

# Powering FPGAs – Improving Undershoot of Voltage Regulators with Asynchronous Pulse Injection



Ryan Wallace

In a perfect world, a power supply provides an ideal voltage and never varies, but in reality, all power supplies have some error associated with the output voltage accuracy at steady state. This can cause serious problems in some applications, occasionally causing logic errors, shutdown or lasting damage. In this post, I take a look at a feature used in fixed-frequency DC/DC buck converters that can help improve transient response during a load current increase, and why devices that use this feature are worth considering for Arria 10 and Stratix 10 FPGA applications.

## Undershoot and Overshoot Explained

As previously mentioned, the output voltage accuracy of power supplies isn't perfect. When the load current increases or decreases, the power supply's output voltage can deviate from the nominal voltage, causing undershoot and overshoot. Undershoot is when the output voltage drops below the nominal voltage and overshoot is when the output voltage rises above the nominal voltage. If a power supply's output voltage deviates in excess of a powered device's input specification, unwanted behavior could result. Excessive undershoot can deprive the powered device of its minimum operating voltage and cause logic errors or shutdown while excessive overshoot can stress and damage sensitive devices.

The fluctuation of output voltage in response to a load change, or transient response, depends on several factors, such as: the rate at which the load current is changing, the magnitude of load current change, passive output stage components, compensation of the control loop, and the time it takes the converter to react.

## Introducing API

For fixed-frequency DC/DC converters, during a sudden rise in load current there is a corresponding drop in output voltage. In order to limit the amount of drop in output voltage, the on-time of the converter is increased, which increases the duty cycle. But the control loop can only increase the on-time either during an unfinished on-time or it must wait until the next switching cycle. So for a fixed-frequency DC/DC converter, if a load transient occurs in the middle of a switching cycle during the off-time, the converter will not react until the next switching cycle. As a worst-case scenario, assume that the transient occurs at the start of this off-time. Consider Equation 1 where I calculate the off-time, or potential wait time (not taking forced dead time into account):

$$\text{Off-time} = \frac{1 - \left(\frac{V_{out}}{V_{in}}\right)}{\text{Switching Frequency}} \quad (1)$$

As I mentioned, one of the contributing factors to transient response is the amount of time it takes for the converter to react. So how can you shorten the reaction time, considering the worst-case scenario? Increasing the switching frequency is one option, but has its own drawbacks, such as lower efficiency. Increasing the duty cycle ( $V_{OUT}/V_{IN}$ ) might not always be feasible since these values are likely predetermined. This is where API comes in... literally.

## How API Works to Minimize Transient Response Time and Undershoot

In order to better explain API, let's look at an example using a simulation in which I'm attempting to meet certain transient response requirements. [Table 1](#) shows a common rail voltage rating for Stratix 10 FPGAs.

**Table 1. Common Voltage Ratings for  $v_{CCIO\_SDM}$ ,  $V_{CCPT}$  And  $v_{CCIO}$  For Stratix 10 Fpgas**

Min	Typ	Max	Unit
1.71	1.80	1.89	V

Figure 1 shows a load transient simulation of a fixed-frequency converter without API. The upper waveform in Figure 1 is the output voltage, which falls below the required minimum operating voltage labelled “Min” at 1.71V.

In addition to the required maximum and minimum values for this particular rail, there is a percentage of error associated with the actual output voltage of the power supply when compared to the target voltage. If we consider the output voltage deviation based on feedback resistors with a 1% tolerance, and a reference voltage with 0.5% accuracy, the resulting output voltage accuracy error percentage is approximately  $\pm 1.27\%$ . This decreases the amount of available tolerance by  $\pm 23\text{mV}$ , which is also shown in Figure 1 labelled “Min w/ %Error” at 1.733V.

The middle waveform in Figure 1 is the switch node; here you can see that the load current (the lower waveform in Figure 1) begins to ramp up in between the two highlighted on-times. Now let’s look at the same simulation, only this time with API enabled.

