Clock Generation and Distribution

AD9510/11/12/13/14/15

PLL Core

- Why do I see reference spurs?
- Why is my phase noise shape changing when I change the PLL settings?
- Why doesn't the PLL make my reference input and the clock outputs line up?
- How do I optimize my PLL loop for the best phase noise and/or jitter?
- My loop is not locking. How do I debug this?
- How long does it take for the PLL to lock?
- Help! My PLL came unlocked over temperature.
- How do I choose between active and passive filter in PLL loop?
- Should I reference the passive filter to ground? or supply?
- How do the PLLs in the AD951x parts compare to other ADI PLLs?
- How does the clock clean-up function of the AD951x parts work?
- Why do I want to run a fast PFD frequency?
- Is it ok for me to connect the same power supply to both the charge pump and distribution power supply pins?

External Loop Filter

- Why can't I use a bandpass filter for my loop filter?
- Should I tie my loop filter to ground or PLL supply?
- The loop filter was working great until I changed the divide ratio in PLL. What happened?

External VCO/VCXO

- How do I use a VCO with a supply greater than 5V?
- What suppliers do you recommend for VCO/VCXOs?
- Do VCXOs have better phase noise and jitter performance than VCOs?
- How do I know which VCO will work best with the AD9510?
- Is there an advantage to running a higher VCO frequency than the output frequency?
- How do I determine if a VCO is good enough for my purpose?
- Is there any difference between the nature of an oscillator's phase noise and the phase noise from a clock chip?

Dividers/Phase Offset

- Do different divide ratios cause variations in jitter?
- I have a clocking scheme which requires several different division ratios simultaneously. I have a frequency plan, but I'm concerned about crosstalk. How much of a problem is this with your clock distribution chips?
- Do divide ratios change the propagation delay?
- I want to use the phase offset feature on the AD9510 dividers to generate two signals 90° out of phase. How accurate is the phase offset?
- On the AD951x clock ICs, does the phase offset (coarse delay) affect the jitter?
- Why doesn't the mini-divider support the divide ratio I want?

Delay Adjust

- I want to use the variable delay adjust, but the jitter is too high. What can I do?
- I changed the coarse phase adjust in the evaluation software, but nothing happened. What's going on?
- What is the difference between the coarse phase adjust and the fine delay adjust?
- What is the fine delay adjust which is available on certain LVDS/CMOS outputs?
- Does the fine delay adjust affect the jitter?
- Why is the fine delay adjust not available on all the outputs?
- Is there a way to cause Input/Output rising edges to be synchronous (zero delay) with the AD9510/11?

Inputs

- Will the AD9510 work without a reference input signal?
- What are the best clock sources for a distribution-only design?
- I am not using the CLK1 input on the AD9510. Can I just leave it floating?
- How good does my input signal need to be?
- I turned off my reference but the Digital Lock Detect (DLD) still says I'm locked.
- Can I shift the threshold on clocks for single-ended inputs?
- The reference input is differential, but my reference is single-ended. Do I need to convert to differential to drive the AD9510?
- Will differential or single-ended inputs/outputs improve my jitter?
- Why should I use differential rather than single-ended?

- How do I feed a single-ended signal into a differential input?
- Why do you recommend AC coupling, rather than DC coupling, at the clock inputs?
- Are the ADI clock parts stand-alone clock sources or do I still have to buy a clock source to drive these parts?
- Which provides better performance - a clock source with sinewave output, or one with differential square wave outputs?
- On the AD9510, what is the relationship between clock output jitter and CLK1/CLK2 input slew rate?
- I'm trying to write to the part in single-byte mode, but I can't write anything. What am I doing wrong?

Outputs

- Can I use the 951X clocks to drive a mixer (RF LO)?
- My applications are RF, not for clocking data converters. Can ADI's 951X ICs be used for RF applications?
- I have an input present at the clock input, but I'm not seeing an output?
- What happens to the AD9510/11 clock outputs if the Reference Input (REFIN) signal goes away?
- What clock frequency comes out of the AD9510 outputs when you first apply power to the device?
- Is it possible to impedance match a clock output if it is heavily loaded? (e.g. C_L=100pF)
- I ran the AD9510 outputs at 1.4 GHz and they seem to work fine. Is there a problem running them at 1.4 GHz?
- What should I do with unused channels on the AD9510?
- Can I tri-state the AD9510 outputs?
- On the AD9510, how can I make sure that the duty cycle of output clocks stays within 40% to 60% duty cycle window?
- What is the effect of distributing harmonically related clocks (on chip or on board) in terms of jitter?
- Is there any reason to use a transformer on a differential clock output to obtain a "clean" single-ended clock output?
- What are some of the advantages/disadvantages of using LVPECL vs. LVDS outputs?
- Does the AD9510 support 2.5V PECL?
- How much bandwidth is required to process a PECL or LVDS output?
- If I use only one of the PECL differential outputs and the unused output is terminated in 50Ω, how will this affect the phase noise or jitter of the single-ended output?
- If I change the level of PECL output, does it affect the jitter?
- What is the best way to terminate LVPECL outputs to get lowest jitter?
- Is it okay to AC-couple PECL or LVDS outputs?
- What is the fan-out capability of the CMOS, LVDS, and LVPECL outputs?
- What is the proper termination (value and location) for outputs?
- Are outputs short-circuit protected?
- Are the CMOS drivers on the clock devices complementary?
- Can I use 100 ohms to improve the slew rate (or jitter)?
- I have pulled SYNCB low, but I still have output from a channel. Why?
- Why can I not get the same output amplitude or rise and fall times as stated in your datasheet?
- The AD9510 datasheet says to use an external pull-up resistor on the FUNCTION pin. Why do I need this and what range of resistors will work?

Top of Section | Top of Page

PLL Core

Why do I see reference spurs?
Reference spurs occur at multiples of the frequency of the PFD and can be caused by any number of things:

- Charge pump up/down mismatch (greatest when the charge pump is operating outside compliance range)
- Coupling on the board
  - reference to output
  - reference to input, weak VCO signal
- Power supply coupling
- Loop filter components
- Loop bandwidth too wide
- Orientation of the loop filter with respect to ground or power supply

If the STATUS output is enabled and the PLL MUX is used to export a signal that is not DC, then this signal may couple to the clock distribution section. It is recommended that the STATUS output be used for static signals (such as digital lock detect) in applications where one is sensitive to PLL spurs.

Why is my phase noise shape changing when I change the PLL settings?
A PLL is a closed-loop feedback system, and as such should be designed to be stable under a certain set of operating condition. Any change in the loop parameters that effects the closed loop bandwidth will change the shape of the phase noise.

Frequently, the loop is designed to be stable by specifying the desired closed loop bandwidth and phase margin, a method supported by ADIsimCLK. When the loop filter is designed for the PLL, the user specifies the charge pump current, PFD frequency and VCO Kvco in order to get a stable design. If the user changes any of the PLL settings that affect these terms, then they are actually changing the closed loop bandwidth of the PLL and the phase margin.

Assuming the loop was designed with sufficient phase margin, small changes in loop parameters usually just result in small changes in the closed loop bandwidth, and hence the shape of the phase noise, without making the loop become unstable (however the user should make sure their design is actually sufficiently stable for these new setting by examining the loop's stability using ADIsimCLK or another PLL design tool). Large changes in loop parameters will result in drastic changes in the loop bandwidth or make the entire loop unlockable. Both of these will change the shape of the phase noise.

Why doesn't the PLL make my reference input and the clock outputs line up?
The PLL will make the reference and input clock edges align at the input of the phase-frequency detector (PFD). The path from the input clock to the output is different than the path from the input clock to the PFD. Therefore, the propagation delay difference between these two paths is not compensated for by the PFD. As such, the AD9510/11 are not "zero-delay" clock parts.

How do I optimize my PLL loop for the best phase noise and/or jitter?
There are many tools one can use to optimize a PLL loop. ADIsimCLK is a good tool to use for the ADI clock parts. Optimizing phase noise and jitter is not necessarily the same thing.

If one has a spec for phase noise at a given offset frequency, then one should supply the VCO and Reference phase noise information to a tool, such as ADIsimCLK, and use this to optimize the closed-loop bandwidth to achieve the desired targets. This process is in essence adjusting the closed-loop bandwidth to trade off the reference and VCO phase noise.

If one has a jitter spec, then the closed loop bandwidth should be adjusted to achieve the lowest jitter, which may not necessarily correspond to what appears to be the lowest phase noise for all offset frequencies.

For example, while it may be possible to achieve low close-in phase noise by extending the closed-loop bandwidth, the resulting jitter may be larger than the minimum possible because the loop is tracking the reference more than is necessary for optimal

jitter. One might achieve lower jitter by reducing the closed-loop bandwidth, allowing the PLL to track the VCO at lower offset frequencies, even though the resulting phase noise plot might show more peaking at the closed-loop bandwidth offset frequency.

My loop is not locking. How do I debug this?
Make sure Vcp is connected. Use the STATUS pin and the PLL MUX output. Check N and R divider outputs to ensure they are at the same frequency and are aligned. Be sure to examine the digital lock detect signal using an oscilloscope. Also use ADIsimCLK to simulate your loop.

How long does it take for the PLL to lock?
The PLL lock time is driven by many elements. The following are some keys items that determine the PLL lock time.

- PLL loop bandwidth is usually the dominant element.
- Phase margin settings - impact settling time by altering the damping (under, over, or critical) of the loop.
- Lock time is also impacted by the value of the control voltage at time zero, and the absolute phase difference between the phase detector inputs at time zero.

