Advanced Clocking Options with the Analog Discovery 3

Introduction

Digilent's third release of the Analog Discovery platform (AD3) takes the concept of a handheld electronics laboratory to a whole new level. Deeper buffers, more powerful processing, and an advanced clock generation scheme enable this miniscule piece of equipment to rival its benchtop counterparts. To fully understand the benefits of these changes, a shallow dive into the basics of analog sampling and external clock synchronization is of value.

Analog Sampling

The conversion of analog signals to their digital representations is one of the most fundamental processes in modern engineering and scientific analysis—and also one of the most difficult to do properly. At the most basic level, as shown in Figure 1, the analog signal of interest is sampled by an analog-to-digital converter circuit at fixed time intervals, otherwise known as the sampling rate. Each sample is represented by a discrete number, and can be used to approximate the original signal in the course of further analysis.

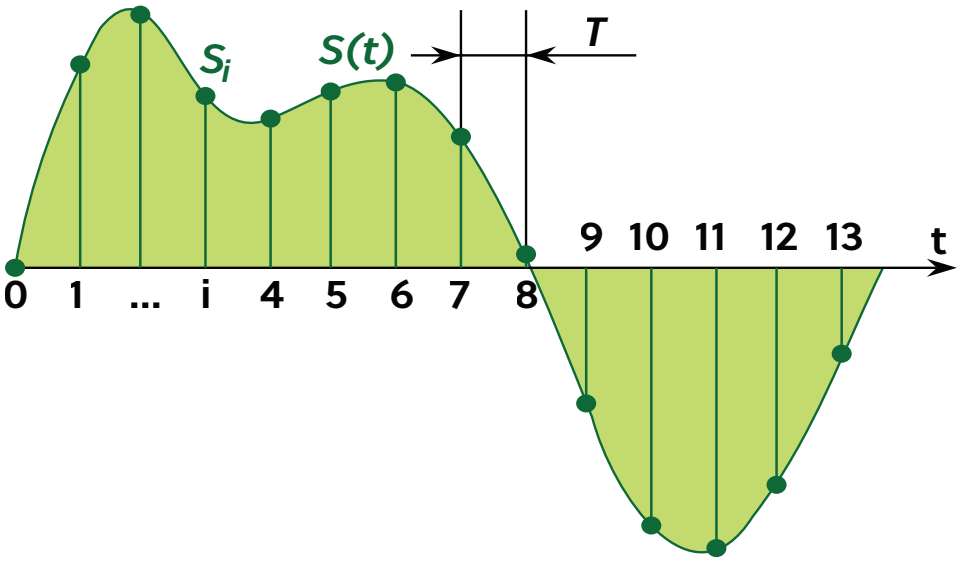

Figure 1: Digital sampling of an analog signal

The difficulty, and the basis for the entire field of signal processing, is in understanding the error introduced by this sampling process. In the y axis, one has to consider issues of quantization and precision. In the x axis, the nuances of bandwidth, Nyquist criteria, and jitter, to name a few, are critically important. However, the equipment that engineers and scientists use on a regular basis for analyzing real world signals accounts for all of these factors, yielding highly sophisticated tools like Digilent's Analog Discovery 3 (AD3).

The Analog Discovery 3 is a small desktop platform for the generation and analysis of both analog and digital signals. Built upon a robust signal processing platform, it includes high speed FPGA, clock generators and PLLs, high speed and precise ADC's and DAC's, and a mature software toolkit. When considering the analog capabilities, the AD3 features a two-channel oscilloscope that rivals standalone benchtop units costing 10x more, offering:

- Two differential channels with 14-bit resolution at up to 125 MS/s per channel with up to +/- 25 V input range, 30+ MHz bandwidth with BNC Adapter
- User-configurable hardware input filters
- FFT, Spectrogram, Lock-In Amplifier, Additional Software Input Filters, Eye Diagram, XY Plot views, and more, in the WaveForms application

The ADC used in this oscilloscope is clocked using a configurable PLL that can be driven from multiple clock sources. This enables very low jitter sampling capabilities with a variable sampling rate up to 125 MS/s. This typical clocking scheme is shown in Figure 2, where the on board 25 MHz clock is used to drive the PLL, which then outputs multiple synchronized clocks for the rest of the AD3 system.

