

Care and Feeding of FPGA Power Supplies: A How and Why Guide to Success

By **Nathan Enger**

Share on   

Introduction

Modern FPGAs are among the most complex integrated circuits ever created. They employ the most advanced transistor technology and cutting-edge architectural structures to achieve both incredible flexibility and the highest performance. Over time, as technology has advanced, this complexity has dictated certain compromises in the design and implementation of systems using the FPGAs. This is nowhere more apparent than in the power supplies, which must be ever more accurate, more agile, more controllable, smaller, more efficient, and more fault aware with each new FPGA generation.

In this article, we look specifically at some of the constraining specifications for the Altera® Arria 10 FPGA, and what they mean for a power supply design. Then we discuss the best power delivery solutions, and lay out the plan to successfully meet all of the specifications and make our FPGA perform at its optimal efficiency, speed, and power level using Analog Devices' complete set of power system management (PSM) ICs, including the [LTC3887](#), [LTC2977](#), and [LTM4677](#).

FPGA Power Requirements (Interpreting the Data Sheet)

Engineers should spend most of their time programming—they don't want to spend time and energy thinking about designing suitable power supplies. Indeed, the best approach to power delivery is to use a robust, flexible, proven design that meets the requirements and expands with the project. Here we take a closer look at some of the important power specifications and what they mean.

Voltage Accuracy

Core power supply voltage is one of the most important keys to balancing FPGA power and performance. The specification documents give a range of acceptable voltages, but the total range is not the complete picture. As with all things, there are trade-offs and optimizations to be made.

Table 1 is an example of core voltage specifications from the popular Altera Arria 10 FPGA.¹ Though these numbers are specific to the Arria 10, they are representative of other FPGA core voltage requirements. The range amounts to a $\pm 3.3\%$ tolerance around the nominal voltage. The FPGA will operate just fine within this voltage window, but the complete picture is more complicated.

Table 1. Altera Arria 10 Core Voltage Specification

Symbol	Description	Condition	Minimum	Typical	Maximum	Unit
V_{cc}	Core voltage power supply	Standard and low power	0.87	0.9	0.93	V
			0.92	0.95	0.98	V
		SmartVID	0.82		0.93	V

Notice the line labeled “SmartVID,” with a range of 0.82 V to 0.93 V. This represents a broad range of voltages that are possible when the FPGA is requesting its own core voltage through the SmartVID² interface (more on this later). This SmartVID specification is an indication of an underlying truth about the FPGA: it can operate at different voltages, depending upon its particular manufacturing tolerance, and upon the particular logic design that it is implementing. The static voltage required by one FPGA might be different from another FPGA. The power supply must be able to respond and adapt.

The goal is to produce just the right performance level to operate the programmed functions without burning unnecessary power. We know from semiconductor physics, as well as published data from Altera, Xilinx® (Figure 1), and others, that both dynamic and static power will increase dramatically with increased core V_{DD} , so the goal is to give the FPGA just enough voltage to meet its timing requirements, but no more. Excess power dissipation does nothing to increase performance. In fact, it makes things worse because transistor leakage current increases with higher temperatures, dissipating even more unwanted power. For these reasons, the imperative is to optimize the voltage for the design and operating point.

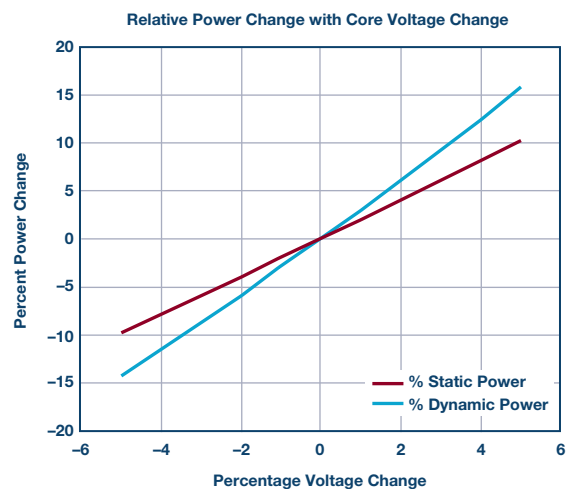


Figure 1. Xilinx Virtex V power vs. core V_{cc} .

This optimization process requires a very accurate power supply for success. Regulator inaccuracy must be baked into the error budget and subtracted from the available voltage range that can be used for optimization. If the core voltage drops below the requirement, the FPGA may fail due to timing errors. If core voltage drifts above the maximum specification, it may damage the FPGA, or it may create hold-time failures in the logic. All of these scenarios must be guarded against by taking into account the power supply tolerance range, and only commanding voltages that are guaranteed to remain within the specification limits.

The problem is that most power regulators are not accurate enough. The regulated voltage may be anywhere within the tolerance range around the commanded voltage, and it may drift with load conditions, temperature, and age. A power supply that guarantees a $\pm 2\%$ tolerance may regulate anywhere within a 4% voltage window. In order to compensate for the possibility that the voltage may be 2% too low, the commanded voltage must be raised by 2% above that required to meet timing. If the regulator then drifts to 2% above the commanded voltage, it will operate 4% above the minimum voltage required at that operating point. This still meets the specified voltage required by the FPGA, but it wastes a lot of power (Figure 2).

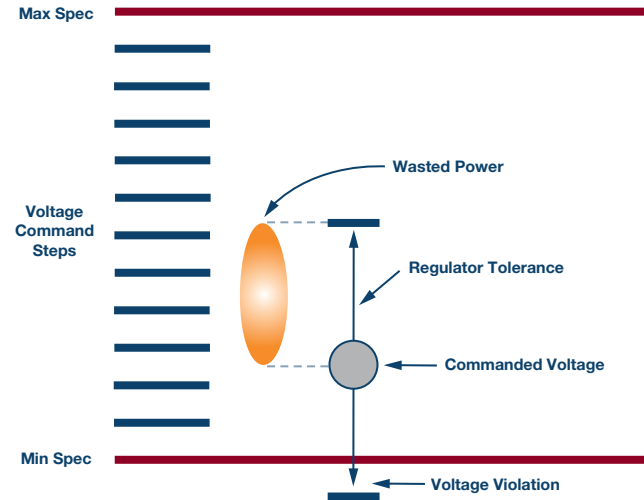


Figure 2. Power supply regulator tolerance trade-offs.

The solution is to choose a power supply regulator that can operate with a much tighter voltage tolerance. A regulator with a $\pm 0.5\%$ tolerance can be commanded to operate much closer to the minimum required specification at the desired operating frequency, and it is guaranteed to be less than 1% from that required voltage. The FPGA will be functional and it will be dissipating the minimum power possible at that operating condition.

The LTC388x family of power supply controllers guarantees regulated output voltage tolerance better than $\pm 0.5\%$ over a wide, configurable voltage range. The LTC297x family of power system managers guarantee a trimmed voltage regulator tolerance of better than $\pm 0.25\%$. With these accuracies, there is a clear path to optimizing the power vs. performance trade-off for any FPGA.

Thermal Management

A more subtle implication of power supply accuracy manifests itself in the thermal budget. Because static power dissipation is far from negligible, the FPGA heats up even when it is doing nothing. The increased temperature causes more static power dissipation, which further increases operating temperature (Figure 3). Adding unnecessary voltage to the power supply only makes this problem worse. An inaccurate power supply requires a guard band in operating voltage to ensure that there is enough voltage to do the job. The power supply voltage uncertainty that results from variability in tolerance, system assembly variability, and operating temperature can create voltages significantly higher than the minimum required. This additional voltage, when applied to the FPGA, can cause thermal complications, or even thermal run-away at high processing loads.

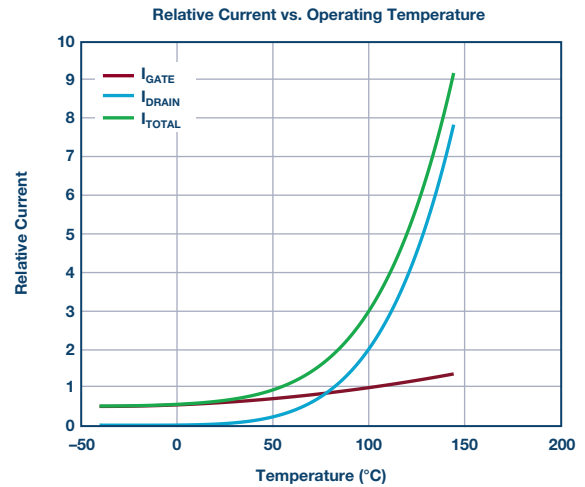


Figure 3. Power supply current vs. operating temperature.