The best way to determine the first order PLL lock time is run ADIsimCLK. It has the capability to run lock time analysis.

Help! My PLL came unlocked over temperature.
This is likely from a loop that was not sufficiently stable to handle changes in the PLL loop parameters over temperature (such as VCO Kvco). The loop was likely marginally stable to begin with (see question regarding phase noise changing with loop parameters). Other possibilities are a VCO that is not capable of guaranteeing the target frequency over temperature.

How do I choose between active and passive filter in PLL loop?
Applications requiring ultra low jitter performance use passive filters vs active filters as passive filters only contribute thermal noise from its components. In an active filter, input referred thermal noise and 1/f noise from transistors is increased by the gain of the circuit. This translates into more noise on the oscillator's voltage control pin which randomly frequency modulates the output signal, resulting in more phase noise.

Active filters are typically used when the oscillator's(VCO/VCXO/TCXO) control voltage range is outside the range of the charge pump supply. In these cases, the charge pump current cannot drive the oscillator's output over its full frequency range. The active filter provides gain or a level shift to exploit the oscillator's full control voltage range.

The AD9510/11 allows the use of a 5V separate Vcp power supply for VCOs requiring a 5V tuning range.

Should I reference the passive filter to ground? or supply?
The passive filter should be referenced according to the oscillator's control pin (tuning voltage). If the oscillator is ground referenced with respect to the control voltage, the passive filter should be grounded. If the oscillator is referenced to the supply, the passive filter should be referenced to the supply side. By doing so, better common-mode rejection is achieved.

For example, if there's noise on the ground to which the passive filter is referenced to, it will couple onto the control voltage input to the oscillator. If the oscillator is ground referenced and the passive filter is grounded, the noise appears as correlated noise across the oscillator input to ground which minimizes the noise impact, else any differential impact between input and ground will degrade performance.

The orientation of the filter can also affect the reference spurs in a similar manner.

Whether the oscillator is referenced to ground or supply is not always available in the data sheets and therefore may require calling the manufacturer.

How do the PLLs in the AD951x parts compare to other ADI PLLs?
The design of the PLL core used on the AD9510/AD9511 began with the ADF4106. There were modifications made to the design to suit it better for its role in the clock ICs. There are some configuration differences between the PLL on the AD9510/11

and the ADF4106, and its maximum VCO frequency is 1600 MHz rather than 6 GHz. But the performance characteristics of the ADF4106 were maintained, so that the PLL on the AD9510/11 provides performance nearly identical to the ADF4106.

How does the clock clean-up function of the AD951x parts work?
The AD9510 and AD9511 include an on-chip PLL core which requires an external VCO or VCXO and a loop filter to configure a PLL loop. This PLL loop can be used to considerably clean up a reference clock signal which has picked up noise as it has been distributed. This has the effect of reducing the phase noise and time jitter on a noisy clock reference.

The PLL loop must be designed around the PLL core characteristics and the selected VCO/VCXO. Often, the VCO/VCXO frequency is higher by some integer multiplier (N) than the reference clock frequency. A suitable loop bandwidth must be selected in order to design the loop filter.

The noise on the reference clock signal will be suppressed outside the PLL loop bandwidth. However, the noise on the reference within the loop bandwidth will be gained up by the ratio 20*log(N), where N is the frequency multiplier of the loop. The amount of cleanup that can be achieved depends on both the narrowness of the loop bandwidth and on the frequency multiplication. The stability of the loop and the settling time of the loop, as well as the practicality of the component values of the loop filter, are also affected by the loop bandwidth.

Outside of the loop bandwidth, the VCO/VCXO phase noise will predominate. So, it is important to select a VCO/VCXO with low broadband phase noise. As with all engineering tasks, a suitable compromise among all of the factors involved must be achieved. A good clock reference cleanup design can result in significant system performance improvements. The locally distributed clock time jitter can often be reduced to the order of 1 ps rms or better.

Remember that ADIsimCLK can help you design your PLL loop, and simulate its performance.

Why do I want to run a fast PFD frequency?
The phase noise from the PFD, reference path, and feedback divider is reduced by running a faster PFD frequency. This can be seen from the PLL FOM (figure of merit) from the datasheet, which gives,

In band PN = FOM + 10log(Fpfd)+20logN-20log(DIV)

Where Fpfd is the PFD frequency and N is the N divider (feedback divider) setting (ratio of VCO frequency to PFD frequency), and DIV is the distribution divider setting. From this it can be seen that while increasing the PFD frequency 2X raises the in-band PN 3dB, the N will also reduce by a factor of 2, giving a net 3dB REDUCTION in the in-band phase noise. Therefore, for the same VCO frequency, every 2X increase in PFD frequency reduces the in-band phase noise 3dB.

Is it ok for me to connect the same power supply to both the charge pump and distribution power supply pins?
The clock distribution section is powered by 3.3V. The max charge pump supply setting is 5V to accommodate 5V VCOs and VCXOs devices. If the designer sets the charge pump supply at 3.3V and good supply decoupling exist, powering the two circuits with one supply would be left up to the designer's discretion. It is always nice to reduce supply domains, however the charge pump is a sensitive node that should be free from excessive noise and switching which could easily transfer to the output. It should be evaluated case by case.

Top of Section | Top of Page

External Loop Filter

Why can't I use a bandpass filter for my loop filter?
A PLL uses a VCO to steer its output frequency in order to obtain a phase locked condition with a reference signal. The VCO generates a frequency proportional to the DC level applied to its control input. In a PLL, the VCO is essentially controlled by the DC level at the output of the loop filter. A bandpass filter, by definition, does not pass DC. Hence, if a band pass loop filter is used there is no DC signal available to control the VCO.

Should I tie my loop filter to ground or PLL supply?
The orientation of the loop filter depends on the design of the VCO. A well-designed supply provides a low impedance path to

ground. Therefore any supply will serve as an AC ground. However, supply variations with respect to the VCO ground can affect the spurious performance of the system. Often, the VCO is ground referenced, so the loop filter should also be referenced to ground. In some cases the VCO will be referenced to a supply (such as from a regulator). In that case, tying the loop filter to the supply MAY provide improved spurious performance.

The loop filter was working great until I changed the divide ratio in PLL. What happened?
A PLL is a closed loop feedback system and is stable under a limited set of conditions. The selection of the loop filter must take into consideration the PLL divider and charge pump settings. Frequently the loop will appear stable over small changes in the dividers and the charge pump setting, however actual stability should be verified using a tool such as ADIsimCLK. Even though the loop may remain stable with these changes, the loop bandwidth and phase margin will most likely change.

Top of Section | Top of Page

External VCO / VCXO

How do I use a VCO with a supply greater than 5V?
The AD9510/11 parts allow the use of a Vcp up to 5V. To use VCOs which require tuning voltages greater than 5V an active loop filter must be used. This architecture makes use of op amps which can provide higher voltage output levels, but require an additional power supply voltage. A caveat though: active loop filters tend to be much noisier than passive loop filters and can contribute to higher jitter levels.

What suppliers do you recommend for VCO/VCXOs?
ADI does not manufacture stand-alone VCO/VCXOs, however, we do utilize products from different manufacturers. We can not guarantee the operation or performance of any other manufacturers' parts, but we can provide a list of manufacturers of VCO/VCXOs with which we are familiar: Epson-Toyocom, Sirenza, Valpey-Fisher, Vectron, Zcomm. Keep in mind there are many others which may be considered.

Do VCXOs have better phase noise and jitter performance than VCOs?
It depends. Different devices have different close-in versus wideband phase noise. Often, this is an observable difference between VCOs and VCXOs. However, an important difference between the two is that VCOs typically have much higher gain (Kvco) than VCXOs. VCXOs usually have a very narrow tuning range. This allows for much narrower loop bandwidth to be easily attained, and therefore better close-in phase noise performance. VCOs, on the other hand, tend to have a lower wideband noise floor. In both cases, this wideband phase noise can be filtered with bandpass filters thereby yielding a lower total jitter.

How do I know which VCO will work best with the AD9510?
The VCO must be well-matched to the frequency and phase noise performance required by the system. The VCO datasheet should give important information such as the gain (Kvco), tuning range, and both close-in and wideband phase noise. Lower gain tends to produce loops with better phase noise performance but will constrain the frequency range. Wideband phase noise causes more jitter than close-in phase noise, because jitter is the result of the total integration of phase noise power over a broad bandwidth. However, system requirements may enforce low close-in phase noise, so this may be a consideration as well.

Is there an advantage to running a higher VCO frequency than the output frequency?
Yes. As a clock input signal traverses the switching region of an input stage, random noise in the input stage adds time jitter to the signal as it is passed along to subsequent stages. A signal which traverses this switching region faster (has higher slew rate) is less affected by the random noise, resulting in less added time jitter, and lower jitter at the clock outputs. For a signal of the same amplitude, a higher frequency signal has a higher (faster) slew rate, and therefore can result in lower time jitter through the clock distribution system.

Any jitter added at the input to a clock distribution system cannot be reduced by subsequent division of the clock frequency (phase noise does reduce with division, but not jitter). So, the jitter should be reduced as much as possible at the input.

This means that in principle there is an advantage to running the VCO at a higher frequency than that required at the clock outputs.

How do I determine if a VCO is good enough for my purpose?

The VCO which I'm considering doesn't have a phase noise graph in its data sheet.