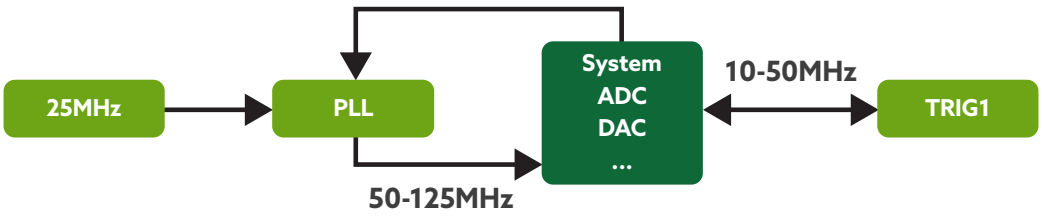

Figure 2: Clocking Scheme

For most common tasks, the Digilent Waveforms software offers simple dropdowns to select desired sampling rates, trigger sources, and the like. For advanced applications, the configuration of this clocking scheme is exposed to the end user through the WaveForms SDK, where even more control can be leveraged. Using this SDK, external clocks can be used to drive the system PLL, opening up some interesting applications that would be impossible to implement with most other test and measurement tools.

External Clocking via Trigger Inputs

At the core of the timing system within the AD3 is the Texas Instruments CDCE6214, a four-channel, ultra-low power clock generator that can create five independent clock outputs from either a crystal or external clock input. Under normal operation, the AD3 uses an onboard 25MHz source to synchronize the ADC, the DAC, and the FPGA. However, as

shown in Figure 3, the Trig1 input can be used to feed an external clock source into the clock generator. This external clock source must be between 10 and 50 MHz, but the resulting performance maintains less than 4 ps of jitter with the same 125 MS/s sampling rate.

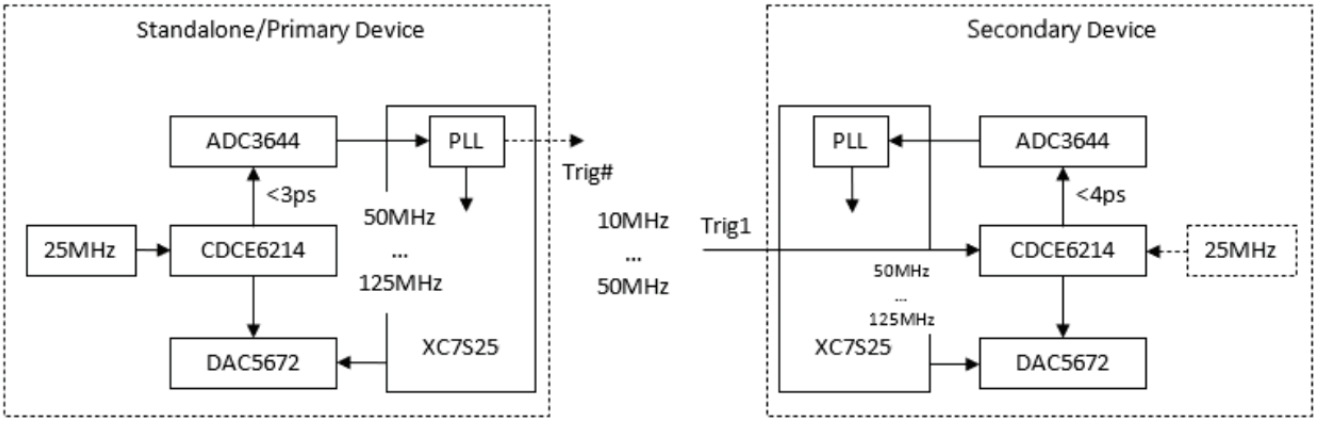

Figure 3: Internal and external clocking configurations in the AD3.

The ability to provide an external clock may seem esoteric, and indeed, it is a feature usually found in only the highest end of test and measurement tools. However, in some very specific use cases where external synchronization is necessary, the AD3 steps in to serve these roles with aplomb and relative simplicity.

