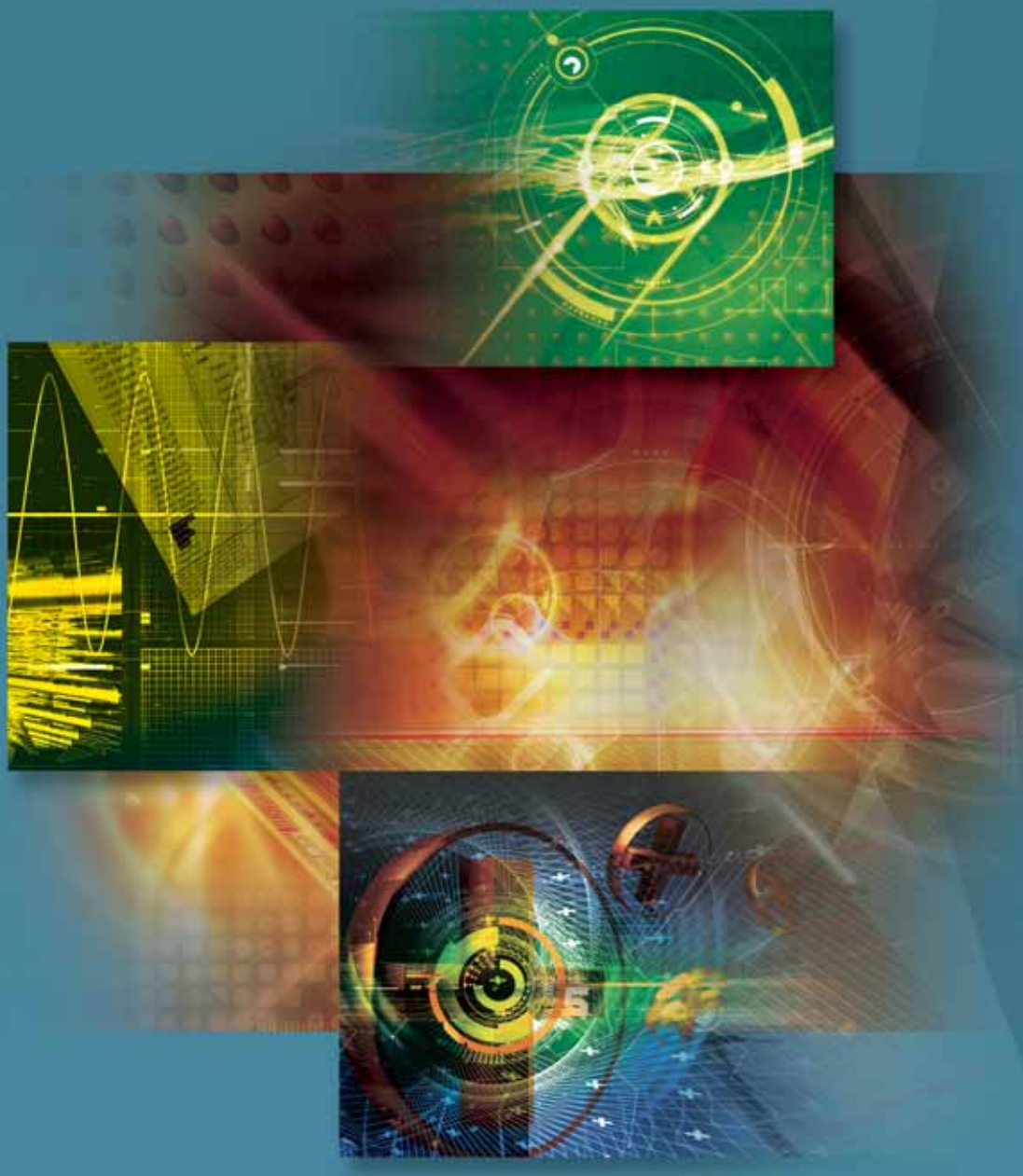
Issue 1
July 2005

Power Management

Solution Guide

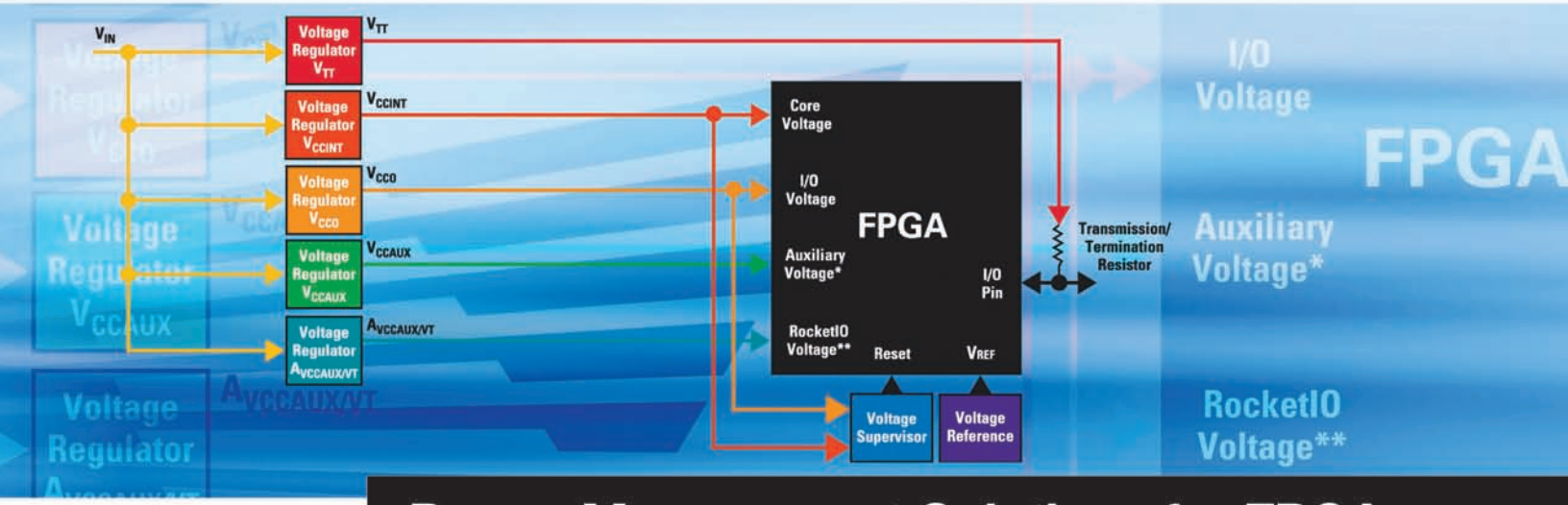


Solving Today's Hot Challenge

INSIDE

XILINX®

Support Across The Board.™



Power Management Solutions for FPGAs

National Devices supported:

- Voltage Regulators
- Voltage Supervisors
- Voltage References

Xilinx Devices supported:

- Virtex™
- Virtex-E
- Virtex-II
- Virtex-II Pro
- Virtex-4FX, 4LX, 4SX
- Spartan™-II
- Spartan™-IIE
- Spartan-3, 3E, 3L

National Semiconductor
The Sight & Sound of Information

XILINX®

Avnet Electronics Marketing has collaborated with National Semiconductor® and Xilinx® to create a design guide that matches National Semiconductor's broad portfolio of power solutions to the latest releases of FPGAs from Xilinx.

Featuring parametric tables, sample designs and step-by-step directions, this guide is your fast, accurate source for choosing the best National Semiconductor Power Supply Solution for your design. It also provides an overview of the available design tools, including application notes, development software and evaluation kits.

Go to em.avnet.com/powermgtguide to request your copy today.

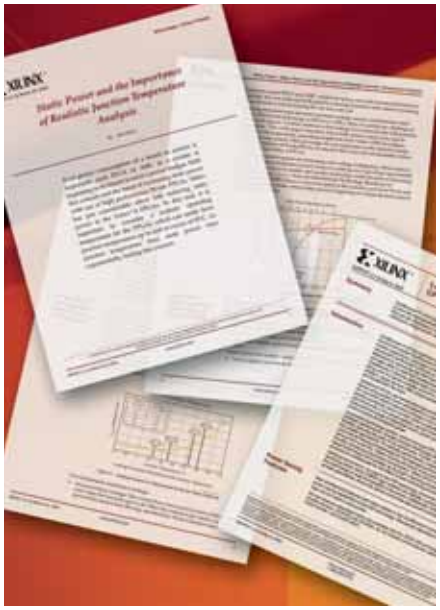
AVNET® electronics marketing

Enabling success from the center of technology™

1 800 332 8638
www.em.avnet.com

NEDA Supplier Authorized Distributor

CONTENTS



"FPGAs are being used increasingly in many applications, so reducing power consumption in FPGAs provides huge benefits to the system design."

ARTICLES

WHITE PAPERS

APPLICATION NOTES

Learn more about power management solutions from Xilinx at: *www.xilinx.com/xcell/power1/*

Power Management Solution Guide

EDITOR IN CHIEF Carlis Collins
editor@xilinx.com
408-879-4519

MANAGING EDITOR Forrest Couch
forrest.couch@xilinx.com
408-879-5270

ASSISTANT MANAGING EDITOR Charmaine Cooper Hussain

XCELL ONLINE EDITOR Tom Pyles
tom.pyles@xilinx.com
720-652-3883

ADVERTISING SALES Dan Teie
1-800-493-5551

ART DIRECTOR Scott Blair

Xcell Publications

Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124-3400
Phone: 408-559-7778
FAX: 408-879-4780

More Power to You

Welcome to the first edition of the Xilinx® *Power Management Solution Guide*. At Xilinx, we have heard a clear message from our customers that power is now a key issue in many system designs. No longer an afterthought, power must now be budgeted and managed just as much as the bill of materials cost or system performance. You have told us that meeting only two of these three key parameters just isn't good enough anymore.

With these thoughts in mind, we have assembled a collection of the most recent – and we hope useful – articles, white papers, and application notes dealing with system power management.

This issue includes several articles and white papers discussing the system power advantages and power management features of the Virtex™-4 FPGA solution. Virtex-4 FPGAs dramatically reduce power consumption compared to other FPGAs in all three key power areas – static power, dynamic power, and in-rush or startup current. They offer as much as 73% lower static power with the industry's first triple-oxide technology; as much as 86% lower dynamic power enabled by embedded IP blocks; and the industry's lowest in-rush current.

These savings are enabled with industry-leading technologies, including the industry-first 90 nm triple-oxide technology, high-performance embedded IP, and power-saving configuration circuitry. In addition to the chip technology itself, Xilinx provides comprehensive tools for power system design, including a Web-based power estimator and the post-implementation XPower tool included as part of the ISE™ design tool suite.

This issue also features a collection of application notes detailing some of the innovative power-saving features of the CoolRunner™-II CPLD. Long the industry's benchmark for low-power CPLDs, the CoolRunner-II family offers both power-saving technology and power-management features that make it the best choice for today's low-power, portable systems.

