with the highest reliability concerns. Adding more pieces of quartz to a design increases the number of points of failure.

Many high-speed SerDes require precision reference clock sources in order to minimize bit error rate (BER), and leverage benefits of differential format clocks such as LVDS, HCSL, or LVPECL. Precision differential reference clocks are commonly required by Ethernet, PCI-Express, ASIC, FPGA, and processor endpoint devices. Quartz crystal oscillators or silicon-based clock generators are commonly used as differential reference clock sources. A quartz crystal oscillator is a mechanical component that contains both a quartz crystal and an oscillation circuit, and has the same susceptibility and reliability limitations as quartz crystals do. Silicon-based clock generators are capable of consolidating the functionality of multiple quartz crystals and/or quartz crystal oscillators into a single IC solution, feature much lower FIT rates for higher reliability, and include rich sets of added features and benefits that system designers can take advantage of to further simplify their system designs. As the size of clock trees grow, clock generators also become much more economical solutions because they can generate both single-ended as well as differential output clocks, consume less board space area compared to a solution comprising of many pieces of quartz-based components, and are often times a cheaper alternative.

Synchronous vs. Free-running Design

There are two types of timing architectures: synchronous and free-running (or asynchronous). Applications that require one or more independent reference clocks that do not need to be synchronized to a common reference frequency are categorized as a free-running clock architecture. Nearly all automotive electronics applications today are free-running. Quartz crystals, quartz crystal oscillators, clock generators, and clock buffers are commonly used in system designs with free-running, clock architecture.

Synchronous clock architecture is more commonly used in applications that require continuous communication and network-level synchronization. Synchronous clock architectures have not traditionally been used in automotive electronics design, however it may be necessary to synchronize the timing between Lidar/Radar sensors and automotive driving ECUs, as well as 5G wireless connectivity in L4/L5 automated driving applications. By synchronizing all SerDes reference clocks to a highly accurate reference clock (e.g. Stratum 3, GPS, IEEE 1588 grandmaster), synchronization is guaranteed across all endpoint nodes within the entire system. In these applications, low bandwidth PLL-based clocks provide wander and jitter filtering (jitter cleaning) to ensure network-level timing synchronization is maintained. Jitter attenuators or discrete VCO (Voltage-Controlled Oscillator)-based PLLs are the preferred clock solution for synchronous clock architecture design.