The remedy is a very accurate power supply that creates just the right voltage, and no more than necessary, which is exactly what the ADI power system management (PSM) devices do well.

SmartVID

SmartVID is the Altera name for the technique of operating each individual FPGA at its optimal voltage, as requested by the FPGA itself. There is a register inside of the FPGA that contains a device-specific voltage (programmed at the factory) at which the FPGA is guaranteed to operate efficiently. A piece of compiled IP inside of the FPGA can read this register and make a request over an external bus to the power supply to provide this exact voltage (Figure 4). Once the voltage is reached, it remains static during operation.

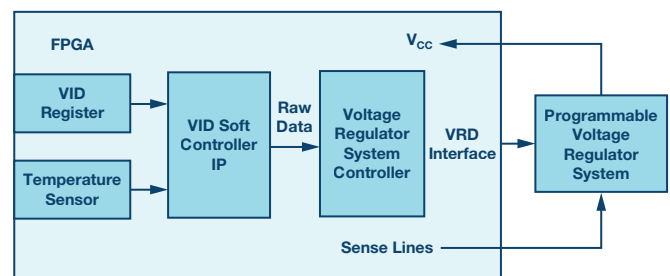


Figure 4. Altera SmartVID structure.

The demands of the SmartVID application on the power supply include the specific bus protocol, voltage accuracy, and speed. The bus protocol is one of several methods that the FPGA uses to communicate its required voltage to the power regulator. Of the available methods, PMBus is the most flexible because it addresses the widest variety of power management ICs. The SmartVID IP uses two PMBus commands: VOUT_MODE, and VOUT_COMMAND, with which it commands the PMBus-compliant power regulator to the correct voltage.

The voltage accuracy and speed requirements for the regulator include an autonomous boot voltage (before the PMBus is active), the ability to accept a new voltage command every 10 ms, the ability to take 10 mV steps every 10 ms during the voltage adjustment phase, and the ability to settle to within 30 mV (~3%) of the target within the 10 ms step time, ultimately ramping to the commanded voltage and remaining static during FPGA operation.

Though Altera uses SmartVID, there are other, similar techniques in use around the industry that accomplish much the same thing. One of the simplest is to test each board at the factory and program into the power supply's nonvolatile memory an exact voltage that optimizes performance for that particular board. This technique does not require any further intervention for the power supply to operate at the correct voltage. This is an advantage of a power supply manager or controller with EEPROM.

All of the requirements for Altera SmartVID can be met by the LTC388x family of power supply controllers. In addition, the LTM4675/LTM4676/LTM4677 μ Module regulators easily meet the requirements and offer a complete solution in a single compact form.

Timing Closure

The computing speed of any logic block depends on its supply voltage. Within limits, a higher voltage gives faster performance. We have already seen why we cannot simply operate at the highest voltage in order to guarantee the best speed. On the other hand, we must operate at a high enough voltage for the application, as shown in Figure 5.

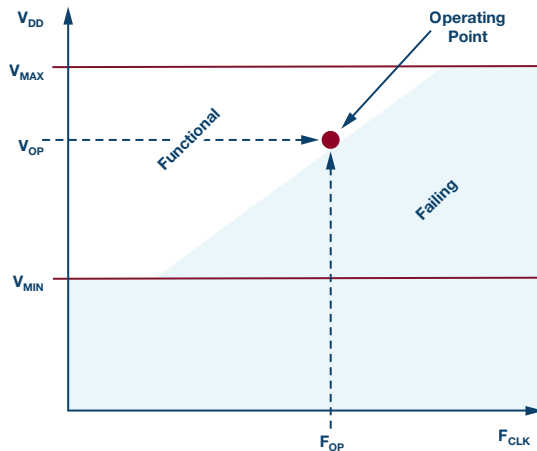


Figure 5. FPGA operating frequency vs. V_{DD} trade-off.

An important implication of Figure 5 is what can be done when a particular design does not meet its logic timing requirements, and falls into the failing area. Often the edge between functional and failing is not well defined before a design is committed to hardware, and the particular voltage at which it will pass timing cannot be predetermined. The only options are to either commit in advance to a voltage that is well above the minimum, thus wasting power to guarantee functionality, or to design a flexible power supply that can adapt to the needs of the hardware at test time, or even, as in the case of SmartVID, at power-up time. The ability to adapt to unknown demands makes the accuracy of ADI PSM devices more valuable, as FPGA designers can trade off power for performance in the real design, and at any stage of development.

Power Supply Sequencing 101

Moore's law drives the trend of shrinking transistors in modern FPGAs and forces the trade-offs involved in using these tiny transistors, which are very fast and small, but much more fragile. A chip containing hundreds of millions of transistors must be segmented into cores, blocks, and partitions that can be designed and managed independently. The practical result of these considerations is an FPGA with many power domains. Some recent FPGAs have upwards of a dozen power supplies that need to be kept happy. This includes, in addition to voltage, current, ripple, and noise, the sequence order during start-up, shutdown, and fault conditions.

Recent FPGA specifications call out specific requirements for sequence order when starting-up and shutting down the power supplies. Both Xilinx and Altera recommend specific ordering and timing to ensure that the FPGA gets reset properly, maintains minimal current draw, and maintains its I/Os in the proper tristate configuration during the power transitions.

Given the number of power supplies for each FPGA, the complexity of the sequencing task is considerable.

The Altera Arria 10 prescription divides the power supplies into three sequence groups (1, 2, and 3), and requires that they sequence up in order 1, 2, and 3, and down in the reverse order: 3, 2, and 1.³

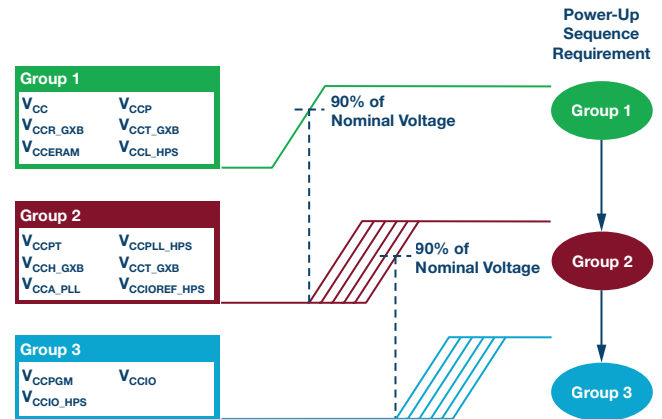


Figure 6. Altera Arria 10 up-sequence group order.

Similarly, the Xilinx recommendation for the Virtex UltraScale FPGA up-sequence is: V_{CCINT}/V_{CCINT_IO} , V_{CCBRAM} , V_{CCAUX}/V_{CCAUX_IO} , and V_{CCO} . Down-sequencing is the reverse of the up-sequence order.⁴

These are just two of the many FPGAs available. Nearly every modern FPGA system has multiple power supply rails, and one of the most obvious questions to ask is, in what order should they turn on and off? Even if there is no explicit sequencing requirement there are good reasons to enforce a deterministic sequence of events. Here are some of the available design options.

- ▶ No sequencing: Let the power supplies rise and fall on their own. What could go wrong?
- ▶ Hardware cascade sequencing: Each rising power supply is hardwired to enable the next one. This only works when the supplies are rising.
- ▶ CPLD-based sequencing: Use programmable logic to create a custom solution. This is flexible, but the entire challenge rests on the designer.
- ▶ Event-based sequencing: Event-based sequencing is similar to cascade sequencing, but more flexible because it can operate both up and down. A dedicated sequencer IC can be programmable and take care of many fault scenarios and corner cases.
- ▶ Time-based sequencing: Time-based sequencing triggers each event at a specified time. Coupled with comprehensive fault management, time-based sequencers can be flexible, deterministic, and safe.

The following sections explore these options in more detail.

No Sequencing

It is possible to turn-on a power supply system with no management at all. When main power becomes available, or the ON switch is activated, the regulators start regulating. When power is removed, or the ON switch turns off, the regulators stop regulating. Of course, the problems with this approach are many. Some more obvious than others.

