

Clock Tree 101

by Linda Lua

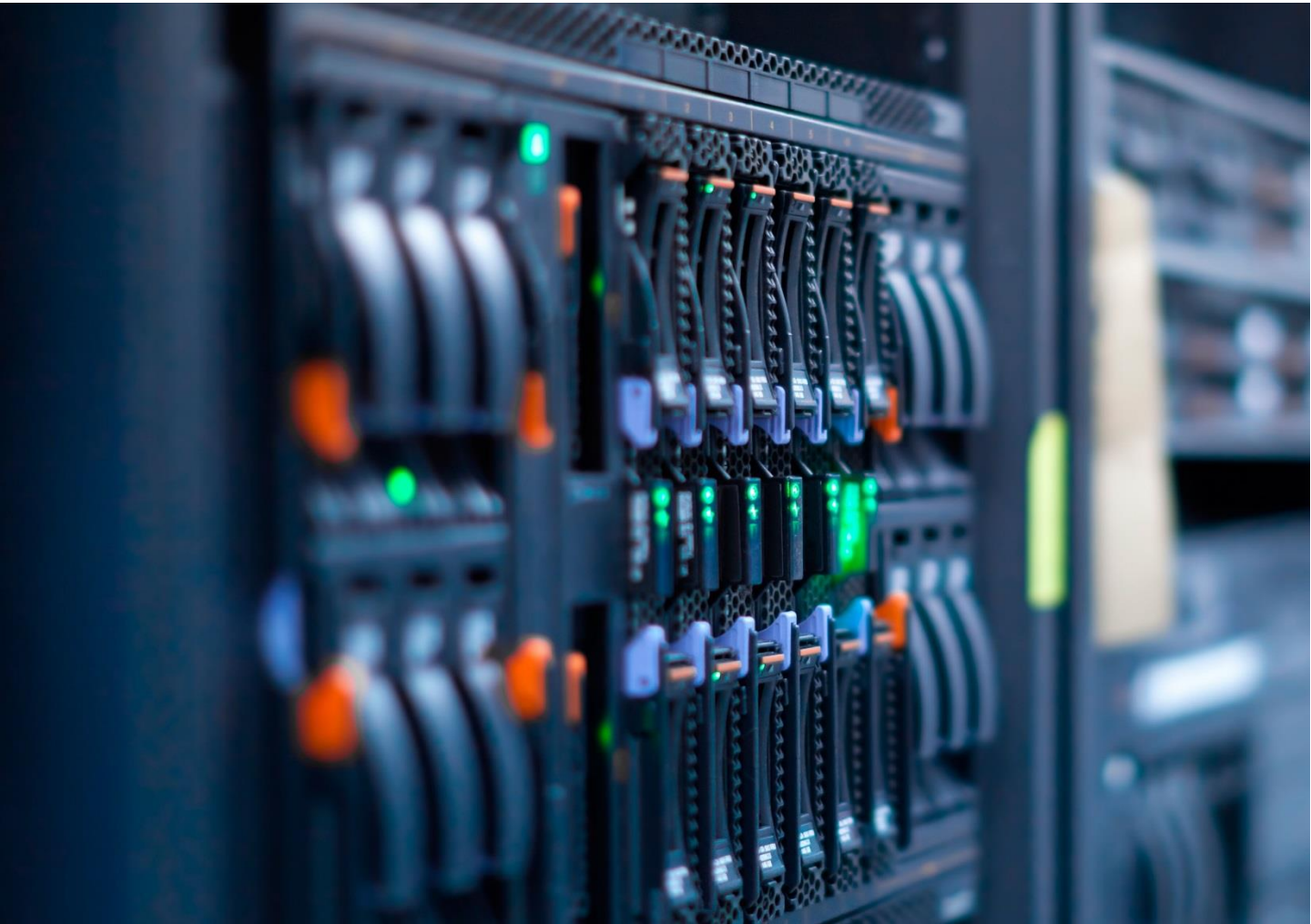



Table of Contents

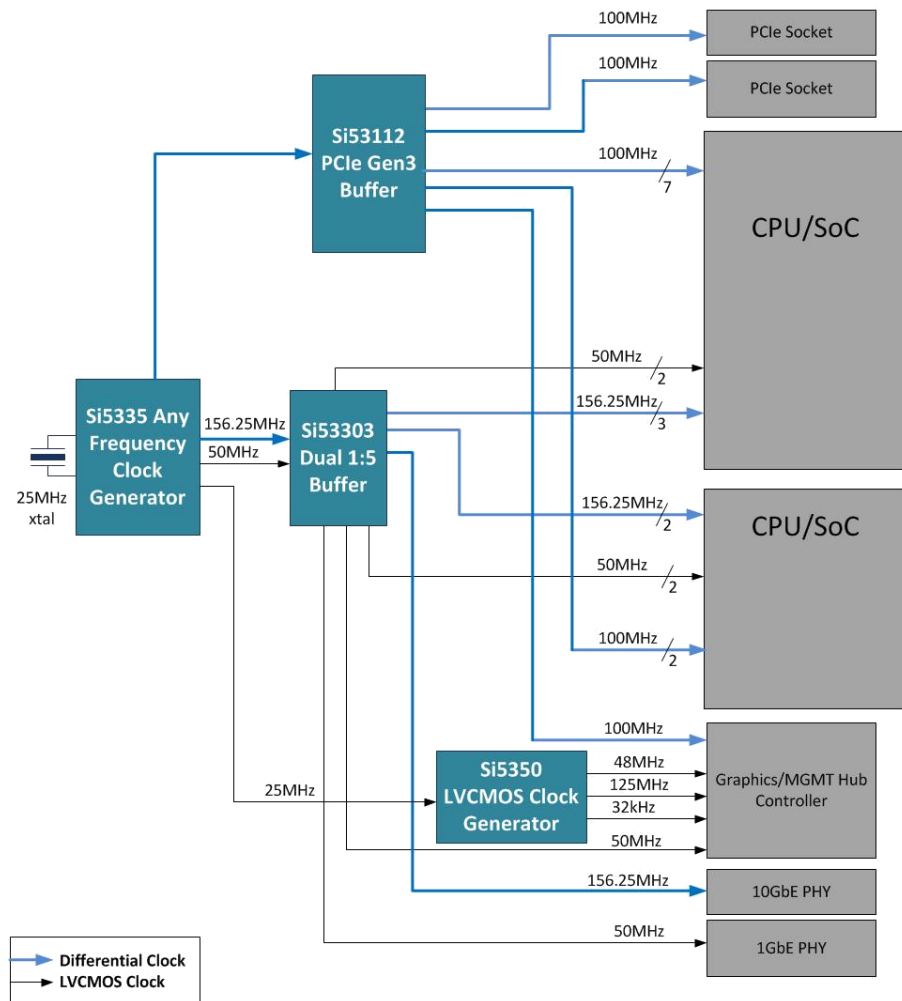
- I. What is a Clock Tree?
 - II. Clock Tree Components
 - I. Crystals and Crystal Oscillators
 - II. Clock Generators
 - III. Clock Buffers
 - IV. Jitter Attenuators
 - III. Clock versus Crystal
 - IV. Free-running versus Synchronous
 - V. Clock Jitter
 - VI. Estimating Clock Tree Jitter
 - VII. Selecting Components
 - VIII. Optimizing Clock Trees
 - IX. Conclusions
 - X. Clock Tree Terminology
- 

What is a Clock Tree?

A clock tree is a clock distribution network within a system or hardware design. It includes the clocking circuitry and devices from clock source to destination.

The complexity of the clock tree and the number of clocking components used depends on the hardware design. Since systems can have several ICs with different clock performance requirements and frequencies, a “clock tree” refers to the various clocks feeding those ICs.

It’s often the case that a single reference clock will be cascaded and synthesized into many different output clocks, resulting in a diagram that looks a bit like a sideways tree trunk. The “trunk” is the reference clock and the “branches” are the various output clocks.



Example “Clock Tree”

Timing Components



Clock trees can be both very complex with many timing components or very simple with a single reference and a few copies. Of course, their complexity depends on the system they support.

While there are many timing component types for many different types of applications, the most common timing components are:

- Crystals – a piece of quartz or other material that resonates in a predictable pattern at a given frequency when used in conjunction with an on-chip voltage oscillator circuit;
- Crystal Oscillators (XOs) – a self-contained resonator and oscillator that outputs a given frequency and format;
- Voltage controlled oscillators (VCXOs) – a self-contained oscillator that varies its output frequency in concert with differing voltages from a voltage reference;
- Clock Generators – an integrated circuit that uses a reference clock or crystal to generate multiple output clocks at one or multiple frequencies;
- Clock Buffers – an integrated circuit that creates copies or derivatives of a reference clock;
- Jitter Attenuators or Jitter Cleaners – an integrated circuit that removes jitter (noise) from a reference clock.