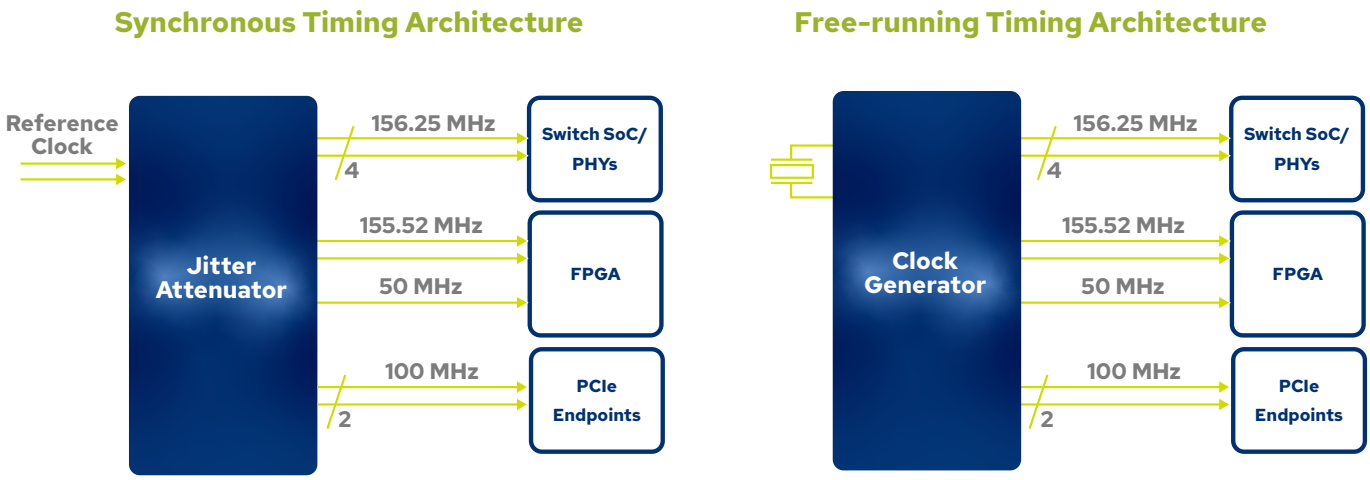


Figure 1. Synchronous vs Free-running Timing Architecture

Reference Clock Jitter

Jitter is the primary performance specification associated with a timing device because excessive clock jitter can compromise system performance. There are three common types of clock jitter. Depending on the application, one type of jitter will be more important than others.

- Cycle-to-cycle jitter measures the maximum change in clock period between any two adjacent clock cycles, typically measured over 1,000 clock cycles.
- Period jitter is the maximum deviation in clock period with respect to an ideal period over a large number of cycles (10,000 clock cycles typical).
- Phase jitter is a ratio of noise power to signal power calculated by integrating the clock single sideband phase noise across a range of frequencies offset from a carrier signal.

Both cycle-to-cycle jitter and period jitter are useful in calculating setup and hold timing margins in digital systems, and they are most commonly used as the type of jitter specified for a single-ended clock. Phase jitter is commonly used as the type of jitter specified for a differential clock, often times for high-speed SerDes applications. Phase jitter is especially critical in FPGA, Ethernet, and PCI-Express clocking applications in which excessive phase jitter can degrade the BER of the high-speed serial interface.

During clock tree design and component selection, it is important to evaluate devices based on maximum jitter performance specifications. Typical jitter specifications do not guarantee device performance over all conditions, including process, voltage, temperature, and frequency variation. Maximum jitter provides a more comprehensive specification inclusive of these factors. If a maximum jitter specification is not provided in a timing component datasheet, it is recommended to request the information from the supplier in order to ensure the performance will meet the requirement of the endpoint that the reference clock is being supplied to.

It is also recommended to ensure the test conditions for jitter specifications are well documented and provide a clear understanding of what the specification applies to. Clock jitter performance varies across a wide range of conditions, including device configuration, operating frequencies, number of enabled output clocks, signal format, power supply and power supply noise. Timing component datasheets will typically list the conditions associated with the jitter specifications provided. Look for devices that include maximum jitter specifications and fully specify jitter test conditions. If there is ambiguity in the way jitter is specified, or if it is only specified for a certain frequency/test condition that does not match the specific timing requirements in your design, it is recommended to contact the manufacturer for clarity and further information.

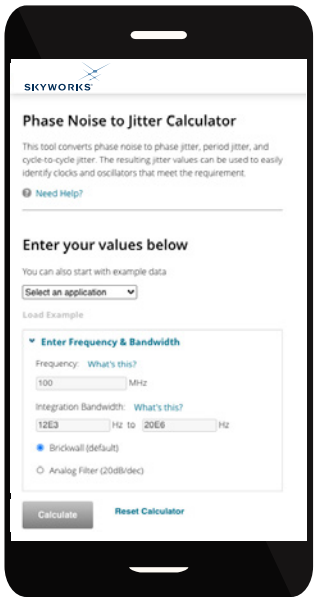
Calculating Clock Tree Jitter

Total clock tree jitter should be estimated to determine if there is sufficient system-level design margin to maximum limits set by endpoint device manufacturers. If there is more than one timing component between the reference source and the endpoint, it is important to note that total clock tree RMS jitter cannot be calculated by simply summing the individual data sheet jitter specifications for each of the timing components. In actuality, it is less than that. The total jitter is defined by the following sum of squares equation:

$$Tj\,(RMS) = \sqrt{J_1{}^2 + J_2{}^2 + \ldots + J_n{}^2} \quad \text{(Equation 1)}$$

Where T_j = Total RMS jitter, J_n = individual device RMS jitter.

