

Analog Dialogue

A forum for the exchange of circuits, systems, and software for real-world signal processing • Volume 49, Number 1, 2015

- 2 Editor's Notes; New Product Introductions
- 3 FPGA-Based Systems Increase Motor-Control Performance
- 11 Powering ICs On and Off
- 15 Design a PLL Loop Filter when Only the Zero Resistor and Capacitor Are Adjustable
- 21 Quickly Implement JESD204B on a Xilinx FPGA
- 25 Power Management for Integrated RF ICs
- 31 Design Reliable Digital Interfaces for Successive-Approximation ADCs



Editor's Notes

IN THIS ISSUE

FPGA-Based Systems Increase Motor-Control Performance

Advanced motor-control systems combine control algorithms, industrial networks, and user interfaces, so they require additional processing power to execute all tasks in real time. Multichip architectures are used to implement modern motor-control systems: a DSP executes motor-control algorithms, an FPGA implements high-speed I/O and networking protocols, and a microprocessor handles executive control. (Page 3)

Powering ICs On and Off

Modern integrated circuits employ sophisticated circuits to ensure that they turn on in a known state; and preserve memory, boot quickly, and conserve power when they are powered down. This two-part article provides tips for using power-on reset and power-down functions. (Page 11)

Design a PLL Loop Filter when Only the Zero Resistor and Capacitor Are Adjustable

A standard procedure uses open-loop bandwidth and phase margin to determine the component values for a PLL loop filter, solving for the pole capacitor and deriving the remaining values. In some cases this capacitor may be integrated, so the standard procedure can't be used. This article proposes an alternative procedure that can be used when the value of the pole capacitor is fixed. (Page 15)

Quickly Implement JESD204B on a Xilinx FPGA

The JESD204 high-speed serial interface connects data converters to logic devices. As the speed and resolution of converters continues to increase, this interface has become common in ADCs, DACs, and RF transceivers. Serializer/deserializer designs in FPGAs implement the physical layer. This article describes how to quickly set up a project using a Xilinx® FPGA to implement the JESD204B interface. (Page 21)

Power Management for Integrated RF ICs

As more building blocks are added to a radio-frequency integrated circuit, more sources of noise coupling arise, making power management increasingly important. This article describes how power-supply noise can affect the performance of RF ICs. The ADRF6820 quadrature demodulator with integrated phase-locked loop and a voltage-controlled oscillator is used as an example, but the results are broadly applicable to other high-performance RF ICs. (Page 25)

Design Reliable Digital Interfaces for Successive-Approximation ADCs

SAR ADCs provide up to 18-bit resolution at up to 5 MSPS. A host processor can access the ADC via a variety of serial and parallel interfaces such as SPI, I²C, and LVDS. This article discusses design techniques for reliable, integrated digital interfaces, including the digital power-supply level and sequence, I/O state during turn on, interface timing, signal quality, and errors caused by digital activity. (Page 31)

Jim Surber [jim.surber@analog.com]

Product Introductions: Volume 49, Number 1

Data sheets for all ADI products can be found by entering the part number in the search box at analog.com.

February

ADC, Σ - Δ , 3-channel, SPI interface ADE7903
ADC, Σ - Δ , 3-channel, SPI interface ADE7923
Amplifier, operational, 200-MHz BW, RRIO ADA4807-1
Codec, audio, four ADCs, two DACs ADAU1372
Data-acquisition system, 4-channel, 16-bit ADAR7251
Data recovery IC, 614.4-Mbps to 10.3125-Gbps ADN2905
Input protection device, high-voltage ADM1270
Mixer, receive, passive, dual, 700-MHz to 3000-MHz ADRF6612
Mixer, receive, wideband, dual, integrated IF amplifiers ADRF6658
Multiplexer, 4-channel, 10- Ω , fault detection/protection ADG5404F
Receiver, IF diversity, 385-MHz BW AD6674
Receiver, optical, 11.3-Gbps ADN3010-11
Regulator, buck, triple, 1800-mA ADP5135

March

ADC, pipelined, dual, 16-bit, 125-MSPS, LVDS outputs AD9655
ADC, pipelined, 14-bit, 1-GSPS/500-MSPS AD9690
Amplifier, operational, quad, EMI/overvoltage protection ADA4177-4
Channel protector, quad, user-defined fault protection/detection ADG5462F
Controller, digital, isolated power supply with PMBus interface ADP1052
DAC, nanoDAC+, octal, 16-bit, I²C interface AD5675
DACs, voltage-output, 12-/16-bit, unipolar/bipolar AD5721/AD5761
DACs, voltage-output, 12-/16-bit, unipolar/bipolar, 2-ppm/ $^{\circ}$ C reference AD5721R/AD5761R
Detectors, voltage, ultralow-power ADM8641/ADM8642
Modulator, Σ - Δ , 16-bit, isolated AD7402
Regulators, switching, 2-channel, bipolar rails ADP5070/ADP5071
Switch, dual SPDT, 10- Ω , fault protection/detection ADG5436F
Switch, power, high-side, 12-V, 2-A, logic-controlled ADP1290
Supervisory ICs, ultralow-power, manual reset ADM8611/ADM8612
Supervisory ICs, ultralow-power, watchdog timer, manual reset ADM8613/ADM8614/ADM8615

Analog Dialogue

Analog Dialogue, www.analog.com/analogdialogue, the technical magazine of Analog Devices, discusses products, applications, technology, and techniques for analog, digital, and mixed-signal processing. Published continuously for 49 years—starting in 1967—it is available in two versions. Monthly editions offer technical articles; timely information including recent application notes, circuit notes, new-product briefs, webinars, and published articles; and a universe of links to important and relevant information on the Analog Devices website, www.analog.com. Printable quarterly issues and ebook versions feature collections of monthly articles. For history buffs, the *Analog Dialogue* archive, www.analog.com/library/analogdialogue/archives.html, includes all regular editions, starting with Volume 1, Number 1 (1967), and three special anniversary issues. To subscribe, please go to www.analog.com/library/analogdialogue/subscribe.html. Your comments are always welcome: Facebook: www.facebook.com/analogdialogue; EngineerZone: ez.analog.com/blogs/analogdialogue; Email: dialogue.editor@analog.com or Jim Surber, Editor [jim.surber@analog.com].

FPGA-Based Systems Increase Motor-Control Performance

By Andrei Cozma and Eric Cigan

Introduction

Used in a wide range of industrial, automotive, and commercial applications, electric motors are controlled by drives that vary the electrical input power to control the torque, speed, and position. High-performance motor drives can increase efficiency and deliver faster, more accurate control. Advanced [motor-control systems](#) combine control algorithms, industrial networks, and user interfaces, so they require additional processing power to execute all tasks in real time. Multi-chip architectures are typically used to implement modern motor-control systems: a digital signal processor (DSP) executes motor-control algorithms, an FPGA implements high-speed I/O and networking protocols, and a microprocessor handles executive control.¹

With the advent of system-on-chip (SoC) devices such as the Xilinx [Zynq All Programmable SoC](#), which combines the versatility of a CPU and the processing power of an FPGA, designers have the means to consolidate motor-control functions and additional processing tasks into a single device. Control algorithms, networking, and other processing intensive tasks are offloaded to the programmable logic, while supervisory control, system monitoring and diagnosis, user interface, and commissioning are handled by the processing unit. The programmable logic can include multiple control cores that run in parallel to implement multiaxis machines or multiple control systems. Having the complete controller on a single chip allows the hardware design to be simpler, more reliable, and less expensive.

In recent years, software modeling and simulation tools, such as [Simulink](#)[®] from MathWorks[®], have allowed model-based design to evolve into a complete design flow—from model creation to implementation.² Transforming the way engineers and scientists work, [model-based design](#) is moving

design tasks from the lab and field to the desktop. Now the entire system—including the plant and the controller—can be modeled, allowing the engineer to tune the controller's behavior before deploying it in the field. This reduces the risk of damage, accelerates system integration, and reduces dependency on equipment availability. Once the control model is complete, it can be automatically translated by the Simulink environment into C and HDL code that will be run by the control system, saving time and avoiding manual coding errors. The risk is further reduced by linking the system model to a rapid prototyping environment that allows observation of how the controller will operate in real-life conditions.

A complete development environment for achieving increased motor-control performance uses the Zynq SoC from Xilinx for implementing the controller, Simulink from MathWorks for model-based design and automatic code generation, and the [Intelligent Drives Kit](#) from Analog Devices for rapid prototyping of the drive system.

Xilinx FPGA and SoC Motor-Control Solutions

Advanced [motor-control](#) systems must execute a combination of control, communication, and user interface tasks, each of which has different processing bandwidth requirements and real-time constraints. The hardware platform chosen to implement such a control system must be robust and scalable while allowing for future system improvements and expansion. The Zynq All Programmable SoC fulfills these requirements by combining a high-performance processing system with programmable logic, as shown in Figure 1. This combination delivers superior parallel-processing power, real-time performance, fast computation, and versatile connectivity. The SoC integrates two Xilinx analog-to-digital converters (XADC) for monitoring the system or external analog sensors.

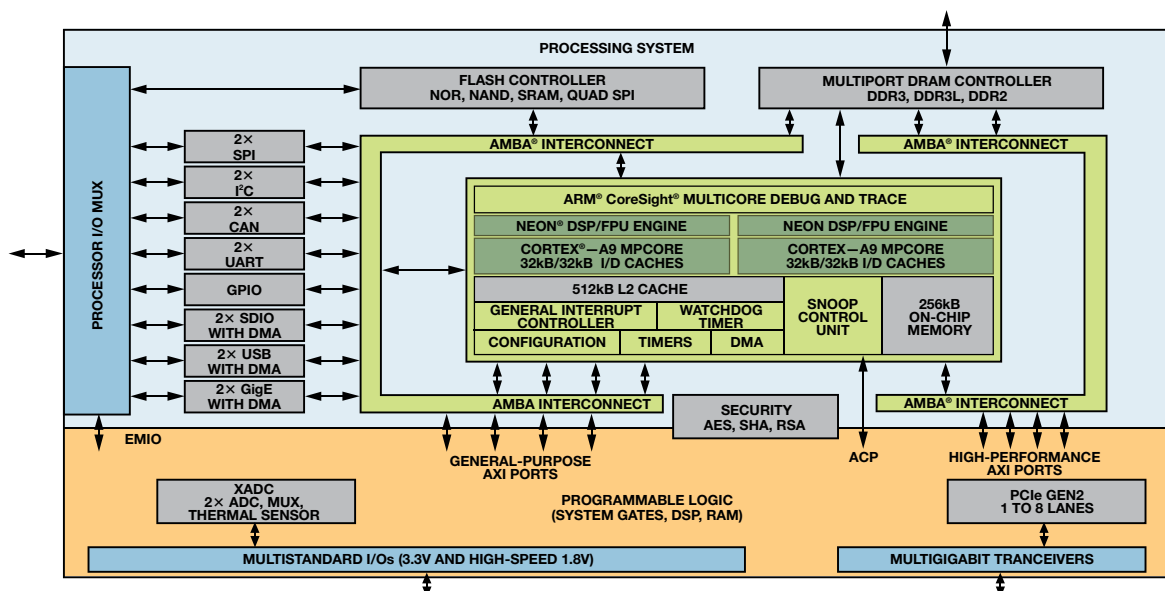


Figure 1. Xilinx Zynq SoC block diagram.

The processing side of the Zynq consists of a dual-core ARM Cortex-A9 processor, a NEON coprocessor, and floating-point extensions that accelerate software execution. The processing system addresses tasks such as supervisory control, motion control, system management, user interface, and remote maintenance functions that are well-suited to software implementation. Embedded Linux or real-time operating systems can be deployed to take advantage of the system's capabilities. The self-contained processor can be used without the need to configure the programmable logic. This allows software developers to write code in parallel with hardware engineers who design the FPGA fabric.

On the programmable logic side, the device has up to 444,000 logic cells and 2200 DSP slices that supply massive processing bandwidth. The FPGA fabric is scalable, so a user can choose from a small device with 28,000 logic cells all the way up to a high-end device, which can tackle the most challenging signal-processing applications. Five AMBA-4 AXI high-speed interconnects tightly couple the programmable logic to the processing system, giving the equivalent of more than 3000 pins of effective bandwidth. The programmable logic is suitable for implementing time critical, processing intensive tasks such as real-time industrial Ethernet protocols, and it can accommodate multiple control cores that run in parallel for multi-axis machines or multiple control systems.

Xilinx All Programmable SoC-based solutions and platforms meet the critical timing and performance requirements posed by today's complex control algorithms such as field-oriented control (FOC) and complex modulation schemes such as the regenerative pulse frequency modulator³ designed by Xilinx and Qdesys.

Model-Based Design Using Simulink from MathWorks

Simulink is a block diagram environment for multidomain simulation and model-based design that is well-suited to simulating systems that include control algorithms and plant models. Motor-control algorithms regulate speed, torque, and other parameters, often for precision positioning. Evaluating control algorithms using simulation is an effective way to determine the suitability of [motor-control designs](#) and reduce the time and cost of algorithm development before committing to expensive hardware testing. Figure 2 depicts an efficient workflow for designing a motor-control algorithm:

- Build accurate controller and plant models, often from libraries of motors, drive electronics, sensors, and loads

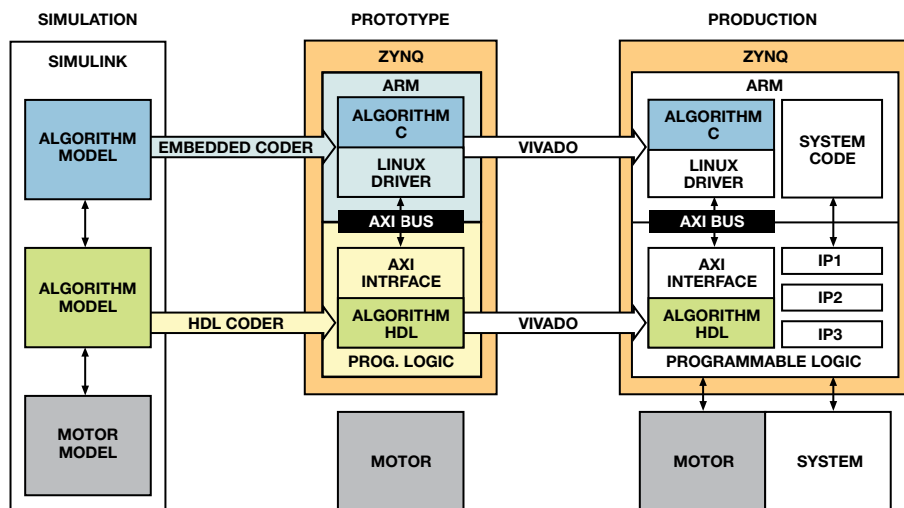


Figure 3. Path from simulation to production.

- Simulate system behavior to verify that the controller is performing as expected
- Generate C code and HDL for real-time testing and implementation
- Test control algorithms using prototyping hardware
- Deploy the controller onto the final production system once the control system has proven to be satisfactory through simulation and testing on prototyping hardware

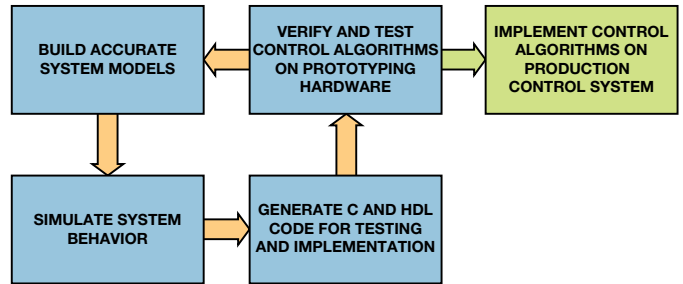


Figure 2. Workflow for motor-control algorithm design.

MathWorks products including the [Control System Toolbox](#),TM [SimPowerSystems](#),TM and [Simscape](#)TM provide industry-standard algorithms and applications for systematically analyzing, designing, and tuning linear control systems, as well as component libraries and analysis tools for modeling and simulating systems spanning mechanical, electrical, hydraulic, and other physical domains. These tools provide the means to create high-fidelity plant and controller models that can verify the behavior and performance of the control system before moving to the physical implementation. The simulation environment is the perfect place to verify functionality corner cases and extreme operating conditions to ensure that the controller is prepared for such situations, and its real-life operation will be safe for both equipment and operating personnel.

Once the control system is fully verified in the simulation environment using the embedded coder and the HDL coder tools, it can be translated into C code and HDL and deployed on prototyping hardware for testing and to the final production system afterwards. At this point software and hardware implementation such as fixed-point and timing behavior requirements are specified. Automatic code generation helps to reduce the time needed to move from concept to actual system implementation, eliminates coding errors, and ensures that the actual implementation matches the model. Figure 3 depicts the real-life steps needed to model a motor controller in Simulink and transfer it to the final production system.

The first step is to model and simulate the controller and the plant in Simulink. At this stage, the controller algorithm is partitioned into blocks that will be implemented in software, and blocks that will be implemented in programmable logic. Once the partitioning and the simulation are complete, the controller model is converted into C code and HDL using the embedded coder and HDL coder. A Zynq-based prototyping system verifies the performance of the control algorithm and helps further tune the controller model before moving to the production stage. In the production stage, the automatically generated C code and HDL are integrated into the complex production system framework. This workflow ensures that once the control algorithm reaches the production stage it is fully verified and tested, thus providing high confidence in the system robustness.

Rapid Prototyping with the Analog Devices Intelligent Drives Kit

Choosing the right prototyping hardware is a major step in the design process. The Analog Devices Intelligent Drives Kit enables rapid, efficient prototyping. Combining the Zynq-7000 All Programmable SoC ARM dual-core Cortex-A9 with 28 nm programmable logic with the latest generation Analog Devices high-precision data converters and digital isolation, the Avnet [Zynq-7000 All Programmable SoC/Analog Devices Intelligent Drives Kit](#) enables high-performance motor control and dual gigabit Ethernet industrial networking connectivity. The kit comes with an Avnet ZedBoard 7020 baseboard and ADI's AD-FMCMOTCON1-EBZ module, which is a complete drive system providing efficient control for multiple motor types. In addition, the kit can be extended with ADI's AD-DYNO1-EBZ [dynamometer drive system](#), a dynamically adjustable load that can be used to test real-time motor-control performance. The AD-FMCMOTCON1-EBZ module consists of a controller and a drive board, as shown in Figure 4.

The controller board is a mixed-signal FPGA mezzanine card (FMC), designed to connect to any Xilinx FPGA or SoC platforms with low pin count (LPC) or high pin count (HPC) FMC connectors. It features:

- Current and voltage measurement using isolated ADCs
- An isolated Xilinx XADC interface
- Fully isolated digital control and feedback signals
- Hall, differential Hall, encoder, and resolver interfaces
- 2-Gb Ethernet PHYs to enable high-speed industrial communication protocols such as EtherCAT, ProfiNET, Ethernet/IP, or Powerlink
- FMC signals voltage adaptation interface for seamless operation on all FMC voltage levels

Isolation, a critical aspect of any motor-control system, is required to protect the controller as well as the user. Full isolation of the analog and digital signals on the controller board ensures that the FPGA platform is always protected from dangerous voltages that can arise on the motor drive side.