Example Use Cases

Many experiments in advanced signal processing demand precise control over the clocks used for sampling and data conversion. These experiments often involve signals, which are correlated, but physically separated in space or time. Two examples demonstrating this requirement are GPS sensor synchronization and equivalent time radar sampling.

GPS synchronization is used when two sensors are positioned with significant distance between them, but the signals of interest are correlated in time. Therefore, the acquisition platform must not only be able to operate from an external clock, but to also provide a method for timestamping the data relative to that clock. AD3, as shown in Figure 4, can accomplish that task. Each measurement site uses a GNSS receiver to create a time-synchronized reference clock. For example, the uBlox ZED-F9T module could be used to output a timing reference with differential timing accuracy of 2.5ns and 4ns of jitter. The resulting data samples can therefore be tightly synchronized in time even with 100's of kilometers between measurement sites.

Figure 4: GPS synchronization for physically separated acquisition sites

The second example application of the external clocking efficacy is the equivalent time sampling of high frequency signals. This time sampling is common in ground penetrating radar (GPR) signal processing: the radar transmitter outputs a wide band, high frequency pulse as shown in Figure 5. The receiver needs to recover the reflected version of this pulse, and sample it with the highest possible fidelity.

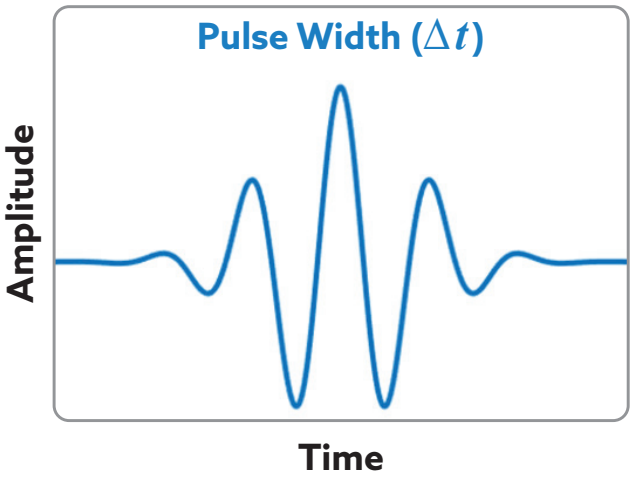

Figure 5: Sample high frequency, wide band GPR pulse

Instead of using extremely high frequency ADC's (gigasamples / second), or attempting to mix the received signal down in frequency, a technique called equivalent time sampling, or semi-real time sampling is typically used. The transmitter sends multiple pulses, and the receiver samples each reflection with a different phase offset in the reference clock. All of the received waveforms are then overlaid to realize the final recovered signal. This technique is shown in Figure 6, in which each of the red sample points is collected by shifting the phase of the sampling clock in subsequent cycles of analysis.