Note: This equation can be applied to calculating total period jitter and phase jitter, assuming the jitter distributions are Gaussian and uncorrelated. This equation should not be applied to cycle-cycle jitter because cycle-to-cycle jitter is expressed as a peak jitter number, not RMS.



Component jitter Jn can be estimated using datasheet jitter specifications or calculated from phase noise data. Skyworks offers an easy-to-use utility for converting clock phase noise to RMS phase jitter. See the Skyworks Phase Noise to Jitter Calculator for more details. As mentioned above, be sure to use maximum jitter specifications to generate a conservative estimate of total clock tree jitter.

Defining the Clock Tree

Selecting an optimal timing solution starts with defining the set of reference clocks needed in a system design, such as processors, FPGAs, switch SOCs, Ethernet PHY, BT/LAN connectivity devices, USB PHYs, PCI-Express endpoints, or ASICs. Table 1 is a summary of reference clock requirements, commonly known as a clock tree. In addition to listing the frequencies a clock tree also includes the clock signal formats (LVCMOS, LVPECL, LVDS, HCSL), voltage levels, and most importantly, the maximum jitter level of each endpoint. Some endpoint devices may specify the need for a crystal input source, but often times, these devices can alternatively use a single-ended LVCMOS clock input. Most of the information needed to generate a clock tree summary can be found by reading through the endpoint device datasheets. If the design is synchronous, information on the input reference clock source(s) should also be summarized in the clock tree.

Clock trees can be drawn out in a block diagram format but are more commonly summarized in a table format. It is recommended to order the rows in the table such that the reference clocks with the most stringent jitter requirements are listed at the top, highlighted as the most important.

Table 1. Clock Tree Requirements

Frequency	Format	Voltage	Max Jitter	Endpoint
40 MHz	LVCMOS (or crystal)	3.3 V	100 ps period	SoC/CPU/FPGA
100 MHz	HCSL	3.3 V	500 fs RMS	SoC/CPU/FPGA
125 MHz	LVDS	1.8 V	800 fs RMS	SoC/CPU/FPGA
156.25 MHz	LVDS	1.8 V	500 fs RMS	1/2.5/10GbE PHY
100 MHz	HCSL	3.3 V	500 fs RMS	PCIe endpoint
100 MHz	HCSL	3.3 V	500 fs RMS	PCIe endpoint
50 MHz	LVCMOS (or crystal)	1.8 V	5 ps period	10/100 PHY
48 MHz	LVCMOS (or crystal)	1.8 V	10 ps period	USB 3.0 PHY

Design Considerations and Selecting an Optimal Timing Solution

Summarizing the clock tree is a very useful first step but system designers should also take other design considerations into account when reviewing timing device options and selecting an optimal solution.

Reliability: As previously noted, quartz crystal and oscillator components are mechanical devices that are prone to shock and vibration failure. Including many quartz crystal and oscillator components in a system design not only increases the bill of materials and total system cost but also increases the number of potential points of failure and creates a reliability concern. A better approach is to replace quartz crystals and oscillators with a highly integrated, silicon-based clock generator. This modern solution significantly reduces the FIT rate associated with the timing portion of the design, while also providing numerous other clock tree design benefits, including spread spectrum clock generation for EMI/EMC mitigation, frequency selection, and fault monitoring.

Frequency Flexibility: High bandwidth FPGA and processor platforms require a diverse mix of reference clocks of varying frequencies, single-ended and differential output signal formats, different output voltages, all with low jitter requirements. Skyworks' patented MultiSynth output divider technology provides 0 ppm synthesis error on both integer and fractional related output frequencies, on up to 12 outputs while maintaining industry best jitter performance. Each clock output can be individually set to a specific output format level and can be tied to different output voltage levels, easily aligning to system design requirements. In some designs, it may be highly beneficial to have the ability to change an output frequency, or fine tune it in ppm increments in a digitally-controlled oscillator (DCO) fashion, which are features that are also available in a clock generator solution.