The drive board contains all the power electronics needed to drive the motors as well as current and voltage sensing and protection circuits. The board features:

- Drives BLDC (brushless dc)/PMSM (permanent-magnet synchronous motor)/brushed dc/stepper motors in the 12-V to 48-V range with 18-A maximum current
- Dynamic braking capability and integrated overcurrent and reverse voltage protection
- Phase-current measurement using isolated ADCs; programmable-gain amplifiers maximize the current measurement input range
- Provides dc bus voltage, phase current, and total current feedback signals to the controller board
- Integrated BEMF zero-crossing detection for sensorless control of PMSM or BLDC motors

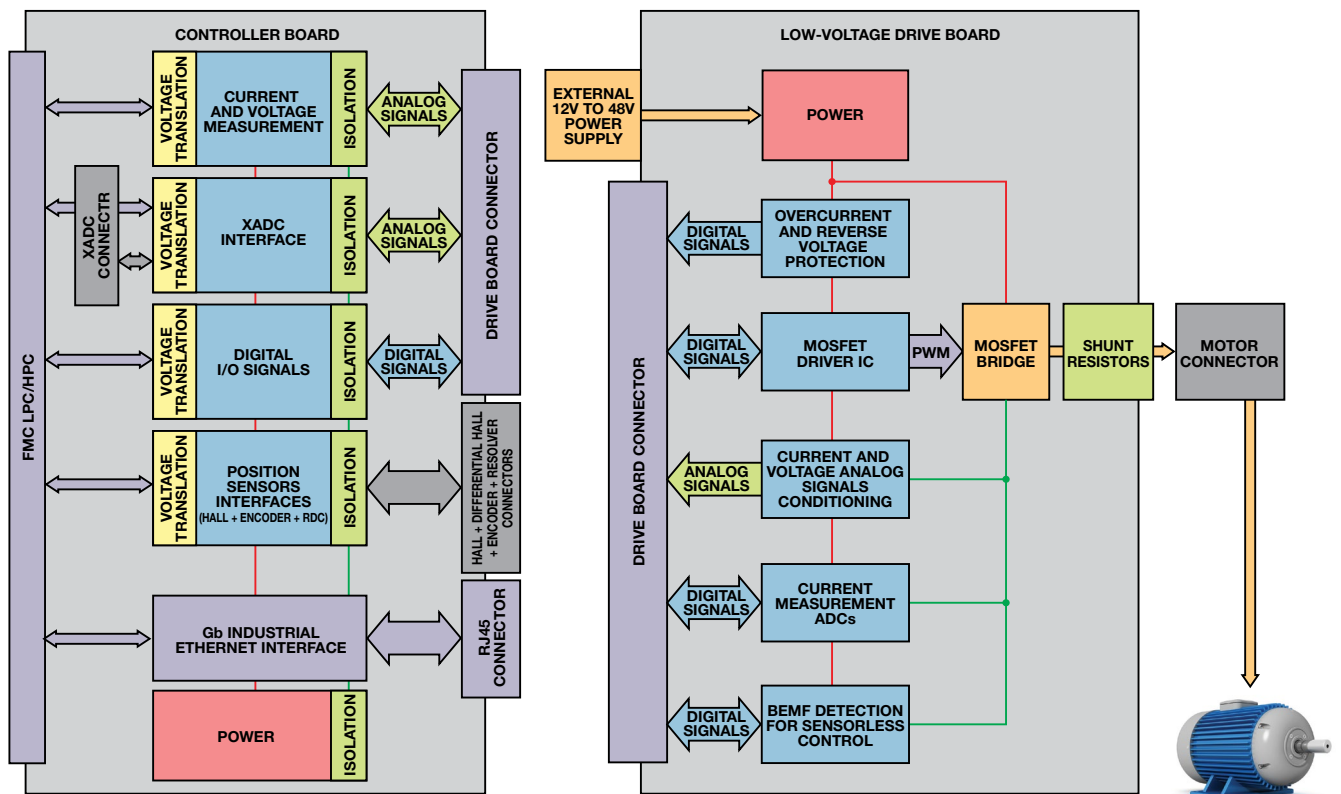


Figure 4. AD-FMCMOTCON1-EBZ block diagram.

The dynamometer, which is a dynamically adjustable load that can be used to test real-time motor-control performance, consists of two BLDC motors directly coupled through a rigid connection. One of the BLDC motors acts as a load and is controlled by the dynamometer's embedded control system; the second motor is driven by the ADI Intelligent Drives Kit, as shown in Figure 5. The system, equipped with a user interface that displays information about the load current and speed, allows different load profiles to be set. External control can be achieved by using the Analog Discovery USB oscilloscope for load signal capture and control directly from MATLAB® using the MathWorks Instrument Control Toolbox.™

The performance of any motor-control system is greatly influenced by the quality of motor current and voltage measurements. By using high-performance analog signal conditioning components and ADCs, the ADI Intelligent Drives Kit provides precise current and voltage measurements. The measurement paths are divided between the controller and the drive board as shown in Figure 6.

The phase currents are sensed by measuring the voltage across shunt resistors. Two possible measurement paths aim to get the best measurement accuracy depending on whether the ADC is close to the shunt resistor or not. When the ADC is

close to the shunt resistor, the signal path is very short and less prone to noise coupling. The small differential voltage on the shunt resistor is measured directly with the AD7401 isolated Σ - Δ modulator without the need for extra interfacing and signal conditioning circuitry. When the ADC is far from the shunt resistor, the signal path is long and prone to noise coupling, especially from power supply switching noise and the motor. Special care must be taken to ensure that the PC board traces and signal conditioning circuitry between the ADC and the shunt resistor are properly shielded. The small differential voltage on the shunt resistor is amplified on the drive board with the AD8207 difference amplifier, which is placed close to the shunt resistor to avoid noise coupling. The signal is amplified from a ± 125 -mV full-scale input range to a ± 2.5 -V range to minimize the effect of the coupled noise. The amplified signal goes through another amplification stage using the AD8251 programmable-gain instrumentation amplifier (PGA), ensuring that the ADC always receives input signals that are properly scaled to fit the input range. The amplified analog signals go through the connector to the controller board. The connector includes shielding for each analog signal to mitigate noise coupling. The analog signals coming from the drive board are shifted back to the AD7401 input range using the ADA4084-2 operational amplifier.

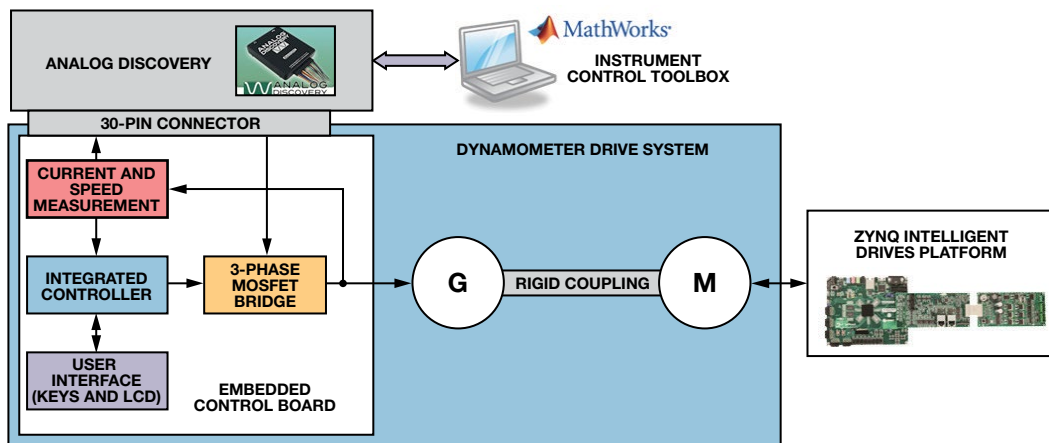


Figure 5. Dynamometer drive system.

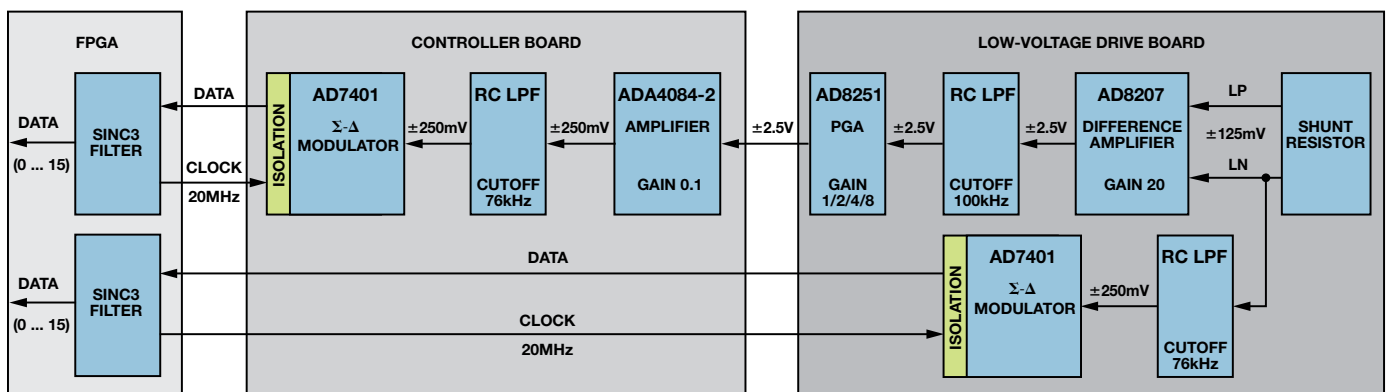


Figure 6. Phase-current signal chain.

The most important part in the current and voltage feedback signal chain is the AD7401A second-order isolated Σ - Δ modulator. This high-performance ADC features 16-bit resolution with no missing codes, 13.3 effective number of bits (ENOB), and 83-dB SNR. The 2-wire digital interface includes a 20-MHz clock input and a 1-bit digital bitstream output. The ADC output is reconstructed using a sinc³ digital filter. A filter model and HDL implementation are provided in the data sheet for a 16-bit output and a 78-kHz sampling rate. The output resolution and sampling rate can be controlled by changing the filter model and decimation. While the 78-kHz sampling rate might be good enough for many applications, some situations require a higher rate. In these cases, filter banks, as shown in Figure 7, can be used to increase the sampling rate of the system up to 10 MSPS of true 16-bit data. The filter bank contains n sinc³ filters with sampling clocks that are delayed by multiples of T , which is the sinc³ filter

propagation time divided by n . The data selector outputs the ADC code with a periodicity equal to T .

Phase-current measurements can be also performed by the Zynq XADC. The XADC signal measurement chain uses the entire path of the regular measurement chain and adds a Sallen-Key analog reconstruction filter after the AD7401 Σ - Δ modulator. This filter is implemented on the controller board using AD8646 operational amplifiers, as shown in Figure 8. The combination of the isolated Σ - Δ modulator and the analog reconstruction filter provides a convenient, low-cost way to achieve analog isolation of the XADC input signals without compromising the quality of the measurement.

The Analog Devices Intelligent Drives Kit comes with a set of Simulink controller models, a complete Xilinx Vivado framework, and an ADI Linux infrastructure, allowing a user to go through all the steps needed to design a motor-control system, starting with simulation, going through prototyping, and finishing with the production system implementation.

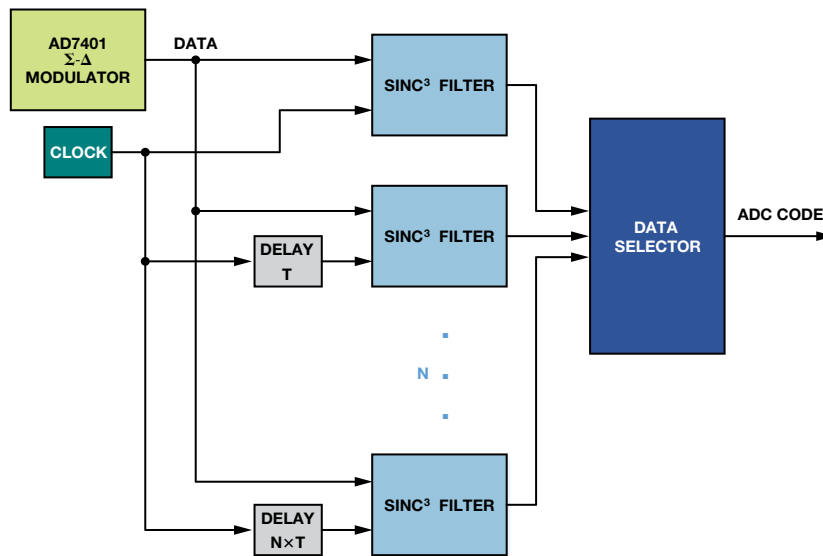


Figure 7. Filter bank.

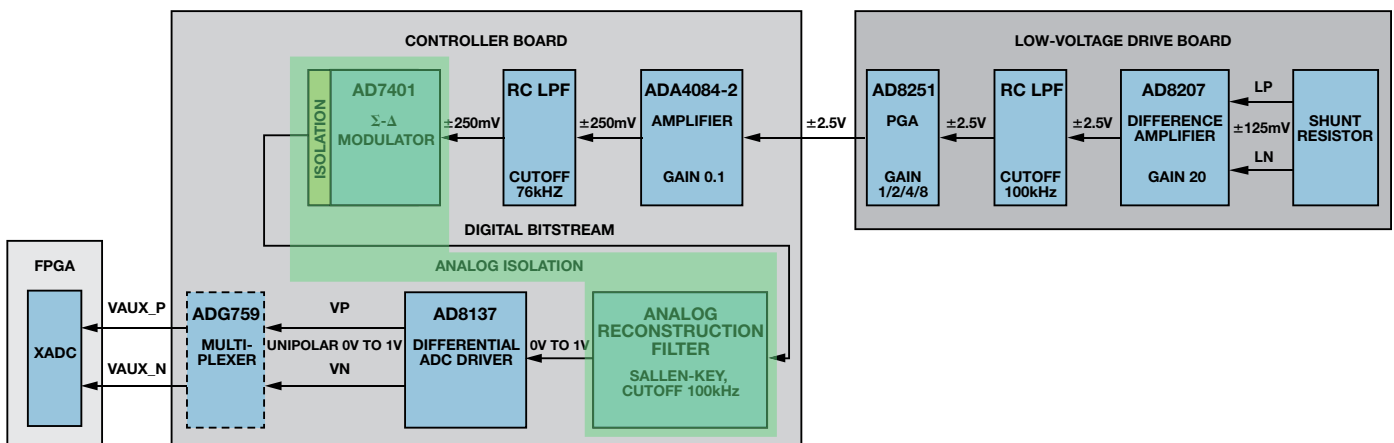


Figure 8. XADC signal measurement chain.

Two controller models—a six-step controller and a PMSM field-oriented controller—can be used to start the design process. Figure 9 shows the top-level views of these two controllers. The six-step controller implements a trapezoidal controller for BLDC motors; the FOC controller provides an FOC core for integration within the control system.

The plant and controller models are created in the simulation stage, and the behavior of the complete system is simulated

to verify that the controller is performing as expected. The controller model is partitioned into components that will be implemented in C code and HDL, and constraints such as timing, fixed-point implementation, sampling rates, and loop times are specified to ensure that the controller model behaves as it will in the hardware implementation. Figure 10 shows the partitioning of the six-step controller between software and HDL.

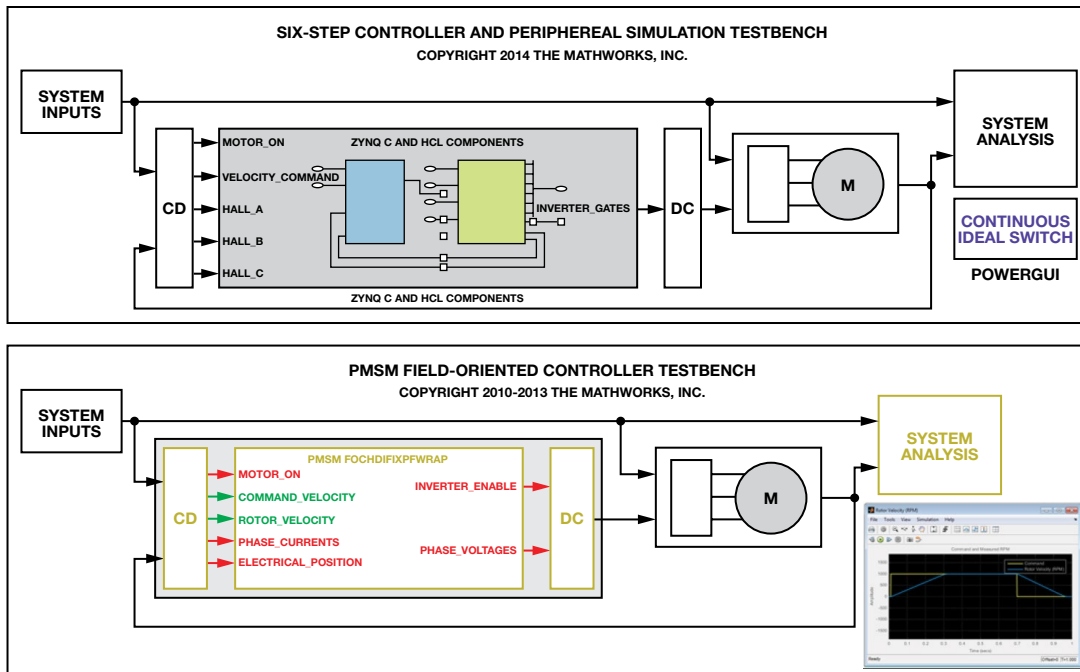


Figure 9. Simulink controller models.

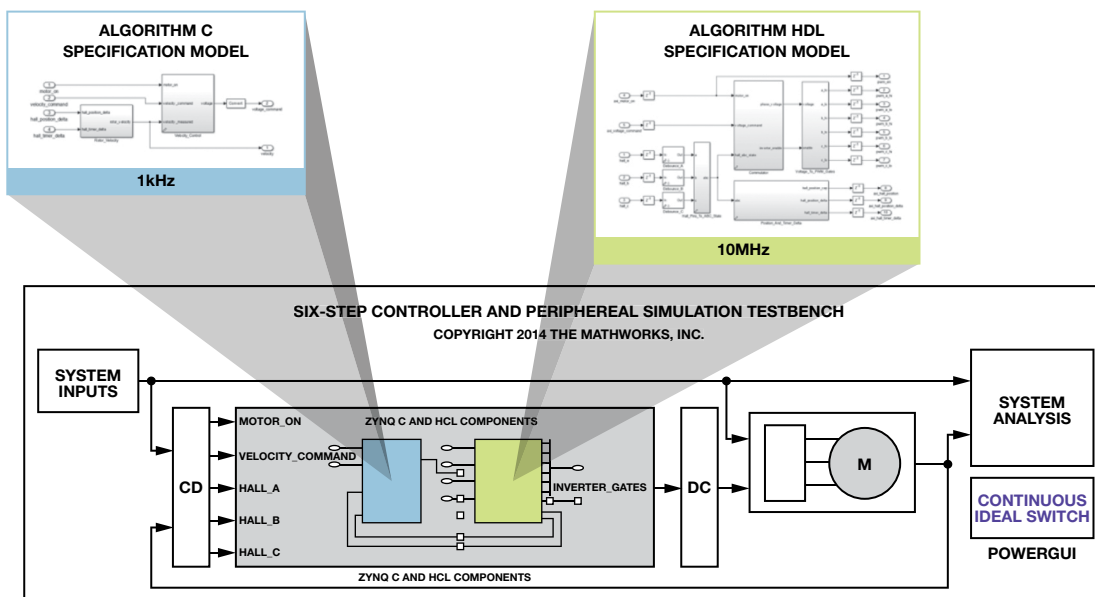


Figure 10. Controller partitioning in C code and HDL.

Once the controller is fully verified in simulation, the next step is to prototype it on the hardware platform. The Zynq SoC guided workflow generates the C code and HDL from the Simulink model partitioned into subsystems targeting the ARM core and the programmable logic. With this workflow, the HDL coder generates the HDL that targets the programmable logic, while the embedded coder generates the C code targeting the ARM. The MathWorks Zynq support package enables the generation of the ARM executable consisting of algorithmic C code from models, which interfaces to the AXI bus, as well as bitstream generation consisting of HDL code from models, which interfaces to programmable logic pins and AXI bus. Figure 11 shows the controller implementation and the relationship with the ADI Intelligent Drive hardware.

Once the bitstream and executable are loaded into the hardware, operational testing of the controller can begin. Hardware-in-the-loop (HIL) testing is performed using an Ethernet link between Simulink and the embedded system running an open-source Linux OS. Motor parameters such as shaft speed can be captured in Simulink and compared against simulation results to make sure that the physical system implementation matches the model. Once the control algorithm testing is complete, the controller can be transferred to the production system.

Together with the Intelligent Drives Kit, Analog Devices provides a complete Vivado framework and a Linux infrastructure that can be used for both prototyping and final production. Figure 12 shows the Zynq Infrastructure that supports the Intelligent Drives Kit. This high-level diagram shows how the ADI reference design is partitioned on a Xilinx Zynq SoC. The programmable logic implements the IP cores for interfacing with the ADCs, position sensors, and the motor drive stage. The HDL, generated by the HDL coder to represent the motor-control algorithm, is integrated into the Analog Devices IP. All the IPs have low-speed AXI-Lite interfaces for configuration and control, and high-speed AXI-Streaming interfaces that allow them to transfer real-time data through DMA channels to the software level. The high-speed Ethernet interfaces can be implemented using either the hard MAC peripherals of the ARM processing system or the Xilinx Ethernet IPs in the programmable logic.

The ARM Cortex-A9 processing system runs Ubuntu Linux, provided by Analog Devices. This includes the Linux IIO drivers needed to interface with the Analog Devices Intelligent Drive hardware, the **IIO Oscilloscope** (Scope) user space application for monitoring and control, a **libiio** server that allows real-time data acquisition and system control over TCP, clients running on a remote computer, and optional user applications that incorporate C code generated by the embedded coder.