Crystals and Crystal Oscillators

Crystals and XOs

Clock
Generators

Clock
Buffers

Jitter
Attenuators

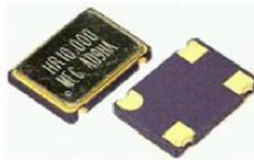
Crystals use quartz, cut at a particular angle and mounted in a protective metal casing, to provide a frequency output when an electrical signal is applied. The output is a single-ended sine wave typically ranging from 32 kHz to 50 MHz. Each output frequency requires a different quartz cut. Crystals require an oscillator circuit to operate. This is generally integrated in the target IC.

Crystal Oscillators (XOs) • Crystal Oscillators (XOs) integrate the crystal with the oscillator circuit, enabling XOs to provide higher frequency outputs. XOs generate a square wave output that is either single-ended or differential. Differential signaling is used in high-speed, jitter sensitive applications. Some specialized XOs provide multi-frequency support either via I2C or pin control. Crystals and XOs are generally very cost effective unless the application requires a variety of clock frequencies. Crystals and XOs are typically used as individual IC reference clocks.

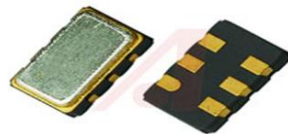
Crystals and XOs are generally very cost effective unless the output requirements are stringent. They are typically used as individual IC reference clocks.



Crystal
single-ended
sine wave
output



LCMOS XO
single-ended
square wave
output



Differential XO
differential or
complementary
square wave
output

Three common types of frequency reference sources

Clock Generators



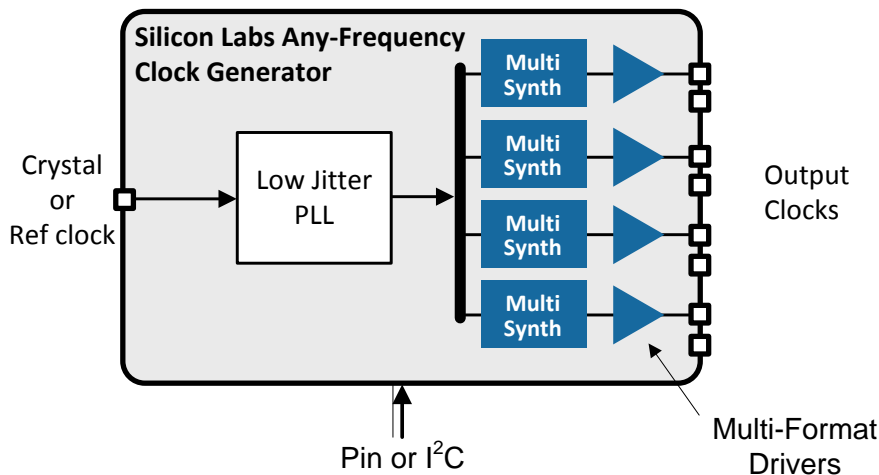
Clock generators are integrated circuits (ICs) that generate multiple output frequencies from a single input reference frequency. The reference frequency may be supplied by a crystal, XO or other clock that may already be present.

Clock generators may also have other features including the ability to turn on/off outputs, skew frequencies, and add spread spectrum to frequencies. They allow feature control through I2C, SPI or pin control.

The clock generator shown below is programmable with up to eight single-ended outputs or four differential outputs. It allows designers to replace eight single-ended crystals or four differential ones.

The perceived challenge with clock generators is in the system layout design. Placing a crystal right next to a target IC is simple and cheap. Routing a signal from a clock generator might not be. There are many points of view, but generally speaking, systems requiring four or more clocks can economically use a clock generator.

Differential signaling, skew control, careful transmission line design, and other techniques can be used to ensure that a centralized clock source provides similar performance as multiple discrete crystals/XOs.