Lack of timing determinism can have various effects in a system. The first is simply that it stresses the sensitive FPGA. This might cause immediate catastrophic failure, or it might cause premature aging that slowly degrades performance. Neither is good. It may also cause unpredictable power-on-reset behavior or indeterminate logic states at power-up, which make system stability questionable and difficult to debug. Questions of fault detection and response, energy management, and debug support are left entirely unanswered in this scheme. In general, to avoid power supply sequencing is to invite disaster.

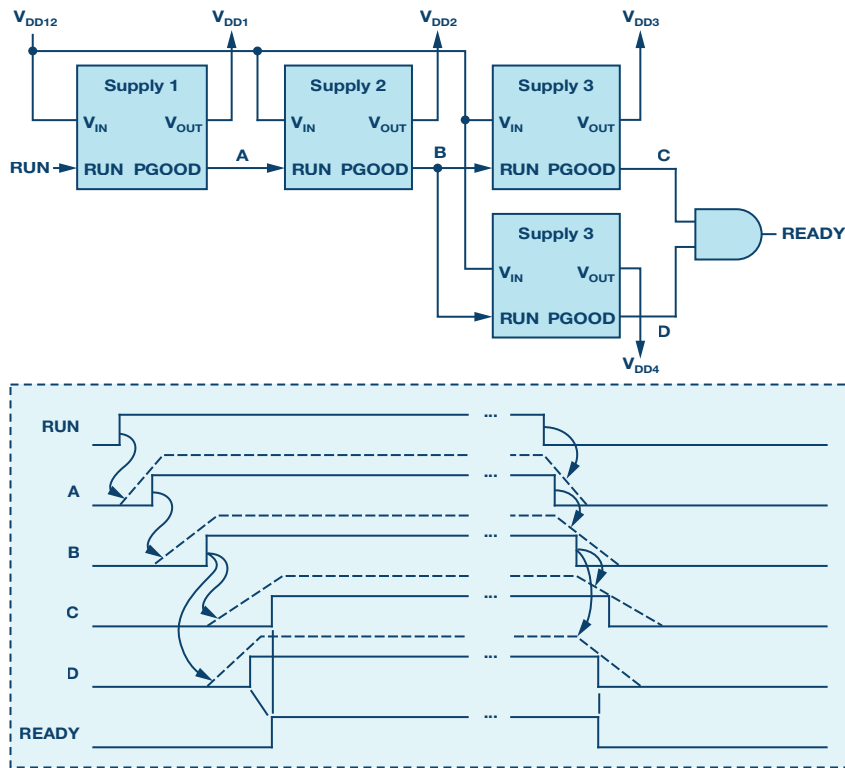


Figure 7. PGOOD-to-RUN cascade sequencing.

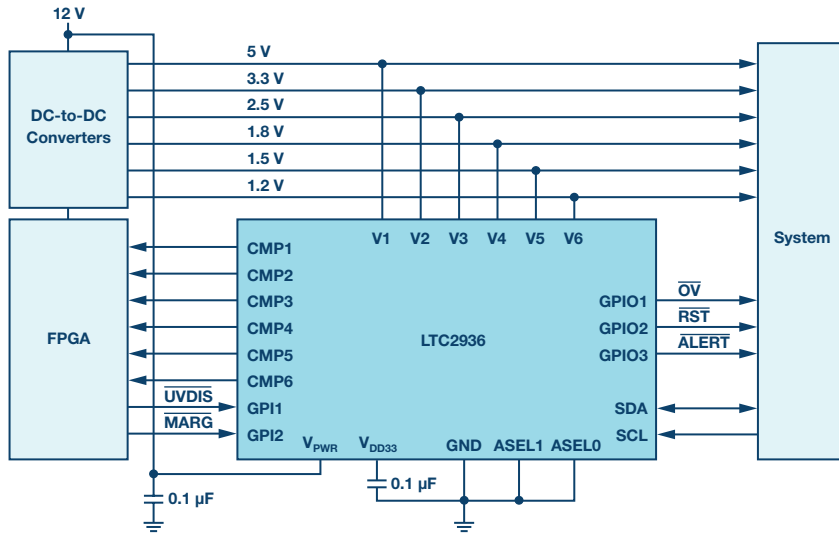


Figure 8. LTC2936 programmable voltage supervisor.

Cascade Sequencing

A slightly more organized approach to sequencing is the classic PGOOD-to-RUN hardwired cascade shown in Figure 7. This is like dominoes falling: each one taps the next in the sequence, which guarantees progress in order. This technique has the benefit of simplicity. Unfortunately, it also has its shortcomings. While it often works adequately for sequencing up a power system, it cannot operate in reverse (or in any other order) for down-sequencing. There can be only one sequence order. Additionally, this scheme cannot gracefully handle faults or manage energy in uncertain operating conditions. It is not smart enough to make any decisions. If one stage of the sequence fails, what happens next? If one operating power supply browns-out, what happens? The answers are undefined, and debugging these problems is not easy.

FPGA or CPLD Sequencing

Using an auxiliary CPLD or FPGA on the board to sequence the supplies is an option that many designers choose. In a system designed by and for digital designers, it has a certain appeal. There is a natural fit to designing

a digital control block that can be programmed into an FPGA to control the power supply of another FPGA. Here the decision can be deceptive because a power supply system is not as simple as it may appear from a digital control perspective.

If designers wish to tackle the power supply sequencing, control, and management problem from top to bottom, they must first thoroughly understand its complexities. We have already discussed many of these, and there are more as well, such as detecting and responding to overvoltage and undervoltage conditions that can happen on the microsecond time scale, detecting hazardous currents and temperatures, logging telemetry and status, and providing bring-up and debug services to make life easier for the hardware guys. All of these considerations require dedicated analog hardware in addition to the digital algorithms.

For the intrepid designers who wish to take this path, Analog Devices provides several analog front-end ICs to help with the task. At the interface between the digital bits and the analog power supply, the LTC2936 provides six rugged, highly accurate programmable threshold analog comparators to

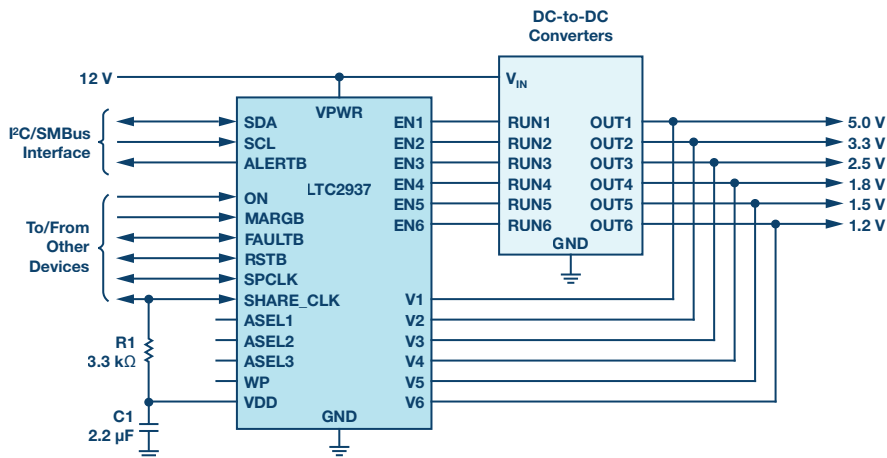


Figure 9. LTC2937 power supply supervisor and sequencer.

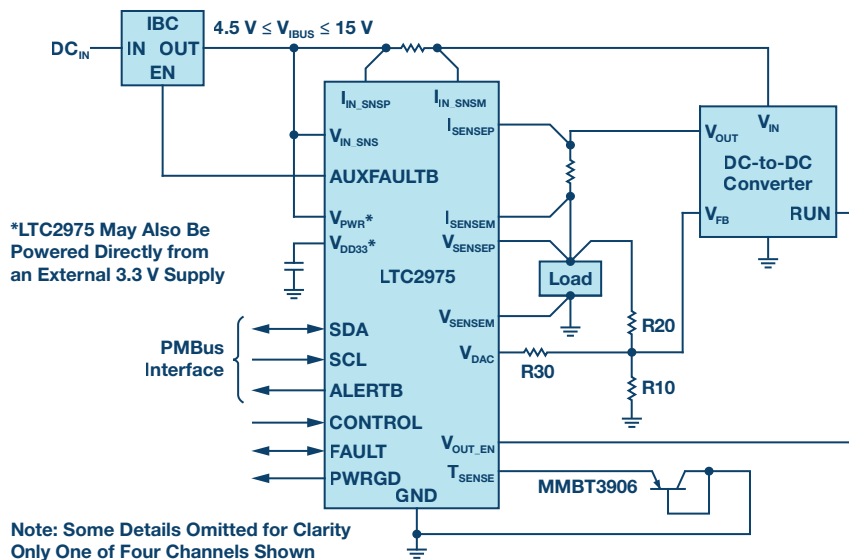


Figure 10. LTC2975 4-channel power system manager.

detect fast events and send digital status to the logic. It also has three programmable GPIO pins for additional functionality. This programmable IC has an EEPROM for nearly instant-on functionality at start-up, and has the ability to store fault telemetry for debugging through its I²C/SMBus interface. A convenient way to use LTC2936 is shown in Figure 8.