We hope you find this solution guide valuable. It is, however, just a sample of the information available to you about this topic. For access to all of the latest on power management, visit our website at *www.xilinx.com/xcell/power1/*.

Thank you,

Sandeep Vij
Vice President
Worldwide Marketing

Conquering the Three Challenges of Power Consumption

Why is power such an issue?

by Steve Sharp
Sr. Manager, Corporate Solutions Marketing
Xilinx, Inc.
steve.sharp@xilinx.com

As chip technology progresses to 90 nm and below, power becomes a burning issue in system design. At this node, leakage plays a more major role in total power; smaller interconnect geometries with new dielectric materials affect dynamic power as well.

According to Jordan Selburn of market research firm iSupply, "Leakage current – essentially insignificant at the 0.35 micron node and earlier – has become a major issue as transistors become increasingly leakier. Studies have shown that at the 90 nm node, leakage power can equal dynamic power consumption and even exceed it at the 65 nm node."

Another factor facing system designers is the tighter power budgets around which they must design. This is not limited to any single type of system, but does affect most designers. Large systems with many boards or modules, as well as portable and consumer products, all face power budgeting issues.

In large systems, power budgeting is typically done for the total system, as well as distributed power regulation on a per-board or per-module basis. With multiple power supplies now on every board, it is not a simple task to increase the power budget for one board without affecting the entire system's power distribution plan.

In line-powered consumer products, the goal is usually to use the smallest and least-expensive power supply possible to keep costs under control. Exceeding the capabilities of a particular model power supply by only a few percent can necessitate the use of a larger, more expensive supply, and this might be unacceptable in light of total system cost. Designers would rather design in more features to differentiate the product than to use a larger power supply.

In portable consumer products, the overwhelming goal is to extend battery life for as long as possible. For these products, longer battery life – both in active and standby modes – is a significant competitive advantage.

With all of these challenges, it's no wonder that power issues are sounding the alarm bells for system designers today. iSuppli's Selburn continues, "On the customer side, chip designers can consider architectural approaches such as parallel processing at reduced clock speeds to reduce dynamic power, or gated clocks that essentially turn off entire sections of the chip when they are not needed. Despite these techniques, power consumption remains a serious issue for a large portion of the core silicon market, an issue that is becoming worse, not better, with time."

System Design Challenges

There are three key areas of power usage and control challenging system designers today: static power, dynamic power, and in-rush power. Each presents different issues and requires different methods to calculate and manage power.

Static power is the power consumed by a device when it is in its quiescent condition with no input signals being exercised. It is also referred to as steady-state or standby power. In today's 90 nm technology devices, leakage currents in the transistors are the biggest contributors to static power. This is usually the key parameter of concern to designers of portable equipment because of its effect on battery life, especially for devices that spend large amounts of time in a standby condition waiting for input from the outside world.

Dynamic power is the power consumed during normal operation. It is also referred to as operating power. Dynamic power is dependant on operating signal frequency, interconnect capacitance, and operating voltage. Because the voltage dependency is a square function, the reduction in voltage when moving to 90 nm devices has substantially reduced operating power in many devices. However, for large, high-performance systems with high operating frequencies, dynamic power is still a significant component of total system power.

In-rush power is the power required at device power-up. It is also referred to as power-up or start-up power, or power-on surge power (or current). Some devices require many times more power to begin operation than they do during normal operation, thereby placing demands on system power supplies. In a consumer system with very tightly controlled power supply size and cost, ensuring that in-rush power is not more than normal operating power is a key design goal.

Higher power levels can affect both manufacturers and end-customers alike, in four key areas:

- Performance. Higher power levels in a chip can limit device and end-system performance by forcing a lower system clock rate to stay within the system power budget.
- Reliability. As power goes up, so does the threat of brown-out and latch-up from high power-on surge. In addition, higher failures-in-time (FIT) rates will be expected due to higher device operating temperatures.
- Cost. As mentioned previously, higher power equals higher cost in the system because of larger, more expensive power supplies and thermal management components such as fans and head sinks.
- End-customer operating expenses. Higher power also impacts end users in the form of higher power bills (which can be significant for large systems) and shorter battery life for portable products.

Higher power levels in a chip can limit device and end-system performance by forcing a lower system clock rate to stay within the system power budget.

How Xilinx Helps Manage System Power

Virtex-4 FPGAs

With a significant reduction in power consumption over that of the competition, the new Virtex™-4 platform FPGAs offer significant benefits for system design, including reduced thermal concerns, easier power-supply design, lower cost power supply, and higher system reliability. Virtex-4 FPGAs dramatically reduce power consumption when compared to other FPGAs in all three key power areas:

- As much as 73 percent lower static power with the industry's first triple-oxide technology
- As much as 86 percent lower dynamic power enabled by embedded IP blocks
- Negligible in-rush current with unique power-saving configuration circuitry

This is enabled with industry-leading technologies such as 90 nm triple-oxide technology, high-performance embedded IP, and power-saving configuration circuitry.

Xilinx also provides comprehensive tools for power system design: Virtex-4 datasheet and user guide; a web-based power estimator; and XPower, included in ISE™ software.

Virtex-4 devices handle the three types of power usage and control in the following ways:

- Static power. As process geometries shrink to 90 nm and lower, the industry expects higher leakage and higher static power when channel length decreases. Working with fab partner United Microelectronics Corp., Xilinx solved this problem by using triple-oxide technology in the Virtex-4 90 nm process, which reduces leakage current significantly. Two-oxide thicknesses are widely used in the industry today, with a thin oxide in the core and thicker oxide in the I/O area. Virtex-4 devices add a third medium-thick oxide transistor used for certain functions in the FPGA. The result is 50% lower static power than that of Virtex-II Pro FPGAs. Other FPGA vendors have gone the other way when migrating to a 90 nm process, with static power increasing more than 2X compared to 130 nm devices.
- Dynamic power. New and existing Virtex-4 embedded functions lower dynamic power by 5 to 20x compared

to Virtex-II Pro FPGAs. This results in as much as 86% lower dynamic power than that of other 90 nm FPGAs.

Note these specific examples:

- PowerPC™ – as much as 86% power reduction
- Block RAM – as much as 82% power reduction
- DSP – as much as 23% reduction with XtremeDSP™ slice
- Ethernet MAC – as much as 83% power reduction
- Logic – although Virtex-4 devices consume similar dynamic power-per-logic cell when compared to other FPGAs, the embedded IP blocks often allow fewer general-purpose logic cells to be used. For example, when building a source-synchronous I/O (SSIO) interface, the new ChipSync™ block reduces the number of logic cells used.

• In-rush power. Other high-performance 90 nm FPGAs have exhibited levels of in-rush power more than four times that of Virtex-4 FPGAs. In Virtex-4 devices, by spending considerable time designing very power-efficient configuration logic, Xilinx has been able to keep in-rush power within 15-20% of the static power requirements and below typical operating power. This removes the need to use a larger power supply just to address in-rush current.

CoolRunner-II CPLDs

When Xilinx designed the CoolRunner™-II family of low-power CPLDs, our goal was to deliver one of the industry's lowest power levels for a programmable logic device. These devices have standby current requirements of less than 20 µA, making them ideal for battery-powered portable devices. Other CPLDs claiming to be low power have standby power 100 to 1000x higher, affecting battery life so significantly that they are unsuitable for portable applications.

The static RealDigital technology used in the logic of CoolRunner-II devices does away with power-hungry sense amplifiers and delivers low dynamic power as good as any other device available today.

In addition to these advantages in the basic circuit design and process technology, CoolRunner-II devices also offer power-management features unique to the CPLD industry, including a DataGate feature to reduce effective logic usage in the device and clock management and input hysteresis features to reduce internal operating frequencies and dynamic power.

Spartan-3 FPGAs

Our customers have told us that in today's cost-conscious consumer products, being forced to put in a bigger supply just to supply a high power-on or in-rush current is not a viable option for their system designs.

Attention to detail when designing the Spartan-3 configuration logic has yielded devices where the maximum quiescent power alone is guaranteed to be sufficient to power up the device. Spartan-3 devices have no in-rush current or power specification. When using these low-cost devices, you can focus on the product features and design without worrying about increased system cost because of high in-rush power requirements.

Power Management Tools

Web Power Tools are pre-implementation tools that estimate a design's power consumption based on the expected utilization of device resources, operating frequencies, and toggle rates.

Once you have implemented your design in the Xilinx software tools, you can use XPower to accurately estimate the power consumption. Actual power consumption must be determined in-circuit under the appropriate operating conditions.

Web Power Tools

The intuitive interface guides you through the steps of the data-entry process and ensures the most accurate estimates possible. The equations and values used by Web Power Tools are based on device characterizations for the family. Web Power Tools are available for the Virtex-4, Virtex-II Pro, Virtex-II, Virtex/Virtex-E, and Spartan-3 FPGA families, as well as CoolRunner-II CPLDs.

XPower

XPower is the first power-analysis software available for programmable logic design, allowing the analysis of total device power, power-per-net, routed, partially routed, or un-routed designs.

Power Management Hardware

Tools for managing power are not just of the software variety. Power-management chips from National Semiconductor, Intersil, Texas Instruments, and Linear Technology are available to make the job of supplying the multiple supply voltages needed by today's FPGAs easier, and they can be valuable companions to Virtex-4 or Spartan-3 devices. Their individual capabilities are highlighted in this issue of the *Xcell Journal* in the following pages.

Conclusion

To conquer the key challenges of power consumption, it takes a combination of good product design, proper device technology, and tools that let you take control of system power management.

Xilinx is an industry leader in power management and now offers many advantages within its programmable solutions:

- Virtex-4 FPGAs consume 1 to 5W less power than competing 90 nm FPGAs.
- Spartan-3 FPGAs are the one of the only low-cost FPGAs in the industry to eliminate power-on surge. In these devices, the maximum quiescent power alone is sufficient to guarantee device power-up.
- CoolRunner-II CPLDs are the world's lowest power CPLDs, ideal for even the most power-critical portable applications.
- Xilinx offers a comprehensive suite of power management tools, from Web Power Tools to the XPower analysis tool integrated into the ISE environment.

You can find comprehensive information on power consumption and solving key power challenges with Xilinx devices, tools, and solutions at *www.xilinx.com/power/.*

The Virtex-4 Power Play

The latest Xilinx FPGA offers revolutionary power innovations.

by Matt Klein
Sr. Staff Engineer, Applications Engineering,
Advanced Products Division
Xilinx, Inc.
matt.klein@xilinx.com

Device power consumption is a primary issue in the semiconductor industry – as process technologies get smaller and faster, they normally consume more power, putting power concerns and performance at odds. The new Virtex-4™ FPGA family from Xilinx® employs innovative architectural features and clever IC design techniques that dramatically reduce power consumption, without compromising performance. This bucks expected trends normally associated with the reduced feature sizes of 90 nm process technology.

In this article, we'll explore how Xilinx IC designers achieved remarkable power efficiency in the high-performance Virtex-4 FPGA.