Silicon Labs Si5338 Clock Generator

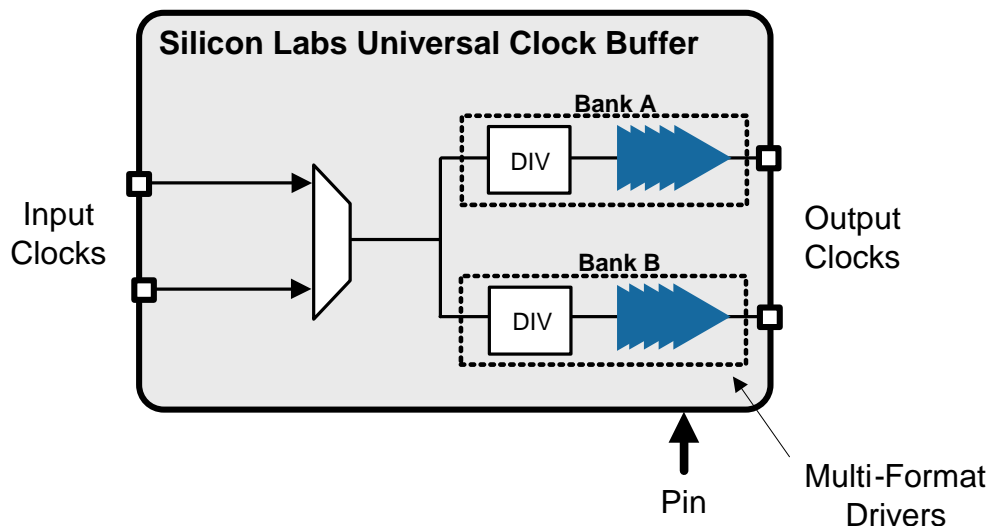
Clock Buffers



Clock buffers are fairly straight-forward ICs for distributing multiple copies of a clock to multiple ICs with the same frequency requirements. A buffer's reference clock can be from a clock generator, an XO or a clock already present. Clock buffers scale from 2 outputs to more than 10 outputs.

Because they are ICs with integrated logic, clock buffers can include functions such as signal level format translation, voltage level translation, multiplexing and input frequency division.

These features save board space and cost by eliminating additional timing components, external voltage dividers or signal level transition circuits.



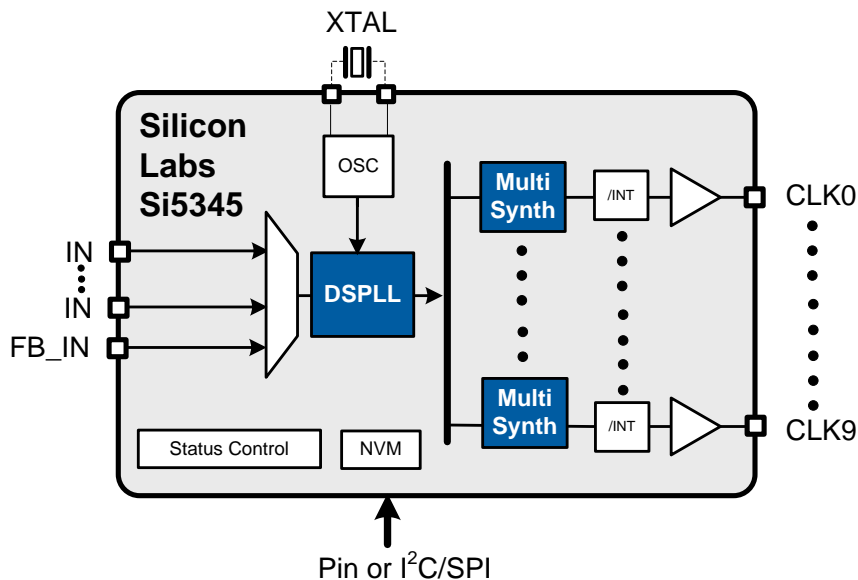
Silicon Labs Si5330x Universal Buffer

Jitter Attenuators



Jitter attenuators are clock generators with specialized circuitry for reducing jitter. They can also be called clock cleaners or jitter cleaners. These highly specialized timing devices remove jitter from incoming reference clocks and minimize jitter in the end application.

Jitter attenuators are typically used in high-speed applications such as Synchronous Ethernet and SDI Video to ensure that all physical layer data transmission is synchronized



Silicon Labs Si5345 Jitter Attenuating Clock

Crystal, XO or Clock Generator?



When to Use a Crystal vs a Clock

When starting a clock tree design, the first step is to inventory all the required clock frequencies, types, and target IC locations on the system board.

Quartz crystals are typically used if the IC has an integrated oscillator and on-chip phase-locked loops (PLLs) for internal timing. Crystals are cost-effective components that exhibit excellent phase noise and are widely available. They can also be placed in close proximity to the IC, simplifying board layout.

One of the drawbacks of crystals is that their frequency can vary significantly over temperature, exceeding the parts-per-million (ppm) stability requirements of some applications.