In addition to the fast comparator functions, there must be an analog-to-digital converter (ADC) to gather telemetry. A proven choice is the [LTC2418](#), which can monitor up to 16 channels of analog signals with its fast-settling 24 bit Σ - Δ ADC and 4-wire SPI interface. The board controller can readily stream measurements and monitor many points of interest in the system.

In general, there are many, many options for using an FPGA or CPLD to control power sequencing. This approach works, but somebody must own the digital and analog designs, including all of the inevitable design bugs, opportunities for unimaginable corner cases and faults, and the unhappy question of support. There are certainly easier ways to build a power system.

Simple Sequencer/Supervisors

Solving the puzzle of robust sequencing and fault handling is the domain of the simple sequencer/supervisors. These do the important job of sequencing the power rails and ensuring that they remain within their specified limits during operation (supervision). The [LTC2928](#) is an easy to use pin-strap configurable sequencer with configurable sequence timing (down is the reverse of up), and configurable supervisor voltage thresholds.

It has the potential to meet the requirements, but has no frills and offers no digital programmability or telemetry.

In the category of programmable sequencer and supervisor with EEPROM is the [LTC2937](#). It features full digital programmability, features time-based and event-based sequencing, and can sequence and supervise any number of power supplies, handling faults, and logging fault status to the EEPROM black box. It is a worthy solution for cases where voltage management and telemetry are not required.

Power System Management

To fully capture all of the benefits of complete PSM, use one of the Analog Devices PSM ICs. These introduce the ability to autonomously sequence up and down any number of power rails; accurately control rail voltages to better than 0.5% (or 0.25% in some cases); measure and report voltage, current, temperature, and status telemetry; cooperatively handle complex fault scenarios; and record detailed fault information to EEPROM.

Sequencing is done by a system of timing handshaking, with all ICs agreeing on time zero and the time base, and all sequence events happening at preprogrammed times (time-based sequencing). This allows any number of rails to sequence up and sequence down autonomously.

The family of PSM ICs includes controllers that have their own switch drivers and analog loop control to handle the switching power supply in all

aspects. Alternatively, power supply managers contain a servo loop that wraps around an external power supply, adding all of the features of power supply management, including sequencing, supervision, and monitoring, to any power rail, from switching power regulators to LDO regulators. An example of a power supply manager is the [LTC2975](#), pictured in Figure 10.

µModule Devices

The most tightly integrated solutions, providing the most functionality per square centimeter, in a BGA or LGA footprint, are PSM µModule® devices. These are complete power supply systems in a single package, including controller ICs, inductors, switches, and capacitors. Some µModule regulators, such as the [LTM4650](#), do not contain digital functions, so they can benefit from additional sequencing and management with the [LTC2975](#). Some µModule regulators, such as the [LTM4676A](#), contain their own PSM functions and can easily integrate with other PSM ICs in the system.

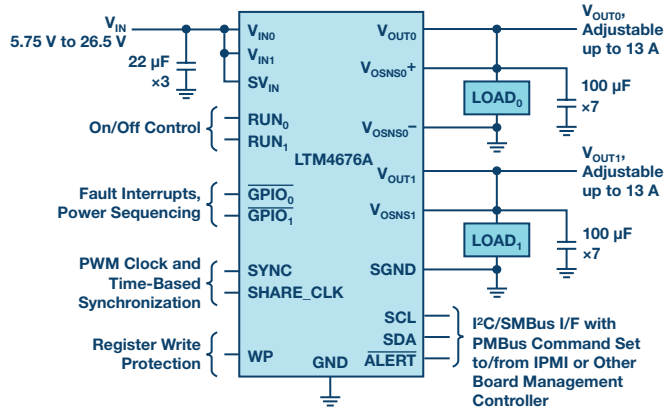


Figure 11. LTM4676A PSM µModule dual 13 A regulator.

Shared Sequencing

The PSM micromodules, manager ICs, and controller ICs all cooperate together in sequencing up and sequencing down by sharing timing information through a simple one-wire bus called SHARE_CLK. Through this single wire all of the PSM ICs share information about when sequencing should begin (time zero), when each tick of the clock occurs, and other status information that affects sequencing. It is enough to simply connect all of the SHARE_CLK pins together in the system to enable this coordination. Each IC has its own programming for sequence timing that can use the shared time base to accurately and reliably time events such as enabling and disabling, ramping, and timing-out in case of a fault.

The SHARE_CLK pin, at its most basic level, is an open-drain, 100 kHz clock pin. The open-drain nature means that an IC can either pull-down actively or let go and allow the bus to float. When all devices on the bus let go, the pull-up resistors pull the voltage to 3.3 V. This allows one device to stop the clock by pulling down until it is ready, and means that all devices must agree before the clock can start: an effective mechanism for communicating time zero, as well as indicating sequencing status by stopping the clock.

Shared Fault Handling

Similar to the SHARE_CLK pin is the FAULT bus. Each of the PSM ICs in the system is connected to the shared FAULT wire, and can either pull it low using its open-drain output, or respond when another device pulls low. This provides a simple, quick way for the entire family of PSM devices to communicate and respond to faults. The behavior is fully configurable, and allows a coordinated response when something goes wrong, either during sequencing or during steady state. The system can be configured to remove power and attempt to re-sequence according to specified timing, while recording black-box information about the state of the system and causes of the fault when it occurred. This EEPROM black-box information is available for later processing over the I²C bus.

Down-Sequencing and Managing Stored Energy

There is an additional consideration when sequencing down the supplies: energy management. Increasingly, it is important to provide deterministic timing for power supplies as they sequence down, and this requires carefully considering where the stored energy in the system is dissipated. A high power supply is likely to have dozens of large electrolytic capacitors as bulk charge storage elements, and these will be charged to the supply voltage, holding enough energy to blow-up an improperly protected device under unfortunate conditions. To avoid this, FPGA manufacturers specify a down-sequence that protects the device. For the Altera Arria 10 this sequence is shown in Figure 12.⁵

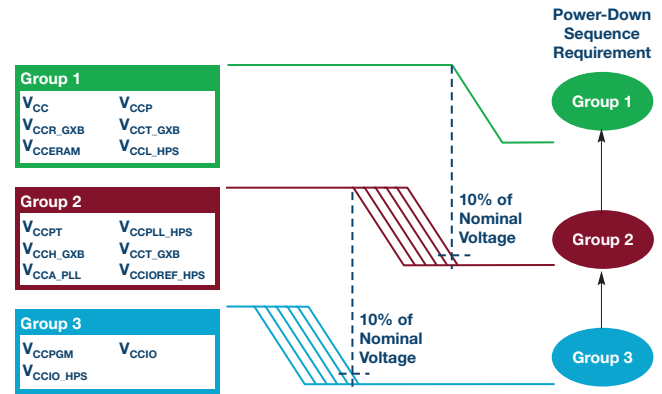


Figure 12. Altera Arria 10 down-sequence group order.

Implicit in this down-sequence is the requirement that all of that stored energy in the capacitors goes somewhere and is safely dissipated. There are several ways to do this. The simplest is a fixed resistor across the capacitor. This resistor always dissipates power while the supply is turned on, but its resistance can be made large enough that the comparative loss is minimal, and the RC discharge time constant is acceptably short. The time that it takes to sufficiently discharge the supply is a multiple (usually 5×) of the RC time constant, and should be optimized to make the static power dissipated in the resistor acceptable (<¼ W, for instance). For a 1 mF capacitance and a 1.0 V supply a resistor value of $R = 4 \Omega$ will have a time constant of $\tau = 4 \text{ ms}$, and will discharge the supply to below 50 mV in just about 13 ms. This approach is sufficient as long as the resistor is rated for at least ¼ W, and the system works with a constant ¼ W loss and a 13 ms discharge time.