Components of Power Consumption

There are two main components to power consumption: static and dynamic. Static or quiescent power is mainly dominated by transistor leakage current. When this current is listed in data sheets, it is listed as I_{CCINTQ} and is the current drawn through the V_{CCINT} supply powering the FPGA core.

Dynamic or active power has components from both the switching power of the core of the FPGA and the I/O being switched. The dynamic power consumption is determined by the node capacitance, supply voltage, and switching frequency and governed by the basic formula $P=CV^2f$.

Both static and dynamic power have been significantly reduced in Virtex-4 devices, even when compared to Virtex-II Pro™ devices.

Dramatic Power Reduction

The Virtex-4 product family has reduced power consumption in several key areas. The power-per-CLB has been cut in half, with static power reduced by 40% and dynamic power reduced by 50% when compared to the 130 nm Virtex-II Pro FPGA and other 90 nm FPGAs. Furthermore, certain hard-logic silicon functions in the Virtex-4 FPGA reduce power consumption by 80-95%, a whopping factor when compared to the same functions implemented in configurable logic blocks and programmable interconnect routing.

Additionally, comprehensive power planning tools are available to help you get an idea, up front, of power consumption for your Xilinx FPGA under its operating conditions.

Reduced Power Consumption Benefits

Reduced power consumption benefits cut across a few areas of product design in reduced thermal concerns as well as eased power supply design (see Figure 1).

- Reduced thermal concerns – When you reduce power consumption in a device or system, you use smaller heat sinks, or no heat sinks at all in some cases. You also have simpler thermal system design from the point of view of reducing airflows and fan size needs.
- Easier power supply design – You can also use smaller supply circuitry and reduce the number of components in the power supply. Using less PCB space allows you to reduce the cost of the power system. Plus, by not having your device consume as much power, you can achieve higher reliability by lowering the temperature of the FPGA die.

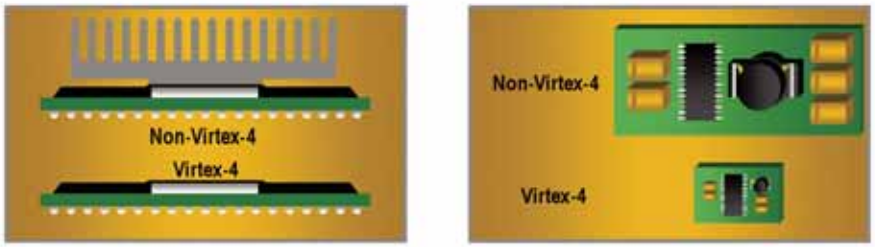


Figure 1 – Virtex-4 devices reduce thermal concerns and simplify power supply design.

Static Power Trends in 90 nm Technology

The reduction in transistor size in 90 nm technology has several effects on power consumption. The biggest potential problem is in the area of static power.

Scaling Trends for Static Power

As we mentioned earlier, static power is dominated by transistor leakage current. Unfortunately, channel leakage increases as transistor size decreases. This is especially true for low V_T transistors where V_T refers to voltage threshold between the gate and drain.

Low V_T transistors are the fastest transistors – the ones with the shortest turn-on and propagation delay – that IC designers use inside the FPGA when the highest speed performance is needed. Regular V_T transistors are also used when less performance is acceptable, but this only helps so much with leakage.

Figure 2 shows that leakage goes up dramatically when moving from 130 nm to 90 nm technology. The Virtex-II Pro device uses 130 nm process technology, whereas the new Virtex-4 device uses 90 nm process technology.

Triple-Oxide – The Savior of Static Power

Triple-oxide simply means that we use a third thickness of oxide in making some of the transistors in the FPGA (two oxide thicknesses are used in devices like the Virtex-II Pro FPGA). Most transistors in the past had a thin oxide layer. Within those transistors could be low V_T, regular V_T, NMOS, or PMOS transistors. Thick-oxide transistors are mostly used for I/O drivers and a few other functions.

Oxide deposition thickness is a very stable and controllable process in the semiconductor industry because it depends on temperature, concentration, and exposure time. Figures 3a and 3b show the Virtex-4 transistor with the middle oxide thickness used in the triple-oxide process. You may notice that the oxide thickness is still very, very thin, but this thicker oxide transistor has much lower leakage than the standard thin-oxide low V_T and regular V_T transistors used in Virtex-II Pro FPGAs and in various parts of Virtex-4 FPGAs.

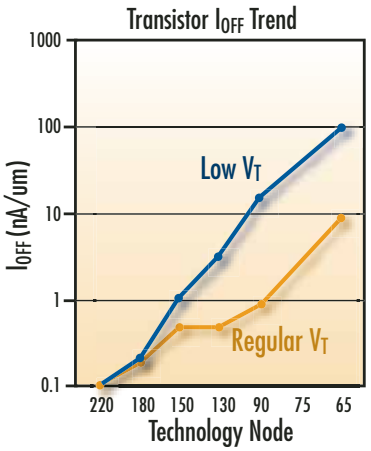


Figure 2 – Transistor leakage trends due to process scaling

Why Doesn't Everyone Use Triple-Oxide?

If triple-oxide is such a great process, why don't other companies like Intel™ or IBM™ use it in their own ASICs?

They probably would if it benefited them. The reason they don't is that all of their transistors need to run at speed; hence, they must use the low V_T leakier transistors for everything. FPGAs can have many different transistor types, which can be selected for function, power, or performance.

FPGAs can use different transistor types for different functions, and Xilinx designers have accomplished this balance.

Optimizing Performance and Leakage

Our IC designers have many things that they can do to adjust the mix to optimize for certain factors. The Virtex-4 FPGA is the first Platform FPGA designed for high speed and low power.

Low V_T transistors are used only where necessary for maximum speed, while the middle thickness of oxide from the triple-oxide process may be used for less aggressive performance with very low leakage. You may use different sizes and types of transistors for performance and function. Combinations are also possible, such as small and medium-sized low V_T fast transistors and small and medium-sized middle oxide thickness transistors. It is not a one-size-fits-all procedure.

Xilinx IC designers were given a directive to reduce power, among other things, in the Virtex-4 platform while maintaining the highest system performance. These transistors are used across the various FPGA functions of LUTs, I/O, interconnect, and configuration memory cells. Even within a given FPGA function, all transistors don't need to be the same, and that is up to the Xilinx IC designers (see Figure 4).

The surprising result of this balancing is that the overall static current in Virtex-4 devices with 90 nm process is reduced by 40% when compared to Virtex-II Pro devices with 130 nm process. Table 1 shows a chart of the weighted average changes to the transistors in the Virtex-4 die compared to Virtex-II Pro die, which allows you to arrive at the reduced transistor leakage in the Virtex-4 FPGA.

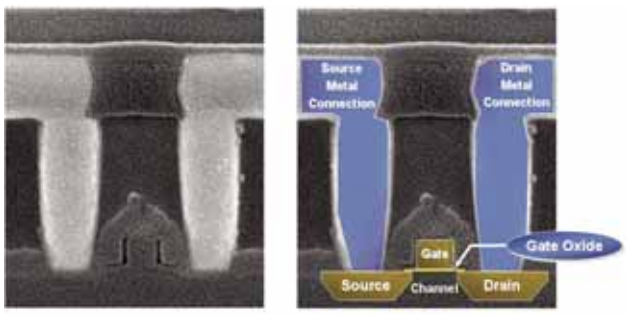


Figure 3a, 3b – Middle oxide thickness Virtex-4 transistor used in triple-oxide process and with highlighted portions of the transistors

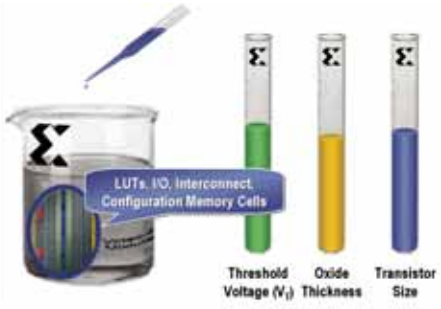


Figure 4 – Optimal transistor mix for minimizing leakage and maximizing performance

Dynamic Power Reduction

Static power reduction, while dramatic, is not the only power winner that you can take advantage of. Dynamic power is also reduced by 50% when compared to Virtex-II Pro FPGAs.

The dynamic power in the FPGA is governed by the following equation:

$$P_{Dynamic} = FPGA_{Core}(CV^2f) + FPGA_{I/O}(CV^2f)$$

The Virtex-4 family of FPGAs has the following:

- Reduced FPGA core dynamic power
  - Internal operating voltage is the dominant factor
  - Secondary scaling by frequency (f) and node capacitance (C)
- Constant FPGA I/O dynamic power
  - Unchanged voltage swing ($V_{I/O}$), toggle rate (f), and pin/pad capacitance (C) for a given I/O standard

So you can see that we may be able to have an effect on dynamic power inside the device, but that dynamic power consumed by I/O switching remains unchanged.

When we go from the 130 nm process of the Virtex-II Pro FPGA to the 90 nm process of the Virtex-4 FPGA, the internal supply voltage changes from 1.5V to 1.2V. This reduces the dynamic power consumption for every internal transistor by 36% ($1-\left[\frac{1.2}{1.5}\right]^2$) of that in the Virtex-II Pro FPGA.

Additionally, the FPGA internal composite capacitance is reduced in the Virtex-4 FPGA. This internal capacitance comprises transistor parasitic capacitances and trace-to-metal and trace-to-trace capacitances for the interconnecting metal traces. Figure 5 shows the capacitance involved relative to their structures.

As mentioned earlier, dynamic power is related to the bulk capacitance and internal voltage levels being switched, $P=CV^2f$. All things being equal, having a lower internal capacitance for the interconnects would be a benefit for dynamic power and reduced resistor-capacitor delay, but other factors contribute to interconnect capacitance, such as distance above the metal plane, interconnect width, and interconnect length.

Additionally, other parasitic capacitances such as gate-to-drain and gate-to-source are also part of the equation. Total capacitance for a path is based on a complex combination of parasitic capacitance

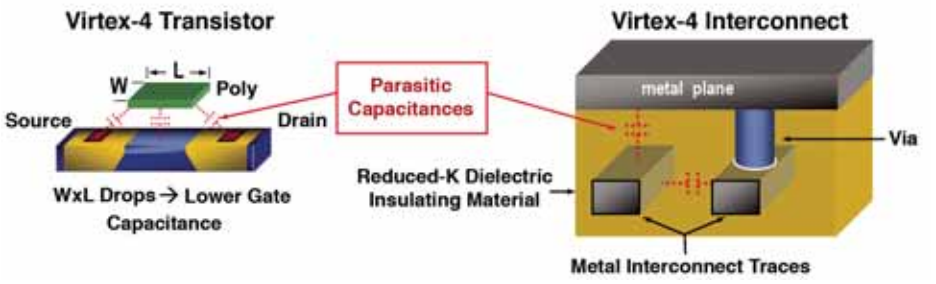


Figure 5 – Internal FPGA capacitance comprises parasitic transistor and interconnect capacitances

Does low-K reduce power? Low-K refers to the dielectric insulating material between the metal traces in the FPGA. Lower K dielectric insulating layers do reduce internal capacitances per unit trace length, but "low-K" is a relative term. Xilinx has reduced-K-insulating materials, and in the past has used low-K itself; we may do so again in the future.