Determining the phase noise performance of a VCO or VCXO can be difficult. If possible, try to get a plot or set of plots of phase noise for the VCO/VCXO from the manufacturer. If this information is not in the product data sheet, you will have to contact the vendor application support to request this.

Alternatively, if you have access to a phase noise measurement system, such as the Agilent E5500, you can make the phase noise plots yourself. This is a pretty involved process, though.

A growing number of available VCOs and VCXOs from the leading vendors have been characterized and included in the library of ADIsimCLK. Check there to see if your VCO/VCXO has been included. If it hasn't yet been included, let Apps Support know so that we can put it on our list for future inclusion in the library.

When the phase noise values for your VCO/VCXO are known, use ADIsimCLK to simulate the jitter performance of the ADI clock distribution IC using that component. For VCO/VCXO parts which are not yet included in the ADIsimCLK library, the phase noise can be manually entered in the appropriate spot in the program's user interface, and the simulations continued. The effect of various amounts of VCO/VCXO phase noise can easily be seen and appreciated in the results given by ADIsimCLK. This will help you to determine a suitable candidate for your VCO/VCXO.

Is there any difference between the nature of an oscillator's phase noise and the phase noise from a clock chip?
Yes. There are different factors contributing to the phase noise in an oscillator as contrasted to an amplifier or buffer. An oscillator consists of a resonator with gain and feedback. There has been much research into the complex nature and causes of micro instabilities resulting in oscillator phase noise. Some of these causal factors are random (thermal) noise and active device noise, but there are many other factors, including multiple resonator modes, acoustic effects, flicker noise, etc. The result of these instabilities and noises is to perturb the phase of the signal produced by the oscillator. These phase perturbations have the effect of spreading the clock signal out in the frequency domain with a distinctive spectral density profile. This spectral density profile is interpreted as the phase noise of the oscillator.

On the other hand, the causes of noise in an amplifier or buffer or other non-resonant stage are somewhat different. As a signal passes through these non oscillating stages, random (thermal) noise and active device 1/f noise adds to and subtracts from the amplitude of the signal at each instant. The result is also to spread out the signal in the frequency domain, but with a somewhat different spectral density profile. The result is an additional background of noise in a continuous spectrum which overlays the original clock signal. The resulting spectral density profile can also be interpreted in terms of phase noise.

In both cases, oscillator and clock system, the observed smearing out of the signal in the frequency domain can be treated as phase noise. In the time domain, the power of the phase noise (integrated over an appropriate bandwidth) results in variations in the time domain. This effect is called time jitter.

Top of Section | Top of Page

Dividers / Phase Offset

Do different divide ratios cause variations in jitter?
Ideally, no. Jitter remains the same as a signal is divided by different ratios. However, phase noise is frequency dependent. With the AD951x parts, there is a slight jitter difference between the divide-by-1 case (divider bypassed) and the 2-32 ratios. When a 2-32 divide-by is chosen there is a slight jitter contribution by the divider, but this is ratio independent. This is because the signal path is different than in the divide-by-1 case which does not incur the additional jitter contribution of the divider.

I have a clocking scheme which requires several different division ratios simultaneously. I have a frequency plan, but I'm concerned about crosstalk. How much of a problem is this with your clock distribution chips?
The AD951x clock chips have been designed to minimize output to output crosstalk. However, there are differences in the level of crosstalk depending on the combination of outputs and output types (LVPECL, LVDS, CMOS) which are being used. The device data sheets give some indication of the crosstalk performance in the specifications table by giving jitter values with and without interferers present. Of course, this is just indicative of the achievable performance, because there are a very large set of different frequencies and output combinations which can be encountered.

In general, though, the LVPECL outputs are virtually unaffected by any other LVPECL outputs. The LVPECL, LVDS and CMOS outputs are more affected by other LVDS and CMOS outputs.

Do divide ratios change the propagation delay?
If the divide ratio is greater than 1, the propagation delay is consistent for all divide ratios up to 32. Propagation time is measured from the 50% point of rising edge of the clock input to the 50% point on the rising edge of the output.

For a divide-by-1, the propagation delay is a little less than when a higher divide ratio is used. For the AD9510/11/12, this amounts to approximately 55 ps difference.

I want to use the phase offset feature on the AD9510 dividers to generate two signals 90° out of phase. How accurate is the phase offset?
Very. The phase difference is actually based on the successive edges of the input clock. The only error will be variations in the propagation delay between the two channels.

On the AD951x clock ICs, does the phase offset (coarse delay) affect the jitter?
No. The phase offset, or coarse delay, adjustment on the AD951x is a function built into the divider for each output. Depending upon the selected divide ratio, there are from 2 to 32 edges of the incoming clock which can be chosen as the trigger edge for the output. This allows for setting a phase offset, or time delay, between outputs. This feature is entirely contained within the divider block itself. These dividers are designed so that this feature does not affect the jitter of the output. The jitter performance of the output will be the same regardless of the phase offset (coarse delay) selected.

Why doesn't the mini-divider support the divide ratio I want?
The reason that not all divide ratios, or other settings, are available on the mini-dividers is that the 11 4-state logic must be decoded internally, and there is a limited decode space available. Because not every single feature could be accommodated a reasonable subset of all the possible device settings is provided. The AD9513 and AD9514 provide all divide ratios from 1 to 32 on OUT2 only. The AD9515 provides all divide ratios from 1 to 32, except for divide-by-13, on OUT2. The other output channels vary in their divide-by values. See the data sheet for a specific product (AD9513, AD9514, AD9515) for explicit details.

Top of Section | Top of Page

Delay Adjust

I want to use the variable delay adjust, but the jitter is too high. What can I do?
Have you thought about using the edge selection (coarse phase adjust)? This feature will allow you to specify which rising edge on the reference clock generates the rising edge in your output. The resolution of this adjustment feature is the system clock period, which can be sufficient for many applications. This feature is available on all channels. Use of the edge select has no impact on the jitter of the output channel.

I changed the coarse phase adjust in the evaluation software, but nothing happened. What's going on?
The coarse phase offset adjust on the AD951x ICs depends on the divide ratio selected. It allows for the selection of which input clock edge is the trigger for the output from the divider. The number of edges which are available for selection is dependent upon the divide ratio. The phase offset is determined by 4 bits in the register for each divider. For a divide-by-2, there are only 2 choices. For a divide-by-15, there are 15 choices. By adding the use of the Start H/L bit, there are more choices, but it becomes more complex.

However, if the divider is bypassed there are no choices. The case of divide-by-1 (divider bypassed) does not allow for a phase adjust. Also, when divide-by-n is set, selecting a phase offset of n or 2n, etc. does not look like it does anything, because the phase offset selected is a whole number of cycles of the input clock signal. For example, when divide by 5 is selected, a phase offset of 5 looks the same as a phase offset of 0.

So, to answer your question: either the divider is bypassed (divide-by-1) or the selected phase offset is a multiple of the divide ratio.

What is the difference between the coarse phase adjust and the fine delay adjust?
On the AD951x clock ICs, each output has a divider which features a coarse phase (or delay) adjust which allows for the selection of a given clock edge within the divide period as a delay or phase offset for that output. This feature adds no jitter to the clock output, but only allows for discrete delays in steps the size of the input clock period.

However, certain outputs also have a feature called a fine delay adjust. This feature is only available on certain LVDS/CMOS outputs, and never on an LVPECL output.

What is the fine delay adjust which is available on certain LVDS/CMOS outputs?
The fine delay adjust is provided as a way to set a variable delay on that particular output with respect to another output or outputs. It is set up by selecting an appropriate full-scale delay, and then selecting a fraction of that full-scale as the actual delay.

The fine delay adjust is implemented in an analog block, which uses a programmable current and a selectable number of capacitors, to create a voltage ramp. When integrated by the capacitor(s), the current will create a voltage which increases linearly with time (a voltage ramp). The higher the current, the faster the voltage will increase. The more capacitors selected, the slower the voltage will increase. By selecting an appropriate current value, and number of capacitors, a ramp with a desired voltage-vs-time ramp is created. This sets the full-scale delay time.

When a fine delay is selected for a given clock output, that output is held until the fine delay period is completed. A delay period begins when an input clock edge triggers the current generator, beginning the voltage ramp. The voltage ramp is then applied to one input of a comparator. A 5-bit DAC creates the other input for the comparator. When the ramp voltage equals the DAC voltage, the delay is terminated, allowing the clock to continue. Therefore the voltage output of the DAC sets the actual time delay, which is a fraction of the full-scale delay. At the end of the delay, the capacitors are discharged and the current source is turned off. The cycle begins again.

The mode of operation of the fine delay block also sets a restriction on the amount of delay which can be obtained. In general a delay of up to one-half of the output clock period is achievable. Also, there is a maximum input clock frequency at which the delay block will no longer reliably reset and allow another trigger. This maximum frequency of the input clock is between 450 and 500 MHz. When this maximum frequency is exceeded the clock output will stop toggling.

Does the fine delay adjust affect the jitter?
Yes. Because this fine delay adjust is an analog function, it contributes a significant amount of jitter to the clock output on which it is used (but only when it is used; there is no effect on jitter when the delay block is off). The longer the delay, the more the jitter. This is why this type of delay is not provided on all of the outputs.

Users can see this relationship using the free ADIsimCLK software available on the website: www.analog.com/ADIsimCLK

This feature is intended to be used to drive digital devices such as ASICs, FIFOs, FPGAs, etc. where a delay is needed in order to synchronize to digital data. It is not recommended for encode clocks on converters, due to the added jitter.