A more complex, but very safe, option is to switch the resistor across the capacitor only when it is time to discharge the supply. This approach pulls charge out of the bulk capacitors when it is time to do so, and safely dissipates it in the resistance of the switch FET and in a supplemental series resistor, but it avoids the continual power drain of a fixed resistor. The circuit is shown in Figure 13.

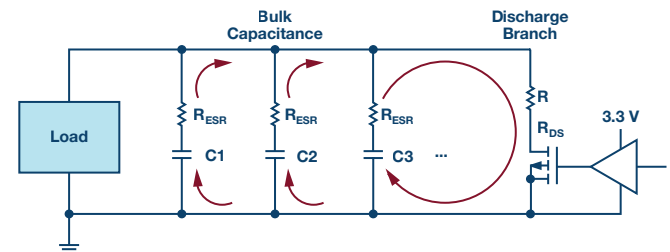


Figure 13. Discharging power supply capacitance with a FET.

There are several considerations with this approach: control, discharge time, and power dissipation. There must be an available signal to command the discharge switch to close at the appropriate time. The switch FET is NMOS, so the control signal must rise above the V_{th} of the FET sufficiently to drive it into saturation. For common FETs, this gate drive voltage may be as much as 3 V to 5 V.

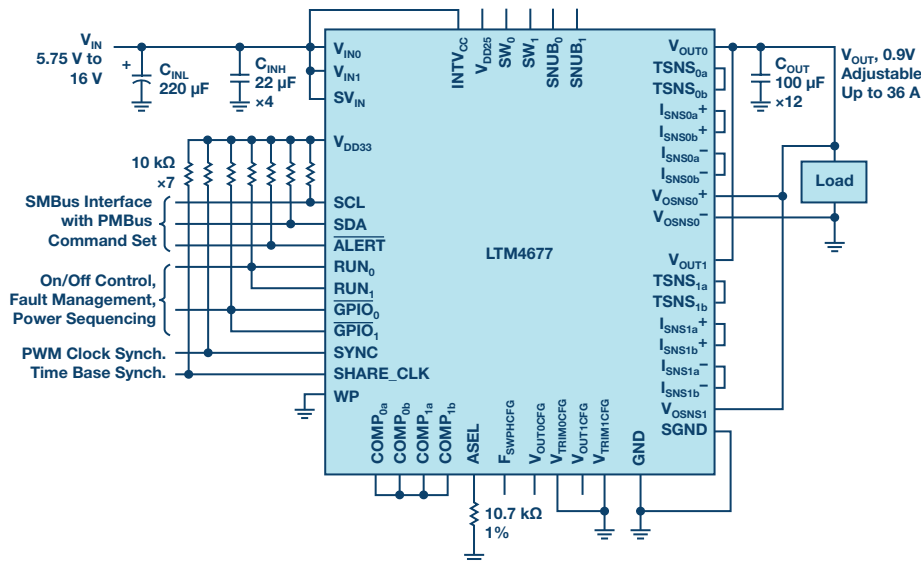
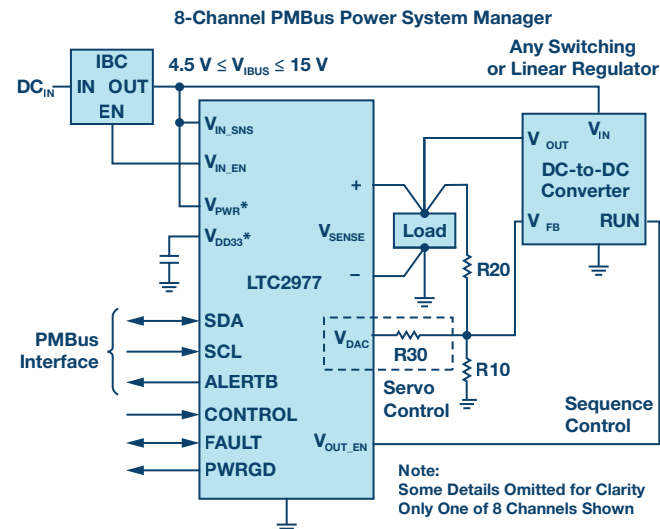


Figure 18. A single LTM4677 providing up to 36 A.

This solution provides the best board space utilization because the integrated μ Module devices require very few external components, and the PMBus interface makes them configurable without hardware modification. Micromodules offer the lowest complexity solution because many of the complex analog considerations, such as power switches, inductors, current and voltage sensing elements, loop stability, and thermals, are included.

Because the LTM4677 module includes PSM, it guarantees that the core power supply will always operate within $\pm 0.5\%$ of the dc voltage target. It also allows for voltage adjustments through the PMBus interface, both from the SmartVID IP inside of the FPGA and from the LTpowerPlay[®] graphical user interface (GUI) that gives full user control over the power supplies.

To manage power supply regulators that do not include their own PSM capabilities, we simply include the LTC2977, which is an 8-channel PMBus-compatible power system manager. Each channel wraps around a power supply to servo the voltage to within 0.25% of the programmed target (Figure 19). It cooperates seamlessly with the LTM4677 μ Module devices for both sequencing and fault response, making the entire power system coherent and easily programmable.



*LTC2977 May Also Be Powered Directly from an External 3.3 V Supply
Figure 20. LTC2977 managing any power supply voltage.

System power sequencing is provided by the cooperative partnership of the LTM4677 core supply, the LTM4676A 3.3 V power supply, and the LTC2977 that manages all of the other power regulators on the board. These ICs have common PMBus timing commands (stored in EEPROM) that easily configure start-up and shutdown sequencing in any order, and with any timing. These guarantee the proper autonomous sequence of events specified for Group 1, Group 2, and Group 3 power supplies (Figure 6).

In addition to the voltage accuracy and sequencing control, the LTM4677, LTM4676A, and LTC2977 on this board provide complete fault handling as well. In the event of an overvoltage, undervoltage, brownout, overcurrent, or complete failure of one or more rails, the system can be configured to respond quickly and automatically, shutting-down to protect the sensitive FPGA, and restarting if it is possible.

Most of the power supply rails in the system require modest currents (less than 13 A) and modest voltage tolerance. These can be supplied by non-PSM devices such as the LTM4620, and sequenced and managed by the LTC2977. This provides a very effective balance between board area, complexity, and cost.

There are also power supply rails, such as PLL and transceiver power, that require lower noise than a switching regulator can provide, and these need a linear regulator. The LTC3025-1 and LTC3026-1 serve these functions well, removing the switching and load-induced noise from their outputs. These, too, can be managed by the LTC2977 to sequence, trim, and handle fault conditions.

LTpowerPlay

The entire family of PSM devices is supported by the comprehensive LTpowerPlay GUI (Figure 21). Because much of the functionality of PSM is accessed through the rich set of configuration registers in the EEPROM of the ICs, one tool can consolidate the entire collection of PSM ICs on the bus into one easy to use view. The LTpowerPlay tool provides a deep set of features to accelerate all phases of design and development. It can operate offline to present a view of the ICs before they are programmed, or in real-time, communicating over the I²C bus with a complete system containing from one to hundreds of power supply rails controlled by many PSM devices. LTpowerPlay simplifies and streamlines complex configurations by providing detailed information about registers and functions. It graphically represents all of the configuration, status, and telemetry information available in the system, making it clear and understandable while the system is operating. It simplifies programming and maintenance of the complete