Parameters	Virtex-II Pro	Virtex-4	Change
Channel Width Ratio	1.00	0.64	-36%
Channel Length Ratio		0.71	-29%
Leakage Current per Unit Width Ratio		1.14	+14%
Leakage Current per Transistor		0.74	-26%
V_{CCINT} Ratio		0.80	-20%
Static Power per Transistor Ration ($I_{LEAKAGE}$ * V_{CCINT})		0.59	-41%

Table 1 – Overall weighted average transistor leakage and parameter comparisons for 90 nm Virtex-4 transistors relative to 130 nm Virtex-II Pro transistors

in the transistors; the architecture of the interconnect paths and actual path lengths; and the number of hops through interconnect switches. Xilinx has reduced the overall capacitance for those components in the Virtex-4 FPGA.

The overall effect is mostly due to reduced gate capacitance and lowers capacitance by 20% for Virtex-4 FPGAs when compared to Virtex-II Pro FPGAs. Table 2 shows a dynamic power reduction of 50% for the Virtex-4 FPGA when compared to the Virtex-II Pro FPGA. We have a 23% reduction in dynamic power when running at a 50% higher frequency.

Because the Virtex-4 FPGA is a much higher performance device than the Virtex-II Pro FPGA, you may need to operate it at higher clock speeds to meet newer demanding performance targets that could never be achieved in previous systems.

Parameters	Virtex-II Pro	Virtex-4	Change
V_{CCINT}	1.5	1.2	-20%
C_{TOTAL} (rel.)	1.0	0.8	-20%
f_{MAX} (rel.)	1.0	1.5	+50%
Power at Same Frequency	2.25	1.15	-49%
Power at f_{MAX}	2.25	1.73	-23%

Table 2 – Chart showing changes in internal FPGA in Virtex-4 devices compared to Virtex-II Pro devices and the effect on dynamic power

Parameters	Virtex-II Pro	Virtex-4	Logic Slice Reduction	Logic Slice Power Reduction
QDR II SRAM Interface	550 slices	125 slices	77%	89%
SPI-4.2 Core	5000 slices	3900 slices	22%	61%

$$\text{Logic slice power reduction} = 100^{*}\left(1 - 0.5\,\frac{\text{Virtex-4 slice count}}{\text{Virtex-II Pro slice count}}\right)\%$$

Note: The factor of 0.5 above comes from the fact that Virtex-4 power per slice is 1/2 of the Virtex-II Pro power per slice because of the 50% dynamic power reduction in Virtex-4 devices compared to Virtex-II Pro devices.

Table 3 – QDR II SDRAM and SPI-4.2 core benefit in reduced power consumption from significant logic cell reduction due to new Virtex-4 ChipSync block

Embedded Blocks

Another major area of improvement in power consumption is in the area of embedded functions. This has always been a strength in Xilinx FPGAs, but it is more so in the Virtex-4 FPGA, even when compared to the feature-rich Virtex-II Pro FPGA.

In Virtex-4 FPGAs you can take further advantage of both static and dynamic power reduction by using the embedded functions, which are built as hard-logic functions.

When embedded functions are implemented as hard-logic functions instead of in configurable logic blocks and programmable interconnects, there is a lot less static and dynamic power consumed. This is because far fewer transistors are used for hard, fixed logic than for programmable logic. Additionally, there are no transistors needed to make connections for interconnects in the embedded functions, because there are no programmable interconnects.

Xilinx has carefully studied some of the functions that engineers like you have struggled with and that we have also found tedious to implement within the FPGA programmable logic. The new embedded functions lower power by 80-95% compared to their configurable logic blocks and routed counterparts in programmable silicon.

Comprehensive Power Planning Tools

Another useful thing in planning power is that Xilinx data sheets show you both typical and maximum power consumption numbers. Maximum numbers are for worst-case process, temperature, and voltage, but many designers are very happy to work with typical numbers, depending on their application and the number of parts being used in one system.

One additional very useful thing that you can take advantage of in planning for power consumption in Xilinx FPGAs are power planning tools. Xilinx web power tools are available for estimating power early in the design cycle. Also, as part of the Xilinx design flow, XPower looks in more detail at a mapped or routed design. Both can be found, along with power application notes, by searching the Xilinx website for the phrase "Xilinx Power Tools."

Conclusion

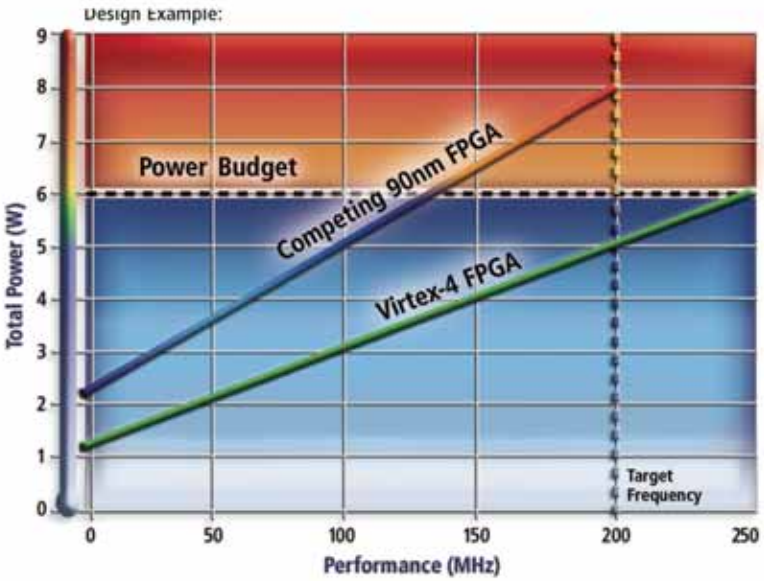
Xilinx has made profound improvements in both static and dynamic power in the Virtex-4 90 nm family of FPGAs when compared to Virtex-II Pro FPGAs – and (we believe) in comparison to our competitors. We have done this through a multi-pronged, purposeful approach in the areas of reduced leakage current, reduced dynamic power consumption, and embedded functions, without compromising performance. These, along with comprehensive power planning tools, make the Virtex-4 device an excellent choice for a high-performance FPGA system.

For more information about power consumption in Virtex-4 and other Xilinx FPGAs, visit *www.xilinx.com/products/design_resources/design_tool/grouping/power_tools.htm.*

Virtex-4 Embedded Functions and Reduction of Dynamic Power

- PowerPC – 50% power reduction compared to Virtex-II Pro PowerPC
- – 10:1 power reduction over FPGA fabric-built version
- DSP – XtremeDSP™ slice greatly reduces logic cells, which previously needed many filtering functions
- – 20:1 power reduction over Virtex-II Pro separated multiply/accumulate functions
- SSIO – New ChipSync™ block reduces logic cell count for SSIO (source synchronous I/O) designs
- – Significant logic cell savings for various memory and networking interface designs leads to reduction in overall power up to 9:1 for selected designs (see Table 3)
- Embedded Ethernet MAC(s) – No need to use logic and interconnect for MAC function, which saves >3,000 logic cells for the Xilinx implementation
- FIFO – SmartRAM™ memory includes built-in FIFO controllers, which can save hundreds of logic cells per FIFO and greatly simplify design as well

With Competing FPGAs, Power has become a Burning Issue.



Design Example: LX60 vs. 2S60, Target Frequency = 200 MHz, Worst-case process.
20K LUTs, 20K Flip-Flops, 1Mbit On-Chip RAM, 64 DSP Blocks, 128 2.5V I/Os
Based on Xilinx tool v4.0 and competitor tool v2.1
For higher density devices, achieve up to 5W lower power

Get 1-5W lower power per FPGA, only with the Virtex-4 family

Check the specs for yourself at realistic operating temperatures ($T_j = 85^{\circ}C$). Different logic architecture or dielectric just won't do it. No competing FPGA comes close to Virtex-4 for total power savings–take it to the lab and see.

- **73% lower static power**
- **Up to 86% lower dynamic power**

Unique Triple-Oxide Technology & Embedded IP

At the 90nm technology node, power is the next big challenge for system level designers. An inferior device can suffer leakage, dramatic surges in static power, and thermal runaway. That's why we designed our Virtex-4 FPGAs with Triple-Oxide Technology™, embedded IP, and power-saving configuration circuitry. Now you can meet your performance goals, while staying within the power budget.

Visit ***www.xilinx.com/virtex4/lowpower*** today, and get the right solution on board before your power issues start heating up.

XILINX®
The Programmable Logic Company℠

www.xilinx.com/virtex4/lowpower

View The TechOnLine Seminar Today

Breakthrough Performance At The Lowest Cost

WP223 (v1.1) May 12, 2005

Power vs. Performance: The 90 nm Inflection Point

By: Anil Telikepalli

The debate over which high-performance 90 nm FPGA has the lowest power is "heating up," and for good reason. The industry has crossed a critical inflection point at 90 nm, where performance competes with power and thermal budgets. Customers want as much performance as possible; increasingly, however, the decision about which FPGA to use is based on which device consumes the least amount of power. This white paper discusses performance versus power consumption in 90 nm FPGAs and how the Virtex™-4 family provides the best of both worlds: high performance and low power consumption.

Introduction

Excessive power is expensive in many ways. It creates the need for special design and operational considerations – requiring everything from heat sinks to fans to sophisticated heat exchangers. Even the cost of building larger power supplies must be taken into consideration. Overall, increased power requires more of everything, including: more area on the PCB, a larger chassis, more floor space, and larger air conditioning systems. The costs continue to compound.

Perhaps the most critical issue is the effect excessive power can have on reliability. As the junction temperatures rise, transistors consume more power, thereby further increasing the temperature of the device. Continuously operating systems with junction temperatures running from 85°C to over 100°C increases reliability issues.

Fortunately, Xilinx encountered the first evidence of this 90 nm inflection point in the early development stages of Spartan™-3 FPGAs, the first Xilinx FPGA family for the 90 nm process. Xilinx began immediately developing new ways to cope with the inherent power issues posed by the 90 nm process. Consequently, when the higher-performance Virtex-4 family was introduced in September 2004, Xilinx was confident that the new family would simultaneously deliver the best of both worlds – the highest performance and lowest power consumption in a 90 nm FPGA.

Reducing Power in FPGAs

The Triple Challenge

There are two major components to power consumption in FPGAs: static power, and dynamic power. Inrush current is another factor that can occur when the FPGA is powered on. Each component poses a unique challenge. For the 90 nm FPGA, the most challenging component is static power.