System Safety Goals: Automotive Safety Integrity Level (ASIL) is a new consideration in ADAS and automated driving platforms. In these applications, reference clock redundancy, fault detection, and health monitoring may be required to meet system-level safety goals. Our AEC-Q100 qualified Si5332-AM clock generators include health status features, such as primary and backup reference inputs for redundancy, input reference health and status monitoring, loss of reference signal fault detection indicators, hardware interface pins to communicate with an ASIL system safety manager IC, and the ability to switch from a primary to a secondary input source.

Fault Detection: ADAS and automated driving platform designs require system-level safety plans to identify faults and take subsequent action to ensure safety of the vehicle and its passengers. Fault detection and health status monitoring features are unavailable in quartz-based solutions. Clock generators like the Si5332-AM offer an I2C interface as well as GPIO pins to provide fault and health monitor signals to a system safety manager IC or ASIL-rated MCU. These hardware pins can also be programmed as input pins, providing the ability to migrate from a primary input reference source to a backup source, in the event a fault occurs on the primary source.

Emissions: Automotive clock trees increasingly include a combination of both differential and single-ended reference clocks. Single-ended LVCMOS reference clocks are common sources of emissions in a high-speed digital design, making it challenging to meet CISPR25 Class-4 and Class-5 emissions requirements. Skyworks has developed device features and layout design guidelines to overcome these challenges. Leveraging complementary LVCMOS output drivers and following the recommended layout design guidelines outlined in AN1237 have proven to show positive results in CISPR25 testing.

Format/Voltage Translation: Reference clock signal format and voltage level are specified within each endpoint device datasheet. Most clock trees will require a combination of different formats as well as voltages, even if the frequencies needed are exactly the same. Clock generators provide the ability to individually program each output driver to a specific output format and voltage, making it easy to satisfy all endpoint reference clock requirements from a single-IC solution.

Power Supply Noise: Jitter performance is directly affected by power supply or board level noise, in both quartz-based timing components as well as silicon-based timing components. As automotive electronics continue adopting higher bandwidth devices, mitigating power supply noise, and its affect on reference clock jitter, is an important design consideration. Even minimal levels of power supply noise will directly translate to significant increases in output clock jitter, often creating significant BER performance degradation within the endpoint SerDes.

The traditional approach to suppressing power supply noise is using RC filters, ferrite beads, or regulators, which add components, PCB area, and cost to a design. A better approach is to implement a timing solution that features on-chip LDOs, which eliminates the need for external capacitors, resistors, and ferrite beads. Power supply noise rejection (PSNR) is the performance parameter associated with a timing device's ability to suppress power supply noise while not impacting output clock jitter. PSNR should be listed in a timing component's datasheet. If PSNR is not listed in the AC specifications of a timing component datasheet, it is a clear indication that the supplier has not taken power supply noise into account and will require the system designer to account for it external to the timing device. Alternatively, clock generators like the Si5332-AM feature on-chip LDOs on each power pin, taking care of noise suppression and ensuring it has no effect on output clock jitter performance.

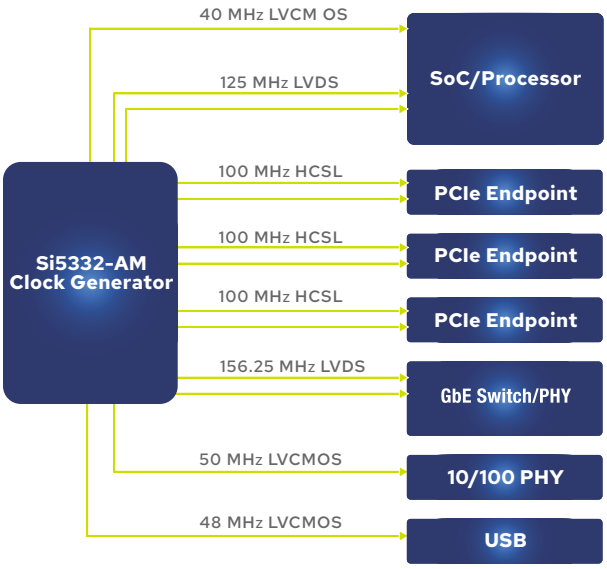
PCIe Clocks: The PCI-Express data bus is now commonly used in IVI, networking gateways, ADAS, and automated driving electronics. The SerDes within a PCIe device endpoint requires a 100 MHz differential clock adhering to the specifications outlined by the PCI-SIG. The latest generations of the PCIe specification require maximum reference clock RMS phase jitter of 500 fs for Gen4 and 150 fs RMS phase jitter for Gen5. The Si5332-AM programmable clock generators, Si5225x PCIe clock generators, and Si5325x PCIe buffers meet PCIe Gen1/2/3/4/5 max clock jitter specs with significant margin, and feature low-power HCSL output drivers with internal termination, capable of directly matching either an 85 ohm or 100 ohm transmission lines without any external resistors. The PCI-SIG specifies unique RMS phase jitter filters that are specific to PCI-Express reference clocks. Properly measuring jitter on a PCIe reference clock can be difficult and confusing. To simplify the process and eliminate confusion, Skyworks developed the PCIe Clock Jitter Tool, which can be downloaded for free by visiting our PCIe Learning Center.