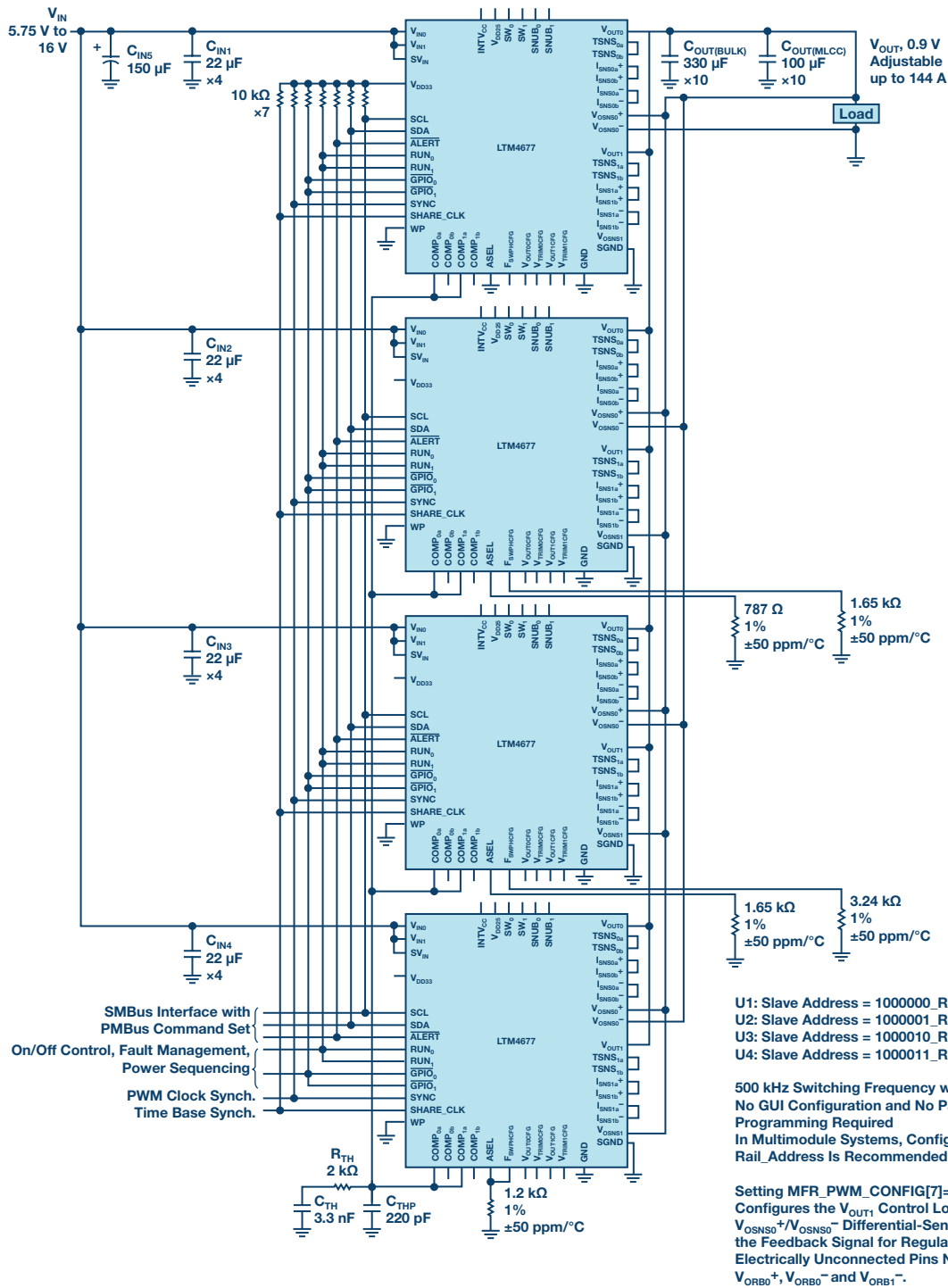


Figure 19. Four LTM4677s providing up to 144 A at 0.9 V.

register sets, providing a simple way to create and save configurations on a Microsoft® Windows® PC. When power supply faults occur, LTpowerPlay makes it easy to see where in the system the fault happened and what the status, telemetry, and black-box information indicate about what happened. It also provides detailed debugging help for common fault scenarios. In the event that someone needs a helping hand, LTpowerPlay also has the ability to call for help, enlisting a live support person who can view the GUI running in real time and see what you see.

Download the free LTpowerPlay tool [here](#).

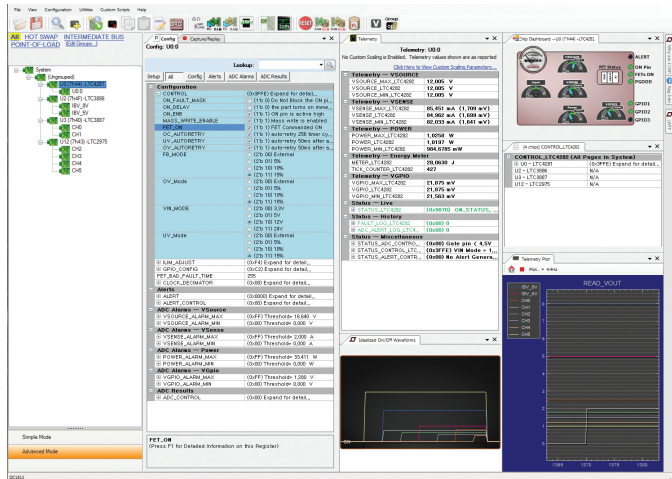


Figure 21. LTpowerPlay graphical user interface.

Analog Devices provides a comprehensive set of demonstration platforms for Altera, Xilinx, and NXP FPGAs. These fully functioning boards are working examples of how PSM provides the cleanest, most flexible, and robust power solutions for FPGA systems. In addition, your local Analog Devices applications engineers can provide detailed help in selecting and using the complete portfolio of PSM ICs. Read more, download reference material, and order the FPGA boards [here](#).

The Journey to FPGAs

Now that we understand how best to power an FPGA system, we can take a whimsical aside and look at why things are this way. In order to understand why things are the way that they are today, we need a brief lesson in history.

Moore's Law

In 1965 Gordon Moore published his famous article in *Electronics Magazine*,⁸ stating his observation that the number of transistors on a single chip had been doubling every year, and his prediction that it should continue to do so, at least through the year 1975. Later enhancements and additional observations of the larger electronics market caused him to revise his model, but the basic principle of a sustained exponential growth rate in the number of transistors on a chip has become an axiom in the electronics industry. This is a curious self-fulfilling prophecy that exists in no other industry, and at no other time in history. In fact, it has become a primary motivator for engineers across the globe, creating innovations, and forcing trade-offs that were as-yet unimagined when Gordon Moore first published his simple observation.

As a result of this technological race against ourselves, the decision-making process has always favored technologies that squeeze more devices into a smaller area at the expense of cost, power, usability, and even durability. In the technology race, size is everything. Some of the implications of this trajectory are that advanced chips use more power, become leakier, are more fragile and more sensitive, and they are much more difficult to manage and to protect.

Transistor Engineering

As transistors have shrunk to feature sizes on the nanometer scale, important side effects have become increasingly dominant. The most obvious is voltage headroom. Whereas 5 V would have been a fine power supply for transistors a few decades ago, such a voltage would break down all of the junctions and oxides in a more recent FET transistor. As transistor features shrink, the internal electric fields become much stronger, and the tolerable operating voltage shrinks to prevent damage. Recent transistor generations can only tolerate about 1.0 V as a maximum power supply voltage. In addition, the absolute voltage tolerance also shrinks proportionally: 2% of 1.0 V is a much smaller range than 2% of 5 V, making accuracy an ever more pressing concern.

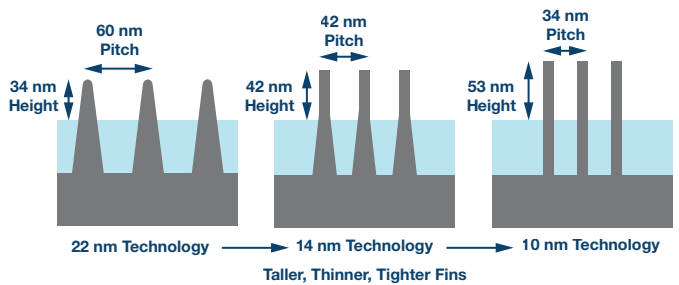


Figure 22. Shrinking transistor sizes.

Along with shrinking voltages comes increasing transistor current drive (I_{DSAT}). Increased drive strength accomplishes at least two purposes. First, it allows a transistor with a smaller gate voltage to drive a significant current—making it strong enough to switch at useful frequencies. Second, it allows a physically smaller transistor. The smaller transistor can be faster. Unfortunately, the increased transistor drive strength comes with its own penalty: leakage current.

There are two kinds of power dissipated by the transistors on a chip. Dynamic power is the familiar cost of switching between Logic 1 and Logic 0 at some frequency, and dynamic power is caused by charging and discharging the tiny parasitic capacitors associated with the transistors themselves and the wires on the chip connecting devices together. Dynamic power is proportional to the frequency of logic transitions and to the square of the power supply voltage.

Less obvious is the power spent by leaking transistors. This power leaks away any time the circuit is powered, regardless of whether it is active or idle, clocking or not. Increased transistor drive strength causes more leakage current because junctions and structures made to conduct more current are harder to turn off. A stronger transistor tends to leak more than a weaker one. With every transistor generation, the effects of leakage have grown. It is only the combination of heroic transistor engineering (chemical, metallurgical, lithographical, and physical) and accurate, flexible power supply management that keeps leakage power under control.