Static Power

Static power consumption occurs as a result of leakage current in the transistors that comprise the FPGA. As transistors get smaller (with each new process), their leakage current increases. This principle is one of the major reasons the 90 nm process crosses a major inflection point (Figure 1).

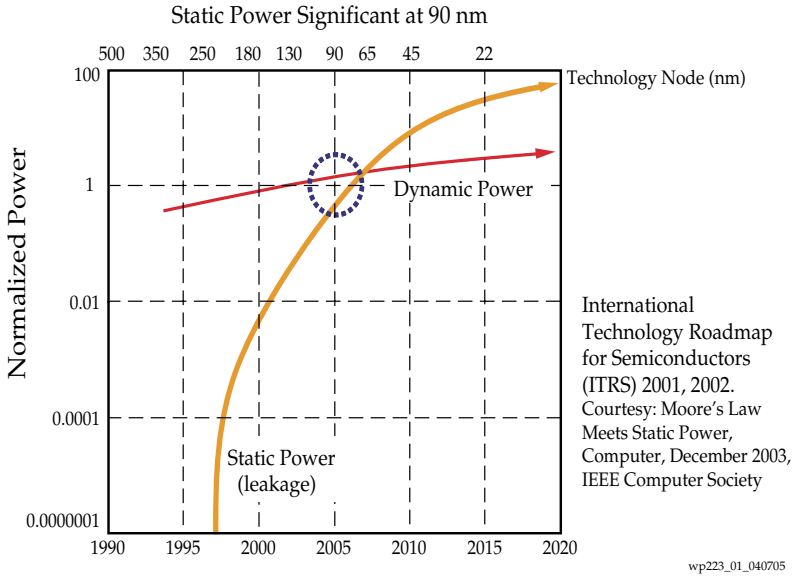


Figure 1: **90 nm Inflection Point**

For the first time, static power is threatening to eclipse dynamic power as the component responsible for the greatest amount of total power consumption in an FPGA. This is partly due to the fact that as processes get smaller, the core voltage decreases; consequently, the rate of increase in dynamic power drops, despite the increase in frequency that regularly accompanies a new process. In contrast, below 0.25 microns static power has grown exponentially with each new process.

This is where the inflection point really becomes a critical factor for the FPGAs and where Xilinx has established a substantial lead. Smaller transistors are faster, but they leak more. However, unlike ASICs, ASSPs, and microprocessors, Xilinx FPGAs do not need all of their transistors to switch at maximum speed. A substantial number of transistors comprise the configuration memory cells used to select logic and routing, and pass transistors used to implement the programmable interconnect routing. Configuration memory cells do not need to be fast, and programmable interconnect transistors only need to be fast from source to drain and not under gate control. These factors allow leakage to be reduced without compromising performance.

Virtex-4 FPGAs take advantage of the abundance of these particular types of transistors to incorporate a new process approach called "triple-oxide technology" to solve the static power problem. Figure 2 shows how Virtex-4 FPGAs consume 50% lower static power than its predecessor, the 130 nm Virtex-II Pro FPGAs.

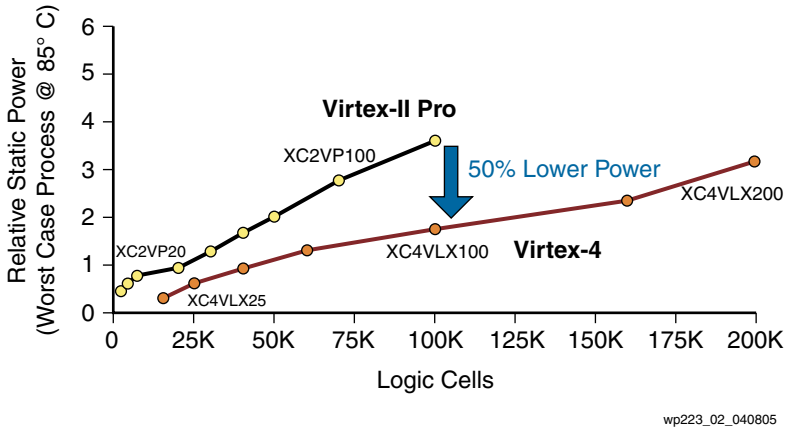


Figure 2: **The use of "triple oxide" technology reverses the trend: the Virtex-4 device actually consumes less static power than its 130-nm predecessor.**

What is Triple-Oxide Technology?

For many years, Xilinx and other semiconductor vendors have used two gate-oxide thicknesses: a standard thin layer used for the vast majority of transistors and a thick oxide layer for I/O drivers. With the introduction of Virtex-4 FPGAs, Xilinx has utilized *triple-oxide*, which refers to a third thickness of gate oxide used in making the configuration memory cells and pass transistors.

Oxide deposition thickness is a very stable and controllable process in the semiconductor industry. Semiconductor manufacturers can accurately set oxide thickness by choosing temperature, concentration, and exposure time. Figure 3 shows the Virtex-4 transistor with the middle oxide thickness used in the triple-oxide process. Although this third oxide layer is still very thin, these transistors exhibit substantially lower leakage than the standard thin-oxide low V_T and regular V_T transistors used in Virtex-II Pro FPGAs and in various other parts of Virtex-4 FPGAs.

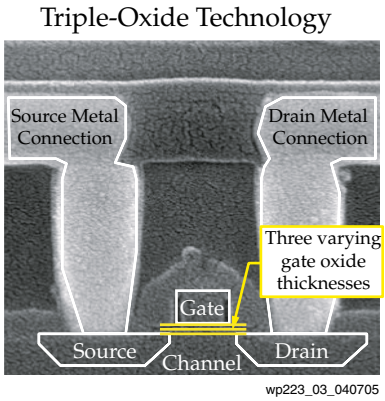


Figure 3: **Virtex-4 Transistor Middle Oxide Thickness**

In addition to using triple-oxide, Xilinx optimized a number of other transistor parameters to balance performance and leakage across I/O, configuration memory, interconnect pass transistors, as well as logic and interconnect buffers (Table 1).

Table 1: **Triple-Oxide Technology and Other Transistor Parameters Used for Power Optimization**

Function	Oxide Thickness	Voltage Threshold (V_T)	Channel Length	Speed	Leakage
I/O	Thick	High	Longest	Fast	Lowest
Config Memory	Medium	Medium	Long	Slow	Low
Interconnect Pass Gates	Medium	Low/Medium	Short	Fast Source→Drain	Low
Logic and Interconnect Buffers	Thin	Low	Short	Very Fast	Medium/High

The use of triple-oxide technology in Virtex-4 devices dramatically reduced the static power component for the entire family.

Other optimizations were made to reduce leakage and hence, static power. These include enhancements to gate length, V_T and oxide thickness. The combination of these effects on leakage and performance is shown in Figure 1.

As mentioned earlier, a comparison of relatively equivalent devices from both the 130 nm Virtex-II Pro family and the 90 nm Virtex-4 family reveals 50% less static power consumed by the Virtex-4 device (Figure 2). This is the first time in FPGA history that static power decreased when moving to a new smaller process. In contrast, Stratix II FPGAs consume two times more power than previous 130 nm Stratix FPGAs (Figure 4).

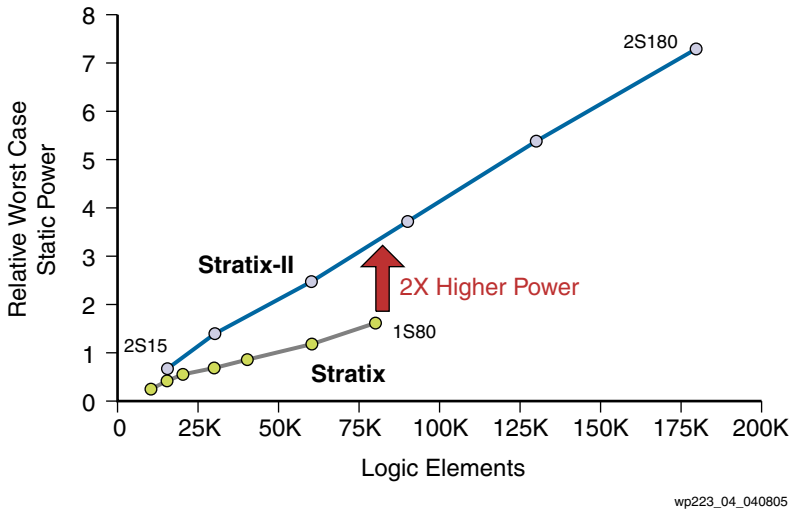


Figure 4: **Stratix-II Exhibits a Typical Increase in Worst-case Static Power Consumption Expected in Transitioning to the 90 nm Process – Without *Triple-Oxide*!**

Dynamic Power

The three contributing elements to dynamic power in the FPGA core are voltage, frequency, and parasitic capacitance. Fortunately, as previously mentioned, the core voltage and capacitance decrease with each new smaller process, which lowers dynamic power. If designs require higher operating frequencies, dynamic power increases. The well-known formula for dynamic power that applies here is:

$$P = CV^2f \qquad \text{Equation 1}$$

In addition, dynamic power is proportional to the data toggle rate.

One major area of improvement in power consumption is in the area of embedded functions. This has always been a strength in Xilinx FPGAs, but it is more so in the Virtex-4 FPGA, even when compared to the feature-rich Virtex-II Pro FPGA.

When embedded functions are implemented as hard-logic functions instead of configurable logic blocks and programmable interconnects, less static and dynamic power are consumed. This is because far fewer transistors are used for hard, fixed logic rather than programmable logic. Additionally, transistors are not needed for interconnects in the embedded functions, because there are no programmable interconnects.

These hard IP cores occupy far less real estate, deliver much higher performance, and consume 80-95% less power than soft IP versions of the same functions. In addition, by making these hard IP cores programmable and parameterizable, Xilinx provides designers with the flexibility they have come to expect from an FPGA. No other FPGA vendor provides so many hard IP cores for common functions.

Functions that Xilinx provides as hard IP cores in Virtex-4 FPGAs include:

- 450 MHz PowerPC™ processors for all microcontroller and embedded processing applications with an APU interface for hardware acceleration. A soft implementation would invariably consume several thousand look-up tables (LUTS) and flip-flops (FFs). Virtex-4 PowerPC processors also contain their own dedicated 16 Kbyte instruction and 16 Kbyte data cache. In addition to providing much better performance, these optimized cache memories consume much less power than building the equivalent-sized cache out of internal FPGA block memory.
- 500 MHz XtremeDSP™ slice for simple math and filters to complex high-performance DSP functions. With 40 op-modes and the ability to cascade multiple

slices without extra logic, each DSP slice functionality implemented as a soft core would cost anywhere from tens to hundreds of LUTs and FFs.