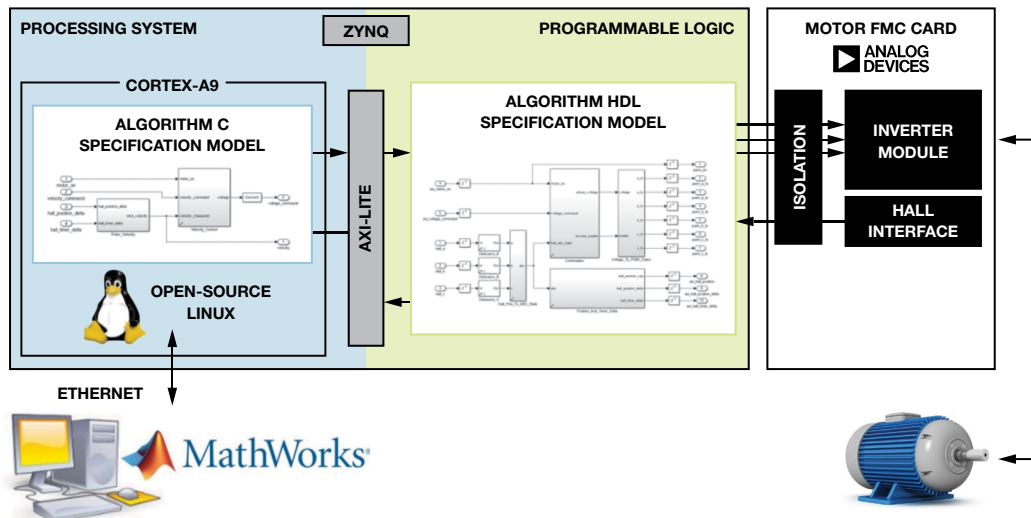


Figure 11. Controller implementation on the prototyping system.

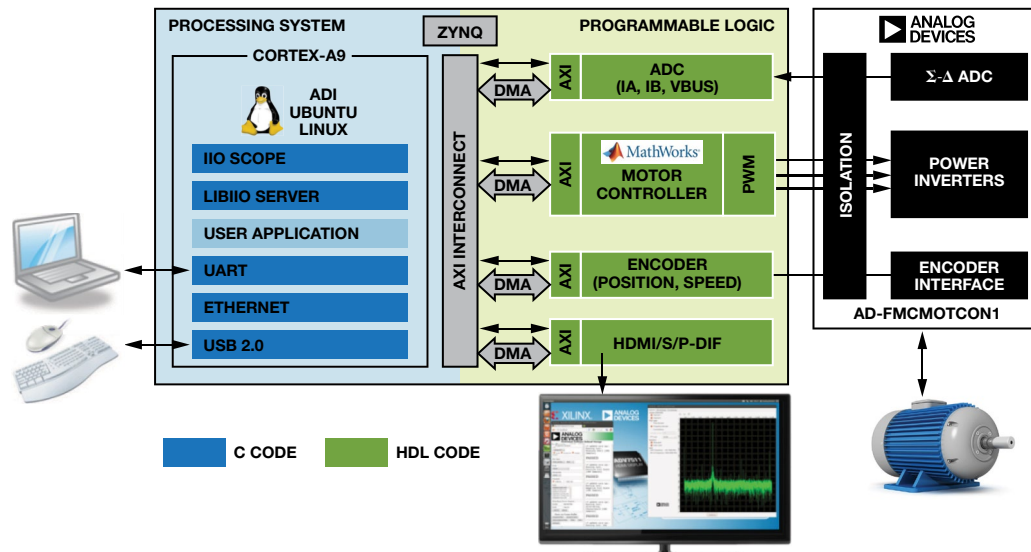


Figure 12. ADI Linux infrastructure.

All the ADI Linux drivers are based on the Linux Industrial I/O (IIO) subsystem, which is now included in all mainline Linux kernels. The IIO Scope, an open-source Linux application developed by Analog Devices that runs on the dual ARM Cortex-A9s inside the Xilinx Zynq, has the ability to display real-time data acquired from any Analog Devices FMC card connected to the Xilinx Zynq platform. The data can be displayed either in the time domain, frequency domain, or as a constellation plot. Different popular file formats, such as comma separated values or .mat Matlab files, are supported to save the captured data for further analysis. The IIO Scope provides a graphical user interface for changing or reading back the configuration of the Analog Devices FMC cards.

The libiio server allows real-time data acquisition and system control over TCP together with clients running on a remote computer. The server runs on an embedded target under Linux and manages real-time data exchange over TCP between the target and a remote client. An IIO client is available as a system object to be integrated in native MATLAB and Simulink applications. An HDMI output is used to display the Linux interface on a monitor while a keyboard and mouse can be connected to the system on a USB 2.0 port.

The Linux software and HDL infrastructure provided by ADI for the Intelligent Drives Kit, together with the tools provided by MathWorks and Xilinx, are ideal for prototyping motor-control applications. They also contain production-ready components that can be integrated into the final control system, thus helping to reduce the time and cost needed to move from concept to production.

Conclusion

This article illustrates the requirements and trends of the modern FPGA-enabled motor-control systems and the tools and systems that MathWorks, Xilinx, and Analog Devices bring to the market in order to meet these constraints and help drive toward more efficient and accurate motor-control solutions. By combining the model-based design and automatic code generation tools from MathWorks with the powerful Xilinx Zynq SoCs and the Analog Devices isolation, power, signal conditioning, and measurement solutions, the design, verification, testing, and implementation of motor drive systems can be more effective than ever, leading to improved motor-control performance and reduced time to market. The Analog Devices Intelligent Drives Kit paired with the Avnet Zynq-7000 All Programmable SoC provides a great prototyping environment for the motor-control algorithms designed using Simulink from MathWorks. The Intelligent Drives Kit comes with a set of reference designs⁴ intended to give a starting point for anyone who wants to evaluate the system and help kick-start any new motor control project.

References

1. Hill, Tom. "Motor Drives Migrate to Zynq SoC with Help from Matlab." *Xcell Journal*, Issue 87, Second Quarter 2014.
2. O'Sullivan, Dara, Jens Sorensen, and Anders Frederiksen. "Model Based Design Tools in Closed Loop Motor Control." *PCIM Europe*, 2014.
3. Corradi, Dr. Giulio. "Fpga High Efficiency, Low Noise Pulse Frequency Space Vector Modulation—Part 1." *EDN Network*, October 04, 2012.
4. AD-FMCMOTCON1-EBZ User Guide.



Andrei Cozma [andrei.cozma@analog.com] is an engineering manager for ADI, supporting the design and development of system level reference designs. He holds a B.S. degree in industrial automation and informatics and a Ph.D. in electronics and telecommunications. He has been involved in the design and development of projects from different industry fields such as motor control, industrial automation, software-defined radio, and telecommunications.



Andrei Cozma

Eric Cigan [Eric.Cigan@mathworks.com] is in MathWorks technical marketing, supporting SoC and FPGA design workflows. Prior to joining MathWorks, he held technical marketing roles at MathStar, AccelChip, and Mentor Graphics. Eric earned a B.S. and M.S. degree in mechanical engineering from the Massachusetts Institute of Technology.



Eric Cigan

Powering ICs On and Off

Modern integrated circuits employ sophisticated circuits to ensure that they turn on in a known state, preserve memory, boot quickly, and conserve power when they are powered down. This two-part article provides tips for using power-on reset and power-down functions.

Power-On Reset

By Miguel Usach Merino

Introduction

The power-on reset (POR) circuit included in many ICs guarantees that the analog and digital blocks initialize in a known state after the power supply is applied. The basic POR function generates an internal reset pulse to avoid race conditions and keep the device static until the supply voltage reaches a threshold that guarantees correct operation. Note that this threshold voltage is not the same as the minimum power-supply voltage shown in the data sheet. Once the supply reaches the threshold voltage, the POR circuit releases the internal reset signal and the state machine initializes the device. Until initialization is complete, the device should ignore external signals, including transmitted data. The only exception is the reset pin, which, if included, would be internally gated with the POR signal. A POR circuit can be represented as a window comparator, as shown in Figure 1. The comparator level, V_{T2} , is defined during the circuit design depending on the device's operational voltage and process geometry.

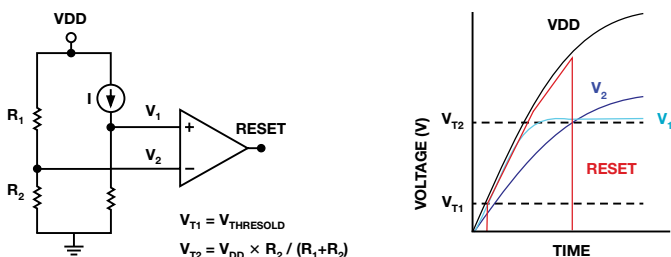


Figure 1. Simplified POR circuit.

POR Strategy

The comparator window is typically defined by the digital supply level. The digital block controls the analog block, and the voltage required for the digital block to be fully functional is similar to the minimum voltage required for the analog block to function, as shown in Figure 2.

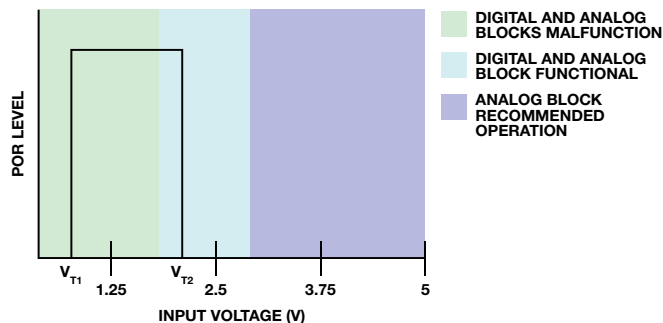


Figure 2. POR threshold voltages.

A higher threshold for V_{T2} is better for the analog block, but making it too close to the minimum recommended supply voltage could inadvertently trigger a reset if the voltage drops slightly. If the device includes separate analog and digital supplies, a strategy to avoid malfunctions is to add a second POR circuit that keeps both blocks reset until the supply voltage is high enough to ensure functionality. For example, in a 3-V IC process, $V_{T1} \approx 0.8$ V and $V_{T2} \approx 1.6$ V.

These voltages can change depending on the process and other design variations, but these are reasonable approximations. The threshold tolerance can be 20% or more; some old designs had up to 40% tolerance. The high tolerance is related to power consumption. The POR must be enabled all of the time, so the ever-present trade-off between accuracy and power consumption is important, as higher accuracy will make the circuit dissipate more power in standby mode without making a real difference in functionality.

Brownout Detector

The POR circuit sometimes integrates a brownout detector (BOD), which avoids malfunction by preventing a reset if the voltage drops unexpectedly for only a short time. From a practical perspective, the brownout circuit adds hysteresis, typically around 300 mV, to the threshold voltages defined in the POR block. The BOD guarantees that once the supply falls above V_{T2} , the POR will not generate a reset pulse unless the supply drops below a different threshold, V_{BOD} , as shown in Figure 3.

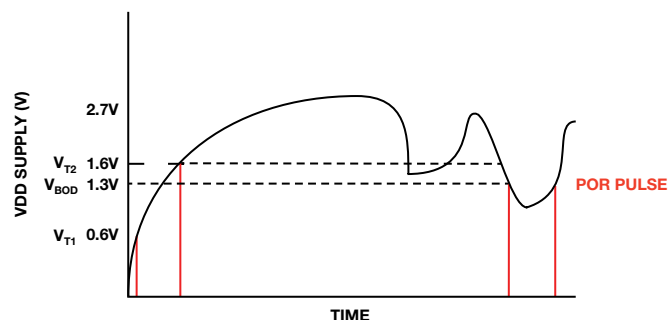


Figure 3. Brownout detector.

The brownout threshold level is high enough to guarantee that the digital circuit retains the information, but is not high enough to guarantee functionality. This allows the controller to halt activity if the supply drops below some level, without requiring the device to be reinitialized if the supply level drops for a limited amount of time.

Correct Device Power Up

Practical POR circuits are more complex than the simple version shown in Figure 1; using MOS transistors in place of resistors, for example. Thus, parasitic models must be considered. Also, the POR circuit requires a start-up block to generate the start pulse, and this could fail under some conditions. Other important considerations are described in the following paragraphs.

It is important to use a monotonic power supply, as a non-monotonic ramp could cause a problem if the deviation is close to any threshold level. The high threshold variation can cause the same nonmonotonic sequence to work in one unit, but fail in others, as shown in Figure 4.

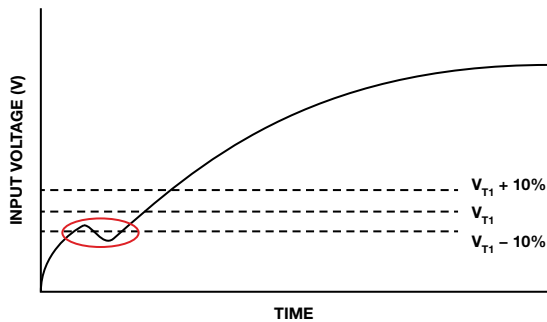


Figure 4. Nonmonotonic supply ramp.

Sometimes, even when the supply is disconnected (LDO disabled), storage capacitors retain some residual voltage, as shown in Figure 5. This voltage should be kept as small as possible to guarantee that the supply drops below V_{T1} or the POR will not reset correctly and the device will not initialize correctly.

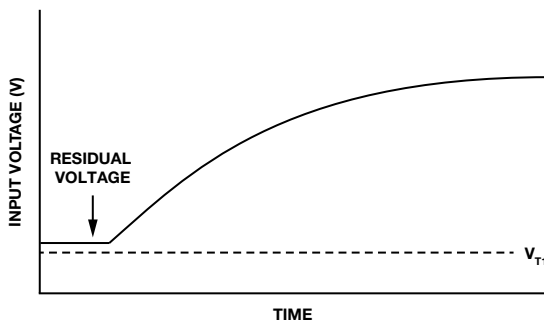


Figure 5. Residual voltage.

Some data sheets define the recommended supply sequence that should be applied to devices that have more than one supply pin. It is important that this sequence be followed. For example, consider a device with two independent supplies. The recommended supply sequence states that the digital supply must be powered before the analog supply (this is common, as the digital block controls the analog block so it must be powered first). The sequence states that the block

must be initialized first. It does not matter which supply starts to ramp first, but the digital supply must cross the threshold before the analog supply, as shown in Figure 6. If the delay between supplies is on the order of 100 μs , the impact should be minor and the device should initialize correctly.

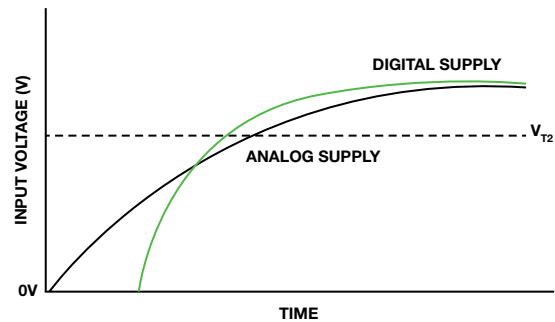


Figure 6. Recommended supply sequence.

Due to internal transistor parasitics, slow supply ramps on the order of 100 ms can cause problems. The POR circuit is evaluated at various slew rates to guarantee correct operation within normal power conditions. The data sheet will state whether a fast supply ramp (100 μs or less) is required.

A poor ground connection, for example, from a board connected to the supply with a thin cable, will have a high ground impedance, which could generate glitches during power up. In addition, in some electromagnetic environments (EME), the parasitic gate capacitance of the MOS transistor can charge, causing the transistor to malfunction until the capacitance is discharged. This could cause a failure in the POR initialization.

Drift and tolerance need to be considered as well. In some cases, discrete components such as capacitors have high tolerances—up to 40%—and high drift vs. temperature, voltage, and time. In addition, the threshold voltages have a negative temperature coefficient. For example, V_{T1} could vary from 0.8 V at room temperature to 0.9 V at -40°C and 0.7 V at $+105^\circ\text{C}$.

Conclusion

This article describes some common problems when powering a board that could cause system problems, and provides basic rules to guarantee correct board initialization. The supply is often overlooked, but both its final voltage accuracy and its transitional behavior are important.

References

Merino, Miguel Usach. "Insight into digiPOT Specifications and Architecture Enhances AC Performance." *Analog Dialogue*, Volume 45, Number 3.

Power Off or Power Down?

By Dushyant Juneja

"Power down, of course!" those alarmed by the question would exclaim. Others might wonder about the difference between the two proposals. Power-down modes often promise memory retention, shorter boot

up time, and ultralow leakage current, while turning off or gating the power does none of these things. But what if these features are not needed? Would the designer be wasting power by keeping the supplies stable and using the power-down mode? Can't we reduce the leakage current by simply shutting off the power? Are there any basic, underlying requirements for power-down modes? Intrigued? Read on.

Temptation and Risk

Modern systems bloom with a rich set of features, achieved through multiple levels of design complexity that often encompass more than one chip. Power is a concern for many applications such as portable medical devices, so these chips often include one or more power-down modes. These modes provide features such as memory retention, peripheral usage, and fast turn-on, all while drawing minimal supply current. An alternative is to do a complete power shutdown. This tactic completely cuts the power supplied to the chip, not allowing any current flow to its supply pins. This reduces the power dissipation, but not without serious side effects.

Consider the example of a complex system comprising multiple chips connecting through a multiplexed bus. If the system is intended for a power-constrained application, it might seem lucrative to simply shut off power to a chip that is not currently being used, especially if the other features offered by power-down modes are not required. Shutting off the supplies reduces the leakage current, but without supplies, the pins can act as low-impedance nodes to incoming signals, resulting in unpredictable operation and potential system-level threats. Tempting as the power-cut option may be, power-down modes offer a fundamental advantage for complex systems: they keep the individual chips in known, desirable states and maintaining safe, reliable operation even as the chip cycles between low-power and high-performance modes. The details can be shown by looking at an I/O node.

A Simple Example

The pin in Figure 7 connects to a multiplexed node, with its operation set by a verified system architecture. As an I/O pin, it has both input and output functions.

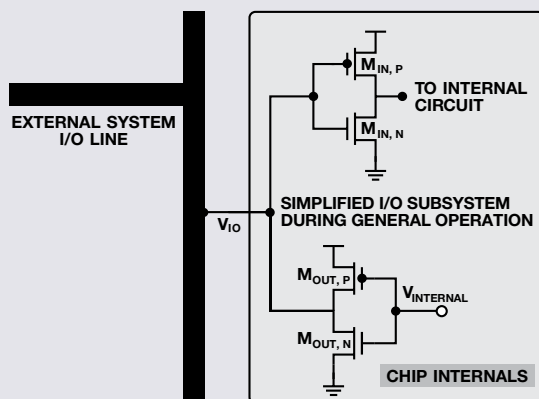


Figure 7. Simplified I/O circuit.

Disregarding issues with the device used for the power switch, turning off the supplies for this chip (assuming that none of the chip operations are required) would lead to the situation shown in Figure 8, with unknown states scattered throughout the chip core. In the worst case, the floating gate output devices ($M_{OUT,P}$ and $M_{OUT,N}$) could be exposed to unexpected external voltages while they are electrically asleep. With a CMOS I/O, as shown in this example, this could lead to a low-impedance connection to ground via the drain connection of the NMOS (highlighted

in red). A high current would ensue, possibly maxing out the drive capacity of the previous stage, causing damage to the MOS circuit in the chip, or both. If it did not damage the system, it could still decrease its performance.

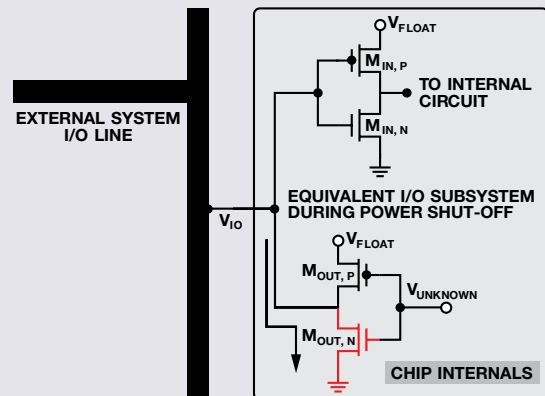


Figure 8. I/O circuit in power-cut mode. Note the unknown state of the internal gates.

Power-Down Mode

Power-down modes equip the chip with an extra layer of protection against these undesired operating conditions. The implementation differs for different modes, product families, and vendors, but the essential focus is providing a safe I/O boundary while the core of the chip sleeps, maintaining a known, dependable, low-power state. The advantage is that I/O operations between system components, with a system-wide multiplexed bus, for example, do not pose a threat to the sleeping device. One implementation could put the I/O pins in a high-impedance state during low-power mode, allowing the internal nodes connecting to the boundary pin to be in a well-defined state. A simplified implementation is shown in Figure 9. Signals will have no impact on the internal circuit, keeping them intrinsically safe. Other implementations, such as light-sleep modes might keep the I/O periphery powered up as well, while ensuring that the interaction between the chip's peripherals and core are verified during power-down mode. This enables the chip to handle active use situations, while keeping power consumption low. In addition, this system reduces the cost of the power switch, which would otherwise need to be a large, low-resistance device that would consume significant leakage and on-state power.