A decade ago, Gordon Moore observed these facts and noted two important points. First, if dynamic power continued to rise at the same pace, then the junction temperature on an operating chip would approximate that of the surface of the sun. Second, if nothing else was done, leakage power would overtake dynamic power as the primary energy dissipation mode, further exacerbating the power dissipation problem (Figure 23). To address these effects, the IC industry adopted several new techniques at that time. One of these was clock management—slowing or stopping the clocks to curb dynamic power—and another was the use of multiple processing cores on a single chip to leverage the growing transistor count.

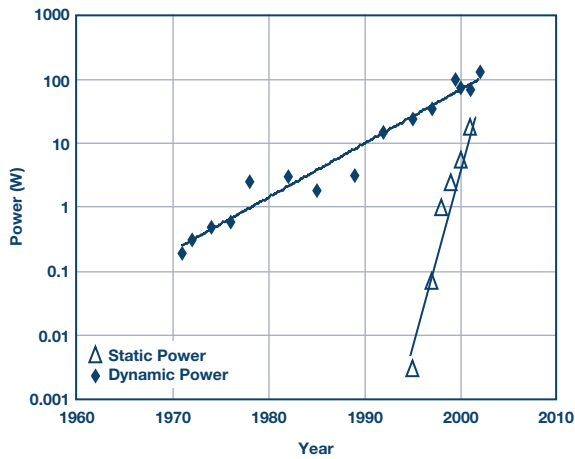


Figure 23. Static and dynamic power growth.⁹

Even with all of this advanced architecture, the problem of leakage power remains troublesome. Transistor engineering is a powerful way to bend the curve downward, but it is not enough. As each smaller transistor generation demands a lowered supply voltage, the problem of dynamic power remains tame, but the resulting increased transistor strength and leakage, coupled with the ever-growing number of devices on the chip, produces a demand for voltage management. Power supply voltage must be tightly controlled, as well as actively adjustable, to meet the needs of each particular device.

Advanced Architectures

Architectural developments up until the turn of the millennium had mainly focused on optimizing a single computing core to perform as many computations possible as quickly as possible. This involved the free technique of raising the clock rate to just under the speed at which the circuit failed: its maximum operating frequency. It also involved architectural optimizations, but these were mainly intended to squeeze more performance out of every clock cycle.

Following the startling realization that power mattered, engineers began to redirect resources away from raw speed and into more subtle kinds of optimization. This new trend showed up in computing architecture first as a plateau in the ever-increasing clock speeds, and a leveling-out of the rate of performance increase per transistor in each generation (Figure 24). This was the most obvious way to tame the beast of dynamic power: stop slopping charge from V_{DD} to V_{SS} so fast.

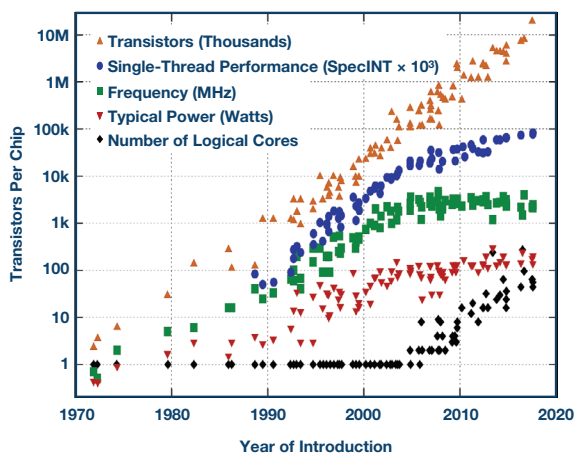


Figure 24. Growth trends in computing hardware metrics.

But the number of transistors on a single chip continued to climb at the inexorable rate predicted (demanded?) by Gordon Moore. Something had to be done with all of these transistors. This necessitated the second great innovation: multicore architectures. At about the same time that the clock speeds stopped growing, the number of cores on a single chip started growing. The advantages of multiple cores include simpler chip design through reuse, simpler software design with familiar building blocks, and the ability to individually throttle each core to meet the demands of the computational load. The multicore revolution began with fixed computing platforms, but one might say that this event was the moment when FPGAs came into their own: when the world realized that maximizing the number of cores is best. In a sense, nothing has more cores than an FPGA with its sea of identical, programmable logic blocks!

Anatomy of an FPGA

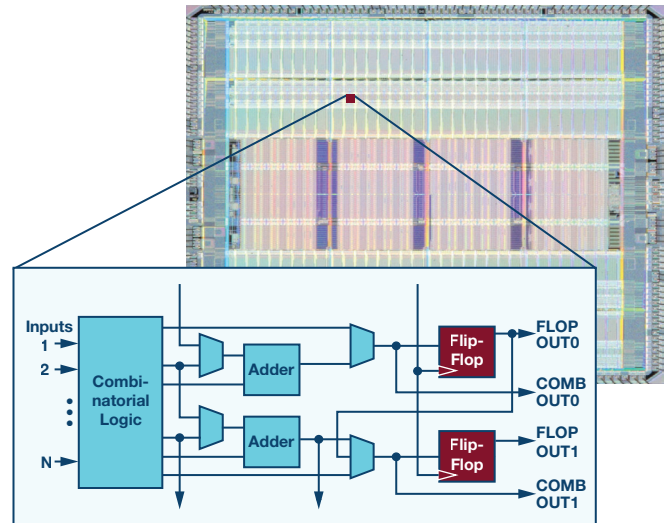


Figure 25. FPGA configurable fabric.

An FPGA, at its most basic level, is a collection of primitive configurable logic cells tied together through a configurable mesh of interconnections. Together with a compiler they form a highly flexible computing fabric that transforms into nearly any imaginable general-purpose digital function, including combinatorial and sequential logic blocks. At the top level, this fabric is surrounded by additional features to support and augment functionality. Some blocks, such as bias circuits, RAM, and PLLs, support functions internal to the chip. Various configurable GPIO cells, high speed communication hard macros (LVDS, DDR, HDMI, SMBus, etc.), and high speed transceivers allow the logic inside of the chip to communicate with the outside world with various voltages, speeds, and protocols. Other blocks, such as integrated CPU and DSP cores, support optimized functions that are commonly needed, and optimized for power, speed, and compactness.

The FPGA core fabric is composed of thousands or millions of primitive cells called configurable logic blocks (CLBs). Each CLB is a collection of combinatorial and sequential logic elements that together can produce an elementary computation and hold the value in one or more flip-flops. The combinatorial logic usually takes the form of a programmable look-up table (LUT) than can translate a few input bits to a few arbitrary output bits. Each LUT performs one basic logic function, as programmed, and passes the result into the configurable interconnect for subsequent processing (Figure 26). The specific CLB and LUT design is one of the secrets that make one FPGA family different from another. Inexpensive FPGAs use simpler CLBs with fewer inputs, outputs, and interconnections, and fewer flip-flops. The highest end FPGAs use much more complex CLBs, each capable of more inputs, more logic combinations, and higher speed. This optimization allows more computation per CLB, and more optimized performance in the compiled design. Naturally, the added inputs and outputs in more complex FPGAs have different dynamic power trade-offs than simpler, less interconnected devices.

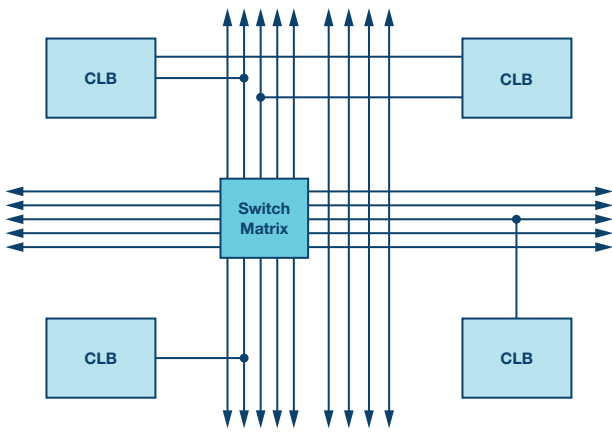


Figure 26. Configurable switch matrix between CLBs.

The fundamental concept of configurable logic functions continues outside of the core fabric itself into the I/O cells, which are also highly configurable to meet a broad array of voltages, drive strengths, and logic styles (push-pull, tristate, open-drain, etc.). Like the configurable LUTs and interconnect matrix, programmable I/Os receive their configuration at boot-up time from configuration memory, which has implications for power supply sequence order.