Why is the fine delay adjust not available on all the outputs?
This feature is intended to be used to drive digital devices such as ASICs, FIFOs, FPGAs, etc. where a delay is needed in order to synchronize to digital data. It is not recommended for encode clocks on converters, due to the added jitter.

Is there a way to cause Input/Output rising edges to be synchronous (zero delay) with the AD9510/11?
The AD9510/11 do not have a zero delay feature. However, it is possible to achieve close to zero phase delay between REFIN and outputs by using one of the outputs as the feedback to the PLL.

The PLL loop will force that output to be very close to zero delay, and the other outputs will track within their skews.

Top of Section | Top of Page

Inputs

Will the AD9510 work without a reference input signal?
If you already have the final master frequency and do not require the PLL portion of the clock distribution chip, yes, the AD9510 will work without a reference. Simply bring the master clock signal to the CLK1 or CLK2 inputs and power down the PLL portion of the chip. Incidentally, the AD9512 is a chip that is distribution only, it has 3 LVPECL outputs and 2 LVDS/CMOS outputs. If this is a sufficient number of outputs, it might make sense to consider the AD9512 as an alternative.

If you do not have the final master frequency, and you intend to use the PLL, then you will need to provide the AD9510 with some sort of stand alone external reference. This reference should be supplied to the AD9510 on the REFIN input.

What are the best clock sources for a distribution-only design?
There are numerous potential sources that could be used to drive a distribution only design: a closed loop PLL employing a VCO or a VCXO, a stand alone oscillator or source, a recovered signal from a CDR, etc. The best clock source will vary based on the application needs. For lowest jitter, VCXO's operating in a closed loop tend to have the best performance.

I am not using the CLK1 input on the AD9510. Can I just leave it floating?
Generally speaking, this is a bad idea. Allowing unused inputs to float, particularly sensitive nodes such as low jitter clock inputs, raises the possibility of external noise coupling into the device. This concern can be resolved by simply applying a .01uF capacitor from CLK1 and CLK1B pins to a clean DC supply or ground plane such as AVDD or AGND. This will provide AC ground locally to the input, ensuring that noise will not be able to couple into the device.

How good does my input signal need to be?
[The clock ICs with a PLL on-chip, along with an external VCO/VCXO and loop filter, allow for the cleaning up of a jittery clock reference, which is then divided, delayed and distributed by the distribution section of the IC. Other questions in this FAQ address the clock cleanup process. This question will address the distribution section only.]

The clock distribution channels of the ADI clock ICs have very low additive phase noise or jitter. However, these circuits can only add noise (jitter and phase noise), not subtract it. The output from the distribution section can be nearly as good in jitter as the input. We specify this as "additive jitter" or "additive phase noise". The additive jitter of an LVPECL output is around 215 fs RMS.

Random noise adds to random noise as the root sum of the squares (RSS). So the input jitter must be RSS'd with the additive jitter of the clock channel to get the resulting clock signal jitter at the output.

So, your input clock signal needs to be good enough so that when the additive jitter of the clock channel is RSS'd with it, that the result is within your system tolerance.

I turned off my reference but the Digital Lock Detect (DLD) still says I'm locked.
The DLD circuit is a clocked circuit that requires the Reference be present in order to operate. It is possible to use the "DLD or Loss of Lock" signal to indicate when the reference has been lost. In this mode the charge pump will also be automatically tri-stated and a user intervention will be necessary to take the charge pump out of tri-state (see datasheet).

Can I shift the threshold on clocks for single-ended inputs?
No. All clock inputs have a differential input. Both inputs have an internal DC bias or common mode voltage which is present on the input pins. Therefore the clock inputs are designed to be AC-coupled with a capacitor (0.1uf or 0.01uf). If a single-ended interface is desired, the unused input should be AC-coupled to ground.

The reference input is differential, but my reference is single-ende*d. Do I need to convert to differential to drive the AD9510?*
Strictly speaking, it is not required that a conversion be used, but it is recommended, especially at frequencies >50 MHz. Using a transformer to convert your single-ended reference to differential will give better noise suppression on the reference. However, if cost and board space are your critical concerns, you could use a .01uF capacitor to couple the unused input to a clean analog supply or ground plane, such as AVDD or AGND.

Will differential or single-ended inputs/outputs improve my jitter?
Yes. In general, using differential inputs and outputs will improve the jitter, due to lower noise susceptibility, higher slew rates, and greater voltage swing.

Why should I use differential rather than single-ended?
Apart from the maximum frequency of each output type, there are general advantages to using differential connections between inputs and outputs. The advantages are due to increased signal swing, higher slew rate, and the use of differential transmission lines. Differential transmission lines offer reduced EMI production, reduced sensitivity to induced or coupled noise, and better impedance matching (less distortion of edges due to signal reflections).

Whenever differential connections are used between devices which are separated by more than a very minimal distance, it is important that controlled impedance coupled differential transmission lines be used for the interconnect. This type of interconnect is easy to implement using microstrip techniques on PCB. It is important to terminate the far-end (receiver side) of a differential transmission line with its characteristic impedance in order to eliminate reverse reflections. In general, the transmission line is NOT matched at the output (transmitter) side.

Whenever a differential clock signal is not available to drive a differential input on one of the AD951x parts, it is possible to drive the input with a single-ended signal. This is done by AC coupling the single-ended signal into one of the differential inputs. The other differential input is connected via a capacitor to a quiet ground. The inputs are self-biased, so no common mode voltage must be set. The performance of the input is not reduced by this technique.

LVPECL and LVDS outputs are differential. CMOS outputs are single-ended (although an inverted output is also available; it is not a common differential driver, which means that the two outputs are not guaranteed to switch simultaneously). The inputs (REFIN, CLK, CLK1, CLK2) are also differential.

How do I feed a single-ended signal into a differential input?
It is possible to drive a differential input with a single-ended clock signal. This is done by AC coupling the single-ended signal into one side of the differential input. The other side of the differential input is connected via a capacitor to a quiet ground. The inputs of the AD951x series of parts are self-biased, so no common mode voltage must be set. The performance of the input is not reduced by coming in single-ended rather than differential.

Alternatively, a single-ended clock signal can be converted to differential by using a balun or transformer. The differential signal should be AC coupled into the inputs because they are internally biased.

Why do you recommend AC coupling, rather than DC coupling, at the clock inputs?
AC coupling is recommended for the clock inputs on the AD951x parts because these inputs are internally self-biased. By AC coupling the input signal, there is no need to match the common mode voltage of the inputs, and there is no need to set the common mode voltage of the inputs. Only the voltage swing (amplitude) of the signal must be considered, lest the input range be exceeded.

DC coupling is possible because the self-biasing internal network has an impedance of about 1.5k ohms. As long as the source common-mode voltage is within the common-mode range of the input, it will work.

Are the ADI clock parts stand-alone clock sources or do I still have to buy a clock source to drive these parts?
While it is true that all ADI clocking products need some reference signal, many of the devices include the necessary circuitry on chip to excite an external crystal reference. When we refer to a device as a clock generator we mean that the device has significant flexibility in terms of the exact fractional relationship between the reference and output frequencies.

Which provides better performance - a clock source with sinewave output, or one with differential square wave outputs?
An important factor in reducing jitter is the slew rate of the input signal. PECL outputs tend toward a square waveform, which, along with their typical 800 mV amplitude, have good slew rates. In addition, PECL is usually used differentially, which improves the slew rate further. This makes PECL a good choice for a clock source.

However, some clock sources are available only in sinewave output versions. Sinewave outputs can certainly be used, either differential (when available) or single-ended. With sinewave signals, the slew rate is dependent on the frequency and the amplitude of the signal. For a given frequency, a higher amplitude signal has a higher slew rate. The amplitude of a sinewave clock source is therefore of importance. Often, the output of a sinewave source is specified in dBm, which is a power output. This can be converted to a voltage swing if the impedance of the operating environment is known.

So, it is OK to use either a differential square wave, or a sinewave, clock source. Slew rate is the key to improved jitter performance, all other things being equal.

On the AD9510, what is the relationship between clock output jitter and CLK1/CLK2 input slew rate?
The higher the slew rate at the CLK1/CLK2 inputs, the lower the jitter, in general. This is because the input signal transitions through the decision point more quickly and therefore is less subject to disturbance by noise. The datasheet specs require a minimum of 1 V/ns slew rate at these inputs.

I'm trying to write to the part in single-byte mode, but I can't write anything. What am I doing wrong?
Remember that a communication cycle consists of 16-bits of instruction and 8-bits of data (data byte). Make certain that when you finish the 1st data byte, you bring the CS-bar pin high, and then low before attempting to write the 2nd data byte (in another communication cycle). If you do not toggle the CS-bar pin, the serial port controller will flush any new writes, and only the very last write (which is the update itself) will be recorded.

Top of Section | Top of Page

Outputs

Can I use the 951X clocks to drive a mixer (RF LO)?
Yes. Most RF mixers do not convey amplitude information from the LO, just the phase and frequency. In this sense, the output of the clock drivers makes a great programmable LO source, up to their maximum frequency capability.

My applications are RF, not for clocking data converters. Can ADI's 951X ICs be used for RF applications?
Yes, in many cases they can. However, our clock parts only cover frequencies up to 1600 MHz or so, depending on the part. The phase noise of the clock parts is very low, comparable to some of the best LO sources. If a sinusoidal output is desired a low-pass filter will convert the square wave output to a sinusoid.