In many stability-sensitive high-speed applications, crystal oscillators (XOs) are a better fit because they guarantee tighter temperature stability.

Use clock generators and clock buffers when several reference frequencies are required and the target ICs are all on the same board or in the same IC or FPGA. In some applications, FPGA/ASICs have multiple time domains for the data path, control plane and memory controller interface and require multiple unique reference frequencies. This is a good place for a clock generator.

A clock generator or buffer is also better when the IC cannot accommodate a crystal input, when the IC must be synchronized to an external reference (source-synchronous application), or when a high-frequency reference is required.

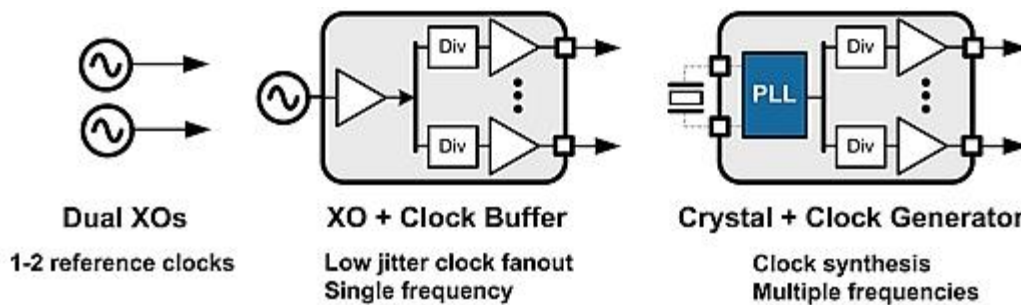
Free Running vs. Synchronous?



Free-Running versus Synchronous Clock Trees (Part 1)

Once the clock inventory has been completed, the next step is to determine and comply with the required timing architecture: free-running or synchronous?

Free running applications require one or more independent clocks without any special phase-lock or synchronization requirements. Example applications are standard processors, memory controllers, SoCs and peripheral components (e.g., USB and PCI Express switches).



Free-Running Clock Tree Examples

Free Running vs. Synchronous?



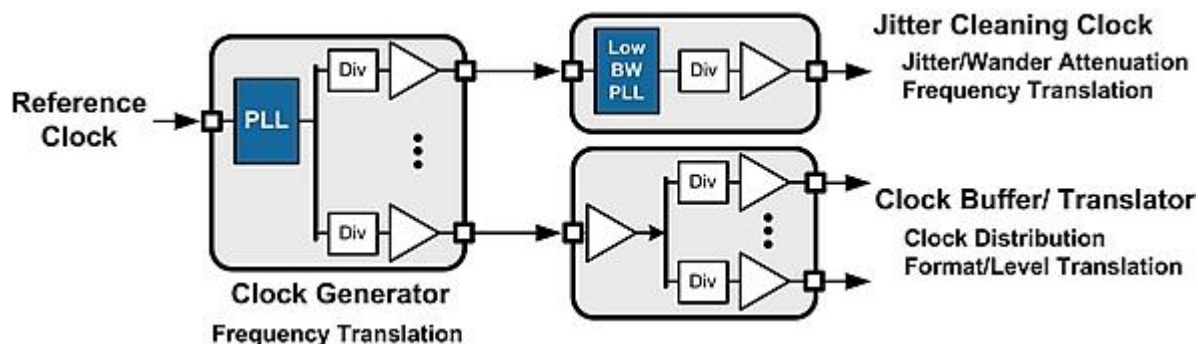
Free-Running versus Synchronous Clock Trees (Part 2)

Synchronous applications require continuous communication and network-level synchronization. Examples are Optical Transport Networking (OTN), SONET/SDH, mobile backhaul, synchronous Ethernet and HD SDI video transmission. These applications require transmitters and receivers to operate at the same frequency.

Synchronizing all SerDes (serialization-deserialization) reference clocks to a highly accurate network reference clock (e.g., Stratum 3 or GPS) guarantees synchronization across all nodes. In these applications, low-bandwidth PLL-based clocks provide jitter filtering to ensure that network-level synchronization is maintained.

Networking line card PLL applications generally use specialized jitter attenuating clocks or discrete PLLs with voltage-controlled oscillators.

For optimal performance, a jitter attenuating clock should be placed at the end of the clock tree, directly driving the SerDes device. Clock generators and buffers can be used to provide other system references.



Synchronous Clock Tree Example

Clock Jitter – What Is It?



Clock jitter is a critical specification for timing components because excessive clock jitter compromises system performance.