Customization and Programmability: Skyworks' ClockBuilder Pro software guides users through an easy, step-by-step process to generate a configuration file specific to each clock tree design. When the device configuration is complete, ClockBuilder Pro generates a unique part number specific to each design, provides an associated datasheet addendum, and saves the file for future use. Once the part number is provided by ClockBuilder Pro, it is available for sample/production orders, with turnaround time for samples as fast as 2 weeks. A field programmer is also available that assists with making changes on boards that have a pre-configured Si5332-AM already mounted.

Clock Tree Example Solution

After summarizing the clock tree and taking all other design considerations into account, the next step is selecting the optimal timing solution.

Optimal Solution Using Clock Generator



Frequency	Format	Voltage	Max Jitter	Endpoint
40 MHz	LVCMOS (or crystal)	3.3 V	100 ps period	SoC/CPU/FPGA
100 MHz	HCSL	3.3 V	500 fs RMS	SoC/CPU/FPGA
125 MHz	LVDS	1.8 V	800 fs RMS	SoC/CPU/FPGA
156.25 MHz	LVDS	1.8 V	500 fs RMS	1/2.5/10GbE PHY
100 MHz	HCSL	3.3 V	500 fs RMS	PCIe endpoint
100 MHz	HCSL	3.3 V	500 fs RMS	PCIe endpoint
50 MHz	LVCMOS (or crystal)	1.8 V	5 ps period	10/100 PHY
48 MHz	LVCMOS (or crystal)	1.8 V	10 ps period	USB 3.0 PHY

The AEC-Q100 qualified Si5332-AM clock generator is an ideal solution for modern automotive electronics designs, capable of synthesizing all required clock frequencies, any mix of differential and complementary LVCMOS signal formats, and output voltages at jitter performance levels well within the max limits specified by end point device datasheets. For the clock tree solution satisfying the requirements outlined above, the Si5332-AM also provides:

- Spread spectrum clock generation for PCIe Gen3/4/5 reference clocks
- Redundant reference inputs with fault detection and communication to system safety manager
- Ability to use complementary LVCMOS drivers to minimize emissions
- On-chip LDOs to ensure power supply noise does not impact output clock jitter performance
- HCSL output drivers with internal termination, eliminating the need for external resistors for PCIe reference clocks and simplifying layout
- A lower cost, smaller footprint, more reliable solution compared to quartz-based crystals and oscillators

skyworksinc.com

SKYWORKS®

Skyworks' Automotive Timing Products and Applications

Skyworks offers the industry's broadest portfolio of AEC-Q100 qualified timing products, with unparalleled performance, integration, and features to help simplify clock tree design in automotive electronics. Learn about the entire family of Skyworks' AEC-Q100 qualified timing solutions and automotive applications.

For technical inquiries, product questions, or to schedule a meeting with Skyworks to learn more about AEC-Q100 qualified timing solutions, please click the "Request Support" button under Contact Technical Support at skyworksinc.com/support-ia. A member of our product management or applications engineering team will reply with an answer and/or to schedule a discussion together.