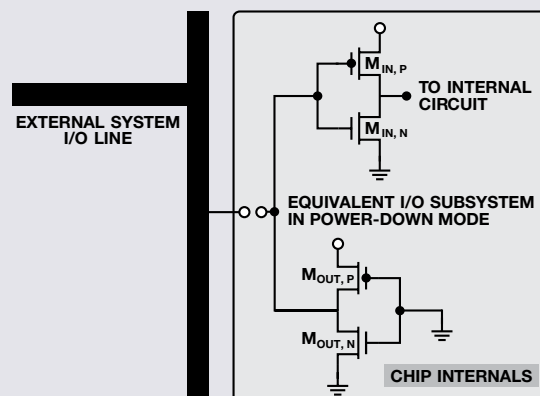


Figure 9. I/O circuit in power-down mode. Note that all internal nodes are well defined.

Power-down modes vary from chip to chip and vendor to vendor, so names such as “light-sleep mode” might not always mean the same thing. Some of these enable memory retention, while others might permit an increased number of interrupts or other similar features. One prominent advantage of these modes is reduced system response time, as compared to full-power shutdown. Some circuits provide separate I/O and core supplies. One advantage of this decoupling is that the board designer can shut off the core supplies to reduce leakage, while keeping the I/O powered. It is always advisable to get the exact details from the data sheet to ensure that the required features and protection methods are supported.

Effect of Shrinking Geometries

Modern IC process technologies offer higher density packaging as a natural consequence of reduced device sizes, making optimal use of power-down modes increasingly important. This also reduces the stress handling capacity of the device, however. For instance, a 28-nm device has a thinner gate oxide than its 180-nm technology counterpart. Thus, the stress applied by the gate voltage in power-cut mode would be more likely to rupture the smaller device. In addition, layout dependent parameters could also cause catastrophic failures in smaller geometry devices.

All of these effects make power-down modes increasingly desirable for modern devices. Packed with features, the modern chip comprises of millions of devices, each of which can contribute to the leakage current when kept on. Optimizing feature usage and powering down unused parts of the chip can save a major portion of this leakage. Make sure that the vendor supports these modes explicitly though, rather than trying to develop your own power-down capability.

A Few More Situations

The power-down puzzle has more pieces. What if we cut the ground connection as well, since that opens another low-impedance path? This is similar to an ESD situation where I/O pins are forced directly without enabling supplies, and if the signal strength is sufficient, it might trigger the ESD protection structure, causing high currents to flow through other connected I/O pins and creating a false power-up situation. A more probable case is a signal that is somewhat weaker, but still powerful enough to reach the supplies through a path, such as the I/O clamp. The signal may not be able to trigger the supply clamp, but could create unexpected ghost voltages on the supplies, which could cause unknown states of operation depending on the topology of the chip. In either case, if the situation persists, the chip might be damaged, unless the previous stage has already stopped supplying high current. If the signal strength is not sufficient to trigger the I/O clamp, it still might stress the first transistor it encounters, possibly damaging it after prolonged operation.

How about disconnecting the supplies and pulling the supply inputs low? Now the chip has no floating supplies and no chance of triggering any ESD structure, but the PMOS drain can reach a higher voltage than the body, forward biasing the drain-to-body diode. The current from the preceding stage would then flow through the PMOS device to ground until the device burns out, the previous stage gives up, or the designer notices the alarm.

Conclusion

Power-down modes result in a faster, safer system-wide response, making them an indispensable feature, especially when looking at the full signal chain in complex systems. Complete power cutoff could be considered if interactions between components are limited, or the system as a whole is simple enough to ensure no complications occur.



Miguel Usach Merino [miguel.usach@analog.com] received his degree in electronic engineering from Universitat de València. Miguel joined ADI in 2008 and works as an applications engineer in the Linear and Precision Technology Group in Valencia, Spain.



Miguel Usach Merino

Also by this Author:

[Insight into digiPOT Specifications and Architecture Enhances AC Performance](#)

[Volume 45, Number 3](#)

Dushyant Juneja [dushyant.juneja@analog.com] is a CAD engineer at Analog Devices. He works predominantly in AMS verification, behavioral modeling, and ESD protection for AMS designs. He received his master's degree in instrumentation engineering from the Indian Institute of Technology Kharagpur (2012), and his bachelor's degree in electrical engineering from the Institute of Technology (BHU) Varanasi (2010).



Dushyant Juneja

Design a PLL Filter when Only the Zero Resistor and Capacitor Are Adjustable

By Ken Gentile

Introduction

As described in the references, a standard procedure can be used to determine the values of R_0 , C_0 , and C_p for a second-order loop filter in a phase-locked loop (PLL). It uses open-loop bandwidth (ω_0) and phase margin (ϕ_M) as design parameters, and can be extended to third-order loop filters to determine R_2 and C_2 (Figure 1). The procedure solves for C_p directly and subsequently derives the remaining values.

In some cases, C_p , R_2 , and C_2 may be fixed-value components integrated within the PLL, leaving only R_0 and C_0 available for controlling the loop response. This nullifies the aforementioned procedure because C_p cannot be adjusted. This article proposes an alternative procedure that can be used when the value of C_p is fixed, and addresses limitations imposed by the inability to control the value of C_p .

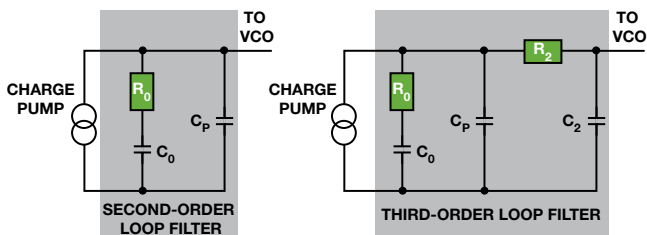


Figure 1. Typical second-order and third-order passive loop filters.

Assumptions

This loop filter design method relies on two assumptions that are typically used in third-order passive filter designs that extend a second-order loop filter design to third-order by compensating for the presence of R_2 and C_2 through adjustment of R_0 and C_0 .

1. The pole frequency resulting from R_2 and C_2 should be at least an order of magnitude greater than ω_0 (the desired open-loop unity-gain bandwidth); specifically $f_0 \leq 0.1/(2\pi R_2 C_2)$, where $f_0 = \omega_0/(2\pi)$.
2. The load of the series combination of R_2 and C_2 on the R_0 - C_0 - C_p network should be negligible.

Transfer Function of a Second-Order Loop Filter

A second-order loop filter has two time constants (T_1 and T_2) associated with its components:

$$T_2 = R_0 C_0 \quad (1)$$

$$T_1 = \left(\frac{C_p}{C_p + C_0} \right) T_2 \quad (2)$$

The loop filter's transfer function, in terms of T_1 , T_2 , and C_p , is important because it plays a significant role in the overall response of the PLL:

$$H_{LF}(s) = \left(\frac{1}{C_p} \right) \left(\frac{T_1}{T_2} \right) \left(\frac{sT_2 + 1}{s(sT_1 + 1)} \right) \quad (3)$$

PLL System Function

The small signal model shown in Figure 2 provides the means for formulating the PLL response and a template for analyzing phase variation at the output resulting from a phase disturbance at the input. Note that the voltage-controlled oscillator (VCO), being a frequency source, behaves like an ideal phase integrator, so its gain (K_V) has a $1/s$ factor (the Laplace transform equivalent of integration). Hence, the small signal model of a PLL has frequency dependence ($s = \sigma + j\omega$).

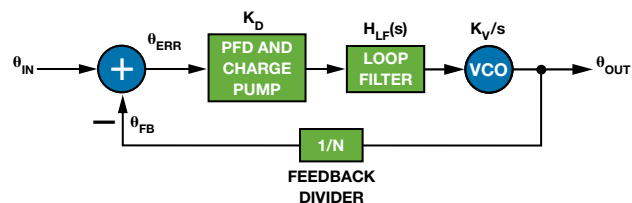


Figure 2. Small signal PLL model.

The closed-loop transfer function (H_{CL}) for a PLL is defined as θ_{OUT}/θ_{IN} . The open-loop transfer function (H_{OL}), defined as θ_{FB}/θ_{IN} , is related to the closed-loop transfer function. It is instructive to express H_{CL} in terms of H_{OL} because the open-loop transfer function contains clues about closed-loop stability:

$$H_{OL}(s) = -K \left(\frac{H_{LF}(s)}{sN} \right) \quad (4)$$

$$H_{CL}(s) = -N \left(\frac{H_{OL}(s)}{1 - H_{OL}(s)} \right) \quad (5)$$

K represents the combined gains of the phase-frequency detector (PFD), charge pump, and VCO—that is, $K = K_D K_V$, where K_D is the charge pump current in amperes and K_V is the VCO gain in Hz/V. H_{OL} , H_{CL} , and H_{LF} are all functions of s . The negative sign in Equation 4 shows the phase inversion implied by the negative feedback to the summing node in Figure 2. Defining H_{OL} as in Equation 4 leads to subtraction in the denominator of Figure 5, which provides an intuitive explanation of closed-loop stability.

Inspection of Equation 5 reveals a potential loop stability problem. Given that H_{OL} is a function of complex frequency ($s = \sigma + j\omega$), it necessarily has frequency dependent magnitude and phase components. Therefore, if H_{OL} simultaneously exhibits unity gain and zero phase shift (or any integer multiple of 2π radians) for any particular value of s , the denominator of H_{CL} becomes zero, the closed-loop gain becomes undefined, and the system becomes completely unstable. This implies that stability is governed by the frequency-dependent magnitude and phase characteristics of H_{OL} . In fact, at the frequency for which the magnitude of H_{OL} is unity, the phase of H_{OL} must stay far enough from zero (or any integer multiple of 2π) to avoid a zero denominator in Equation 5.

The frequency, ω_v , at which the magnitude of H_{OL} is unity, holds great importance. The phase of H_{OL} at ω_0 defines the phase margin of the system ϕ_M . Both ω_0 and ϕ_M can be derived from H_{OL} .

Defining R_0 and C_0 in Terms of ω_0 and ϕ_M

Using the design parameters ω_0 and ϕ_M to determine the values of R_0 and C_0 requires expressions containing those four variables and other constant terms. Start with Equation 4, because it defines H_{OL} . This includes H_{LF} , which includes R_0 and C_0 via T_1 and T_2 . Since H_{OL} has magnitude and phase, it stands to reason that ω_0 and ϕ_M can be incorporated as well.

Substituting Equation 3 into Equation 4 and rearranging terms yields Equation 6, which presents H_{OL} in terms of T_1 and T_2 along with constants K , N , and C_P :

$$H_{OL}(s) = -\left(\frac{K}{s^2 N C_P}\right) \left(\frac{T_1}{T_2}\right) \left(\frac{s T_2 + 1}{s T_1 + 1}\right) \quad (6)$$

Evaluation at $s = j\omega$, yields the frequency response of H_{OL} :

$$H_{OL}(j\omega) = -\left(\frac{K}{(j\omega)^2 N C_P}\right) \left(\frac{T_1}{T_2}\right) \left(\frac{j\omega T_2 + 1}{j\omega T_1 + 1}\right) \quad (7)$$

The $(j\omega)^2$ term in the denominator simplifies to $-\omega^2$:

$$H_{OL}(j\omega) = \left(\frac{K}{\omega^2 N C_P}\right) \left(\frac{T_1}{T_2}\right) \left(\frac{j\omega T_2 + 1}{j\omega T_1 + 1}\right) \quad (8)$$

The magnitude and phase of H_{OL} are:

$$|H_{OL}(j\omega)| = \left(\frac{K}{\omega^2 N C_P}\right) \left(\frac{T_1}{T_2}\right) \left(\frac{1}{1+(\omega T_1)^2}\right) \sqrt{(1 + \omega^2 T_1 T_2)^2 + \omega^2 (T_2 - T_1)^2} \quad (9)$$

$$\angle H_{OL}(j\omega) = \arctan(\omega T_2) - \arctan(\omega T_1) \quad (10)$$

Keep in mind that T_1 and T_2 are shorthand expressions for algebraic combinations of R_0 , C_0 , and C_P . Evaluating Equation 9 at $\omega = \omega_0$ and setting $|H_{OL}| = 1$ defines the unity-gain frequency, ω_v as the frequency at which the magnitude of H_{OL} is unity.

$$1 = \left(\frac{K}{\omega_0^2 N C_P}\right) \left(\frac{T_1}{T_2}\right) \left(\frac{1}{1+(\omega_0 T_1)^2}\right) \sqrt{(1 + \omega_0^2 T_1 T_2)^2 + \omega_0^2 (T_2 - T_1)^2} \quad (11)$$

Similarly, evaluating Equation 10 at $\omega = \omega_0$ and setting $\angle H_{OL} = \phi_M$ defines the phase margin, ϕ_M , as the phase of H_{OL} at frequency ω_0 (the unity gain frequency).

$$\Phi_M = \arctan(\omega_0 T_2) - \arctan(\omega_0 T_1) \quad (12)$$

It is a trivial matter to expand Equation 11 and Equation 12 by substituting Equation 1 for T_2 and Equation 2 for T_1 , which brings R_0 and C_0 into the equations. Hence, we have succeeded in relating ω_0 and ϕ_M to the variables R_0 and C_0 along with constants K , N , and C_P .

Simultaneously solving the resulting equations for R_0 and C_0 is no trivial task. The symbolic processor available in Mathcad® can solve the two simultaneous equations, but arccos must be substituted for arctan. This transformation enables the symbolic processor to solve for R_0 and C_0 , yielding the following solution sets (R_{0A} , C_{0A} ; R_{0B} , C_{0B} ; R_{0C} , C_{0C} ; and R_{0D} , C_{0D}). See the Appendix for details on transforming Equation 12 to use the arccos function.

$$\begin{aligned} R_{0A} &= \frac{\omega_0 K N \sqrt{1 - \cos^2(\Phi_M)}}{K^2 + 2K C_P N \omega_0^2 \cos(\Phi_M) + (C_P N \omega_0^2)^2} & C_{0A} &= -\left(\frac{K^2 + 2K C_P N \omega_0^2 \cos(\Phi_M) + (C_P N \omega_0^2)^2}{N \omega_0^2 (C_P N \omega_0^2 + K \cos(\Phi_M))}\right) \\ R_{0B} &= -\left(\frac{\omega_0 K N \sqrt{1 - \cos^2(\Phi_M)}}{K^2 + 2K C_P N \omega_0^2 \cos(\Phi_M) + (C_P N \omega_0^2)^2}\right) & C_{0B} &= -\left(\frac{K^2 + 2K C_P N \omega_0^2 \cos(\Phi_M) + (C_P N \omega_0^2)^2}{N \omega_0^2 (C_P N \omega_0^2 + K \cos(\Phi_M))}\right) \\ R_{0C} &= \frac{\omega_0 K N \sqrt{1 - \cos^2(\Phi_M)}}{K^2 - 2K C_P N \omega_0^2 \cos(\Phi_M) + (C_P N \omega_0^2)^2} & C_{0C} &= -\left(\frac{K^2 - 2K C_P N \omega_0^2 \cos(\Phi_M) + (C_P N \omega_0^2)^2}{N \omega_0^2 (C_P N \omega_0^2 - K \cos(\Phi_M))}\right) \\ R_{0D} &= -\left(\frac{\omega_0 K N \sqrt{1 - \cos^2(\Phi_M)}}{K^2 - 2K C_P N \omega_0^2 \cos(\Phi_M) + (C_P N \omega_0^2)^2}\right) & C_{0D} &= -\left(\frac{K^2 - 2K C_P N \omega_0^2 \cos(\Phi_M) + (C_P N \omega_0^2)^2}{N \omega_0^2 (C_P N \omega_0^2 - K \cos(\Phi_M))}\right) \end{aligned}$$

This result is problematic because the goal was to solve for R_0 and C_0 given ω_0 and Φ_M , but this indicates four possible R_0, C_0 pairs instead of a unique R_0, C_0 pair. However, closer inspection of the four results leads to a single solution set as follows.

Note that in the context of modeling a PLL, all of the variables in the above equations possess positive values, including $\cos(\Phi_M)$ because Φ_M is constrained to values between 0 and $\pi/2$. As a result, C_{0A} and R_{0B} are clearly negative quantities. Therefore, solution sets R_{0A}, C_{0A} and R_{0B}, C_{0B} are immediately ruled out because negative component values are not acceptable. The R_{0C}, C_{0C} and R_{0D}, C_{0D} results require further analysis, however.

Note that the four equations involving R_{0C}, C_{0C} , and R_{0D}, C_{0D} possess the common factor:

$$K^2 - 2KC_P N \omega_0^2 \cos(\Phi_M) + (C_P N \omega_0^2)^2 \quad (13)$$

Closer inspection reveals that Expression 13 has the form $a^2 - (2ac)\cos(\beta) + c^2$. Equating this with the arbitrary quantity, b^2 , yields:

$$b^2 = a^2 + c^2 - (2ac)\cos\beta \quad (14)$$

Equation 14, the Law of Cosines, relates a , b , and c as the lengths of the three sides of a triangle with β being the interior angle of the vertex opposite side b . Since b^2 is the square of the length of one side of a triangle, it must be a positive quantity, which implies the right side of Equation 14 must also be positive. Thus, Expression 13 must be a positive quantity, which means the denominator of R_{0D} is positive. The numerator of R_{0D} is also positive, therefore R_{0D} must be negative, which rules out the R_{0D}, C_{0D} solution set. This leaves only the R_{0C}, C_{0C} pair as a contender for the simultaneous solution of Equation 11 and Equation 12.

$$R_0 = \frac{\omega_0 K N \sqrt{1 - \cos^2(\Phi_M)}}{K^2 - 2KC_P N \omega_0^2 \cos(\Phi_M) + (C_P N \omega_0^2)^2} \quad (15)$$

$$C_0 = \frac{K^2 - 2KC_P N \omega_0^2 \cos(\Phi_M) + (C_P N \omega_0^2)^2}{N \omega_0^2 (K \cos(\Phi_M) - C_P N \omega_0^2)} \quad (16)$$

Constraints on R_0 and C_0

Although Equation 15 and Equation 16 are contenders for the simultaneous solution of Equation 11 and Equation 12, they are only valid if they result in positive values for both R_0 and C_0 . Close inspection of R_0 shows it to be positive—its numerator is positive, because the range of $\cos^2(x)$ is 0 to 1—and its denominator is the same as Expression 13, which was previously shown to be positive. The numerator of C_0 is also the same as Expression 13, so C_0 is positive as long as its denominator satisfies the following condition:

$$K \cos(\Phi_M) > C_P N \omega_0^2 \quad (17)$$

This is depicted graphically in Figure 3, in which the left and right sides of Equation 17 are each equated to y (blue and green curves) with the horizontal axis being ω_0 and Φ_M . The intersection of the two curves marks the boundary condition for ω_0 and Φ_M . The condition under which Equation 17 is true appears as the red arc. The portion of the horizontal axis beneath the red arc defines the range of Φ_M and ω_0 that ensures C_0 is positive. Note the point on the horizontal axis directly below the intersection of the blue and green curves establishes Φ_{M_MAX} , the maximum value of Φ_M to ensure C_0 is positive.

$$\Phi_{M_MAX} = \arccos\left(\frac{C_P N \omega_0^2}{K}\right) \text{ radians} \quad (18)$$

Equation 18 requires that $C_P N \omega_0^2$ be less than K in order to satisfy the constraints of the arccos function for Φ_{M_MAX} between 0 and $\pi/2$. This establishes ω_{0_MAX} , the upper limit on ω_0 to ensure C_0 is positive.

$$\omega_{0_MAX} = \sqrt{\frac{K}{C_P N}} \text{ radians/s} \quad (19)$$

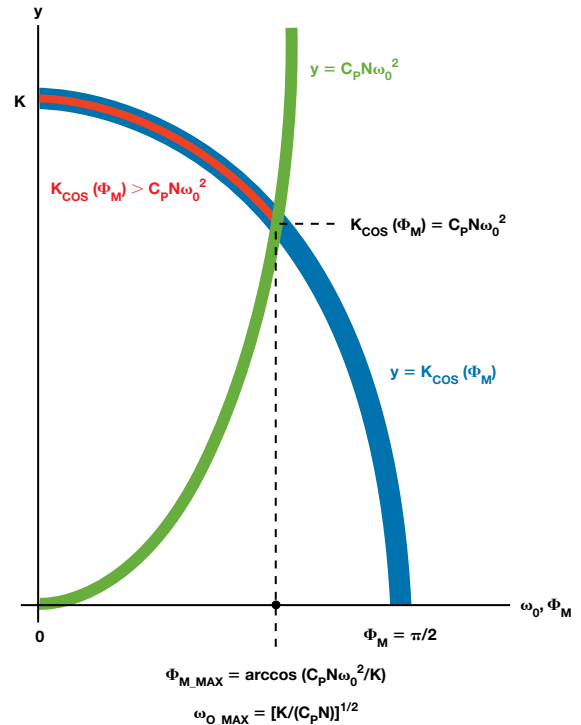


Figure 3. Constraint on C_0 denominator.