There are three common types of clock jitter, and depending on the application, one type of jitter will be more important than another.

- **Cycle-to-cycle jitter** measures the **maximum change** in the clock period between any two adjacent clock cycles, typically measured over 1,000 cycles.
- **Period jitter** is the **maximum deviation** in clock period with respect to an ideal period over a large number of cycles (10,000 is typical).
- **Phase jitter** is the figure of merit for demanding, high-speed SerDes applications. It 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.

Silicon Labs provides a detailed investigation of timing jitter in the Timing Jitter Dictionary and Technical Guide available at the button below.

[Timing Jitter Dictionary & Technical Guide](#)

Selecting Components



It is important to evaluate devices based on maximum (MAX) jitter performance.

Typical (TYP) data sheet specifications **do not guarantee device performance** over all conditions. The device performance can change across manufacturing process, supply voltage, temperature and frequency variation.

Take special care to closely read the test conditions on data sheets.

Clock jitter performance varies across a wide range of conditions including device configuration, operating frequency, signal format, input clock slew rate, power supply and power supply noise.

Look for devices that fully specify jitter test conditions since they guarantee operation over real world operating conditions.

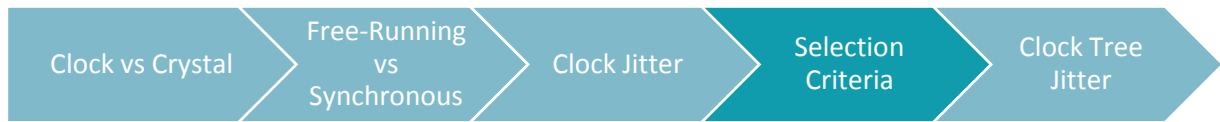
			MIN	TYP	MAX	
RMS Phase Jitter ³	J _{GEN}	Integer Mode 12 kHz to 20 MHz	—	0.090	0.140	ps RMS
		Fractional Mode 12 kHz to 20 MHz	—	0.130	0.165	ps RMS

Notes:

1. Measured as time from valid VDD/VDD33 rails (90% of their value) to when the serial interface is ready to respond to commands. Measured in SPI 4-wire mode, SCLK = 10 MHz.
2. Actual loop bandwidth might be lower; please refer to CBPro for actual value for your frequency plan.
3. Jitter generation test conditions: $f_{IN} = 19.44$ MHz, $f_{OUT} = 156.25$ MHz LVPECL, loop bandwidth = 100 Hz.

Example of MAX Jitter Specification with Test Specifications

Selecting Components



The table below summarizes many other selection criteria used for both free-running and synchronous clock trees.

More information on these specifications is at <http://www.silabs.com/timing>.

Function	Crystal	XO	Clock Generator	Clock Buffer	Jitter Attenuator
Free-run operation	No	Yes	Yes	Yes	Yes
Synchronous operation	No	No	Yes	Yes	Yes
Clock multiplication	No	No	Yes	No	Yes
Clock division	No	No	Yes	Yes	Yes
Jitter cleaning	No	No	No	No	Yes
Design complexity	Low	Low	Medium	Low	Medium
Integration	Low	Low	High	High	High
Key features that simplify clock tree design	Small form factor		Any-frequency, any-output	Format/level translation	Any-frequency clock synthesis
	Placement next to IC		Format translation	Integrated output mix	Integrated loop filter
			Glitchless switching btw clocks at different frequencies	Output voltage translation	Hitless switching
			VDD level translation	Synchronous output clock disable	Hold over on lock loss

Estimating Clock Tree Jitter



The total clock tree jitter should be estimated to determine if there is sufficient system-level design margin *before* the clock tree is committed.

A component with poor clock performance can compromise the whole system's performance if its jitter is too high or poorly specified.

It is fundamentally important to note that a clock tree's jitter is not simply the sum of the MAX specifications of each component. It is the root of the sum of the squares of each device's MAX RMS jitter.

$$T_j(RMS) = \sqrt{J_1^2 + J_2^2 + \dots + J_n^2}$$

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

You can also start with example data
Networking

▼ **Enter Frequency & Bandwidth**
Frequency: [What's this?](#)
 MHz
Integration Bandwidth: [What's this?](#)
 Hz to Hz
 Brickwall (default)
 Analog Filter (20dB/dec)

▼ **Enter Phase Noise**
Number of Phase Noise Data Pairs: [What's this?](#)