I have an input present at the clock input, but I'm not seeing an output?
If the output is not switching, then one of the following could be the problem:

- The FUNCTION pin must be driven high or else the part is in reset mode. The FUNCTION pin has an internal 30k ohm pull-down resistor. The pin must be driven high to keep the part out of reset mode.
- The output is not terminated correctly. Refer to data sheet for proper termination.
- Coupling to the clock input is incorrect. The clock inputs must be ac-coupled unless otherwise specified.
- The output stage could have been inadvertently powered-down. Re-program.
- The output has been shorted to ground and damage has occurred to the output stage. Note, LVPECL and CMOS outputs are not short-circuit protected.

What happens to the AD9510/11 clock outputs if the Reference Input (REFIN) signal goes away?
When the reference input is lost, the PLL will go out of lock. However, as long as power remains to the VCO, it will continue to drive the CLK2 input. Therefore, the clock outputs will have a signal but the frequency will not be fixed. It is possible for the user to monitor the reference externally and put the charge pump in tri-state upon loss of reference. This is a form of holdover which allows the control voltage to float. However, the PLL frequency will drift slowly. The drift in frequency is set by the range of the oscillator, so a VCXO will have much less drift range than a VCO.

What clock frequency comes out of the AD9510 outputs when you first apply power to the device?
The PLL is disabled on power-up and CLK1 input is selected. If there is no signal on CLK1 there can be chatter which will show up on the enabled outputs. This can be prevented (if CLK1 will not be used) by tying the inputs somewhat apart in voltage so that it will not pick up noise and cause chatter. If CLK1 is the desired input and a signal is present at power-up, then this signal will drive the outputs and the frequencies present will be according to the preset divide ratios.

Is it possible to impedance match a clock output if it is heavily loaded? (e.g. C_L=100pF)

In the classical (microwave) sense of the phrase "impedance match," the answer is no. Generally impedance matching is viable over relatively narrow bandwidths. The rich (odd) harmonic spectrum of a nearly ideal square wave, which is highly desirable in most clocking applications due to the high slew-rate edges, occupies a wide bandwidth. A narrowband impedance match would effectively filter the square wave, causing the final clock output to become more sinusoidal, and consequently reduce the slew-rate.

Heavy loads, such as C_L=100pF, are best avoided. Devices with high input capacitance should be discarded and careful PCB layout is necessary to reduce board parasitics. Clock distribution parts such as the AD9510 can be used to drive several devices from a single reference, thereby solving the fan-out issue.

Top of Page | Main Index

I ran the AD9510 outputs at 1.4 GHz and they seem to work fine. Is there a problem running them at 1.4 GHz?
The 1.2 GHz spec limit guarantees operation over process and temperature variations. Of course at non-extreme temperatures the outputs will operate to higher frequencies. However, ADI does not assume responsibility for any performance beyond the datasheet limits.

What should I do with unused channels on the AD9510?
Unused channels can be turned off individually. These "off" channels do not have to be terminated.

Can I tri-state the AD9510 outputs?
Not exactly. The LVPECL channels have both a "safe" and "unsafe" power-down mode. The unsafe mode is essentially tri-state, but care must be taken not to exceed the reverse bias limits on this node or device damage may occur. In the safe power-down mode, the outputs are protected, but not tri-state. The LVDS and CMOS outputs when powered-down are effectively in tri-state.

On the AD9510, how can I make sure that the duty cycle of output clocks stays within 40% to 60% duty cycle window?
A divide-by-3 will produce a duty cycle of 33.3%-66.6%. A divide-by-5 produces exactly 40%-60%. Higher odd divide ratios (7, 9, 11...) produce duty cycles within the 40%-60% window.

What is the effect of distributing harmonically related clocks (on chip or on board) in terms of jitter?
Harmonically related clocks in general do not produce extra jitter. However, the variation in propagation times can cause edges to occur noncoincidentally. This can possibly produce jitter if the amplitudes of the various edges are too high at a given node.

Is there any reason to use a transformer on a differential clock output to obtain a "clean" single-ended clock output?
Generally differential signaling is used for noise immunity. However, some systems require a single-ended input. If a transformer is selected that has adequate bandwidth and low insertion loss, the slew-rate and signal power of the differential signal can be approximately maintained.

What are some of the advantages/disadvantages of using LVPECL vs. LVDS outputs?
LVPECL has the advantage of swing amplitude over LVDS (800mV versus 350mV). This produces a higher effective slew rate at a receiver and can result in lower jitter. LVPECL also offers higher clock rate than LVDS (AD9510/11/12: 1.2 GHz LVPECL versus 800 MHz LVDS).

LVDS offers the advantages of lower power dissipation and simpler termination versus LVPECL. Both have the advantage of being fully differential.

Does the AD9510 support 2.5V PECL?
No, 3.3V only. However this is compatible with 2.5V PECL receivers with AC coupling.

Top of Page | Main Index

How much bandwidth is required to process a PECL or LVDS output?
Assuming a first-order system, the bandwidth can be determined by accounting for the 10-90% rise/fall time of the signal:

$$f_{BW} = \frac{2.2}{2\pi\Delta t}$$ where f_{BW} is the bandwidth, Δt is the rise/fall time

If I use only one of the PECL differential outputs and the unused output is terminated in 50Ω, how will this affect the phase noise or jitter of the single-ended output?
The output signal swing is now halved. This can have phase noise or jitter implications for the receiving circuitry. It is important to note that the rise/fall time of the single-ended PECL output are not equivalent; generally the rise time is much smaller than the fall time. If the receiving circuit uses the single-ended falling edge for critical timing this may be a problem.

If I change the level of PECL output, does it affect the jitter?
Yes, the slew rate of a clock signal can affect jitter. The amount of jitter due to an input stage (receiver) is reduced by a faster slew rate at the input.

A higher amplitude (signal voltage swing) can result in a faster slew rate. The amplitude of the AD9510/11/12 LVPECL outputs can be selected as 810 mV, 660 mV, 500 mV, or 340 mV. The default swing is the highest, 810 mV. This swing gives the highest slew rate, and therefore should result in the lowest jitter. Setting the LVPECL output swing to a lower amplitude will not improve the jitter.

What is the best way to terminate LVPECL outputs to get lowest jitter?
The data sheets for the ADI clock ICs show both a load resistor with AC-coupled termination, and the "traditional" Thevenin termination scheme for LVPECL. There seems to be little difference between the two termination methods in jitter performance; layout and other considerations are more important. Both termination schemes are capable of giving the same jitter performance when all other factors are considered.

Is it okay to AC-couple PECL or LVDS outputs?
Generally yes, but one must be aware of the high-pass filter formed by the coupling capacitor and the input resistance of the circuit driven by the PECL or LVDS signal:

$$A_V = \frac{sR_{IN}C}{1+sR_{IN}C}$$

Therefore, the gain from the PECL or LVDS output to the receiving circuit input is approximately unity for $f >> 1/(2\pi R_{IN}C)$.

LVPECL requires near-end termination which must be before any coupling capacitor.

What is the fan-out capability of the CMOS, LVDS, and LVPECL outputs?
The clock distribution chips were designed with the intention to provide enough outputs to allow one output for each receive path. The best jitter performance requires a good match between the output and the driven input.

CMOS, of course, can drive multiple CMOS gates. However the number depends on the frequency of the clock, the capacitive loading (mostly parasitic), the termination, the input impedance of the driven devices, the input current (static) of the driven devices, the distance that the signal is being driven, the matching between paths, and probably even more factors. In general, though, a CMOS output can be expected to drive several CMOS (high impedance) devices, if the distances and parasitic loading is reasonable.

LVDS outputs can conceivably drive multiple loads, but this is not usually part of the language of LVDS. That is, since an LVDS output drives 3.5 mA through a 100 ohm termination load, it is probably OK to split the load into two 200 ohm resistors and drive two differential inputs which are in very close proximity. However, the potential for causing reflections on the 100 ohm differential transmission line is increased. This is not necessarily recommended, but is at least feasible.

LVPECL could also be expected to drive 2 or 3 differential inputs if the inputs are very close together. There are multiple ways of terminating the "far end" (that is, at the receivers). The "far end" termination must not be so heavy as to load down the driver much more than a single terminated receiver, but impedance matching to a transmission line can become very tricky. Of course,

the "near end" termination must remain the same as is normally recommended – 180 to 200 ohms to ground on each side of the differential output to provide the path for the required amount of emitter current.

What is the proper termination (value and location) for outputs?
The "proper" termination for an output depends on the type of output (LVPECL/LVDS/CMOS) and the environment in which it is being used. The AD951x data sheets show recommended, or example terminations, for each of the types of outputs.

CMOS outputs are generally not terminated - that is, they drive a high impedance input. CMOS is usually single-ended, and so differential transmission lines are not used. Even if it is differential, controlled impedance transmission lines are not usually used.

LVDS outputs are differential by nature. LVDS normally drives a controlled-impedance differential transmission line, terminated at the pins (or on-chip) of the receiver in the characteristic impedance of the transmission line (usually 100 ohms). No termination is needed at the output of the source; only at the input of the receiver. AC coupling may or may not be used.