- 500 MHz Digital Clock Managers (DCM) and Phase Matched Clock Dividers (PMCD) that support clock synthesis, clock management, and phase matching. Unlike embedded PLLs, DCMs and PMCDs do not require special supply rails.
- ChipSync™ block in every I/O with built-in SERDES and data-alignment function to simplify source-synchronous interfaces in memory, networking, and telecom applications. Implemented as soft IP, each ChipSync block control circuitry would consume tens to hundreds of LUTs and FFs.
- 622 Mb/s - 10.3125 Gb/s RocketIO™ transceivers with built-in Physical Coding Sublayer (PCS) and Physical Media Attachment (PMA). With support for 8B/10B, 64B/66B, custom coding, elastic buffers, clock-data recovery, and multiple equalization options, implementing this functionality in soft IP would cost thousands of LUTs and FFs per transceiver. An external SERDES typically consumes much more power because the FPGA to external SERDES is typically high-performance parallel I/O. These parallel I/O can burn far more power than the Serial I/O channels.
- Tri-mode Ethernet MACs that run at 10/100/1000 Mb/s and can interface directly with RocketIO transceivers. Here again, a soft IP implementation would consume a few thousand LUTs and FFs.
- Smart RAM memory with distributed RAM and block RAM. Each LUT has built-in circuits to turn it into a distributed RAM/ROM, as well as a 16-bit shift-register. Each block RAM has built-in ECC for error correction and built-in control circuits to implement an asynchronous FIFO. All this functionality implemented in soft IP would consume hundreds of LUTs and FFs per block RAM.

Besides the obvious advantages associated with moving these commonly used blocks into hard IP, one must not overlook the inherent contribution the Xilinx Advanced Silicon Modular Block (ASMBL) architecture makes to the Virtex-4 dynamic power advantage. Because each of the three families – the LX, FX, and SX – is meant to satisfy distinct requirements for a particular application domain (logic, embedded processing, and signal processing), their standard ratio of logic cells, memory, I/O, DSP, processors, etc., has been optimized for that domain. Consequently, the Virtex-4 device is the first FPGA to offer domain-optimized dynamic power consumption (Table 2).

Table 2: **The Virtex-4 Domain-optimized Functions that Substantially Improve Dynamic Power Efficiency**

Functions	Virtex-4 LX	Virtex-4 SX	Virtex-4 FX
Logic Cells	14 to 200K	23 to 55K	12 to 140K
Embedded Memory	0.9 to 6 Mb	2.3 to 5.7 Mb	0.6 to 10 Mb
DCMs	4 to 12	4 to 8	4 to 20
XtremeDSP Slices	32 to 96	125 to 512	32 to 192
SelectIO Signalling	240 to 960	320 to 640	240 to 896
RocketIO Transceivers	N/A	N/A	0 to 24 channels
PowerPC Processors	N/A	N/A	1 or 2 processors
Ethernet MACs	N/A	N/A	2 or 4 embedded blocks

Inrush Current

When voltage is initially applied to an FPGA, its internal circuitry undergoes states of ambiguity while configuration storage latches and other circuits are programmed sequentially. The internal contention that occurs during powerup can produce spikes of inrush current that measure in multiple amps. This inrush current often requires the

use of expensive regulators, and larger, more expensive power supplies. To properly configure the FPGA, the power supply must be able to handle the large in-rush currents even if the device operates at much lower dynamic current levels during operation. This is especially true if the clock frequency of the FPGA during normal operation does not push the maximum limits.

Xilinx eliminated nearly all inrush current in Virtex and Spartan series FPGAs a few years ago by embedding innovative *housekeeping circuitry* to prevent this contention from occurring.

Power Estimations

The heat of competition in the FPGA marketplace can sometimes make it difficult for customers to obtain objective, fair-handed assessments of performance and power claims. So how does one determine which high-performance 90 nm FPGA family consumes the lowest total power, including static and dynamic power measured at the design's operating temperature?

In an attempt to keep a level playing field and the comparisons on an apples-to-apples basis, the Xilinx Web Power Tool v4.1 based on extensive characterization data was used to estimate power for the Virtex-4 devices. Altera's PowerPlay 2.1 power estimation tool was used to do the same for relatively equivalent logic density Stratix II parts.

To keep the static power comparison simple, only the core voltage supply is examined. Temperature invariant auxiliary voltage supplies (V_{CCAUX} and V_{CCPD}) are intentionally excluded. Stratix II FPGAs require additional voltage supply rails and power for every PLL used – these are also not included in the comparison.

If a hard IP block is available in Stratix II, it is used in the comparison; if not, the equivalent soft IP, using the logic utilization published for Stratix II implementations, is used. Because Virtex-4 FPGAs have more hard IP functions than Stratix II, and each hard IP function has more programmable features and options than Stratix II, it is difficult to compare "equivalent" functions; however, the comparison was executed on as much of an "apples-to-apples" basis as driven by the products.

The following results are offered for consideration, and since both tools are readily available, everyone is encouraged to perform the same or similar comparisons themselves. The results are compelling.

66% - 73% Less Static Power

Figure 5 compares the static power at T_j = 85°C using Xilinx WPT v4.1 vs. Stratix II PowerPlay v2.1. Table 4 shows the detailed data.

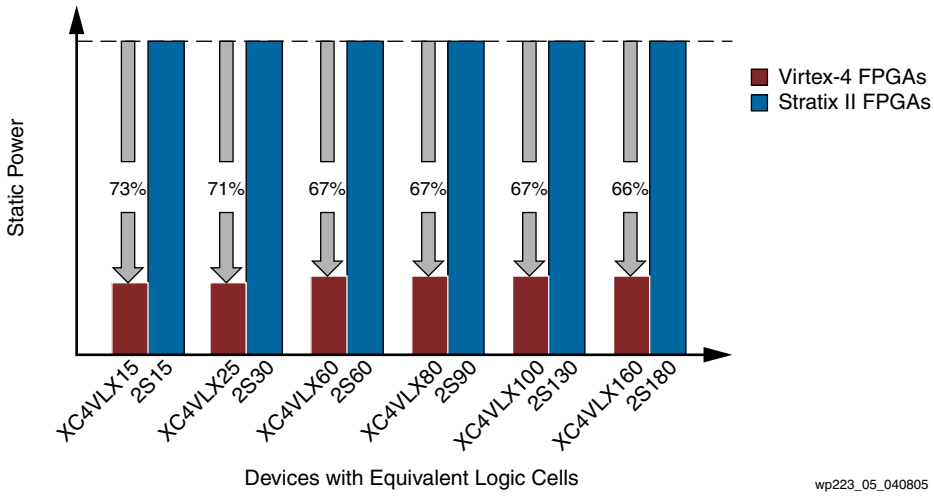


Figure 5: **Static Power Estimations**

Hard IP = Less Dynamic Power

Figure 6 compares the dynamic power at T_j=85°C using Xilinx WPTv4.1 vs. Stratix II PowerPlay v2.1. Table 5 contains detailed information.

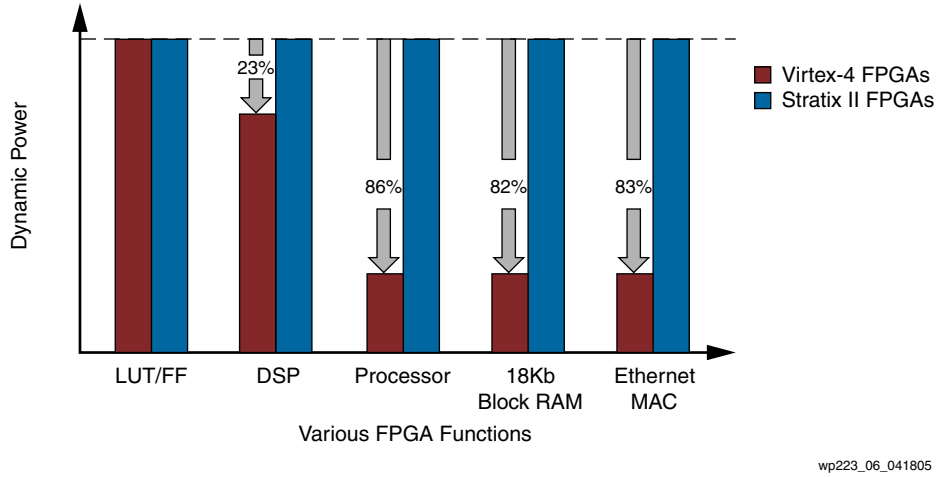


Figure 6: **Dynamic Power Estimations**

Measured Results

Recognizing the variability encountered when using *power estimators* as new characterization data becomes available from the vendor, Xilinx conducted static power tests on equivalent parts across the full commercial temperature range – an important consideration for many applications.

At a typical operating temperature of 85°C, Virtex-4 devices consume one-fourth the static power of Stratix II – a comparative relationship that remains constant as the temperature increases to 100°C (Figure 7).

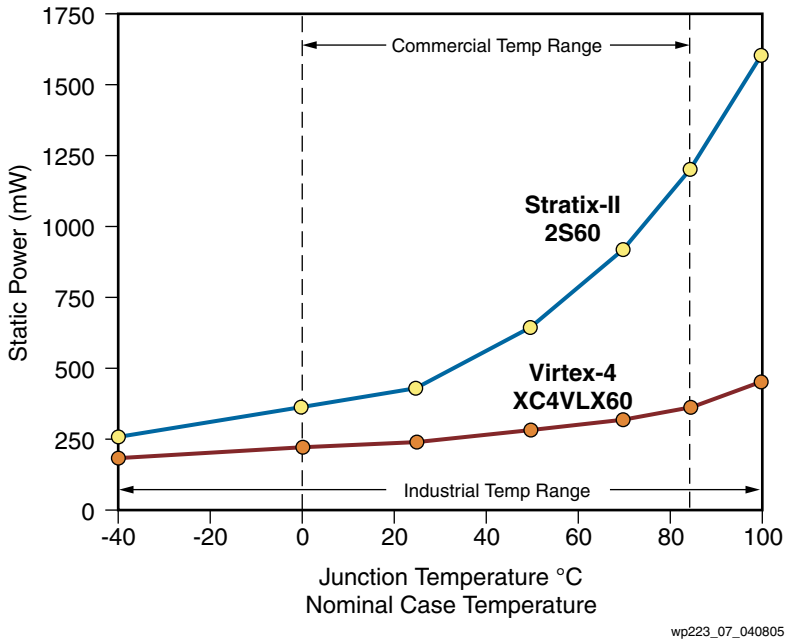


Figure 7: **Static Power Measurements Viewed Across Realistic Junction Temperature Ranges**

The measured results of the dynamic power tests demonstrate equivalent dynamic power consumption (Figure 8) for Virtex-4 and Stratix II devices when looking at just

LUTs and FFs. A more dramatic result (Figure 9) is obtained when testing block RAM configurations of equivalent sizes. In large part, the difference is due to the availability of optimally sized 18K bit memories from Xilinx. The smaller Stratix II 4K bit block RAMs consume nearly the same amount of power as the Virtex-4 18K bit block RAMs. Most customer designs either need ultra-small distributed RAM or medium-level memories built using 18K bit blocks – prompting Xilinx to move from 4K bit memories three product generations ago.