Compensating for R_2 and C_2 (Third-Order Loop Filter)

In the case of a third-order loop filter, components R_2 and C_2 introduce additional phase shift, $\Delta\Phi$, relative to the second-order loop filter:

$$\Delta\Phi = -\arctan(\omega_0 R_2 C_2) \quad (20)$$

To deal with this excess phase shift, subtract it from the desired value of Φ_M :

$$\Phi_{M_MAX} = \Phi_M - \Delta\Phi = \Phi_M + \arctan(\omega_0 R_2 C_2) \quad (21)$$

Applying ϕ_{M_NEW} to Equation 15 and Equation 16 results in different values for R_0 and C_0 than for the second-order solution, with the new values compensating for the excess phase shift introduced by R_2 and C_2 . The presence of R_2 and C_2 also affects ϕ_{M_MAX} , the maximum allowable value of ϕ_M . The new maximum value of ϕ_M ($\phi_{M_MAX_NEW}$) is

$$\Phi_{M_MAX_NEW} = \Phi_{M_MAX} + \Delta\Phi = \arccos(\omega_0^2 N C_P / K) - \arctan(\omega_0 R_2 C_2) \quad (22)$$

Conclusion

This article demonstrates using open-loop unity-gain bandwidth (ω_0) and phase margin (ϕ_M) as design parameters for second-order or third-order loop filters when only components R_0 and C_0 are adjustable. Simulation of a PLL with a second-order loop filter using R_0 and C_0 yields an exact match to the theoretical frequency response of H_{OL} and the resulting phase margin, thereby validating the equations. The parameters ω_0 and ϕ_M have upper bounds for a second-order loop filter per Equation 19 and Equation 18, respectively.

The procedure for determining R_0 and C_0 assumed a second-order loop filter, but is extendible to third-order loop filter designs by adjusting the desired phase margin (ϕ_M) to a new value (ϕ_{M_NEW}) per Equation 21, yielding a new upper bound ($\phi_{M_MAX_NEW}$) per Equation 22.

Although simulations using a second-order loop filter validated Equation 15 and Equation 16, validating the equations that extend the design procedure to third-order loop filter designs requires a redefinition of the loop filter response, $H_{LF}(s)$, to include R_2 and C_2 as follows:

$$H_{LF}(s) = \frac{sR_0C_0 + 1}{s(s^2R_0R_2C_0C_2C_P + sR_2C_0C_2 + sR_0C_0C_P + sR_2C_2C_P + sR_0C_0C_2 + C_0 + C_2 + C_P)}$$

The incorporation of this form of H_{LF} into the H_{OL} and H_{CL} equations enables simulations of third-order loop filter designs using R_0 and C_0 . Such simulations reveal the calculated values of R_0 and C_0 deviate from the theoretical frequency response and phase margin associated with H_{OL} for a PLL when using a third-order loop filter. This is predominantly due to the effect of R_2 and C_2 on H_{OL} in a third-order loop filter.

Recall that the formulas for R_0 and C_0 assume a second-order loop filter, but R_2 and C_2 do not exist in a second-order filter, so including them as part of the loop filter constitutes a source of error in spite of adjusting R_0 and C_0 to compensate for the phase shift introduced by R_2 and C_2 . Even in the presence of this error, however, simulation indicates that using the adjusted values of R_0 and C_0 , but limiting the choice of ω_0 to a maximum of $\frac{1}{4}$ of the value dictated by Equation 19 yields acceptable results. In fact, the simulated open-loop bandwidth and phase margin results deviate only slightly from the design parameters (ω_0 and ϕ_M) for a PLL using a third-order loop filter.

Simulation Results

The following is the result of running four simulations of a PLL with a third-order loop filter. The simulations all have the following fixed-loop filter components and PLL parameters:

$$C_P = 1.5 \text{ nF}$$

$$R_2 = 165 \text{ k}\Omega$$

$$C_2 = 337 \text{ pF}$$

$$K_D = 30 \text{ }\mu\text{A}$$

$$K_V = 3072 \text{ (25 ppm/V at 122.88 MHz)}$$

$$N = 100$$

Simulation 1 and Simulation 2 use $\omega_0 = 100$ Hz, which is near the calculated upper limit of 124.8 Hz (ω_{0_MAX}). As such, Simulation 1 and Simulation 2 deviate from the design parameter values (ω_0 and ϕ_M) by nearly 10%. On the other hand, Simulation 3 and Simulation 4 use $\omega_0 = 35$ Hz, which is approximately $\frac{1}{4}$ the upper limit. As expected, Simulation 3 and Simulation 4 hold much closer to the design parameters (ω_0 and ϕ_M), yielding an error of only about 1%.

Table 1 summarizes the simulation results and also includes the calculated values of R_0 , C_0 , ω_{0_MAX} , and ϕ_{M_MAX} for the given design parameters, ω_0 and ϕ_M . Note that for the purpose of comparison it would be preferable for both Simulation 1 and Simulation 3 to use $\phi_M = 80^\circ$, but Simulation 1 must satisfy the constraint imposed by Equation 22 of $\phi_M < 48^\circ$ (hence the choice of 42°).

Table 1: Simulation Summary

	Simulation 1		Simulation 2		Simulation 3		Simulation 4	
Parameter	ω_0	ϕ_M	ω_0	ϕ_M	ω_0	ϕ_M	ω_0	ϕ_M
Design	100 Hz	42°	100 Hz	30°	35 Hz	80°	35 Hz	30°
Simulation	93.1 Hz	38.7°	92.5 Hz	27.1°	34.9 Hz	79.0°	34.7 Hz	29.3°
R_0	969.6k k Ω		1118 k Ω		240.1 k Ω		139.9 k Ω	
C_0	14.85 nF		3.670 nF		225.5 nF		21.24 nF	
ω_{0_MAX}	124.8 Hz		124.8 Hz		124.8 Hz		124.8 Hz	
ϕ_{M_MAX}	48.0°		48.0°		84.8°		84.8°	

Figure 4 and Figure 5 show the open- and closed-loop response for each simulation.

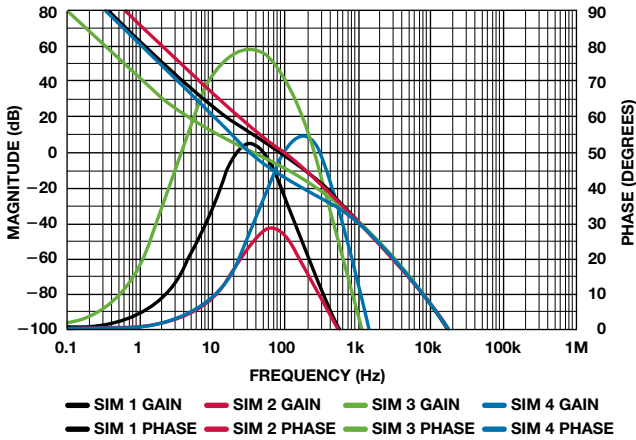


Figure 4. Open-loop gain and phase.

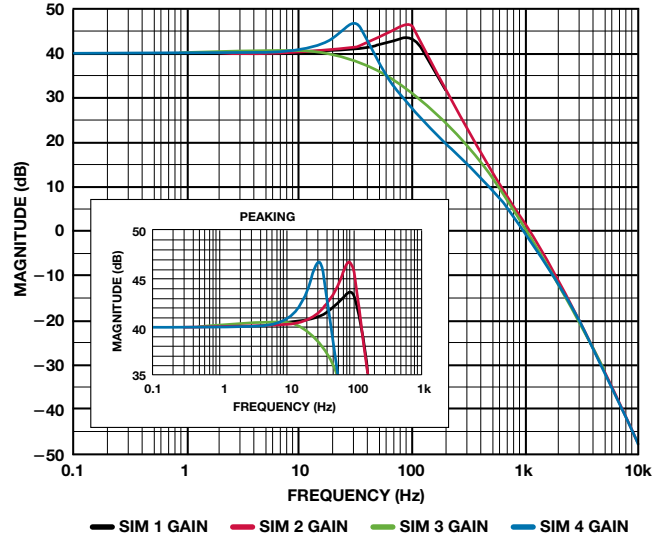


Figure 5. Closed-loop gain.

Appendix—Converting the Discontinuous Arctan Function to the Continuous Arccos Function

Equation 10 demonstrates that the angle ϕ is the difference between angle θ_2 and angle θ_1 , where $\theta_2 = \arctan(\omega T_2)$ and $\theta_1 = \arctan(\omega T_1)$. Furthermore, ωT_2 is expressible as $x/1$ and ωT_1 as $y/1$:

$$\Phi = \theta_2 - \theta_1 = \arctan\left(\frac{x}{1}\right) - \arctan\left(\frac{y}{1}\right)$$

This implies the geometric relationship shown in Figure 6, with θ_1 and θ_2 defined by the triangles of Figure 6 (b) and (a), respectively. Figure 6 (c) combines these two triangles to show ϕ as the difference between θ_1 and θ_2 .

The law of cosines relates an interior angle (θ) of a triangle to the lengths of the three sides of the triangle (a , b , and c) as follows:

$$c^2 = a^2 + b^2 - 2ab \cos(\theta)$$

(θ is the angle opposite side c)

Applying the law of cosines to angle ϕ in Figure 6 (c) yields:

$$(x - y)^2 = (\sqrt{1 + x^2})^2 + (\sqrt{1 + y^2})^2 - 2\sqrt{1 + x^2}\sqrt{1 + y^2} \cos \Phi$$

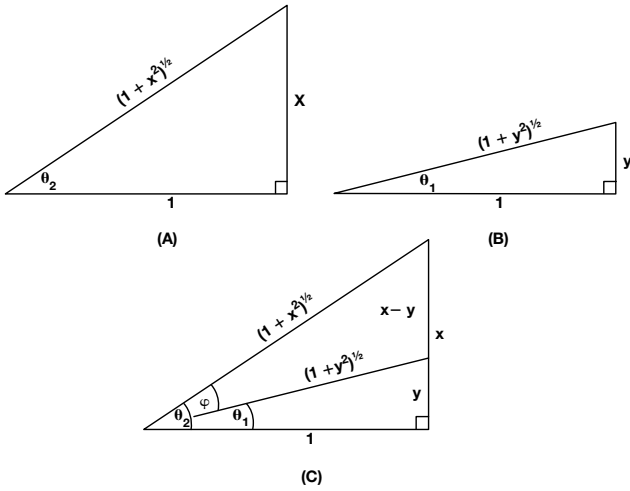


Figure 6. Geometric representation of Equation 10.

Solving for ϕ :

$$\Phi = \arccos\left(\frac{1 + xy}{\sqrt{(1 + x^2)(1 + y^2)}}\right)$$

But, $x/1 = \omega T_2$ and $y/1 = \omega T_1$, allowing ϕ to be expressed in terms of T_1 and T_2 .

$$\Phi = \arccos\left(\frac{1 + \omega^2 T_1 T_2}{\sqrt{[1 + (\omega T_2)^2][1 + (\omega T_1)^2]}}\right)$$

References

Brennan, Paul V. *Phase-Locked Loops: Principles and Practice*. McGraw-Hill, 1996.

Keese, William O. AN-1001, National Semiconductor Application Note, *An Analysis and Performance Evaluation of a Passive Filter Design Technique for Charge Pump Phase-Locked Loops*. May 1996.

MT-086: Fundamentals of Phase-Locked Loops (PLLs).

PLLs/PLLs with Integrated VCOs.



Ken Gentile [ken.gentile@analog.com] joined Analog Devices in 1998 as a system design engineer with the Clock and Signal Synthesis product line in Greensboro, NC, where he specializes in direct digital synthesis, analog filter design, and writing GUI-based engineering tools in MATLAB. Ken holds 10 patents. He has published 14 articles in various industry trade journals and over a dozen ADI application notes, as well as having presented at ADI's annual General Technical Conference (GTC) in 2001, 2005, and 2006. He graduated with honors in 1996 with a B.S.E.E. from North Carolina State University. In his spare time, Ken enjoys reading, mathematical puzzles, and most anything related to science, engineering, and "backyard" astronomy.

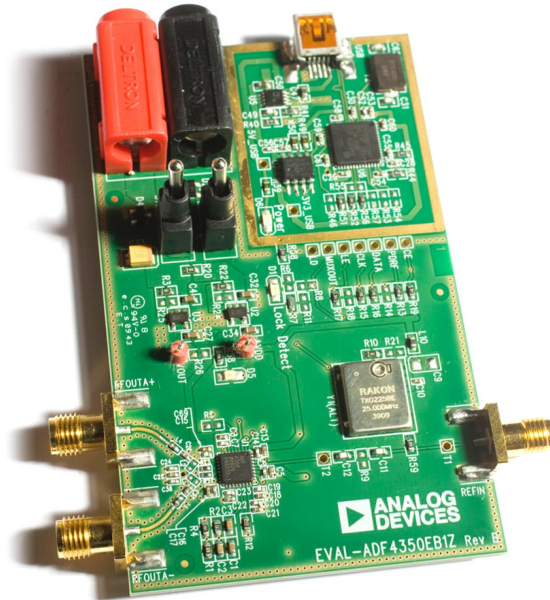


Ken Gentile

Also by this Author:

[Reconstruct a DAC Transfer Function from Its Harmonic Spectral Content](#)

Volume 43, Number 1



EVAL-ADF4350EB1Z evaluation board.

Quickly Implement JESD204B on a Xilinx FPGA

By Haijiao Fan

Introduction

JESD204 is a high-speed serial interface for connecting data converters (ADCs and DACs) to logic devices. Revision B of the standard supports serial data rates up to 12.5 Gbps and ensures repeatable, deterministic latency on the JESD204 link. As the speed and resolution of converters continues to increase, the JESD204B interface has become ever more common in ADI's high-speed converters and integrated RF transceivers. In addition, flexible serializer/deserializer (SERDES) designs in FPGAs and ASICs have naturally started to replace the traditional parallel LVDS/CMOS interface to converters, and are used to implement the JESD204B physical layer. This article describes how to quickly set up a project using a Xilinx FPGA to implement the JESD204B interface, and provides some application and debug suggestions for FPGA designers.

JESD204B Protocol Implementation Overview

The JESD204B specification defines four key layers that implement the protocol data stream, as shown in Figure 1. The transport layer maps the conversion between samples and framed, unscrambled octets. The optional scrambling layer scrambles/descrambles the octets, spreading the spectral peaks to reduce EMI. The data-link layer handles link

synchronization, setup, and maintenance, and encodes/decodes the optionally scrambled octets to/from 10-bit characters. The physical layer is responsible for transmission and reception of characters at the bit rate.

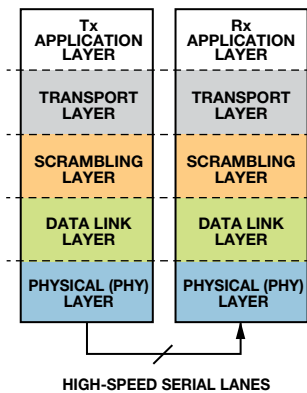


Figure 1. Key layers of JESD204B standard.

Different JESD204B IP vendors may implement the layers in different ways. Figure 2 and Figure 3 illustrate how the JESD204B transmit and receive protocols are implemented by ADI.

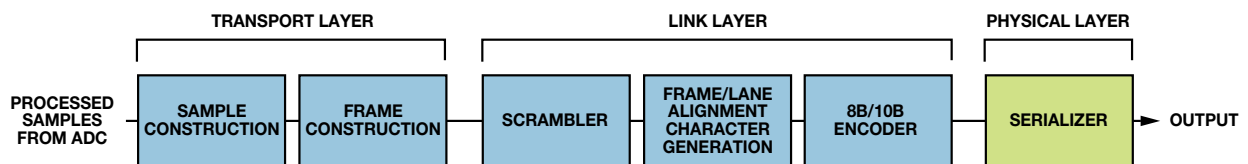


Figure 2. JESD204B transmitter implementation.

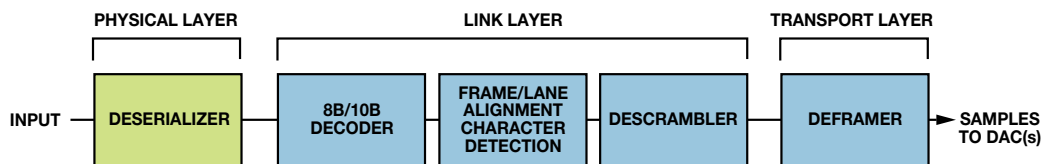


Figure 3. JESD204B receiver implementation.

The transport layer implementation depends strongly on the specific converter's configuration and how it maps between samples and frames, so most FPGA vendors exclude it from their JESD204 IP. In addition, highly configurable, tightly integrated SERDES transceivers are integrated in FPGAs. These can be used to support all kinds of serial protocols, including PCIe, SATA, SRIO, CPRI, and JESD204B. Thus, a logic core that implements the link layer, combined with a configurable SERDES that realizes the physical layer, forms the basis for a JESD204B link. Figure 4 and Figure 5 show block diagrams of a JESD204B transmitter and receiver on a Xilinx FPGA. The transmitter/receiver lanes implement the scramble and link layers; the 8B/10B encoder/decoder and the physical layer are implemented in the GTX/GTX/GTH gigabit transceivers.

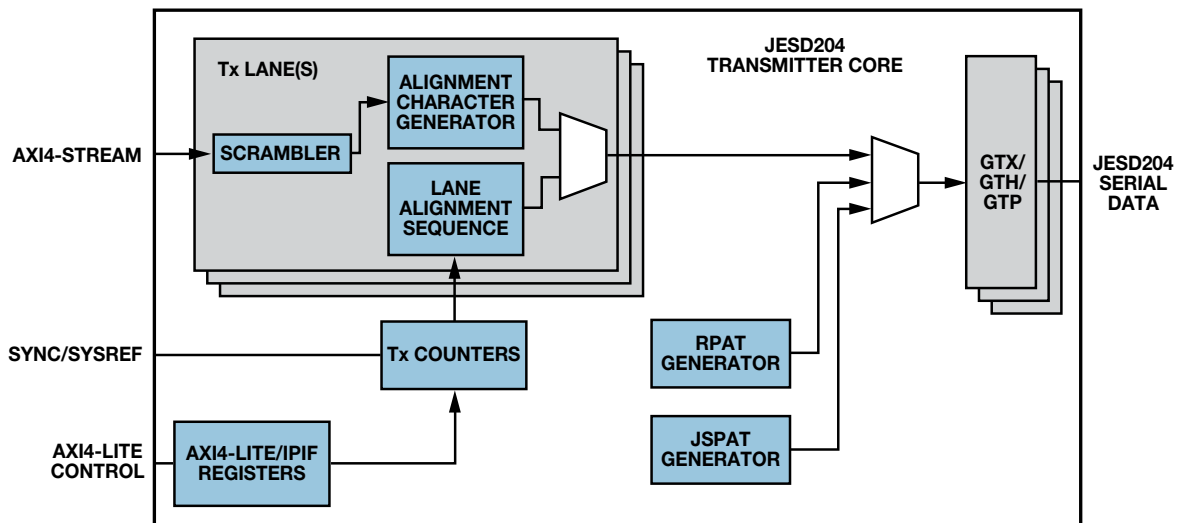


Figure 4. JESD204B transmitter implementation using a Xilinx FPGA.

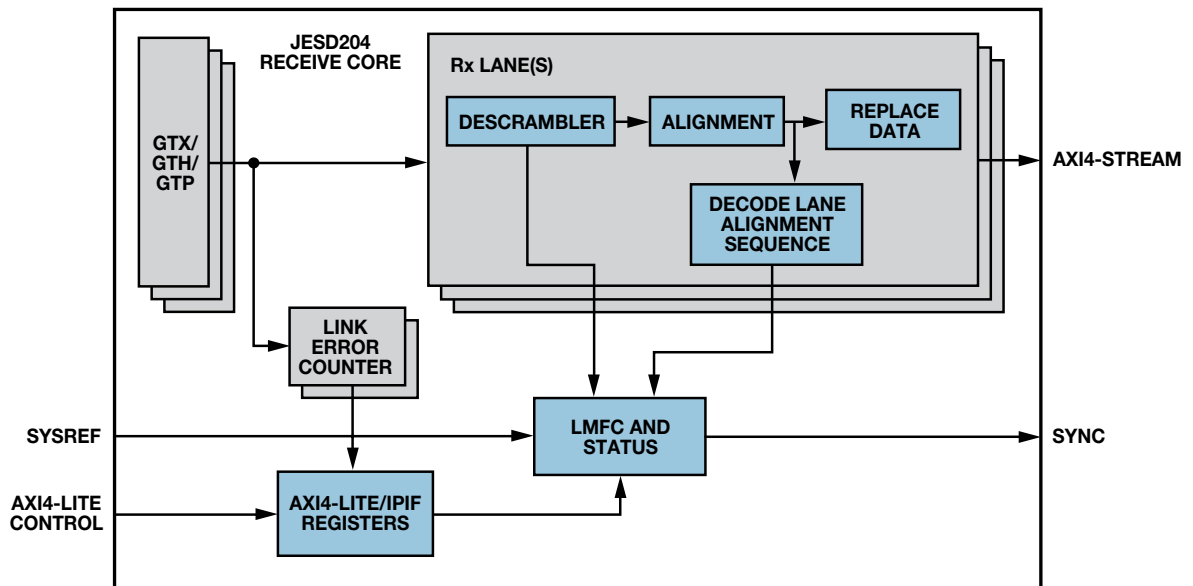


Figure 5. JESD204B receiver implementation using a Xilinx FPGA.