LVPECL outputs are differential, but can be used as single-ended or differential. The LVPECL output driver is an emitter-follower, and must have current flowing at all times in order to keep the output impedance low. If current cannot flow, the output is high impedance. This means that an LVPECL output must always be terminated in such a way that current flows. This is done in one of three ways:

- Termination through 50 Ω to Vs - 2 V (this is the classic termination)
- Termination by a Thevenin equivalent of the above. This consists of a resistor divider connected to Vs and GND, with the output tied to the junction of the two resistors. This provides an equivalent termination as the above, but does not require a Vs - 2 V supply.
- Termination by: resistors from the output to ground; AC coupling; controlled impedance coupled differential transmission line; and far-end termination in characteristic impedance of transmission line.

The load resistors (typically 200 Ω) set the current in the emitter followers. The AC coupling capacitors separate the common mode of the outputs from the common mode of the receiver. The differential coupled transmission line is typically 100 Ω. The far-end termination must match the characteristic impedance of the transmission line (100 Ω). If the receiver inputs are self-biased, no further steps need to be taken. However, if the common mode voltage of the receiver inputs must be set, this can be done by splitting the far-end termination resistor and feeding the Vcm at the center point.

Are outputs short-circuit protected?
LVPECL and CMOS outputs are NOT short circuit protected. If these outputs are grounded, internal damage may occur to the output stages.

The only outputs that are short-circuited protected are LVDS outputs. There's actually a specification in the data sheet for the maximum amount of current that flows under this condition.

Are the CMOS drivers on the clock devices complementary?
No, they are 180 degrees out of phase with each other (OUTN/OUTNB), but not truly complementary in that their crossover is not guaranteed to be at half swing.

Some of the schematics in the AD951x data sheets show an LVPECL termination scheme which is different from the classic termination often seen (50 Ω to Vs - 2V, or the Thevenin equivalent thereof). How does this work, and how did you chose 200 Ω for the resistors? Can I use 100 ohms to improve the slew rate (or jitter)?
There are various practices for terminating LVPECL outputs. Some of these hark back to the old days of ECL, where both sides of the output were driving 50 ohm single-ended transmission lines. This resulted in the "standard" ECL termination being 50 ohms to a stiff power rail at (Vsh-2) volts. (Remember that for ECL, Vsh - the high rail - was ground, or zero volts; and the Vsl - the low rail - was -5.2 V.) Later on, since the requirement for a -2V power rail was a problem, users adopted the Thevenin equivalent form of that termination, which consists of a resistor divider between Vsh and Vsl. This is where the 130 ohm to Vsh and 82 ohm to Vsl divider. This is still terminating to the high rail minus 2V with 50 ohms.

These days we recommend microstrip differential 100 ohm transmission lines to couple LVPECL outputs to their driven device

inputs. This allows a somewhat different termination scheme. The resistors to ground on each output set the emitter current of the output drivers, and determine the output impedance of the drivers (re=1/gm=Vt/Ic). So, for the recommended 200 ohm resistors, the emitter current for a logic high output is (3.3 - 1)/200 = 11.5 mA and for a logic low output (3.3 - 1.8)/200 = 7.5 mA. Assuming a reasonable beta (>200), this gives Re(h) = Vt/Ic = 0.026/0.0115 = 2.26 ohms, and Re(l) = 0.026/0.0075 = 3.47 ohms. Therefore the differential output resistance = 5.73 ohms.

This relatively low differential output impedance drives a balanced, fully differential 100 ohm transmission line. The power transfer is not maximum, but the voltage swing is maintained at 800 mV differential. The far end (receiver end) is terminated in the transmission line characteristic impedance (100 ohms across the differential line) to minimize reflections in the reverse direction. Overall, this is a very good and clean termination scheme.

Now, if the near-end termination resistors are reduced below 200 ohms, the emitter current increases, and the signal fall time improves (because an emitter follower cannot "pull down" a node, it depends on the emitter resistor to discharge any capacitance on the node). But, the voltage swing does not change, and the receiver does not see any bigger signal. Also, it is important to keep the emitter current below the value set by the allowable current density in the output device. This is set by the design and process parameters, and is about 18 mA in the case of the AD951x parts. Therefore, it is not recommended to increase the LVPECL output current to a point where it may exceed 18 mA during a static (outputs not switching) output logic high (2.3 V) condition. This limits the LVPECL near-end resistor to no less than 128 ohms.

The reason for recommending 200 ohms for the LVPECL near-side terminations is: 1) it keeps the static current (output not switching) well within the allowable current density rating of the part; 2) it provides for a suitably low output impedance for the LVPECL driver.

So, our recommendation for the LVPECL near-end termination resistors to ground is for values between 150 and 200 ohms for reliability and overall performance.

I have pulled SYNCB low, but I still have output from a channel. Why?
There are two possible reasons for this:

- The channel is divide-by-1. For the AD9510/11/12, a divide-by-1 channel is not subject to synchronization.
- The channel has its "no sync" bit set in the register map. The channels which can be sync'd are held at a fixed level while the SYNCB is held low, therefore there will be no output from that channel.

Why can I not get the same output amplitude or rise and fall times as stated in your datasheet?
There are many variables which may be encountered. Some of these are:

- Termination configuration. Is it the same as we specify in the spec table? Also, are you double terminating?
- Bandwidth of measurement equipment. Remember, at the "rated" bandwidth of an oscilloscope, the signal is already rolled off by 3 dB.
- As a further extension of the Bandwidth consideration above, remember that very fast rise and fall times require very high bandwidth measurements. Rise and fall times in the 100 ps range, for example, require measurement bandwidths of several GHz (5 - 7 GHz) in order to be measured accurately.

The AD9510 datasheet says to use an external pull-up resistor on the FUNCTION pin. Why do I need this and what range of resistors will work?
The AD9510/11/12 have a 30k ohm internal pull-down resistor on the FUNCTION pin. If this pin is not driven externally, it sits at a low logic level. The default function of the FUNCTION pin is RESETB. This means that the part will not function until this pin is driven high. This pin can be tied directly to the positive supply (or a small value resistor <10k) if not used. However, if it is used it must be actively driven.

Top of Section | Top of Page

AD9540, DDS-Based Clocking

Top of Section | Top of Page

May I use the AD9540 for spread spectrum clocking?
The AD9540 is capable of doing a frequency sweep function which can be used for the EMI reduction technique called Spread Spectrum Clock Generation. A triangular waveform is easy to achieve using the frequency accumulator available on that product. Non-triangular waveforms can also be constructed using the profile registers to drive changes to the "Delta Frequency Tuning" word which is used by the frequency accumulator.

Can I get two clock outputs from the AD9540?
Yes, application note AN-769, "Generating Multiple Clock Outputs from the AD9540" is available online in the Clock Library.

What's the advantage of a DDS-based clock generator?
Direct Digital Synthesis has long been used for clock generation. The primary reasons designers choose to go with a DDS-based clock generation solution are:

- Frequency resolution – a 48 bit FTW provides granularity on the µHertz level, orders of magnitude finer than the finest available from fractional-N PLL products.
- Easy, well controlled phase offset capability up to 14 bits of phase offset enables designers to place edges very accurately; since it is a digital function, the repeatability (low part to part variation) is unparalleled in the PLL domain.
- Multi-octave tunable – DDS products can be programmed over a wide range of frequencies (DC to 40 [or even 45%] of their system clock). PLL tuning range is typically less than an octave.
- DDS can be used as a high resolution divider in a PLL loop. In this configuration, though, the resolution benefit is retained, but not the multi-octave tuning range.

Why does the AD9540 require special filtering on its analog output. What are the requirements of this filter?
To use a DDS-based device as a clock driver, the output sinewave must be converted to a rectangular wave with fast edge transitions (high slew rate). In order to do this, it is necessary to pass the AC output signal through a high- speed comparator, with an output of the correct voltage level.

An important factor in creating a clean, low jitter clock signal is the reconstruction filter, which is required between the DAC output and the comparator. The sampled sinewave output of the DDS must be low pass (or bandpass) filtered in order to remove the sampling artifacts (images) from the AC signal. Another way to look at this is that the filter acts to smooth the stairsteps in the DAC output. This smoothed sine wave is then applied to the input of a high-speed comparator. The duty cycle of the resulting clock signal can be varied by adjusting the reference voltage on other input of the comparator, if desired.

For applications requiring the lowest jitter, a bandpass filter may provide better performance than a low pass filter. Also, differential filters offer performance advantages over single-ended ones.

Several of the ADI DDSs have high-speed comparators integrated with the DDS in order to make clock signal generation simple. It is also always possible to use an external, high-performance, high-speed comparator from ADI. But, in all cases – don't forget the low pass (band pass) filter.

Top of Section | Top of Page

Clock Design, Tools and Applications

Phase Noise/Jitter

- I'm working with optical networks - SONET/SDH. Do ADI's clock chips support these applications?
- On my board, I can't get the same low jitter numbers that are shown in the datasheet. Am I doing something wrong?
- How do you determine the bandwidth over which phase noise is integrated to obtain jitter?
- Using the "ADC SNR method", what is the equivalent bandwidth for the jitter specification?
- How do harmonic spurs in the output spectrum affect jitter (random or deterministic)?
- When a jitter number is specified without an associated bandwidth, what bandwidth should be assumed?
- How do you specify jitter?
- How do I use the clock part for jitter clean-up?
- If jitter can be calculated from phase noise measurements, is it possible to calculate phase noise from jitter numbers?
- Does jitter vary with different clock frequencies? How about phase noise?
- I sure can't measure jitter with femtosecond resolution on my scope! How do you do it? How much confidence do you have in the jitter figures that you are quoting for these parts?