Although Stratix II FPGAs also have DSP blocks that can be connected using adder trees built out of logic resources, the Virtex-4 XtremeDSP slice provides the unique advantage of cascading multiple slices without additional logic. For many DSP functions, this gives a big advantage on speed and power. To test the dynamic power for DSP designs, a 64-tap FIR filter is used. The filter has 63 section asymmetrical taps with 18-bit data stream and fixed 18-bit coefficients. The Virtex-4 FPGA uses 63 XtremeDSP slices in a single column; one slice is used for stimulus. The Stratix II FPGA uses four tap sections in each DSP block, which are added together using nine 3-input adder tree structure. Stratix II ALMs are used for stimulus. Figure 10 shows the Virtex-4 FPGA consuming 2.35x (or 60%) lower dynamic power than Stratix II for DSP designs.

The more astonishing result is found for the V_{CCINT} core voltage and the V_{CCAUX} (2.5V) in the Virtex-4 device and the V_{CCPD} (3.3V) supply in Stratix II. The V_{CCAUX} in the Virtex-4 FPGA is used for configuration memory (some in DCMs, and some in I/O pre-drivers). The V_{CCPD} supply in Stratix II is used for configuration and I/O pre-drivers. In order to test I/Os, 500 LVCMOS I/Os were tested at all voltage levels (1.5V, 2.5V, and 3.3V). All outputs are in DDR mode, and loads are terminated to ½ V_{CCIO}. DC to 400 MHz frequency range was used. Shown in Figure 11, the Virtex-4 device consumes lower power at all voltage levels.

In Figure 12, V_{CCPD} supply consumes four times the power of the V_{CCAUX} supply — translating into Watts of power difference! The V_{CCINT} supply in Stratix II also consumes three times the static power and 50% more dynamic power

The conclusion about the tests is that the Virtex-4 device consumes much less power than Stratix II.

Dynamic Power - Fabric

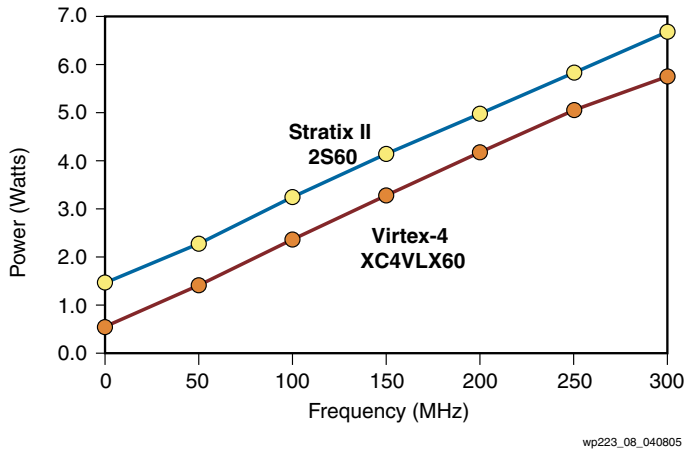


Figure 8: **Fabric Test with 25,000 LUT/ALUTs and 21,000 FFs (High Toggle Rate). All measurements were taken at T_j = 85°C.**

Dynamic Power - Memory

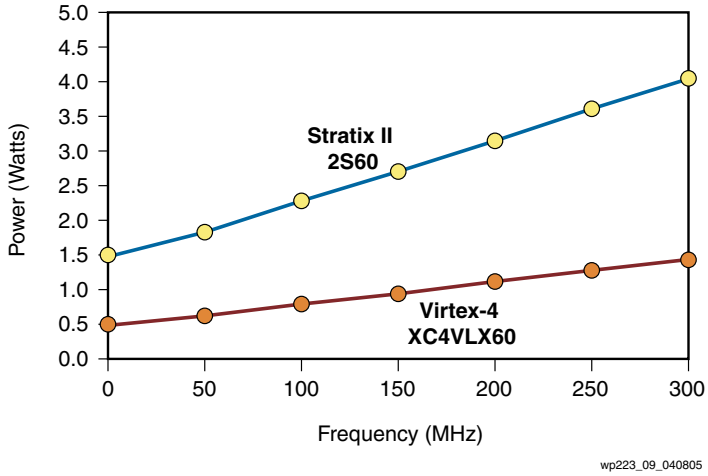


Figure 9: **FIFO/Block RAM test with 252 M4K in Stratix II and 63 Block RAM in Virtex-4 FPGAs (same total storage). All measurements were taken at T_j = 85° C.**

Dynamic Power - DSP (64-Tap FIR Filter)

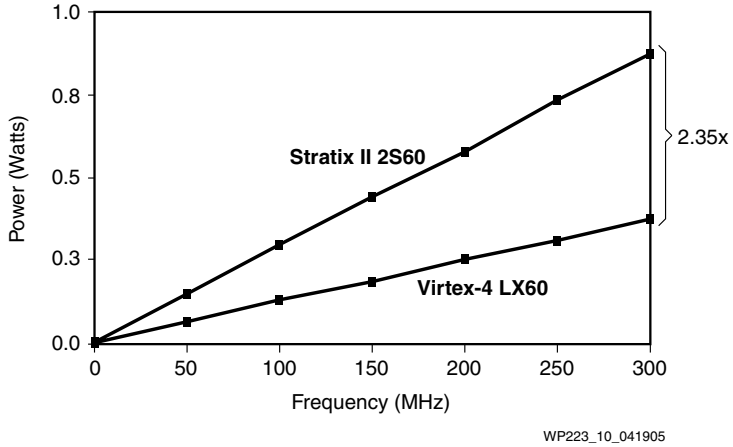


Figure 10: **DSP Test: 64-Tap FIR Filter**

Dynamic Power - I/O .

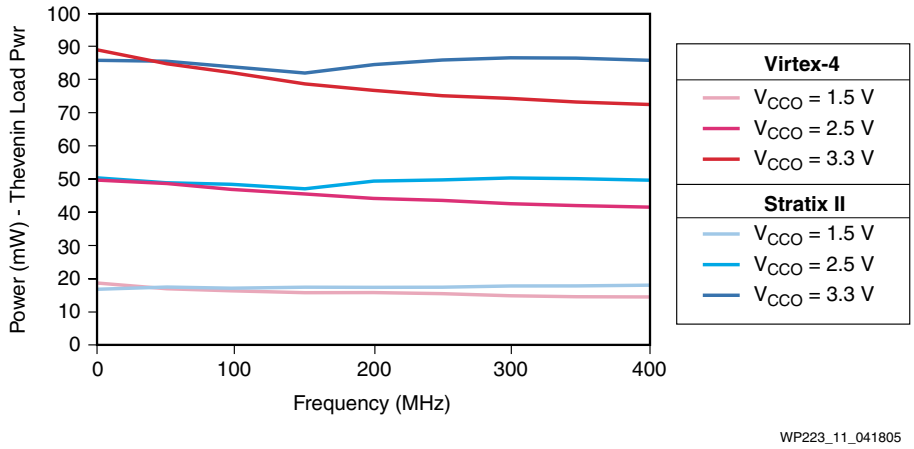


Figure 11: **Virtex-4 FPGAs Consume Less Power at All Three V_{CCIO} Voltages**

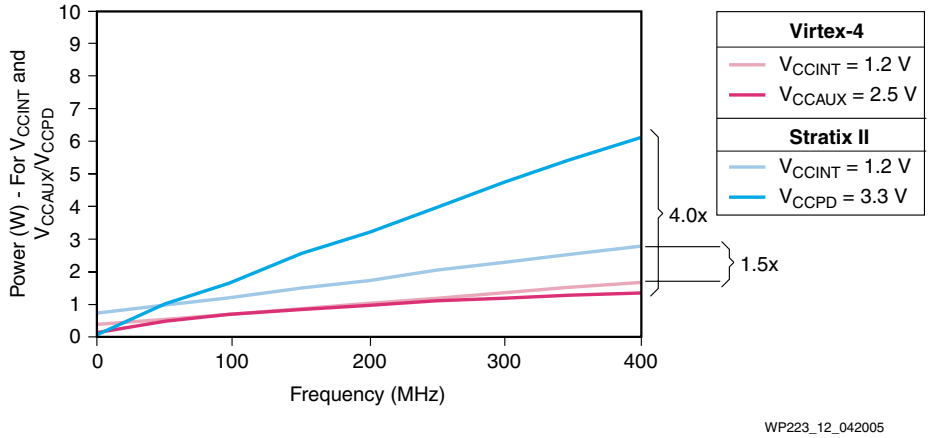


Figure 12: **I/O Test: Stratix II shows 1.5x the V_{CCINT} Dynamic Power and 4x the V_{CCPD}/V_{CCAUX} Power compared to the Virtex-4 FPGA**

Total Power Budget Comparison

Another simple way to obtain a high-level comparison of power consumption in FPGAs is to assess the available performance and power headroom for a given power budget and target frequency. For this comparison, popular applications for high-performance FPGAs at a sweet-spot density of 60,000 logic cells were considered, e.g., multi-service provisioning platforms in networking and telecom access markets; blade servers in storage and server markets; line cards in wireless base stations; and control cards in industrial and military/aerospace systems. Operating at an average target frequency of 200 MHz, with operating temperatures in the range of 65-100ºC, the power budget for these applications is typically 6W per FPGA.

Using these applications as an example, the Virtex-4 FPGA meets the target frequency, well within the power budget at 4.9W. On the other hand, Stratix II FPGAs exceed the power budget, consuming 7.9W at the target frequency. Consequently, when constrained by the 6W power budget, Stratix II delivers only 133 MHz performance (Figure 13).

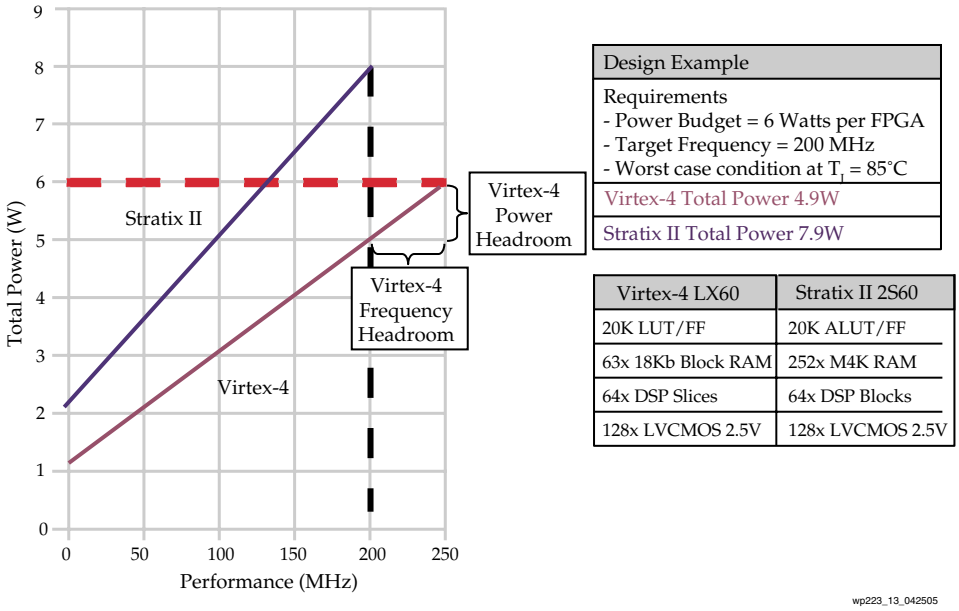


Figure 13: **Customers Select the High-performance FPGA that Best Meets Their Power Budget**

Xilinx Telecom Customer's SPI4.2 Design Comparison

Intrigued by the power consumption advantages in Virtex-4 devices, one Xilinx customer used the web power estimation tools to compare the power consumption for a SPI4.2 core implemented on Stratix II and Virtex-4 FPGAs.