JESD204B Design Example Using a Xilinx FPGA

The latest [Xilinx JESD204 IP](#) core is delivered and encrypted as a black box via the [Vivado® Design Suite](#). Xilinx also provides a Verilog example design using the [Advanced eXtensible Interface](#) (AXI), but this example project is overdesigned for most applications. Users typically have their own configuration interfaces and do not need to integrate an extra AXI for JESD204B logic. Figure 6 shows a simplified JESD204 design, which is intended to help FPGA users understand the structure of JESD204 and quickly start their own FPGA-based JESD204 project.

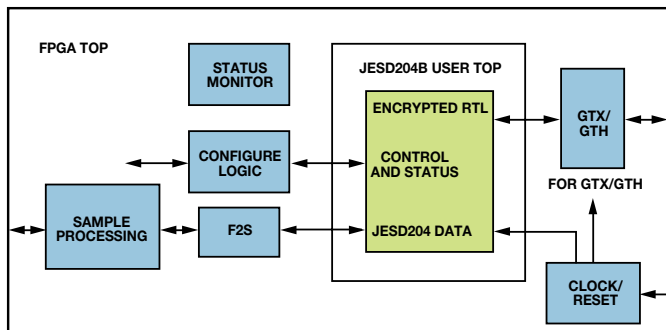


Figure 6. JESD204B design example.

The encrypted [register-transfer language](#) (RTL) block—the JESD204 logic IP core generated by Vivado—is equivalent to the transmitter and receiver modules shown in Figure 4 and Figure 5. The encrypted interface definition can be found in the Xilinx example design files. Then the encrypted RTL is wrapped into the JESD204B user top. Control, configuration, status, and JESD data interfaces from the encrypted RTL go through the wrapper to connect with user logic and the GTX/GTH transceiver data. Symbol alignment configuration for GTX/GTH is optimized and updated to make the transceivers work more robustly.

Dedicated pads should be used for the GTX/GTH reference clock to the SERDES transceivers. Careful attention must be paid to the global clock design for the FPGA logic, including clocks for the internal PLL, parallel interface, JESD204 logic core, and user-specific processing logic. In addition, the master system reference (SYSREF) input for the JESD204B logic core (Subclass 1) must be captured accurately to guarantee the deterministic delay of the JESD204 link.

The reset sequence for the GTX/GTH transceivers and the JESD204 core is crucial for reliable JESD link initialization, so the JESD204 core should be in reset state until the internal PLL in the GTX/GTH transceiver is locked and the GTX/GTH has been reset.

A frame-to-samples (F2S) module is needed to implement the transport layer of JESD204, which maps samples to or from frames according to the specific JESD204B configuration. The samples are then processed by application-specific logic. An auxiliary module monitors the JESD204 logic and physical layer (PHY) status for system debug.

Symbol Alignment in the Xilinx SERDES Transceiver

In the SERDES receiver, serial data must be aligned to symbol boundaries before it can be used as parallel data. To align the data, the transmitter sends a recognizable sequence, usually called a comma. The receiver searches for a comma in the incoming serial data stream and moves it to a symbol boundary once found. This enables the received parallel words to match the transmitted parallel words. The comma is usually a K, which is a special character in 8B/10B table used for [control symbols](#). For JESD204B applications, the transmitter will send a stream of $K = K28.5$ symbols for code group synchronization (CGS). The FPGA can therefore use $K28.5$ as a comma to align symbol boundaries and users can specify whether a comma match consists of either a comma plus ([running disparity](#) is plus) or a comma minus (running disparity is minus), or both. The JESD204B default setting for GTX/GTH comma detection allows either a comma plus or a comma minus to align the comma.

In some applications, the default comma setting may result in symbol realignment, or alignment to the wrong symbol boundary. This can cause messy 8B/10B decoding errors and broken JESD204B links. Combined comma plus and minus is more robust, forcing the comma align block to search for two commas in a row, detecting a comma only when the received data has a comma plus or minus followed by a comma minus or plus with no extra bits in between. This helps to maintain symbol boundaries and link stability when the line rate is high or the system has excessive noise.

Design Consideration for JESD204 Projects on FPGA

A synchronous, active low sync signal from the JESD204 receiver to the transmitter indicates the state of synchronization. Link reinitialization during normal operation will cause messy samples data, so the link status must be monitored in real time. In particular, a continuous low on sync means the receiver cannot identify at least four consecutive $K28.5$ symbols in the received data stream. If this occurs, check the transmitter/receiver SERDES configuration or make sure that the transmitter is sending $K28.5$. A continuous high on sync means the link has established and maintained stability. When sync goes from high to low and back to high, the duration of the low state should be counted. If it is longer than five frames plus nine octets, the receiver has detected a large error and sends a request to reinitialize the JESD204 link. If the duration is equal to two frame clocks, the receiver has detected a small error, but does not trigger a link reinitialization. This function can significantly ease system debug and further link monitoring, so users should include it in their designs.

8B/10B decoding errors can lead to JESD204B link reinitialization, but they are not the only cause, so user designs should have the ability to count decoding errors of each lane to determine the cause of link resynchronization. Also, the SERDES link quality can be determined in real time by the 8B/10B decoding error status.

[Pseudorandom bit sequences \(PRBS\)](#) provide a useful resource for measuring signal quality and jitter tolerance in high-speed links. The SERDES transceiver in most FPGAs has a built-in PRBS generator and checker, so no extra FPGA resources are needed. Thus, do not forget to instantiate this function, which should be used when the bit-error rate (BER) or eye diagram is evaluated.

A buffer is always used in SERDES transceivers to change the internal clock domain. A bad clock design for the transmitter and receiver or the wrong clock and data recovery (CDR) setting can cause buffer overflow or underflow. Some link errors may occur in this case, so monitoring the buffer status is meaningful. An interrupt record for buffer overflow and underflow is useful for system debug, so other internal buffers that are not allowed to underflow or overflow in user logic should also be monitored.

Conclusion

This article showed how to quickly implement a JESD204 block on a Xilinx FPGA, but the method can be applied to other FPGAs as well. First, understand the function and interfaces of the JESD204 logic core and transceiver provided by the FPGA vendor, then instantiate them and wrap them into your logic. Second, globally design the FPGA clock tree and reset sequence for your entire project. Third, carefully define the interfaces between the JESD204 logic core, user logic, and transceivers. Finally, add necessary debug resources. Following these steps will help you to achieve a quick and successful design for your JESD204 interface.

References

[JESD204B Survival Guide](#)

[JESD204 Serial Interface JEDEC Standard for Data Converters](#)

[High-Speed Analog-to-Digital Converters](#)

[High-Speed Digital-to-Analog Converters](#)

[Integrated Transceivers, Transmitters, and Receivers](#)

[Alliances and FPGA Reference Designs](#)

[Demystifying the JESD204B High-Speed Data Converter-to-FPGA Interface](#)

Beavers, Ian. [“Demystifying Deterministic Latency Within JESD204B Converters.”](#) *Electronic Design*, 2/25/2014.

Beavers, Ian. [“Prototyping Systems: JESD204B Converters and FPGAs.”](#) *Electronic Design*, 1/23/2014.

Beavers, Ian and Jeffrey Ugalde. [“Designing JESD204B Converter Systems for Low BER, Part 1.”](#) *EDN*, 10/22/2014.

Beavers, Ian and Jeffrey Ugalde. [“Designing JESD204B Converter Systems for Low BER, Part 2.”](#) *EDN*, 10/28/2014.

Harris, Jonathan. [“Understanding Layers in the JESD204B Specification: A High-Speed ADC Perspective, Part 1.”](#) *EDN*, 9/24/2014.

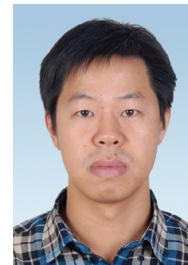
Harris, Jonathan. [“Understanding Layers in the JESD204B Specification: A High-Speed ADC Perspective, Part 2.”](#) *EDN*, 10/2/2014.

Jones, Del. [“JESD204B Subclasses \(Part 1\): Intro and Deterministic Latency.”](#) *EDN*, 6/18/2014.

Jones, Del. [“JESD204B Subclasses \(Part 2\): Subclass 1 vs. 2, System Considerations.”](#) *EDN*, 6/25/2014.



Hajjiao Fan



Hajjiao Fan [hajjiao.fan@analog.com] is an applications engineer at Analog Devices in Beijing, China, focusing on JESD204 protocol evaluation and integrated RF transceivers application and support. He received his B.S.E.E. in 2003 and M.S.E.E. in 2006 from Northwestern Polytechnical University, China. Prior to joining ADI in July 2012, Hajjiao worked as an FPGA and system engineer for over six years.

Power Management for Integrated RF ICs

By Qui Luu

As more building blocks are added to a radio-frequency integrated circuit (RFIC), more sources of noise coupling arise, making power management increasingly important. This article describes how power-supply noise can affect the performance of RFICs. The [ADRF6820](#) quadrature demodulator with integrated phase-locked loop (PLL) and voltage-controlled oscillator (VCO) is used as an example, but the results are broadly applicable to other high-performance RFICs.

The power-supply noise can degrade linearity by creating mixing products in the demodulator and degrade phase noise in the PLL/VCO. A detailed power evaluation is accompanied by recommended power designs using low-dropout regulators (LDOs) and switching regulators.

With its dual supply and high level of RF integration, the ADRF6820 provides an ideal vehicle for discussion. It uses a similar active mixer core as the [ADL5380](#) quadrature demodulator and the identical PLL/VCO cores as the [ADRF6720](#), so the information presented can be applied to those components. In addition, the power-supply design can be applied to new designs requiring 3.3-V or 5.0-V supplies with similar power consumption.

The ADRF6820 quadrature demodulator and synthesizer, shown in Figure 1, is ideally suited for next-generation communication systems. The feature-rich device comprises a high-linearity broadband I/Q demodulator, an integrated fractional-N PLL, and a low-phase-noise multicore VCO. It also integrates a 2:1 RF switch, a tunable RF balun, a programmable RF attenuator, and two LDOs. The highly integrated RFIC is available in a 6-mm × 6-mm LFCSP package.

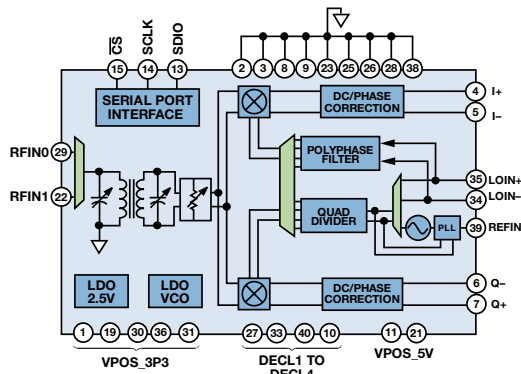


Figure 1. ADRF6820 simplified block diagram.

Power-Supply Sensitivities

The blocks most affected by power-supply noise are the mixer core and the synthesizer. Noise coupled into the mixer core creates unwanted products that degrade linearity and dynamic range. This is especially critical for a quadrature demodulator because the low-frequency mixing products fall within the band of interest. Similarly, power-supply noise can degrade the phase noise of the PLL/VCO. The effect of unwanted mixing products and degraded phase noise are common to most mixers and synthesizers, but the exact

level of degradation is determined by the architecture and layout of the chip. Understanding these power-supply sensitivities allows a more robust power design that optimizes performance and efficiency.

Quadrature Demodulator Sensitivities

The ADRF6820 uses a double-balanced [Gilbert cell](#) active mixer core, as shown in Figure 2. Double-balanced means that both the LO and RF ports are driven differentially.

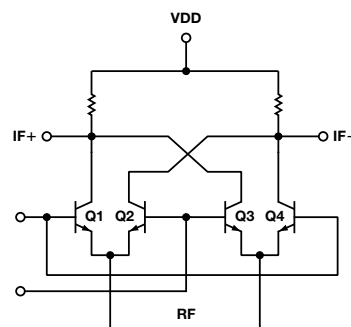


Figure 2. Gilbert cell double-balanced active mixer.

After a filter rejects the high-order harmonics, the resulting mixer outputs are the sum and difference of the RF and LO inputs. The difference term, also called the IF frequency, lies within the band of interest, and is the desired signal. The sum term falls out of band and gets filtered.

$$V(t) = \frac{2V_{RF}}{\pi} [\cos(\omega_{RF}t - \omega_{LO}t) + \cos(\omega_{RF}t + \omega_{LO}t)]$$

Ideally, only the desired RF and LO signals are presented to the mixer core, but this is rarely the case. Power-supply noise can couple into the mixer inputs and manifest itself as mixing spurs. Depending on the source of the noise coupling, the relative amplitudes of the mixing spurs may vary. Figure 3 shows a sample mixer output spectrum and where the mixing products may reside due to power-supply-noise coupling. In the figure, CW corresponds to a continuous wave or sinusoidal signal that couples onto the power-supply rail. The noise may be the clock noise from a 600-kHz or 1.2-MHz switching regulator, for example. The power-supply noise can cause two different problems—if the noise couples to the mixer outputs, the CW tone will appear at the output with no frequency translation; if the coupling occurs at the mixer inputs, the CW tone will modulate the RF and LO signals, producing products at $IF \pm CW$.

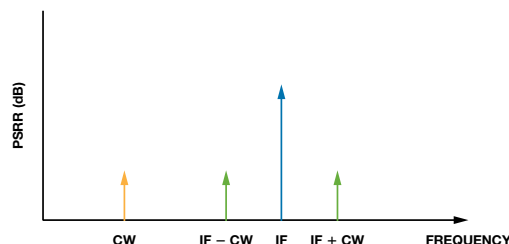


Figure 3. Sample mixer output spectrum with power-supply noise coupling.

These mixing products can be close to the desired IF signal, so filtering them out becomes difficult, and dynamic range loss is inevitable. This is especially true for quadrature demodulators because their baseband is complex and centered around dc. The demodulation bandwidth of the ADRF6820 spans from dc to 600 MHz. If a switching regulator with noise at 1.2 MHz powers the mixer core, undesired mixing products will occur at $IF \pm 1.2$ MHz.

Frequency Synthesizer Sensitivities

The references provided at the end of the article offer valuable information on how power-supply noise affects integrated PLLs and VCOs. The principles apply to other designs with the same architecture, but nonidentical designs will need their own power evaluation. For example, the integrated LDO on the ADRF6820's VCO power supply offers more noise immunity than a PLL power supply that does not use an integrated LDO.

ADRF6820 Power-Supply Domains and Current Consumption

To design the power-management solution, first examine the RFIC's power domains to determine which RF blocks are powered by which domain, the power consumption of each domain, the operational modes that affect the power consumption, and the power-supply rejection of each domain. Using this information, sensitivity data for the RFIC can be collected.

The major functional blocks of the ADRF6820 each have their own power pins. Two domains are powered from the 5-V supply. VPMX powers the mixer core, and VPRF powers the RF front-end and input switches. The remaining domains are powered from the 3.3-V supply. VPOS_DIG powers an integrated LDO, which outputs 2.5 V to power the SPI interface, the PLL's Σ - Δ modulator, and the synthesizer's FRAC/INT dividers. VPOS_PLL powers the PLL circuitry, including the reference input frequency (REFIN), phase-frequency detector (PFD), and the charge pump (CP). VPOS_LO1 and VPOS_LO2 power the LO path, including the baseband amplifier and dc bias reference. VPOS_VCO powers another integrated LDO, which outputs 2.8 V to power the multicore VCO. This LDO is important for minimizing the sensitivity to power-supply noise.

The ADRF6820 is configurable in several operational modes. It consumes less than 1.5 mW in normal operational mode with a 2850-MHz LO. Decreasing the bias current reduces both power consumption and performance. Increasing the mixer bias current makes the mixer core more linear and improves IIP3, but degrades the noise figure and increases power consumption. If noise figure is of key importance, the mixer bias current can be reduced, decreasing the noise within the mixer core and reducing power consumption. Similarly, the baseband amplifiers at the output have variable current drive capabilities for low impedance output loads. Low output impedance loads require higher current drive and consume more power. The data sheet provides tables showing power consumption for each of the operational modes.

Measurement Procedure and Results

Noise coupling on the power rail produces undesired tones at CW and $IF \pm CW$. To mimic this noise coupling, apply a CW tone to each power pin and measure the amplitude of the resulting mixing product relative to the input CW tone. Record this measurement as the power-supply rejection in dB. The power-supply rejection varies with frequency, so sweep the CW frequency from 30 kHz to 1 GHz to capture the behavior. The power-supply rejection over the band of interest determines whether filtering is required. The PSRR is calculated as:

$$\text{CW PSRR in dB} = \text{input CW amplitude (dBm)} - \text{measured CW feedthrough at I/Q output (dBm)}$$

$$(\text{IF} \pm \text{CW}) \text{ PSRR in dB} = \text{input CW amplitude (dBm)} - \text{measured IF} \pm \text{CW feedthrough at I/Q output (dBm)}$$

$$(\text{IF} + \text{CW}) \text{ in dBm} = (\text{IF} - \text{CW}) \text{ dBm, as CW tones modulated around the carrier have equal amplitudes.}$$

Lab Setup

Figure 4 shows the lab setup. Apply a 3.3-V or 5-V dc source to the network analyzer to produce a swept continuous sinusoidal signal with a 3.3-V or 5-V offset. Apply this signal to each of the power rails on the RFIC. Two signal generators provide the RF and LO input signals. Measure the output on a spectrum analyzer.

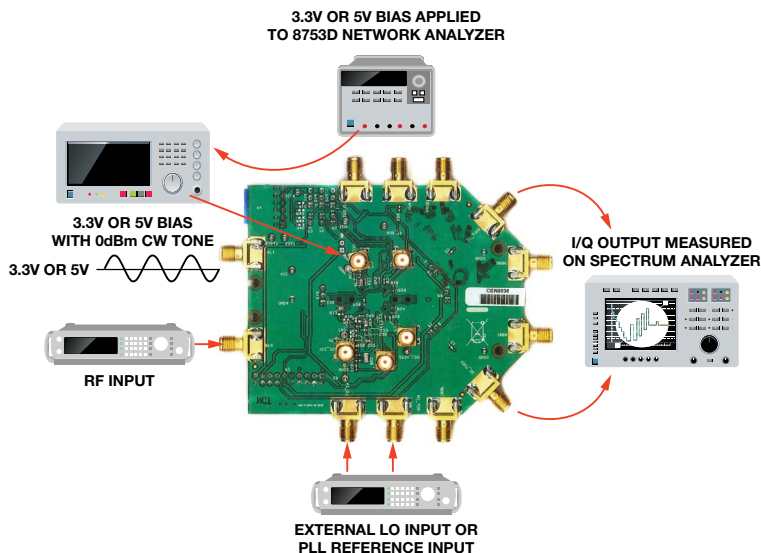


Figure 4. ADRF6820 PSRR measurement setup.

Measurement Procedure

The amplitude of the undesired mixing products depends on the chip's power-supply rejection, and the size and location of the decoupling capacitors on the evaluation board. Figure 5 shows the amplitude of the (IF + CW) tone at the output given a 0-dB sinusoidal signal on the power pin. With no decoupling capacitors, the amplitude of the undesired tone was between -70 dBc and -80 dBc. The data sheet recommends a 100-pF capacitor adjacent to the device on top of the board and a 0.1- μ F capacitor on the back. The resonance of these external decoupling capacitors can be seen in the graph. The transition at 16 MHz is due to the resonance of the 0.1 μ F capacitor with a 1-nH parasitic inductance. The transition at 356 MHz is due to the resonance of the 100-pF capacitor with 2 nH of parasitic inductance from both capacitors. The transition at 500 MHz is due to the resonance of the 100-pF capacitor with a 1-nH parasitic inductance.