There are also functional blocks that cannot, or should not, be implemented using general-purpose CLBs and GPIOs. These are the so-called hard macros. They are functions that benefit from optimization, or that simply cannot be made fast enough or small enough, and need dedicated circuits. These include gigabit transceivers, arithmetic logic and DSP elements, specialized controllers, memories, and dedicated processor cores. These are hard macros in contrast to soft blocks that can be compiled like software and loaded into the configurable fabric. Hard macros usually have their own power supplies, specific voltages, and timing requirements.

All of these various functional blocks have diverse power needs that the power supply system must accommodate. The core fabric usually requires both the lowest voltage and the highest power on the chip. In a modern FPGA, the fabric, when fully utilized, may demand more than 100 A from a power supply operating at 0.85 V. Similar voltages are found in the CPU core, but at different currents, and with different sequencing requirements. Other on-chip analog functions may be powered by 1.8 V or 3.3 V and must be energized before anything else. At the same time, the GPIO banks may operate at 3.3 V or 1.8 V, and must not be energized until the power-on reset for the core fabric is complete. Each of these power supply sequence requirements must be enforced by the system.

The final piece of FPGA architecture is the tool chain (Figure 27). In order to transform the blank slate of configurable logic fabric into a high performance circuit, a comprehensive set of tools exists to translate a set of Verilog or VHDL code into logical blocks, assign clock, reset, and testability resources; optimize the functions for speed, power, or size constraints; then load the result into the FPGA's configuration EEPROM. Without these tools the FPGA would never achieve its full potential. In fact, the tools and programming languages are so important that they often overshadow the fundamental circuit design that enables the FPGA to function. Engineers spend most of their time programming and don't want to spend time and energy thinking about delivering suitable power supplies. Often overlooked, however, are the power supply requirements implied by the tools. Because so much effort goes into the digital design, it is only late in the game when the compiled design comes together that power requirements are known, and problems with power supplies may be discerned. Here in the digital design and software tools, just as much as in the hardware design, flexible power supply architecture is crucial to success.

History, economics, and human factors continue to drive the trends in transistors and architectures that create FPGAs. At every level, and every design stage, the power supply plays a critical, unseen role in the success of the FPGA. The best choice of power supply is one that is accurate, robust, flexible, compact, and easy to use. In all of these qualities, Analog Devices' family of PSM products set the standard for the industry.

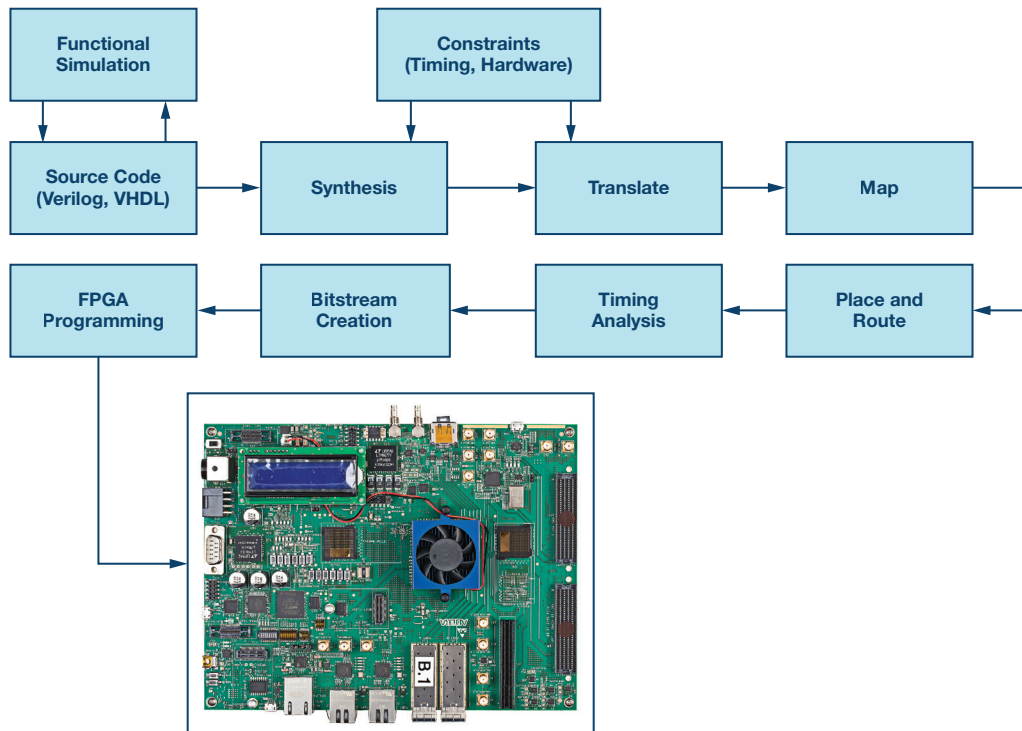


Figure 27. FPGA design flow (hardware comes last).

Acknowledgements

We would like to thank M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, I. Hammond, C. Batten, and K. Rupp¹¹ for their contributions in collecting and plotting the data in Figure 24.

Appendix A: Care and Feeding?

There may be some readers who question the title of this article, especially those who are not familiar with English language vernacular. At first it may seem entirely out of place to refer to the care and feeding of an FPGA. The answer to this objection, however, is quite simple: English is a funny language. While no one agrees on the precise moment in history that the term “care and feeding” came into popular usage, it is well understood that the term originates in the agrarian roots of a simpler time, and has come into popular use (abuse) to refer to just about anything that may be fragile or temperamental. In this case we have hit the nail on the head. While it is arguable whether one must “feed” an FPGA, one certainly must “care” for it!

While the term “care and feeding” is broadly applied in our modern internet era to such things as infants, children, husbands, bosses, expatriates, scientific data, and even digital pulse-shaping filters, one of the earliest, and perhaps the oddest, references can be found in this classic text, which is interesting not only for its title, but for the fact that it is instantly available on the internet.

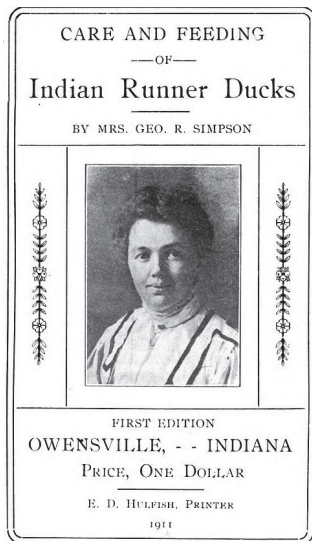


Figure 28. Care and Feeding of Indian Runner Ducks by Mrs. Geo. R. Simpson.

References

- ¹ Intel Arria 10 device data sheet. Intel, June 2018.
- ² AN 711: Power Reduction Features in Intel Arria 10 Devices. Intel, July 2018.
- ³ Intel Arria 10 Core Fabric and General Purpose I/Os Handbook. Intel, August 2018.
- ⁴ Virtex UltraScale FPGAs data sheet: DC and AC Switching Characteristics. Xilinx, January 2018.
- ⁵ AN 692: Power Sequencing Considerations for Arria 10 and Stratix 10 Devices. Intel, April 2018.
- ⁶ DMN1032UCB4 data sheet. Diodes Incorporated, January 2015.
- ⁷ Joachim Von der Ohe. Application Note: Pulse Load on SMD Resistors: At the Limit. Vishay Beyschlag, August 2015.
- ⁸ Gordon Moore. “Cramping More Components onto Integrated Circuits.” *Electronics Magazine*. Vol. 38, No. 8. April 19, 1965.
- ⁹ Gordon Moore. “No Exponential is Forever: But ‘Forever’ Can Be Delayed!” International Solid-State Circuits Conference, 2003.
- ¹⁰ Samuel Fullers and Lynette Millett. *The Future of Computing Performance: Game Over or Next Level?* National Academy of Science, Washington D.C.
- ¹¹ Karl Rupp. “40 Years of Microprocessor Trend Data.” February 2018.

Nathan Enger [nathan.enger@analog.com] is a mixed-signal applications engineer at Analog Devices in Colorado Springs, Colorado. Nathan has a B.S.E.E. and an M.S.E.E. from the University of Colorado and Colorado Technical University, respectively. He has 23 years of mixed-signal IC design experience, including high speed PLL and transceiver development at Ford, Intel, and Marvell Corporations.



Nathan Enger