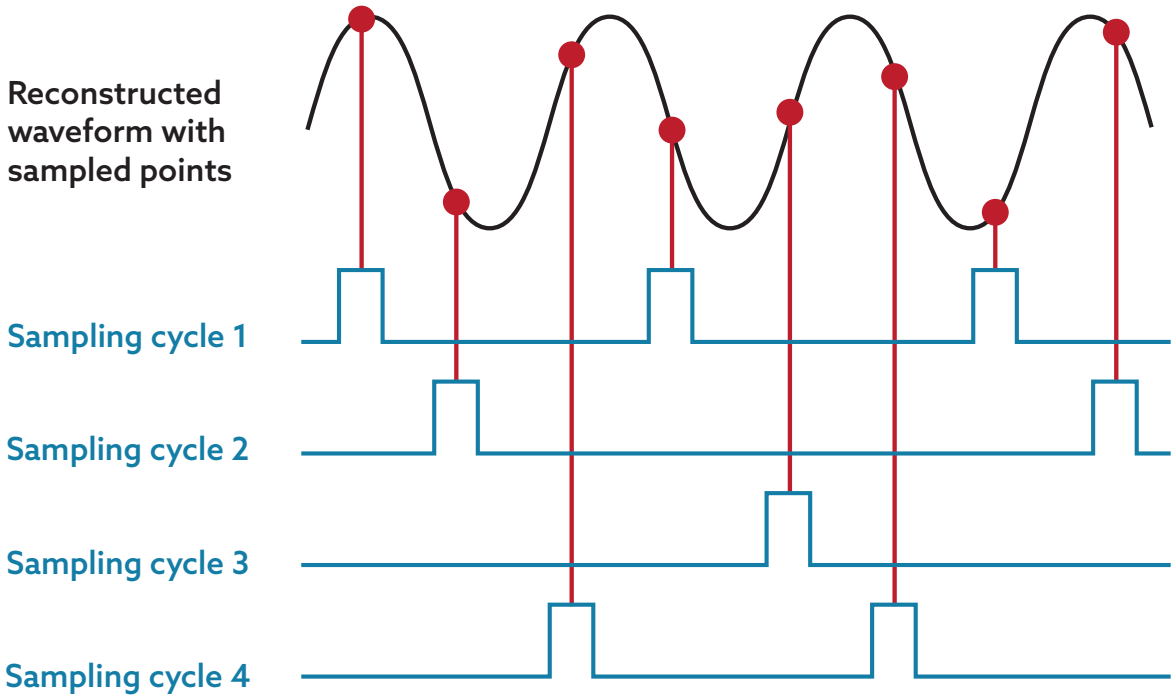

Figure 6: Equivalent time sampling technique for high frequency repetitive signals

The AD3 can be used for this exact type of sampling using the external clocking feature. As long as the signal to be sampled is repetitive with a known time trigger, the external clock phase can be adjusted iteratively to capture signals with equivalent sampling rates higher than 125 MS/s (as long as bandwidth limitations are observed).

The WaveForms SDK

For many experiments, controlling the AD3 programmatically can be helpful to craft complex scripts to capture and analyze data in an automated way. Digilent offers a software development kit (SDK) for this exact purpose. We will focus on the Python implementations of the SDK, though versions exist for C(++) as well. The SDK itself is included as part of the Waveforms software download. On Mac OS, the framework is installed by simply dragging the dwf.framework icon into the user Frameworks folder, as shown in Figure 7.

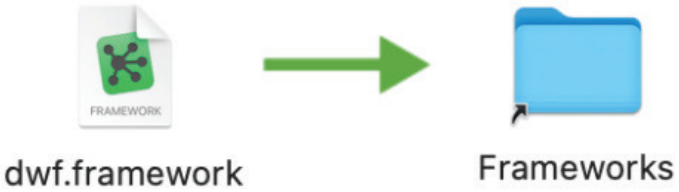

Figure 7: Installing the dwf framework for Mac OS

Similar installation steps can be followed for Windows and Linux machines, as explained in Digilent's **getting started guide**.

Once the dwf framework is properly installed, Digilent provides a wealth of useful examples to get started with. These examples are located in both the dwf SDK folder and in a **GitHub repository**. The easiest way to locate this folder is to click on the "WaveForms SDK" link in the upper right hand corner of the Welcome tab in the WaveForms GUI. Each example follows a similar structure: the AD3 device is opened first, then configured, then executed to collect data, and finally, the data is processed for visualization or storage. Every aspect of the AD3 can be controlled programmatically in this way, including the oscilloscope, waveform generator, logic analyzer, and power supplies. An example is shown in Figure 8, in which the housekeeping and data analysis have been removed for simplicity.

```
# initialize the scope with default settings
scope.open(device_data)

# set up triggering on scope channel 1
scope.trigger(device_data, enable=True, source=scope.trigger_source.analog, channel=1, level=0)

# generate a 10KHz sine signal with 2V amplitude on channel 1
wavegen.generate(device_data, channel=1, function=wavegen.function.sine, offset=0, frequency=10e03, amplitude=2)

sleep(1)    # wait 1 second

# record data with the scopeon channel 1
buffer = scope.record(device_data, channel=1)
```

Figure 8: Python script example for generating a waveform and recording it on the oscilloscope

Externally Clocking the AD3

To demonstrate the external clocking capability of the AD3 using the programmatic SDK, a test setup was assembled as shown in Figure 9. A signal generator was used to create a two-cycle wavelet based on a 10 MHz sine wave, along with a 10 MHz square wave clock, ensuring that the clock and the wavelet are synchronized in time (like the radar example above). The wavelet is connected to the Analog 1 channel of the AD3 for capturing on the oscilloscope, and the square wave is fed into the Trigger 1 channel for external clocking.