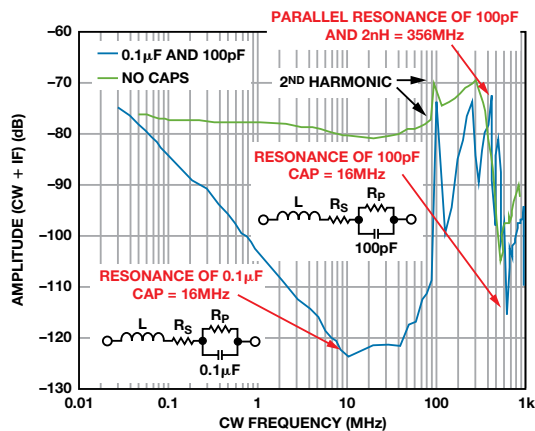


Figure 5. Effects of decoupling capacitor resonance on $IF \pm CW$.

Results

The amplitudes of the interfering signal (CW) on the power-supply rail and the modulated signals ($IF \pm CW$) were measured at the baseband outputs. Noise was introduced to the power rail under test, while the other power supplies remained clean. Figure 6 shows the amplitude of the ($IF \pm CW$) tone when a 0-dB sinusoidal signal was injected on the power pin and swept from 30 kHz to 1 GHz. Figure 7 shows the feedthrough from the CW tone to the baseband outputs.

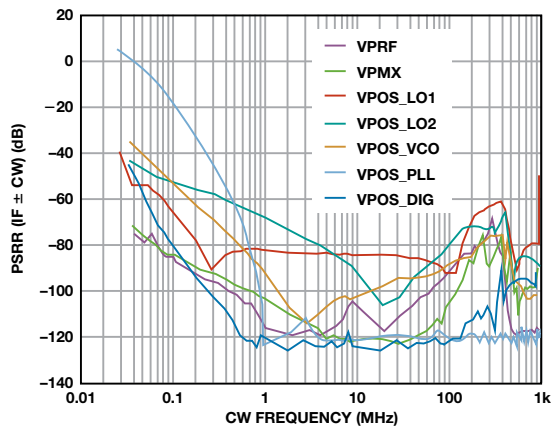


Figure 6. PSRR of the ($IF \pm CW$) tone.

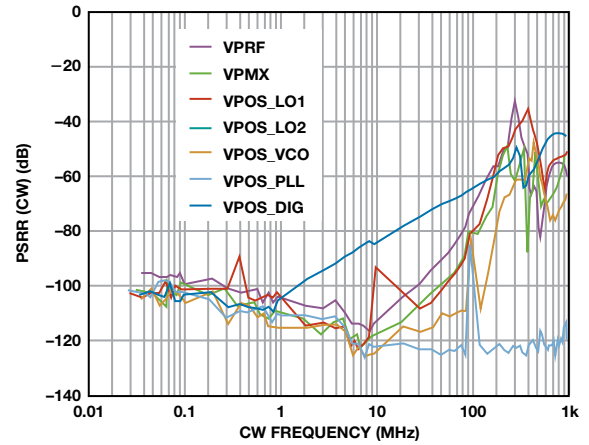


Figure 7. PSRR of the CW tone.

Analysis

The plots provide invaluable data on the supply sensitivities at each power pin. VPOS_PLL has the worst power-supply rejection and is therefore the most sensitive power node. This power pin powers the PLL circuitry, including the reference input frequency, phase-frequency detector, and the charge pump. These sensitive function blocks determine the accuracy and phase performance of the LO signal, so any noise coupled on them propagates directly to the output.

Under the same reasoning, it can be argued that the VCO power-supply is also a critical node. The plots show that VPOS_VCO has much better rejection than VPOS_PLL. This is a result of the internal LDO that actually powers the VCO. The LDO isolates the VCO from noise on the external pin and also provides it with a fixed-noise spectral density. The PLL power supply has no LDO, making it the most sensitive power rail. Thus, isolating it from potential noise coupling is critical for optimal performance.

The PLL loop filter attenuates high CW frequencies, so the sensitivity on VPOS_PLL is poor at low frequencies and slowly improves as the frequency sweeps from 30 kHz to 1 GHz. At higher frequencies the amplitude of the interfering tone gets attenuated and the power level injected into the PLL is substantially lower. Thus, VPOS_PLL shows better high-frequency power-supply rejection than the other power domains. The loop filter components were configured for 20 kHz, as shown in Figure 8.

The power rails, listed from most sensitive to least sensitive, are: VPOS_PLL, VPOS_LO2, VPOS_VCO, VPOS_LO1, VPOS_DIG, VPMX, and VPRF.

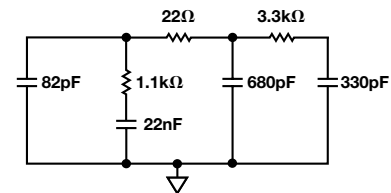


Figure 8. PLL loop filter configured for a 20-kHz loop bandwidth.

Power-Supply Design

With a good understanding of the maximum power consumption of the ADRF6820 in its various modes and the sensitivity of each power domain, power-management solutions were designed using both switching regulators and LDOs to determine the feasibility of both power solutions. First, a 6-V source was regulated to 5 V and 3.3 V for the ADRF6820 power rails. Figure 9 shows the power design for the 5-V power-supply for VPMX and VPRF. The [ADP7104](#) CMOS LDO can deliver up to 500 mA load current. The [ADP2370](#) low quiescent current step-down (buck) switching regulator can operate at 1.2 MHz or 600 kHz. Additional filtering was added to the switching regulator output to

attenuate the switching noise. The ADP2370 can deliver up to 800 mA load current. The ADRF6820's 5-V rail can be sourced by either the ADP7104 or the ADP2370. Additional decoupling and filtering is applied to each power pin.

Figure 10 shows the 3.3-V power design. The source voltage is still 6.0 V, but an additional LDO steps the source down to an intermediate voltage before it further gets regulated down to 3.3 V. The extra stage is required to reduce power loss, as a 6-V source regulated directly down to 3.3-V would operate at 55% maximum efficiency. An intermediate stage is not necessary for the switching regulator path because its pulse-width modulation (PWM) architecture minimizes power loss.

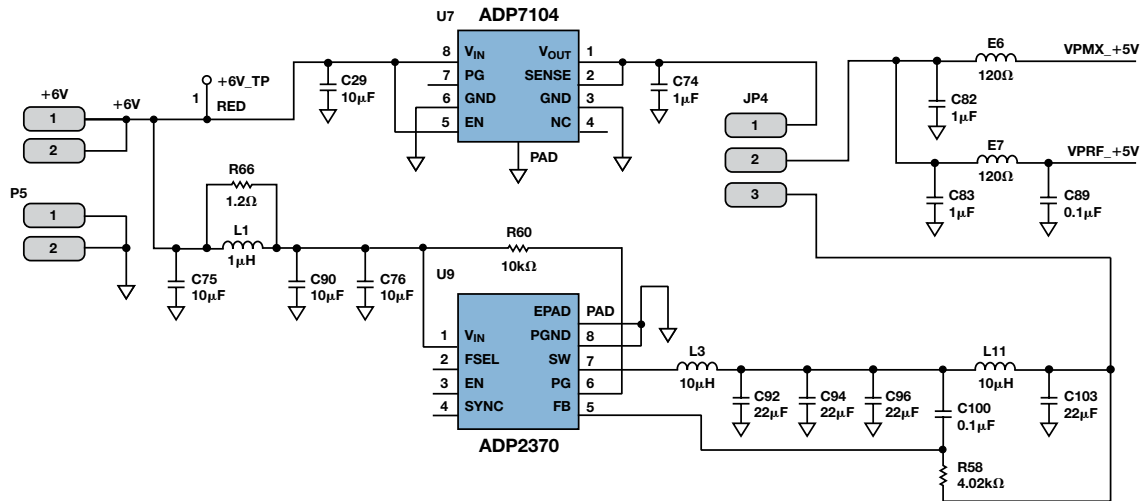


Figure 9. 5-V power design.

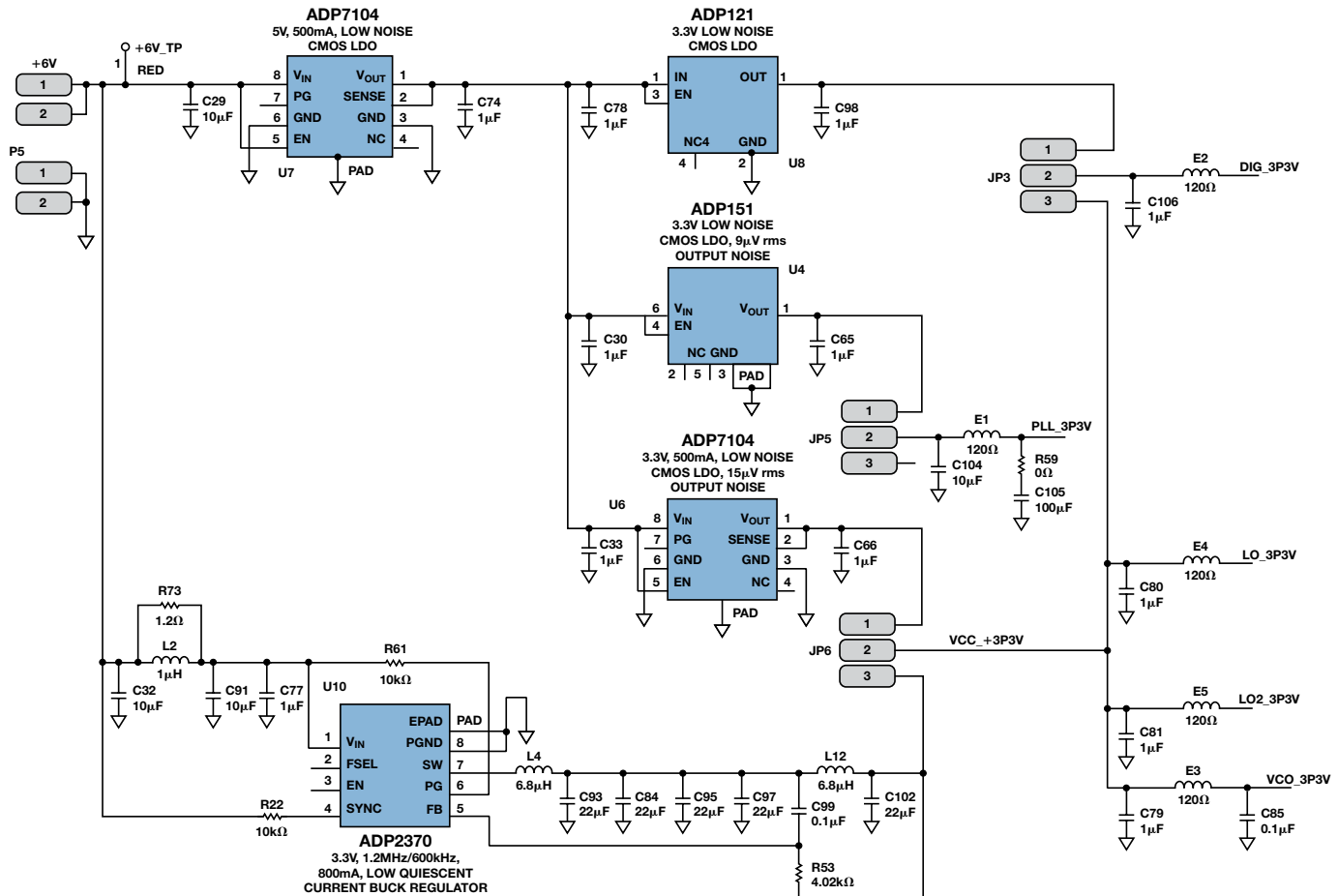


Figure 10. 3.3-V power design.

The 3.3-V design allowed for more experimentation. In addition to sourcing the 3.3-V rail with either an LDO or a switching regulator, the VPOS_PLL rail has additional LDO options and the VPOS_DIG rail has an optional isolated LDO. As the PLL power supply is the most sensitive, three power solutions were tried, each with different output noise: the ADP151 3.3-V ultralow-noise CMOS LDO with 9 μ V rms output noise; the ADP7104 3.3-V low-noise CMOS LDO with 15 μ V rms output noise; and the ADP2370 3.3-V buck regulator. We want to determine the highest level of power-supply noise that will still maintain the required phase-noise performance. Is the highest performance, lowest noise LDO an absolute necessity?

The ADP121 3.3-V low-noise CMOS LDO was also tried on the VPOS_DIG power rail to determine if digital noise would affect performance. The digital power rail tends to be noisier than the analog supplies due to switching on the SPI interface. We want to determine if the digital 3.3-V power supply will require its own LDO or if it can be coupled directly to the analog power-supply. The ADP121 was chosen as a low-cost solution.

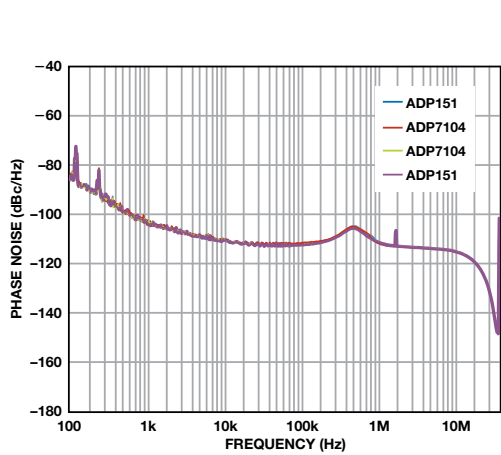


Figure 11. Integrated phase noise using the ADP151 and ADP7104.

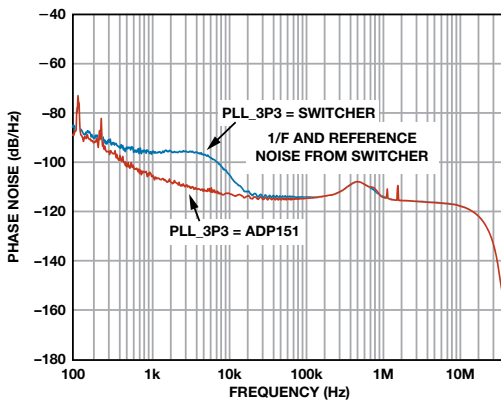


Figure 12. Integrated phase noise using the ADP151 and ADP2370.

Conclusions and Power Design Recommendations

For VPOS_PLL, the most sensitive power-supply rail, the low-cost ADP151 LDO achieves the same phase noise as the ADP7104 high-performance, low-noise LDO, as shown in Figure 11. Performance was degraded when the ADP2370 switching regulator was used, however, as shown in Figure 12. The noise hump is caused by the switching regulator, and can be seen on its output, as shown in Figure 13. Thus, VPOS_PLL can tolerate up to 15 μ V rms noise with no degradation in integrated phase noise, but a switching regulator cannot be used to power this pin. No benefit is obtained by using a higher performance, lower noise LDO.

Good phase-noise performance is maintained when either a switching regulator or an LDO powers the remaining supply rails, as shown in Figure 14. The 5-V power-supply pins, VMPX and VPRF, can both be tied together and sourced with a single supply. The 3.3-V power-supply pins, VPOS_LO1, VPOS_LO2, and VPOS_VCO, can also be tied together and sourced by a single supply. VPOS_DIG does not require an independent LDO and can be tied to the analog 3.3-V power supply.

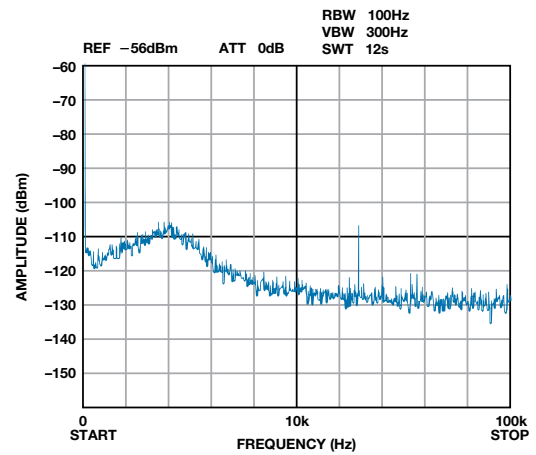


Figure 13. Output spectrum of the ADP2370.

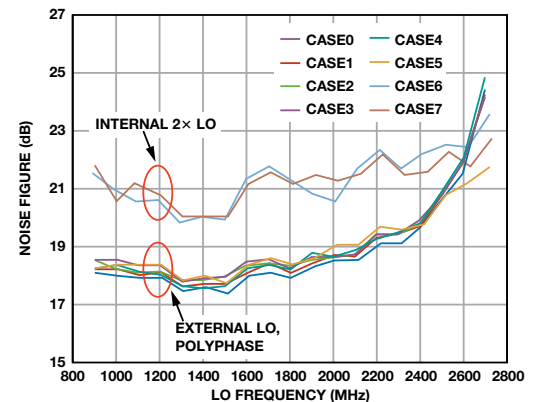


Figure 14. Switcher vs. LDO noise figure.

With a 6-V source voltage, the recommended power-supply design, shown in Figure 15, includes the ADP7104 5.0-V and the ADP7104 3.3-V LDOs. This solution uses only LDOs because the source voltage is close to the required supply voltages. The power efficiency is acceptable, so the added cost of filtering components and switching regulators is unnecessary.

With a 12-V source, the recommended power-supply design, shown in Figure 16, includes two switching regulators and an LDO. The source voltage is much larger than the required supply voltages, so the switching regulators are used to improve power efficiency. All of the power pins except for the sensitive VPOS_PLL supply can be sourced from the switching regulators. Either the ADP7104 or ADP151 can be used for VPOS_PLL.

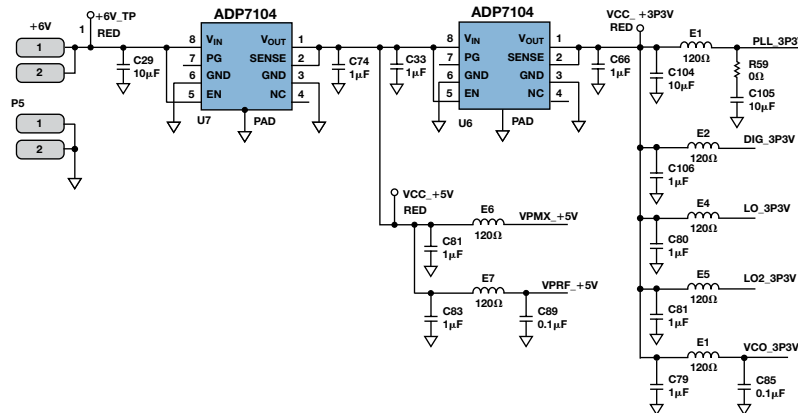


Figure 15. Recommended power-supply design for a 6-V source voltage.

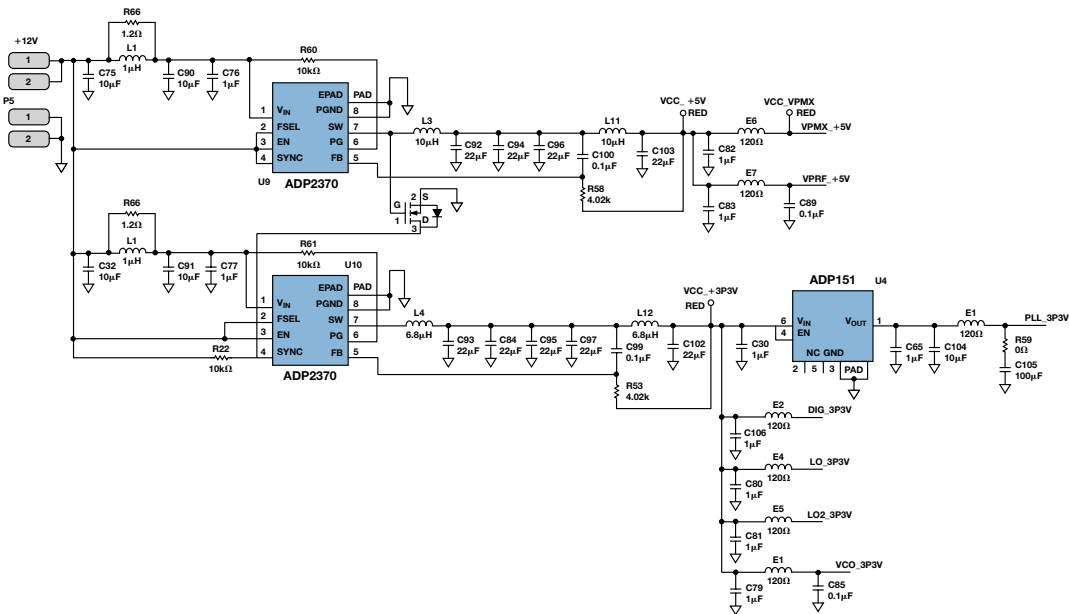


Figure 16. Recommended power-supply design for a 12-V source voltage.

References

Circuit Note CN0147. *Powering a Fractional-N Voltage-Controlled Oscillator (VCO) with Low Noise LDO Regulators for Reduced Phase Noise*. Analog Devices, Inc., 2010.

Collins, Ian. *Integrated PLLs and VCOs [Part 2]*. Radio-Electronics.com, Nov 2010.