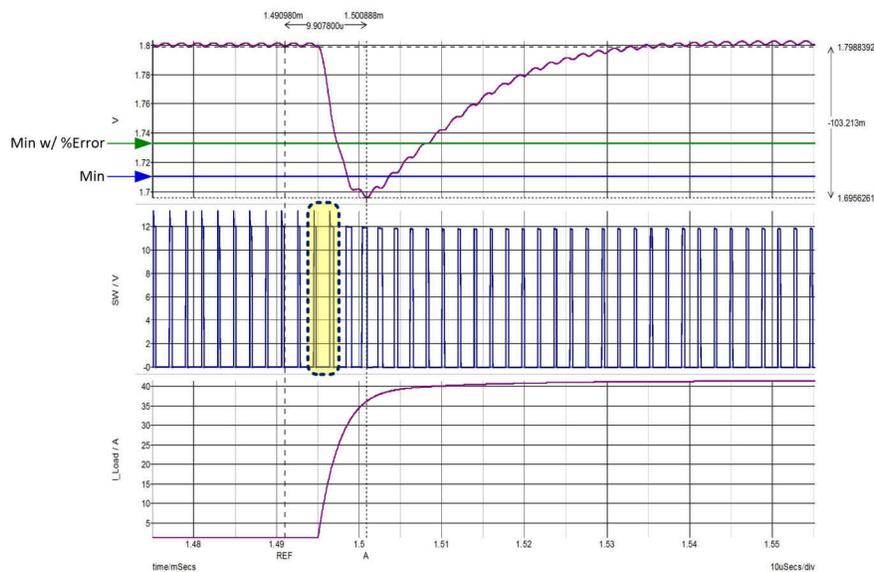
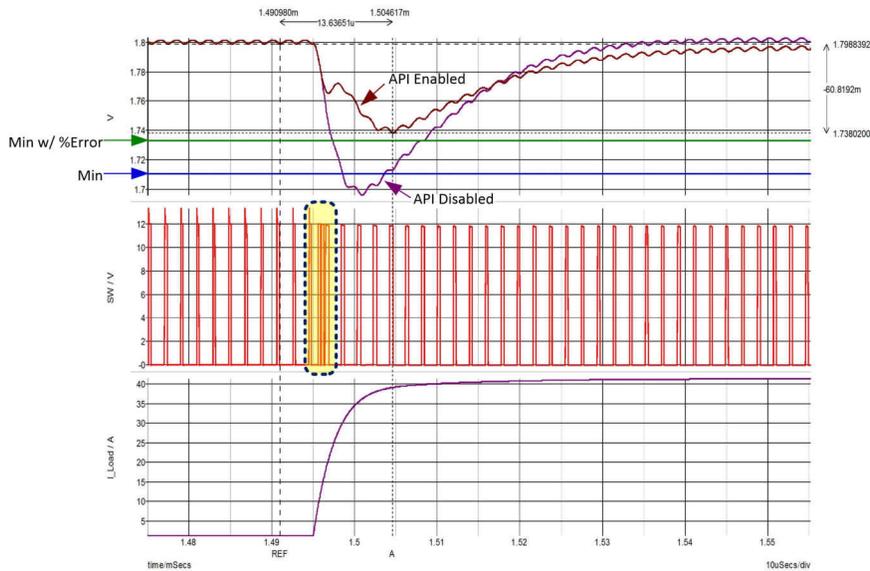

**Figure 1. A Load Transient Simulation of a Fixed-frequency Converter without API**

Figure 2 shows a load transient simulation of a fixed-frequency converter with API enabled. The upper waveform in Figure 2 is the output voltage, which now remains within the operating range of the expected voltage values listed in Table 1 labeled “Min”, as well as the minimum taking output voltage accuracy into account labeled “Min w/ %Error” at 1.733V. The middle waveform in Figure 2 is the switch node; here you can see that the load current (the lower waveform in Figure 2) ramps up at the same point in respect to the switching cycle compared to the previous simulation. Only now you can see that there are additional on-times caused by the injected pulse-width modulation (PWM) pulse.



**Figure 2. A Load Transient Simulation of a Fixed-frequency Converter with API, with the Switch Node Highlighted**

There is over 40mV of undershoot reduction in the simulation that used API for this particular model. During the load transient, the API pulses increased the duty cycle and switching frequency and reduced the delay between the occurrence of the event and the modulator’s reaction. The amount of pulses and sensitivity will depend on the specific converter with this feature and your design choices. After injecting the API pulse(s), the converter increases the on-time over a fixed frequency in order to bring the output voltage back to the target value as it normally would. One consideration when using API is that if one or more pulses are injected during a load transient, the switching frequency will vary temporarily.

**Undershoot in a Device with and without API Enabled**

Figure 3 shows the difference with and without API based on actual bench results of the [TPS543C20](#), a 40A device that’s capable of 80A when stacked. The test configuration was a 12V input voltage, 0.9V output voltage, 500kHz switching frequency and 15A step up/step down with a slew rate of 50A/μs. When comparing the same device with and without these features enabled, undershoot was approximately 36mV less with API enabled.



**Figure 3. Undershoot with API Enabled/disabled**

As shown in the simulation and bench testing enabling API resulted in less deviation from the nominal output voltage during load transients, and improved the converter’s response time, when compared to API being disabled. This feature offered in the [TPS543B20](#) and [TPS543C20](#) devices that can reduce the amount of

---

undershoot during a load current increase with the same amount of output capacitance, which is something to consider when designing in Arria 10 and Stratix 10 FPGAs.

## Conclusion

You might be wondering, what about the overshoot caused by a load current decrease? These same devices offer a feature to improve transient response during a load current decrease as well. I will be taking a look at it in my next blog: [Improving overshoot of voltage regulators for Intel Arria 10 and Stratix 10 FPGAs](#).

## Additional Resources

- Get more information on powering the [Arria 10](#) and [Stratix 10](#) FPGAs.
- Estimate actual load current and transient specifications for Altera devices with the [Early Power Estimator \(EPE\)](#) and [Power Delivery Network \(PDN\)](#).
- Learn more about [TI power management for Altera FPGAs](#).

## IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#) or other applicable terms available either on [ti.com](https://www.ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2023, Texas Instruments Incorporated