ADIsimCLK

- Do you guarantee performance shown in ADIsimCLK?
- Who do I contact for technical support on ADIsimCLK?

Power

- Should I use the minimum charge pump current settings in order to minimize power?
- Can I run CMOS outputs at 5V?
- Can I use different power supply voltages for the PECL output drivers?
- Is .01 uF sufficient for power supply pin bypass?
- My application has pretty tight power consumption requirements. I am very interested in the capabilities of the AD9510, but I don't need every feature. Is it possible to turn off the unused features and save power?
- Why don't you spec psrr and cmrr in the datasheet?

Synchronization

- How do I get two AD951x (with PLL) to synchronize to the same reference input edge?
- I really need >10 clock outputs. Can I use multiple chips together and still guarantee that all output clocks are synchronized to REFIN?
- How do I synchronize multiple clock devices?

Thermals/Reliability

- What happens if I run the part in an ambient environment which exceeds 85C?
- How can I determine the die temperature of your device?

PCB / Evaluation Board

- My circuit board has both an analog GND and a digital GND. How should I connect the AD9510 pins labeled GND?
- What PCB layout recommendations do you have for the of the exposed paddle on the bottom side of the LFCSP package?

Phase Noise/Jitter

I'm working with optical networks - SONET/SDH. Do ADI's clock chips support these applications?
Yes, the jitter specs and frequencies covered by our clocks are suitable for SONET/SDH applications.

On my board, I can't get the same low jitter numbers that are shown in the datasheet. Am I doing something wrong?
Many factors contribute to the overall jitter performance. This includes: the jitter of a reference source, the clock input, the design of the PLL loop, the VCO/VCXO choice, noise on the power supplies and bypassing, slew rate of input signals, output termination configurations, and signal routing. Differences in any of these can affect jitter performance. Also, the method of jitter measurement is critical because jitter at these low levels is very difficult to measure.

See AN-501 and AN-756 for more information, available from the Clock Technical Library.

How do you determine the bandwidth over which phase noise is integrated to obtain jitter?
It is a little tricky to specify the bandwidth over which phase noise should be integrated in order to calculate the jitter which will actually be observed when that clock signal is used to clock a converter. There are many variables which are seldom known with accuracy – such as the inherent bandwidth of the sample clock circuit on the converter. Also, it is very difficult to actually measure the broadband phase noise of a clock signal beyond an offset of a few MHz. If there is an application spec which defines the bandwidth (as is the case, for example, with the SONET/SDH optical networking specs) then use those limits. However, for a true "broadband" jitter calculation some assumptions and simplifications must be made. One assumption made by ADIsimCLK, for example, is that the upper offset integration limit is one-half the clock frequency. The lower offset integration limit is assumed to be between 100 Hz and 1 kHz.

The match between the ADIsimCLK calculation based on these phase noise offset limits is in good agreement with the jitter measurements made using an entirely different whole-system method based on measuring an actual ADC SNR when driven by a clock signal with a measured phase noise spectrum (insofar as that measurement extends in offset). The match between calculated and measured jitter using ADIsimCLK is less than 2 dB.

Using the "ADC SNR method", what is the equivalent bandwidth for the jitter specification?
The signal-to-noise ratio is given by:

SNR = -20*log(2*pi*Fa*Tj); where Fa is the analog BW, Tj is the time jitter as an RMS value.

The bandwidth over which the phase noise is integrated to obtain Tj(rms) is the encode bandwidth. This is not simply Fs/2; it is the bandwidth over which aliased noise components of the encode clock contribute significantly in the equivalent analog input signal bandwidth (Nyquist band). Please see application note AN-756 for more information.

How do harmonic spurs in the output spectrum affect jitter (random or deterministic)?
The short answer is that harmonic spurs have no impact on jitter.

However, what about non-harmonic spurs? Non-harmonic spurs do affect jitter, but they do so in a periodic (i.e., deterministic) manner. This is because a non-harmonic spur causes deviations in edge timing that wander about the ideal timing point in a sinusoidal manner; unlike random noise, which causes the edge timing to wander unpredictably.

Non-harmonic spurs become an important consideration when the signal is the result of a sampled system (e.g., a DAC). This is because harmonics of the fundamental signal may be sufficiently high enough in frequency to appear as an aliased image. Unfortunately, the frequency of an aliased image is generally not harmonic. For example, suppose a DAC operates at a sampling frequency of 100MHz, which puts the Nyquist frequency at 50MHz. Furthermore, suppose that the DAC's fundamental output frequency is 15MHz. In this case, the 4th harmonic occurs at 60MHz. However, 60MHz is greater than the Nyquist frequency of 50MHz, which means that the 4th harmonic actually appears at the aliased image frequency of 40MHz (100MHz-60MHz). Note that 40MHz is NOT a harmonic of 15MHz. Therefore, in this case, the 4th harmonic produces a non-harmonic spur, which means that the 4th harmonic would produce deterministic jitter.

When a jitter number is specified without an associated bandwidth, what bandwidth should be assumed?
Jitter is a time domain phenomenon which is the result of all the noise which affects the node of interest. This noise is usually assumed to be random noise, which is broadband in nature, although narrowband spurs and noise can certainly contribute to jitter. The most natural bandwidth is the actual bandwidth of sensitivity of the node where the jitter is of concern - such as at the encode node of an ADC. This bandwidth is not always known in practice. So, when a measurement of jitter is made it is referred to as a "broadband" measurement. That is, all of the broadband noise which affects the node is accounted for in the measured jitter – it has not been limited by an intentional filter.

Because ADCs and DACs are time-sampled data systems, they are also Nyquist systems. This means that the broadband noise of the sampling clock is folded back into the Nyquist bandwidth (between 0 and Fs/2). Therefore, the Nyquist bandwidth is a natural bandwidth over which to measure phase noise of the sampling clock in order to calculate the jitter to be expected in a converter.

Often the broadband phase noise of a high quality clock source falls below the system noise floor long before reaching the Nyquist frequency (Fs/2). In this case, the noise floor is dominant in determining the jitter over the Nyquist bandwidth. However, close-in phase noise of some sources rises significantly at small offsets from Fs, and can have an impact on the jitter. In this case, a low frequency limit for the phase noise measurement should be specified, unless the total jitter due to the close in phase noise is significant in the system.

How do you specify jitter?
Jitter is specified statistically as the root mean square of the time deviation (over many cycles) between the actual clock edges and an ideal clock of the same frequency. The rms value corresponds to one sigma, or a standard deviation, of a normal distribution. This applies only to truly random jitter, and not to any pattern jitter which may be present due to crosstalk between outputs. A good estimate of the peak-to-peak jitter (as normally used) is to multiply the rms value by six.

How do I use the clock part for jitter clean-up?
The distribution portion of the chip does not offer jitter cleanup. The PLL portion does offer cleanup, so the PLL must be employed. Using the PLL filters the clock applied to the distribution section. Any reference clock noise outside the PLL loop bandwidth is filtered, reducing the phase noise and the jitter of the reference clock.

If jitter can be calculated from phase noise measurements, is it possible to calculate phase noise from jitter numbers?
No. Only the total integrated phase noise power over the bandwidth of integration can be calculated from the jitter number. The shape of the phase noise spectrum, and the offset limits used, are lost in the calculation of the phase jitter.

Does jitter vary with different clock frequencies? How about phase noise?
All other things being equal, the jitter does not vary with different clock frequencies. The phase noise does vary as 20*log(F1/F2). When the equations for phase offset vs time variation as a fraction of the unit interval (1 cycle) are examined, it will be seen that this is a direct consequence of the math.

I sure can't measure jitter with femtosecond resolution on my scope! How do you do it? How much confidence do you have in the jitter figures that you are quoting for these parts?
Well, we don't have an oscilloscope with (usable) femtosecond time resolution either. Femtosecond level jitter has to be inferred from other measurable values. In our case, there are two types of measurements which allow us to determine jitter at the femtosecond level:

- We can calculate expected jitter based on observed phase noise characteristics. Phase noise measurements require specialized measurement equipment – we use an Agilent E5500 phase noise measurement system – and careful measurement technique.
- We can infer jitter magnitudes from measurements made when driving an actual ADC converter with the clock whose jitter is being investigated. Briefly, the jitter of the sampling clock can be calculated from the observed SNR of a conversion.

Top of Section | Top of Page

ADIsimCLK

Do you guarantee performance shown in ADIsimCLK?
No, we do not. ADIsimCLK is intended to provide a fair representation of typical performance which can be expected in the defined configuration. All results are based on actual observed data, not strictly theoretical calculations. The results given by ADIsimCLK should be within 1-2dB of what you can expect to see.

Who do I contact for technical support on ADIsimCLK?
Support is available from the users' forum and bug reporting forum on the Applied Radio Labs website.

Top of Section | Top of Page

Power

Should I use the minimum charge pump current settings in order to minimize power?
While a lower charge pump setting will reduce power, the savings are likely to be negligible. The reason for this is that when the loop is in lock, the charge pump only turns on for a time equal to the anti-backlash pulse width plus any static phase errors.

Can I run CMOS outputs at 5V?
No, the CMOS outputs are limited to 3.3V LVCMOS levels.

Can I use different power supply voltages for the PECL output drivers?
This depends on the device being used. With the AD9510/11/12/13/14/15 - no; there are no separate power supply pins for the LVPECL outputs of these devices. However, with the AD9516 this is possible.