Modulators/Demodulators

Linear Regulators

Switching Regulators



Qui Luu

Also by this Author:

[RF-to-Bits Solution Offers Precise Phase and Magnitude Data for Material Analysis](#)

Volume 48, Number 4



Qui Luu [qui.luu@analog.com] is an RF applications engineer at Analog Devices since June 2000. Qui received her B.S.E.E. from Worcester Polytechnic Institute in Worcester, MA, in 2000 and her M.S.E.E. at Northeastern University in Boston, MA, in 2005.

Design Reliable Digital Interfaces for Successive-Approximation ADCs

By Steven Xie

Introduction

Successive-approximation analog-to-digital converters, called SAR ADCs due to their successive-approximation register, are popular for applications requiring up to 18-bit resolution at up to 5 MSPS. Their advantages include small size, low power, no pipeline delay, and ease of use.

A host processor can access or control the ADC via a variety of serial and parallel interfaces such as SPI, I²C, and LVDS. This article discusses design techniques for reliable, integrated digital interfaces, including the digital power-supply level and sequence, I/O state during turn on, interface timing, signal quality, and errors caused by digital activity.

Digital I/O Power-Supply Level and Sequence

Most SAR ADCs provide a separate digital I/O power-supply input, V_{IO} or V_{DRIVE} , which determines the operating voltage and logic compatibility of the interface. This pin should be at the same voltage as the host interface (MCU, DSP, or FPGA) supply. The digital inputs should generally be between $DGND - 0.3\text{ V}$ and $V_{IO} + 0.3\text{ V}$ to avoid violating the absolute maximum ratings. Decoupling capacitors with short traces should be connected between the V_{IO} pin and $DGND$.

ADCs that operate with multiple supplies may have well-defined power-up sequences. Application Note AN-932, [Power Supply Sequencing](#), provides a good reference for designing supplies for these ADCs. To avoid forward biasing the ESD diodes and powering up the digital core in an unknown state, turn on the I/O supply before the interface circuitry. The analog supplies are usually powered before the I/O supply, but this is not the case for all ADCs. Read and follow the data sheet to ensure the proper sequence.

Digital I/O State During Turn On

For proper initialization, some SAR ADCs require certain logic states or sequences for digital functions such as reset, standby, or power-down. After all power supplies are stable, the specified pulse or combination is applied to guarantee that the ADC starts in the intended state. For example, a high pulse on RESET, with a duration of at least 50 ns, is required to configure the [AD7606](#) for normal operation after power up.

No digital pin should toggle until all the power supplies have been fully established. For SAR ADCs, the convert start pin, CNVST, may be sensitive to noise. Figure 1 shows an example in which the host cPLD brings CNVST high while AV_{CC} , DV_{CC} and V_{DRIVE} are still ramping. This might put the [AD7367](#) into an unknown state, so the host should keep CNVST low until the supplies are all fully established.

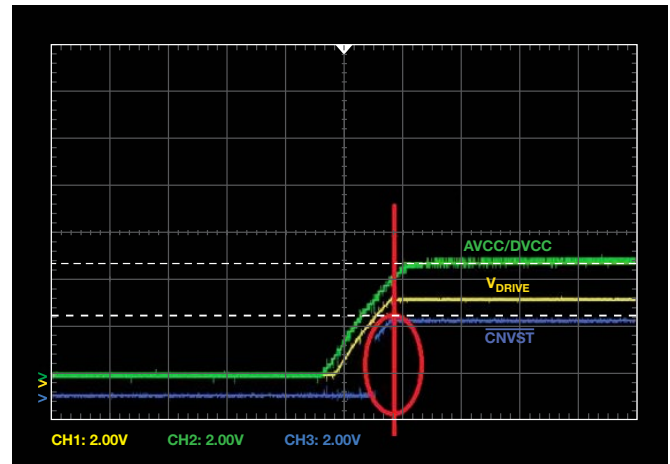


Figure 1. Bringing CNVST high during power-supply ramp up could result in an unknown state.

Digital Interface Timing

After a conversion has finished, the host can read the data via a serial or parallel interface. To read the data correctly, follow a specific timing strategy, such as which mode to use for the SPI bus. Do not violate the digital interface timing specifications, especially the setup and hold times of the ADC and the host. The maximum bit rate is determined by the whole cycle, not just the minimum specified clock period. Figure 2 and the following equations show an example of how to calculate the setup and hold timing margin. The host sends the clock to the ADC and reads the data output from the ADC.

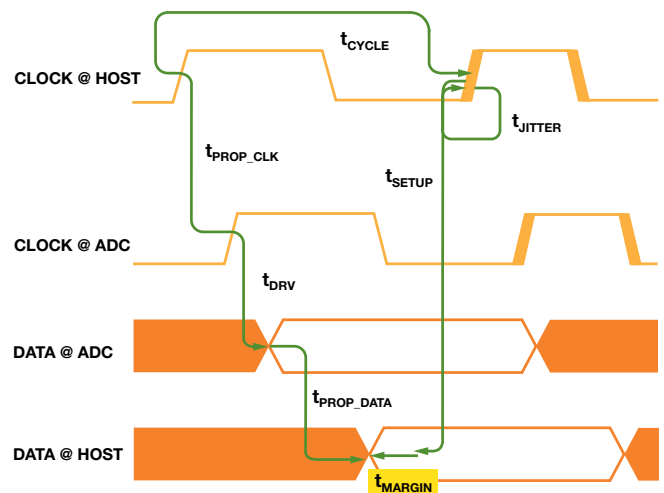


Figure 2. Setup and hold timing margin.

$$t_{\text{CYCLE}} = t_{\text{JITTER}} + t_{\text{SETUP}} + t_{\text{PROP_DATA}} + t_{\text{PROP_CLK}} + t_{\text{DRV}} + t_{\text{MARGIN}}$$

$$t_{\text{CYCLE}}: \text{Clock period} = 1/f_{\text{CLOCK}}$$

t_{JITTER} : Clock jitter

t_{SETUP} : Host setup time

t_{HOLD} : Host hold time

$t_{\text{PROP_DATA}}$: Data propagation delay along the transmit line from ADC to host

$t_{\text{PROP_CLK}}$: Clock propagation delay along the transmit line from host to ADC

t_{DRV} : Data output valid time after clock rising/falling edge

t_{MARGIN} : Margin time, ≥ 0 means the setup time or hold time is met; < 0 means the setup time or hold time is not met.

The setup margin for the host is

$$t_{\text{MARGIN_SETUP}} = t_{\text{CYCLE_MIN}} - t_{\text{JITTER}} - t_{\text{SETUP}} - t_{\text{PROP_DATA}} - t_{\text{PROP_CLK}} - t_{\text{DRV_MAX}}$$

The setup time equation defines the minimum clock period time or maximum frequency in terms of the maximum system delay terms. It must be ≥ 0 to meet the timing specifications. Increase the period (reduce the clock frequency) to handle excessive system delays. For buffers, level shifters, isolators, or other additional components on the bus, add the extra delay into $t_{\text{PROP_CLK}}$ and $t_{\text{PROP_DATA}}$.

Similarly, the hold margin for the host is

$$t_{\text{MARGIN_HOLD}} = t_{\text{PROP_DATA}} + t_{\text{PROP_CLK}} + t_{\text{DRV}} - t_{\text{JITTER}} - t_{\text{HOLD}}$$

The hold time equation defines the minimum system delay requirements to avoid logic errors due to hold violations. It has to be ≥ 0 to meet the timing specifications.

Many of ADI's SAR ADCs with an SPI interface clock the MSB from the falling edge of $\overline{\text{CS}}$ or CNV, while the remaining data bits follow the falling edge of SCLK, as shown in Figure 3. When reading the MSB data, use t_{EN} in the equations instead of t_{DRV} .

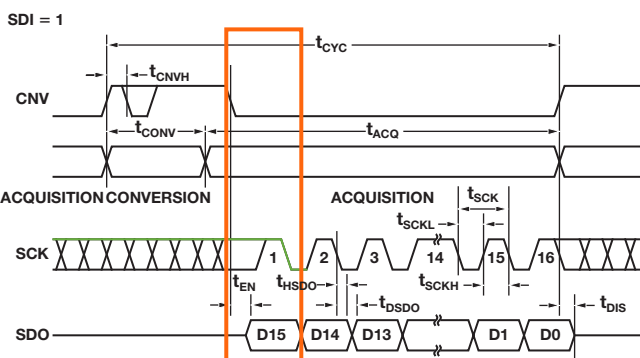


Figure 3. SPI timing of AD7980 3-wire $\overline{\text{CS}}$ mode.

Thus, in addition to the maximum clock rate, the maximum operating speed of the digital interface also depends on the setup time, hold time, data output valid time, propagation delay, and clock jitter.

Figure 4 shows a DSP host accessing the AD7980 in 3-wire $\overline{\text{CS}}$ mode, with $V_{\text{IO}} = 3.3 \text{ V}$. The DSP latches the SDO signal on the falling edge of SCLK. The DSP specifies 5 ns minimum setup time and 2 ns minimum hold time. For a typical FR-4 PC board, the propagation delay is about 180 ps/in. The propagation delay of the buffer is 5 ns. The total propagation delay for CNV, SCLK and SDO is

$$t_{\text{PROP}} = 180 \text{ ps/in} \times (9 \text{ in} + 3 \text{ in}) + 5 \text{ ns} = 7 \text{ ns.}$$

$$t_{\text{JITTER}} = 1 \text{ ns. The host runs SCLK at } 30 \text{ MHz, so } t_{\text{CYCLE}} = 33 \text{ ns.}$$

$$t_{\text{SETUP_MARGIN}} = 33 \text{ ns} - 1 \text{ ns} - 5 \text{ ns} - 7 \text{ ns} - 11 \text{ ns} - 7 \text{ ns} = 2 \text{ ns}$$

$$t_{\text{HOLD_MARGIN}} = 11 \text{ ns} + 7 \text{ ns} + 7 \text{ ns} - 1 \text{ ns} - 2 \text{ ns} = 22 \text{ ns}$$

The setup and hold margins are both positive, so the SPI SCLK can run at 30 MHz.

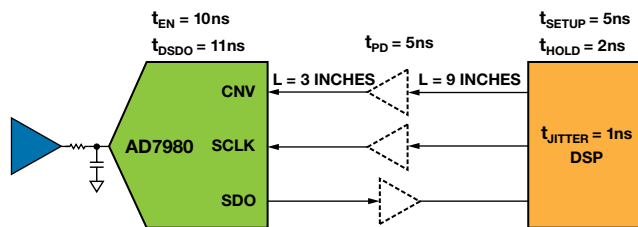


Figure 4. Digital interface between DSP and AD7980.

Digital Signal Quality

Digital signal integrity, which includes both timing and signal quality, ensures that signals: are received at specified voltage levels; do not interfere with one another; do not damage other devices; and do not pollute the electromagnetic spectrum. Signal quality is specified by many terms, as shown in Figure 5. This section will introduce overshoot, ringing, reflection, and crosstalk.

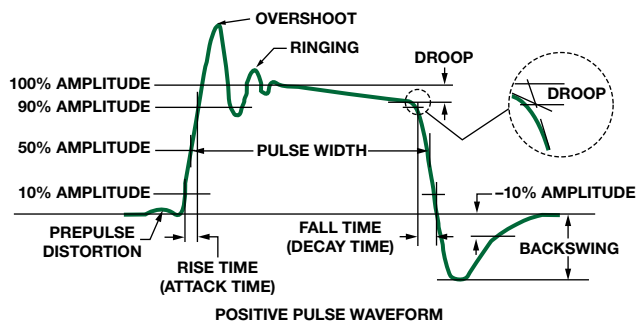


Figure 5. Common specifications of signal quality.

Reflection is a result of impedance mismatch. As a signal travels along a trace, the instantaneous impedance changes at each interface. Part of the signal will reflect back, and part will continue down the line. Reflection can produce overshoot, undershoot, ringing, and nonmonotonic clock edges at the receiver.

Overshoot and undershoot can damage the input protection circuitry or shorten the IC's life span. Figure 6 shows the absolute maximum ratings of the AD7606. The digital input voltage should be between -0.3 V and $V_{\text{DRIVE}} + 0.3 \text{ V}$. In addition, ringing above V_{IL} maximum or below V_{IH} minimum may cause logic errors.

ABSOLUTE MAXIMUM RATINGS $T_A = 25^\circ\text{C}$, UNLESS OTHERWISE NOTED.

PARAMETER	RATING
V_{CC} TO AGND	-0.3 V TO $+7 \text{ V}$
V_{DRIVE} TO AGND	-0.3 V TO $V_{\text{CC}} + 0.3 \text{ V}$
ANALOG INPUT VOLTAGE TO AGND	$\pm 16.5 \text{ V}$
DIGITAL INPUT VOLTAGE TO DGND	-0.3 V TO $V_{\text{DRIVE}} + 0.3 \text{ V}$
DIGITAL OUTPUT VOLTAGE TO GND	-0.3 V TO $V_{\text{DRIVE}} + 0.3 \text{ V}$
REFIN TO AGND	-0.3 V TO $V_{\text{CC}} + 0.3 \text{ V}$
INPUT CURRENT TO ANY PIN EXCEPT SUPPLIES	$\pm 10 \text{ mA}$

Figure 6. Absolute maximum ratings of AD7606.

To minimize reflection:

- Make the trace as short as possible
- Control the characteristic impedance of the trace
- Eliminate stubs
- Use an appropriate termination scheme
- Use solid metal with a small loop area as the return current reference plane
- Use lower drive currents and slew rates

Many software tools or webs are available for calculating the characteristic impedance of a trace, such as the Polar Instruments Si9000 PCB transmission line field solver. These make it easy to get the characteristic impedance by selecting a transmission line model and setting parameters such as dielectric type and thickness, and trace width, thickness, and separation.

IBIS is an emerging standard used to describe the analog behavior of an IC's digital I/O. ADI provides IBIS models for SAR ADCs. Prelayout simulation checks clock distribution, chip package type, board stack-up, net topology, and termination strategies. It can also check the serial interface timing constraints to guide placement and layout. Post-simulation verifies that the design meets all guidelines and constraints, and checks for violations such as reflection, ringing, and crosstalk.

Figure 7 shows one driver connected to SCLK1 through a 12" microstrip line, and a second driver connected to SCLK2 through a 43-Ω resistor in series with the microstrip.

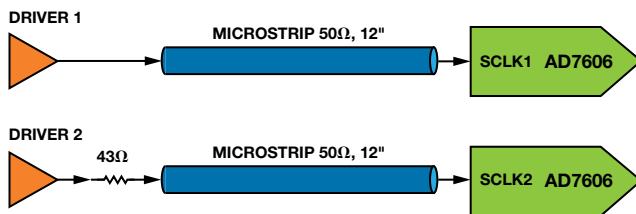


Figure 7. Drive AD7606 SCLK.

Figure 8 shows a large overshoot on SCLK1 that violates the -0.3 V to +3.6 V absolute maximum rating. The series resistor reduces the slew rate on SCLK2, keeping the signal within the specifications.



Figure 8. AD7606 IBIS model simulation of overshoot.

Crosstalk is the coupling of energy between parallel transmission lines via mutual capacitance (electric field) or mutual inductance (magnetic field). The amount of crosstalk depends on the signal's rise time, the length of the parallel lines, and the spacing between them.

Some common practices to control the crosstalk are:

- Increase the line spacing
- Minimize parallel runs
- Keep the traces close to the reference metal planes
- Use an appropriate termination scheme
- Reduce the signal's slew rate

Performance Degradation Caused by Digital Activity

Digital activity can degrade the SAR ADC's performance, with the SNR decreasing due to a noisy digital ground or power supply, sampling clock jitter, and digital signal interference.

Aperture or sampling clock jitter sets the limit for SNR, especially for high-frequency input signals. System jitter comes from two sources: aperture jitter from the on-chip track-and-hold circuitry (internal jitter) and jitter on the sampling clock (external jitter). Aperture jitter is the conversion-to-conversion variation in the sampling time, and is a function of the ADC. The sampling clock jitter is usually the dominant source of error, but both sources cause varying analog input sampling times, as shown in Figure 9. Their effects are indistinguishable.

The total jitter produces an error voltage, with overall SNR of the ADC limited by

$$SNR = 20 \log_{10} \left[\frac{1}{2 \pi f t_j} \right]$$

$$Total \text{ jitter} = t_j \text{ (rms)},$$

$$Total \text{ jitter} = \sqrt{(ADC \text{ aperture jitter})^2 + (sampling \text{ clock jitter})^2}$$

where, f is the analog input frequency and t_j is the total clock jitter.

For example, with a 10 kHz analog input and 1 ns total jitter, the SNR is limited to 84 dB.

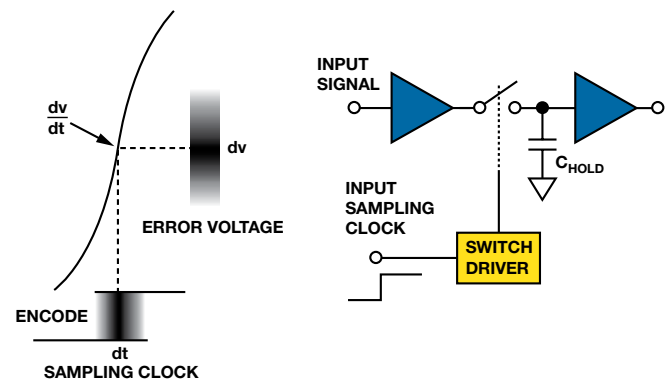


Figure 9. Error voltage caused by sampling clock jitter.

Power-supply noise caused by digital outputs switching should be isolated from the sensitive analog supplies.

Separately decouple the analog and digital power supplies, paying careful attention to ground return current paths.

High-precision SAR ADCs can be sensitive to activity on the digital interface, even when the power supply is properly decoupled and isolated. Burst clocks often perform better than continuous clocks. The data sheet usually shows quiet time when the interface should not be active. Minimizing digital activity during these times—typically the sampling instant and when critical bit decisions occur—can be challenging at higher throughput rates.

Conclusion

Pay careful attention to digital activity to ensure valid conversions from SAR ADCs. Digitally induced errors may take SAR ADCs into an unknown state, cause malfunctions, or degrade performance. This article should help designers investigate root causes and provide solutions.

References

- Kester, Walt. “[Data Converter Support Circuits.](#)” *Data Conversion Handbook*, Chapter 7. Analog Devices, Inc., 2004.
- Brannon, Brad. AN-756 Application Note. *Sampled Systems and the Effects of Clock Phase Noise and Jitter*. Analog Devices, Inc., 2004.
- Ritchey, Lee W. *Right the First Time: A Practical Handbook on High-Speed PCB and System Design*, Volume 1. Speeding Edge, 2003.
- Usach, Miguel. AN-1248 Application Note. *SPI Interface*. Analog Devices, Inc., 2013.
- Casamayor, Mercedes. AN-715 Application Note. *A First Approach to IBIS Models: What They Are and How They Are Generated*. Analog Devices, Inc., 2004.



Steven Xie [steven.xie@analog.com] has worked as an ADC applications engineer with the China Design Center in ADI Beijing since March 2011. He provides technical support for SAR ADC products across China. Prior to that, he worked as a hardware designer in the Ericsson CDMA team for four years. In 2007, Steven graduated from Beihang University with a master's degree in communications and information systems.



Steven Xie

Also by this Author:
[Successive-Approximation ADCs: Ensuring a Valid First Conversion](#)
Volume 47, Number 4

Notes



©2015 Analog Devices, Inc. All rights reserved.
Trademarks and registered trademarks are the
property of their respective owners.

Ahead of What's Possible is a trademark of
Analog Devices.

M02000491-0-6/15

analog.com/analogdialogue

**Analog Devices, Inc.
Worldwide Headquarters**

Analog Devices, Inc.
One Technology Way
P.O. Box 9106
Norwood, MA 02062-9106
U.S.A.
Tel: 781.329.4700
(800.262.5643,
U.S.A. only)
Fax: 781.461.3113

**Analog Devices, Inc.
Europe Headquarters**

Analog Devices, Inc.
Wilhelm-Wagenfeld-Str. 6
80807 Munich
Germany
Tel: 49.89.76903.0
Fax: 49.89.76903.157

**Analog Devices, Inc.
Japan Headquarters**

Analog Devices, KK
New Pier Takeshiba
South Tower Building
1-16-1 Kaigan, Minato-ku,
Tokyo, 105-6891
Japan
Tel: 813.5402.8200
Fax: 813.5402.1064

**Analog Devices, Inc.
Asia Pacific Headquarters**

Analog Devices
5F, Sandhill Plaza
2290 Zuchongzhi Road
Zhangjiang Hi-Tech Park
Pudong New District
Shanghai, China 201203
Tel: 86.21.2320.8000
Fax: 86.21.2320.8222



AHEAD OF WHAT'S POSSIBLE™