Reducing System Power

Ten Steps to Minimize Power Consumption in Your System

Introduction

Lower system power consumption is a requirement for more applications than ever due to the dramatic increase in portable and battery-operated applications. Today, FPGA technology is being used more frequently in low power applications such as consumer and automotive electronics. This makes achieving lower system power an increasingly important challenge. FPGAs have been adopted widely in recent years due to advanced technology that lowered the unit price; however, the price reductions have come at the cost of higher power due to higher transistor leakage at smaller process geometries.

Minimizing Power Consumption When Designing with FPGAs

When creating a new design, it is important to consider the following: bill-of-materials costs, power, board size, time to market, security, and reliability. After prioritizing the initial requirements and determining if the system has specific power needs, the designer should consider several factors before selecting a specific FPGA to implement the design. To minimize total system power, perform the following steps before selecting an FPGA:

1. Evaluate total system power contributed by the FPGA and additional components required.

   In some cases, implementing solutions with a certain FPGA technology may require additional components for support functions. For example, memory or a microcontroller may be required for boot-up of an FPGA based on volatile memory (SRAM), whereas nonvolatile FPGA devices offer lower power single-chip implementation. Neglecting the power contributed by the additional components hides the true power consumed by the FPGA.
2. Profile your design. How long will the FPGA be running at high speed versus low speed or with stopped clocks?
   - Can burst mode processing at a higher clock frequency, but with more device "sleep" time, achieve the required system throughput?
   - Is it better to run the entire design at a lower clock frequency for longer periods of time?

   Most FPGA suppliers provide power analysis and prediction tools to help in this process. Be careful, however, as some tools can be overly optimistic.
3. Calculate power consumption for each product state.

   It is very important to account for power consumed by the device in all states over the product lifetime or expected battery operation time. At the very least, consider power-up, standby, idle,

dynamic, and power-down states. For example, an FPGA in a consumer handheld device with WiFi communication may have the following power profile: 5% active, 20% static, and 75% sleep.

Base calculations on datasheet and power estimator numbers, and ask how those numbers were obtained. Do the numbers also take into account silicon variations? Remember that what is measured on the bench today may vary significantly from what is shipped as a low power device tomorrow. Be careful when basing power calculations solely on measured data.

4. For each board component, calculate the worst-case static power consumption.

   Newer FPGA technologies may have significantly higher static power consumption than designers are aware of, particularly over extended temperature ranges. Make sure core, I/O, and any auxiliary power supplies are taken into consideration. When calculating static power consumption, P = I * V for each component, and power supply is crucial in determining the power consumed by the FPGA device.

5. Analyze the temperature and voltage changes expected over the product power profile.

   Effects such as heat and voltage changes over the product operation time can dramatically change the product power profile and need to be accounted for in the component power profile. For example, SRAM-based FPGA power consumption at 50ºC may be ten times or more than power consumption at room temperature.

6. Determine battery operation time for each system alternative.

   Further calculation can be done to determine the battery operation time for each system alternative (for example, high performance for short periods of time versus lower performance for longer periods of time). This analysis will show which alternative offers the longest battery operation time, satisfying the power target and improving customer satisfaction.

7. Optimize component power consumption using low power modes.

   Components designed for low power applications usually offer low power modes that are easy to use and put the device in sleep mode or lower standby power mode. The characteristics of each mode are device-dependent, as are the method and timing required to enter or exit the power mode. Some power saving modes require board considerations for implementation, so the design should be able to accommodate them. Some modes cannot be used due to complex implementation or a long period of time for the application to enter or exit the mode. For example, low power modes offered by SRAM FPGAs or SRAM Hybrid FPGAs require device reconfiguration. The power surge associated with configuration could reach one watt and last hundreds of milliseconds.

   Other FPGAs require the user to stop the incoming clock externally and prevent I/Os from toggling to achieve the low power specifications, which adds complexity and delay to the system.

8. Optimize component power consumption with design techniques.

   When creating clock regions using local or regional clock resources, use "enabled" logic to disable clock transitions in the system. User static RAM can sink excessive power; therefore, look for techniques to minimize RAM usage. I/Os can also sink a great deal of board power. Try to use LVTTL/LVCMOS standards and lower I/O voltages. Serial LVDS chip-to-chip data transfers using double data rate registers may save power over parallel off-chip buses.

   Check whether components can be integrated or functionality minimized to save even more on power. A bigger FPGA may be able to accommodate a soft MCU and save significant power over a two-chip solution, while reducing the toggle rate and total system resources used. Chip-to-chip communication can significantly increase system power consumption due to large capacitive loads on PCB traces.

9. Optimize system power consumption.

   To accommodate the power saving modes for all system components, board and software modifications must be done. This will optimize the system and minimize power consumption according to the calculated product profile. Additionally, sharing a common power rail with another component (i.e., using a low power 1.2 V FPGA with a 1.2 V–based MCU) may enable the elimination of a DC/DC converter, saving cost, power, and area.

10. It may take several iterations to achieve the lowest power possible. Zero in on the best system implementation and actual FPGA device.

Consider a design example using a programmable logic device component. We will compare an FPGA based on SRAM technology with an FPGA based on Flash technology.

Following step 1, we can see that using an SRAM-based FPGA forces us to add more components to the board or accommodate the FPGA's needs with resources on the board. More components and more system activity lead to more power dissipation.

In SRAM FPGAs, which are based on 90 nm, current leakage is dramatically higher than in Flash FPGAs by as much as a factor of 10, even though both types of technologies use 1.2 V core voltage.

For a product profile of 5% active, 20% static, and 75% sleep, it is clear that the static and sleep components determine the majority of power consumed by the device, assuming the active power (design-dependent) is comparable, since both Flash and SRAM FPGAs can operate with 1.2 V core voltage. Under typical conditions, a 1 million system gate SRAM device may consume 2.4 mW in sleep mode and 68 mW in static mode. The million gate Actel IGLOO™ AGL1000 device consumes 160 µW in static mode under typical conditions by utilizing its Flash*Freeze™ technology. Assuming 100 mW active power consumption and calculating all system stages, we see that the SRAM FPGA consumes more than 20 mW/hour, while the IGLOO device consumes 5.2 mW/hour. This is approximately one-fourth the power consumption of the SRAM FPGA.

The active component is design-dependent but can be considered as comparable; however, the static component contributes to higher active power on the SRAM FPGA device. In this example, Actel IGLOO devices show an advantage of over 2 orders of magnitude in static power consumption compared to the 1 million system gate, 90-nm SRAM solution. This is without accounting for the additional power consumed by the external components, which are required to assist the SRAM FPGA in configuration and brownout.

For your next design, use the most power-friendly FPGA technology to lower system power consumption, without compromising performance and cost.

For more information, visit our website at **www.actel.com**

www.actel.com

Actel Corporation

2061 Stierlin Court
Mountain View, CA
94043-4655 USA
Phone 650.318.4200
Fax 650.318.4600

Actel Europe Ltd.

Dunlop House, Riverside Way
Camberley, Surrey GU15 3YL
United Kingdom
Phone +44 (0) 1276 401 450
Fax +44 (0) 1276 401 490

Actel Japan
www.jp.actel.com
EXOS Ebisu Bldg. 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150 Japan
Phone +81.03.3445.7671
Fax +81.03.3445.7668

Actel Hong Kong
www.actel.com.cn
Suite 2114, Two Pacific Place
88 Queensway, Admiralty
Hong Kong
Phone +852 2185 6460
Fax +852 2185 6488

55900055-1/8.06