Figure 9: Test configuration to demonstrate external clocking capability

Referencing the WaveForms SDK manual, one finds the following information (Figure 10) on externally clocking the AD3. Basically, parameter number 10 must be set to the value of 2 to convert the Trigger 1 channel into a clock input and direct the AD3 PLL to use it as a reference.

DwfParamFrequency	8	Adjust system frequency in Hz (default 100MHz) Supported by Digital Discovery, ADP3X50, Analog Discovery 3
DwfParamExtFreq	9	Specify for input or set reference output frequency in Hz (default 10MHz) Supported by ADP3X50, Analog Discovery 3
DwfParamClockMode	10	0 = use internal oscillator (default) 1 = enable reference output on Trigger IO 1 2 = use reference input from Trigger IO 1 3 = use Trigger IO 1 as reference input-output (only ADP3X50) Supported by ADP3X50, Analog Discovery 3
DwfParamTempLimit	11	Specifies the over temperature threshold in degree Celsius on devices which support such option.
DwfParamFreqPhase	12	Specifies the system clock phase which is useful for device synchronization when reference input clock is used.

Figure 10: Digilent reference manual, FDwfParamSet() function parameters

This is accomplished in Python using a single line executed before the device is opened.

```
dwf.FDwfParamSet(DwfParamClockMode, c_int(2))
```

If using the graphical UI in the WaveForms software, the settings dropdown can be used to change the clock source from internal to external. An example is shown in Figure 11.

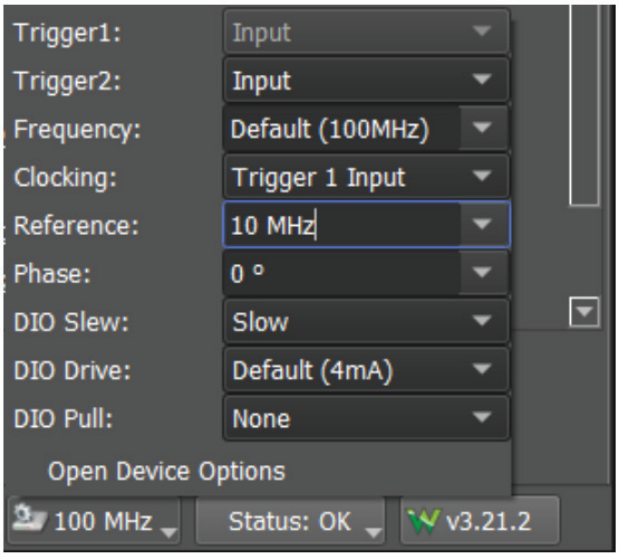

Figure 11: WaveForms control panel for setting the AD3 clock source

The remainder of the code follows the test_scope-wavegen.py example with the wavegen portion commented out. The resulting capture is shown in Figure 12, where the two-cycle wavelet is nicely sampled using the external clock reference scaled to 125 MS/s.