Is .01 uF sufficient for power supply pin bypass?
Often it is sufficient; however, sometimes it may be desirable to use both a 0.01 uF in parallel with a much smaller value with a higher self-resonant frequency. This applies at the device pins. It is advisable to put a larger value (10 uF) at the point where the power supply enters the board.

My application has pretty tight power consumption requirements. I am very interested in the capabilities of the AD9510, but I don't need every feature. Is it possible to turn off the unused features and save power?
Yes. It is very possible to turn off unused sections of the AD951x chips. There is a high degree of configuration which allows unused inputs, outputs, and the PLL to be powered off when not needed or not used. The programming register table shows the bits which control these power downs.

Why don't you spec psrr and cmrr in the datasheet?
PSRR and CMRR is meaningful and well understood in the case of an op amp. However, its meaning for a clock chip is much less clearly stated, so we do not spec these for our clock devices. As stated in the data sheets, the power supply voltage must be kept within 5% of the nominal voltage (3.3V) in order to meet the data sheet specs. We don't specify a small signal noise limit on the power supply. Rather, we indicate that good power supply design and bypassing is assumed.

It is important that power supplies be as quiet as practicable. At the clock device, power supply pins should be well bypassed by adequate capacitors of low ESR, as near to the power supply pins as possible. It is possible that high frequency noise on the power supply pins could show up on the output if the power supply pins are inadequately bypassed. Also, any noise on the common ground can affect all of the circuits on a board, and can couple to the outputs. Therefore, it is important to follow good layout and bypassing practice on all of the circuits on the board to minimize the injection of noise into the common ground.

ADI clocking devices feature differential inputs and outputs (LVPECL and LVDS, but not CMOS). The conscientious use of differential paths greatly reduces the susceptibility to any common-mode noise.

Top of Section | Top of Page

Synchronization

How do I get two AD951x (with PLL) to synchronize to the same reference input edge?
The PLL reference divider's are not synchronized by the SYNCB pin. To synchronize two 951x parts to the same reference input edge it is recommended that the user set the R divider to R=1 and use the SYNCB pin to synchronize the dividers.

I really need >10 clock outputs. Can I use multiple chips together and still guarantee that all output clocks are synchronized to REFIN?
Yes, many ADI clock parts feature multi-chip synchronization. This requires careful attention to layout (equal length traces, etc.) to keep the delays constant across all chips.

How do I synchronize multiple clock devices?
The clock devices have a FUNCTION pin or SYNCB that can be configured and used to reset the clock outputs across multiple devices. In addition, the configuration requires a common input clock and common SYNC signal to all devices. See the datasheet for more information.

Top of Section | Top of Page

Thermals / Reliability

What happens if I run the part in an ambient environment which exceeds 85C?
ADI does not guarantee specs on parts operating outside the temperature range listed in the product datasheet. The user may find that devices operate outside rated temperatures, but ADI does not recommend exceeding datasheet limits.

How can I determine the die temperature of your device?
By knowing the θ_{JA} of the package, ambient temperature, and the power dissipation, you can calculate the die temperature.

Example: Power = 2 Watts and ambient = 25 C

What's the die temperature?

$T_{die} = T_A + \theta_{JA} * Pwr$
$= 25\ C + 24C/W * 2W$
$= 25\ C + 48\ C$
$T_{die} = 73\ C$

Please note that thermal impedance is a complex subject. Reference materials on ADI's website for more detail.

Top of Section | Top of Page

PCB / Evaluation Board

My circuit board has both an analog GND and a digital GND. How should I connect the AD9510 pins labeled GND?
All AD9510 ground pins should be connected to the analog ground.

What PCB layout recommendations do you have for the of the exposed paddle on the bottom side of the LFCSP package?
We have an comprehensive application note (AN-772) that covers layout recommendations. In addition, some users import our gerber files of the evaluation board as a guide in their layout.

Application notes are available from the Analog Devices website.

Top of Section | Top of Page

Glossary

Top of Section | Top of Page

Wideband Jitter:
Phase noise is usually measured as a single-sideband power within a bandwidth ranging between a low and a high frequency offset from the fundamental frequency component of a clock signal. There is a quantitative relationship between the cumulative phase noise power in a clock signal and the time jitter of that clock signal.

The noise bandwidth of a clock signal can be limited by bandpass filtering, in which case the time jitter is the result of the phase noise power within the bandwidth of the filter. The amount of time jitter due to the phase noise power within a given bandwidth can be calculated by known formulas. However, in many cases, there is no intentional limiting (filtering) of the bandwidth of a clock signal. For example, a clock signal with fast rise and fall times (that is, a wide bandwidth clock signal) is used to drive the encode node of an ADC. The ADC encode node has its own frequency response, which may or may not be known. In this case the resulting jitter is due to the total phase noise power in the signal as seen by the encode node. The jitter measured under these conditions is referred to as "wideband jitter", since it is not the result of the intentional limiting of the phase noise to a well defined bandwidth which can be stated.

Wideband jitter on the sampling clock has the effect of setting a limit to the SNR performance of an ADC or DAC.

Additive Phase Noise:
Any component in a signal chain has some amount of noise at its output that is generated internally, which is not due to externally applied noise at its input. It is useful to measure the phase noise at the output of a clock device in such a way that the phase noise of any external source is cancelled out. The phase noise so measured can be called the "additive phase noise" of the device. That is, it is the amount of phase noise that the device adds to the signal chain.

By considering only the phase noise due to the device itself, it is then possible to predict the degree to which the device impacts the total system phase noise when used in conjunction with various other components (such as oscillators and clock sources), each of which contribute to the phase noise of the total system. In many cases, the phase noise of one element in the chain dominates the system phase noise.

Residual Phase Noise:
Residual phase noise is sometimes used in the same sense as "additive phase noise" (see Additive Phase Noise). It is called "residual" because it is the noise left over after the noise of the source driving the device is cancelled out or subtracted. In other words, it is the phase noise due to the device itself, and not to the source driving it. This allows for calculating the impact of the device on the total signal chain.

Jitter (rms):
Jitter is a time domain phenomenon. In the context of a regularly repeating clock signal, jitter is the displacement of the clock signal edges from their expected times of occurrence. Jitter can be caused by either random or non- random noise sources. Non-

random noise can cause what is known as pattern jitter (sometimes also called deterministic jitter). On the other hand, random phase noise in the frequency domain causes random time jitter. Non-random jitter usually has its origins in the system layout and design, rather than in the clock device itself. Random jitter is due to stochastic phenomena such as thermal noise, and is very much a characteristic of the clock device itself.

Being a result of stochastic causes, and having the characteristics of a random distribution, jitter is usually specified in statistical terms. This means that the clock edge time displacements from the expected values display a distribution (usually considered Gaussian, or normal) in time. A normal distribution can be specified by its sigma. This is numerically equivalent to the root of the mean square (rms) of the errors from the expected. Therefore, a useful way to give a jitter value is as the one sigma of its time distribution. Accordingly, one sees jitter as, for example, 200 fs rms, where 200 fs is the one sigma value of the observed distribution.

Jitter (pk-to-pk):
See Jitter (rms). Since random jitter is a random distribution (Gaussian or normal) it has unbounded tails of low probability. So, it is impossible to precisely state the peak-to-peak excursion of the edge timing error. To do so would imply the necessity of an infinitely long (large) series of measurements. However, a convenient rule-of-thumb for stating an effective peak-to-peak value for a normal distribution is to consider it as 6 X sigma. So, a clock device with a specified 200 fs rms jitter can be considered to exhibit a 1.2 ps pk-to-pk jitter.

Jitter (cycle-to-cycle):
Jitter is the variation in time of the occurrence of clock edges with respect to the expected (or ideal) time of occurrence. There are various ways of looking at, or specifying, jitter. One of these is to measure the length of each period, measured from one rising clock edge to the next. Then the difference in period from one cycle to the next is calculated. The resulting collection of errors (differences) may be said to give the cycle-to-cycle jitter.

Time Jitter:
Phase noise is a frequency domain phenomenon. In the time domain, the result of phase noise power in a regular signal (that is, a signal of a "constant" frequency) is exhibited as displacements in time, or jitter. In the case of a sinusoidal wave, the times of successive zero crossings vary from their ideal points in time. In the case of a square wave, transition edges are randomly displaced from their ideal (regular) times of occurrence. In both cases, the variations in timing from the ideal are the time jitter.

In this specific usage of the term "time jitter", the errors in timing are variations from the ideal (determined by the number of cycles in a period of time, evenly divided). This is in contrast to cycle-to-cycle jitter in which the errors are the differences in period from one cycle to the next. (The two points of view can be numerically related, but are somewhat different.)

Since these variations (due to stochastic causes) are random (characterized by a Gaussian or normal distribution), the time jitter is specified in units of seconds root mean square (rms) or 1 sigma of the Gaussian distribution. See Jitter (pk-to-pk); Jitter (rms).

Additive Time Jitter:
Additive (or residual) time jitter is the amount of time jitter that is attributable only to the device or subsystem being measured. The time jitter of any external oscillators or clock sources has been subtracted. This makes it possible to predict the degree to which the device will impact the total system time jitter when used in conjunction with various oscillators and clock sources, each of which contribute their own time jitter to the total. In many cases, the time jitter of the external oscillators and clock sources dominates the system time jitter. Time jitter of cascaded components adds up as the root of the sum of the squares of the rms additive time jitter of each component.

See Additive Phase Noise; Residual Phase Noise.

Top of Section | Top of Page

Last Updated: 11/2006