Click here for Silicon Labs' free "Phase Noise to Jitter Calculator" tool

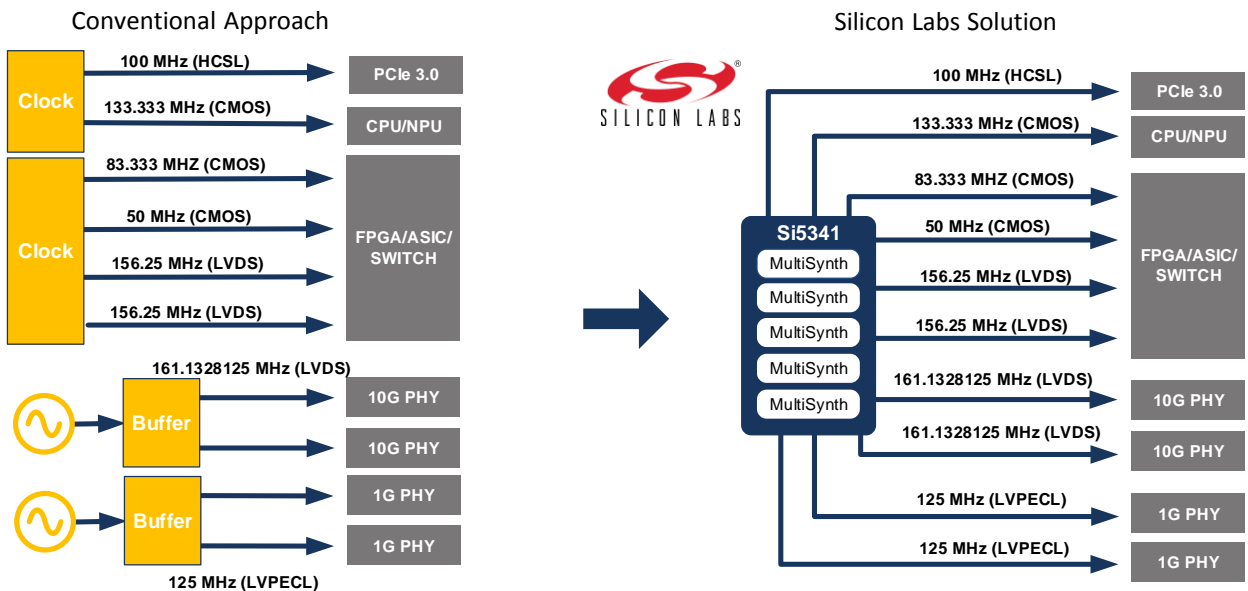
Optimizing Clock Trees – Example One

Clock trees can be highly complex or relatively simple, but in all cases they provide a fundamentally important part of the system and must be optimized for performance and cost.

Silicon Labs offers a comprehensive portfolio of timing products for all ranges of applications, from the most demanding to the most cost conscious.

Silicon Labs' unique MultiSynth IP allows for any-frequency input to generate any-frequency output to maximize flexibility and minimize cost.

Here is a real-world example of a traditional clock tree that Silicon Labs simplified into a single component, reducing space and cost while maintaining or even improving performance.



Clock Tree Challenges

- FPGA/ASIC/PHY require diverse mix of frequencies, formats
- High-speed 10G+ clocks must have very low jitter

Silicon Labs Solution

- MultiSynth generates any combination of frequencies
- Best-in-class jitter (100 fs RMS)
- 4–10 clock outputs