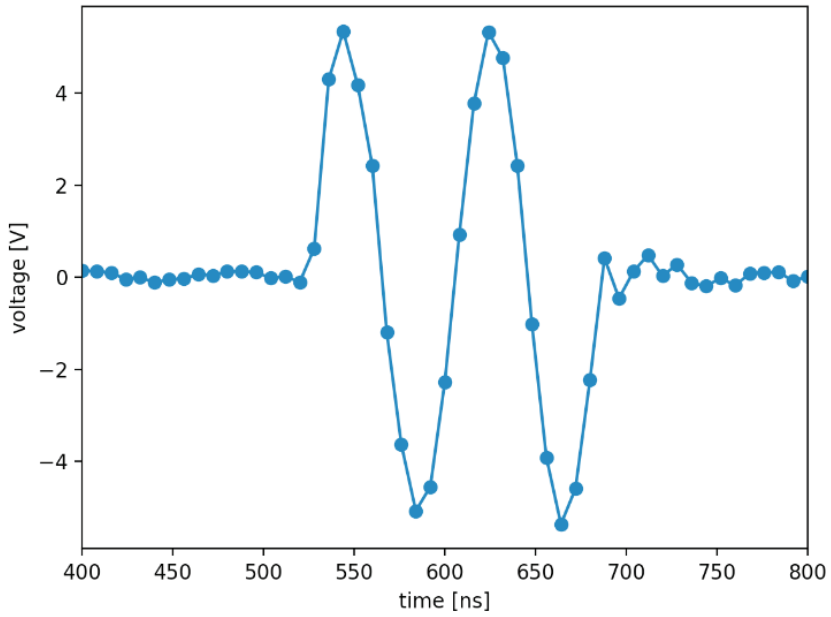

Figure 12: AD3 capture of 10 MHz wavelet with external clock

Since the input wavelet is repetitive, the clock phase offset can be adjusted on the signal generator to shift all of the samples slightly in time. The aforementioned equivalent sampling technique is used in that way to increase sampling rates beyond the fundamental limit of the ADC. As shown in Figure 13, the red set of samples was collected on the second pass of the repetitive input wavelet with approximately 10 degrees of phase shift.

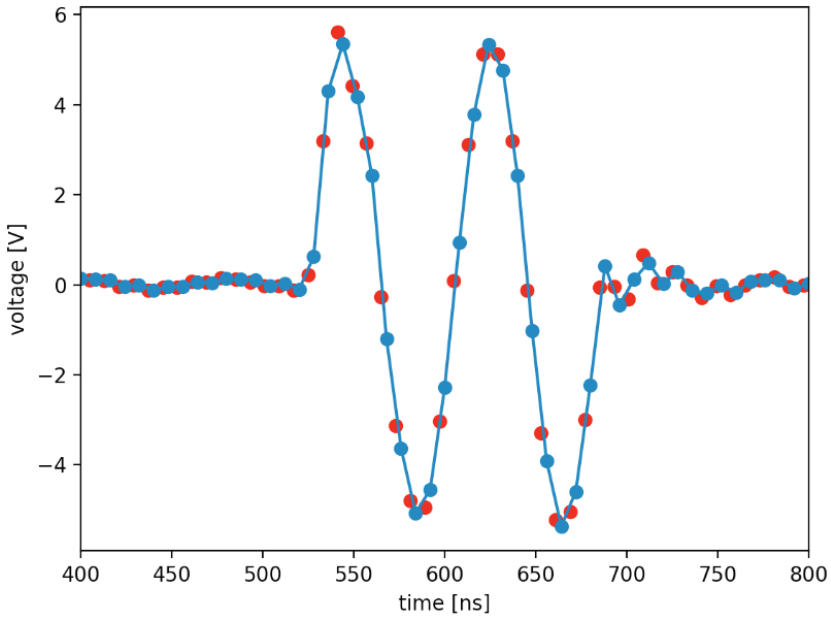

Figure 13: Equivalent time sampling of wavelet using AD3 and external phase shifted clock

The superposition of the red samples with the blue samples creates an artificially enhanced equivalent sampling rate. Further iteration of the clock phase shift could fill in additional samples until the limitations of jitter result in diminishing returns.

Learning More With Digilent's AD3

The experiment demonstrated in this article highlights the power of being able to control the external reference clock—indispensable in certain applications. The flexibility of the AD3 hardware, along with Digilent's willingness to expose these features through the SDK makes for a powerful test and measurement tool, saving on costs and without relying on several larger desktop devices.

To learn more, visit the **AD3 landing page**, and explore the **SDK Python repository** on GitHub.

Since 2000, Digilent (a wholly owned subsidiary of National Instruments) has provided embedded engineers, researchers, scientists, and students with cost-optimized products, tools, and application information for innovative, FPGA and SoC based hardware-software systems. Our customizable and flexible solutions will accelerate development time for even the most experienced professionals, while maintaining low barrier to entry for advancing engineers, students, and the perpetually curious.

From our competitive pricing to the portability of our products and comprehensive documentation, we value delivering accessibility and lowering barriers to progress for our customers.

We specialize primarily in Xilinx-based FPGA/SoC development boards/kits and portable USB test and measurement devices, all designed to be owned by and used from an engineer's or student's desk. We also offer a variety of expansion modules (Pmods and Zmods) to create flexible I/O options for our other products.