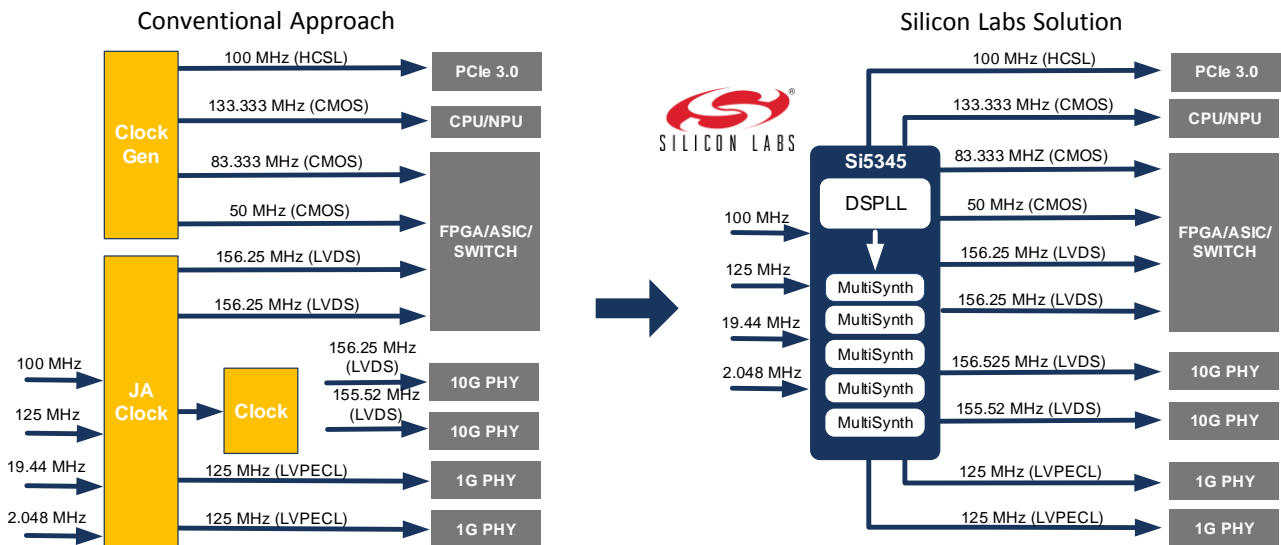
Optimizing Clock Trees – Example Two

Clock trees can be highly complex or relatively simple, but in all cases they provide a fundamentally important part of the system and must be optimized for performance and cost.

Silicon Labs offers a comprehensive portfolio of timing products for all ranges of applications, from the most demanding to the most cost conscious.

Silicon Labs' unique MultiSynth IP allows for any-frequency input to generate any-frequency output to maximize flexibility and minimize cost.

Here is a real-world example of a traditional clock tree that Silicon Labs simplified into a single component, reducing space and cost while maintaining or even improving performance.



Clock Tree Challenges

- Jitter cleaning
- FPGA/ASIC/PHY require diverse mix of frequencies, formats
- High-speed 10G+ clocks must have very low jitter

Silicon Labs Solution

- DSPLL accepts any frequency and cleans clocks
- MultiSynth generates any combination of frequencies
- Best-in-class jitter (100 fs RMS)

Conclusion

Silicon Labs' comprehensive timing portfolio provides optimized clock trees for the most demanding applications and the most cost-conscious applications.

Our solutions are easy to configure and customize, with most samples available immediately or within less than two days.

Our free tools will assist you in creating the right clock tree for your application.

And our experienced customer service experts are happy to help.

Contact us for your timing needs. We make timing easy.



Clocks

Single-chip clock generators, PCIe, jitter attenuators and network synchronizers.



Buffers

Differential, PCIe, LVCMOS, and Universal any-format buffers.



Oscillators

Any-frequency, ultra-low-jitter oscillators with short lead times.



About the Author

Linda Lua is the Silicon Labs product manager for datacenter timing products, managing the datacenter clock generators and clock buffers portfolio, new product launches, new product initiatives and marketing promotions.

Prior to joining Silicon Labs, Ms. Lua was at ISSI, responsible for High Speed Memory products, and at IDT Inc., responsible for timing products business development and product management in networking and the communications market.

Ms. Lua holds a BS in Electrical Engineering from Iowa State University and MBA from the University of Texas at Dallas.

Clock Tree Terminology

Before learning about clock tree design fundamentals, we should first take a moment to define common concepts.

Fanout--Fanout is a term that defines the maximum number of digital inputs that the output of a single logic gate can feed. Most transistor-transistor logic (TTL) gates can feed up to 10 other digital gates or devices. Thus, a typical TTL gate has a fan-out of 10.

LVPECL—LVPECL stands for Low-Voltage Positive Emitter-Coupled Logic, and it is a power optimized version of PECL or Positive Emitter-Coupled Logic. It uses a positive 3.3 V power supply.

LVDS—LVDS is Low-Voltage Differential Signaling, and it is only a physical layer specification, but a data link layer is often added by communication standards and applications.

CML—Current Mode Logic transmits data at speeds between 312.5 Mbit/s and 3.125 Gbit/s across standard circuit boards.

HCSL—High-Speed Current Steering Logic is differential logic with two output pins that switch between 0 and 14 mA.

LVC MOS—LVC MOS stands for Low Voltage Complementary Metal Oxide Semiconductor, and its goal is to reduce the device geometries of integrated circuits, with resulting reduction in operating voltage.