The Optical Internetworking Forum's SPI4.2 parallel interface appears in numerous 10-Gb/s networking and telecom applications. Because virtually all networking and telecom equipment has stringent power and thermal budgets, the customer was pleased to discover that the estimators showed Virtex-4 devices consuming 50% lower total power than Stratix II for this application.

Satisfied but surprised by the results, the customer decided to perform real measurements on actual implementations to verify the findings. Once again, this time with actual lab results in hand, the customer found that Virtex-4 FPGAs consumed 50% lower total power for the SPI4.2 implementation than did the Stratix II version.

The Xilinx customer's test has been recreated using similar conditions, with similar results. The results are presented in Figure 14 and Table 3.

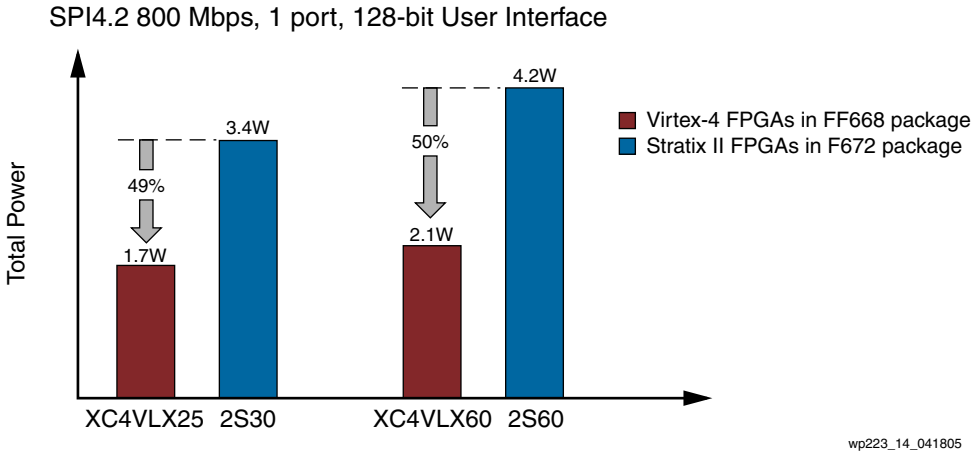


Figure 14: **Customer Analysis on SPI4.2 Power Consumption**

Table 3: **Virtex-4 vs. Stratix II Implementation**

	Virtex-4 Implementation	Stratix II Implementation
Protocol	SPI4.2 v7.2, 800 Mb/s	SPI4.2 v2.2.2, 800 Mb/s
User I/F	128-bit, 1 port	128-bit, 1 port
Logic	3750 Slices: • 4893 LUTs • 25% toggle rate used	Tx and Rx: • 10844 ALUTs (calculated from 10968 + 2588 LEs) • 25% toggle rate used
Memory	306 Kb total – 17 18Kb-block RAMs used: • 13 - 512 x 36 • 2 - 36 x 512 write and 18 x 1024 read • 2 - 1024 x 9 • 25% toggle rate used	244 Kb total – 62 block RAMs used: • 61 - M4K blocks • 1 - M512 blocks • 25% toggle rate used
LVDS	17 LVDS pairs for Rx and Tx: • 35 pF average load • Additional differential clock output	17 LVDS pairs for Rx and Tx: • 35 pF average load • Additional differential clock output
Clock Frequency (Assumption)	200 MHz single clock	200 MHz single clock

Note: Limited data available on the Stratix II implementation. For fair comparison, some assumptions had to be made about clocks. See Answer Record 20430 on Xilinx.com for Virtex-4 SPI4.2 power.

1 to 5 Watts lower power per FPGA

Having demonstrated substantial power savings – both in static power as a result of triple-oxide technology and in dynamic power using embedded IP – and having presented corroborating results from both power estimators and lab measurements using various designs, one might well ask, "What does it all mean for my design?"

The simplest example provides the best perspective. Using an equivalent amount of generic logic and memory in Virtex-4 and Stratix II devices of equivalent density, with no consideration of other embedded IP, the Virtex-4 FPGA saved 1 - 5W in power. A design should realize at least this much power savings per every Virtex-4 FPGA used, if not better.The results are shown in Figure 15. To present the lowest bar in power savings, benefits from extensive hard IP in Virtex-4 FPGAs were not included.

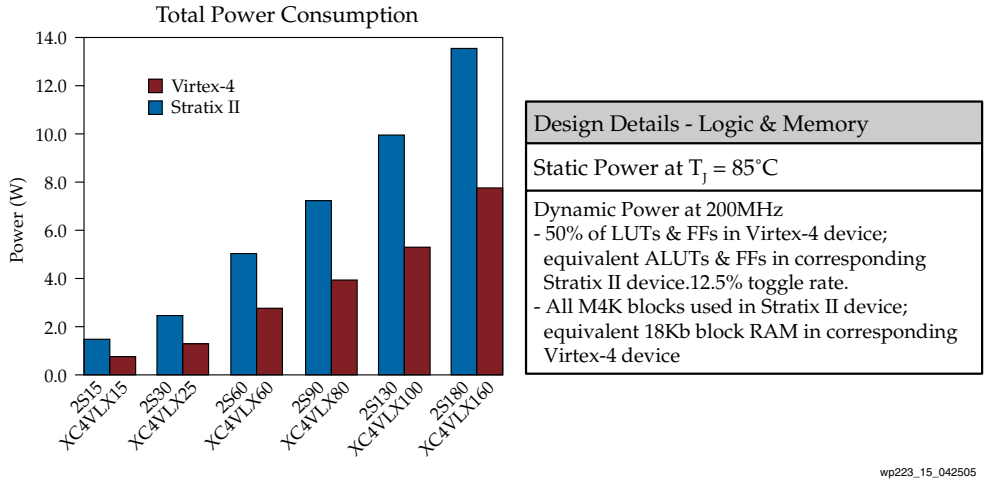


Figure 15: **Virtex-4 Designs Consume 1 to 5 Watts Lower Power per FPGA**

Power Struggle

The battle to deliver maximum performance with minimum power expenditure has taken center stage in the evolution of the FPGA. Power conservation impacts every budget, whether technological or financial. Product acceptability, reliability and profitability depends as much or more on power efficiency as it will on performance.

Virtex-4 FPGAs not only have a far superior feature set, but also exhibit a real power consumption advantage. The choice should be very clear for designers who require lower power consumption, and high-performance and functionality. Initial test results tell the story:

- Up to 73% lower static power with the industry's first triple-oxide technology
- Up to 86% lower dynamic power with high-performance embedded IP

The competition in the market does not end with 90 nm devices. Some interesting new dynamics arise when moving into a 65 nm node and below. Fortunately for Xilinx, one inherent value of using triple-oxide technology is that it scales nicely with each new process.

As for the value of embedding hard IP wherever appropriate, it is practically an industry axiom. Xilinx has incorporated the right amount of programmable embedded IP with programmable logic to make the whole solution more flexible with higher-performance, and lower power. In the long term, customers will use only Platform FPGAs that provide the best of performance and power.

Appendix

Static Power Comparison Data

Table 4 gives the comparison data for static power.

Table 4: **Static Power Comparison Data**

Virtex-4 Static Power		Stratix II Static Power		Virtex-4 Power Reduction Typical 25°C/85°C
Virtex-4 Device	Typical 85°C T_j	Stratix II Part	Typical 85°C T_j	
XC4VLX15	138 mW	2S15	515 mW	73%
XC4VLX25	231 mW	2S30	792 mW	71%
XC4VLX60	493 mW	2S60	1,478 mW	67%
XC4VLX80	640 mW	2S90	1,960 mW	67%
XC4VLX100	863 mW	2S130	2,630 mW	67%
XC4VLX160	1,117 mW	2S180	3,310 mW	66%

- Core static power = I_{CCINTQ} x V_{CCINT}. Altera data is based on PowerPlay tool v2.1 on www.altera.com. Static power for V_{CCPD} (the Stratix II V_{CCAUX} supply) and V_{CCIO} does not change with temperature and is not shown. Stratix II also consumes additional power for PLL supply rails that are not shown.
- Devices of equivalent logic density are compared.

Dynamic Power Comparison Data

Table 5 gives the comparison data for dynamic power.

Table 5: **Dynamic Power Comparison Data**

Virtex-4 Dynamic Power (Room Temperature)		Stratix II Dynamic Power (Room Temp)		Virtex-4 Power Reduction
Function at 200 MHz	**Power**	**Function at 200 MHz**	**Power**	
1 LUT and 1FF • 30% toggle rate • Medium Routing	0.15 mW	1 ALUT and 1 FF • 30% toggle rate • Routing data unavailable	0.15 mW	~same
1 XtremeDSP™ Slice • 18x18 MAC • Registered in/out • Medium (55%) toggle rate	6.6 mW	1 DSP Block • 18 x18 MAC • Registered in/out • 50% toggle rate	8.57 mW	23% ⇓
PowerPC™ Processor • 16Kb I-Cache and D-Cache • 2 DCMs	120 mW	Soft Processor • 1 - 6K I-Cache and D-Cache	879 mW	86% ⇓
18 Kb block RAM • Width = 18, Depth = 1 • 50% read and 50% write	6.16 mW	Equivalent M4K RAM • Width = 18, Depth = 1 • 50% read and 50% write	33.98 mW	82% ⇓
Ethernet MAC - embedded • 1 Gb/s, 125 MHz	27 mW	Ethernet MAC - soft • 1 Gb/s, 125 MHz • 2500 ALUTS and 2500 FFs	160 mW	83% ⇓

Power Across the Board

Optimize Design of Xilinx FPGA, ASIC, and Point-of-Load Power Applications

LM2852 Features

- Best-in-class 60 mΩ switches provide over 95% efficiency
- Factory-programmable EEPROM for any output voltage between 0.8V to 3.3V
- Up to 1.5 MHz internal compensation eases designing with a variety of output capacitors
- Available in thermally enhanced, ETSSOP-14 package

LM2745 Features

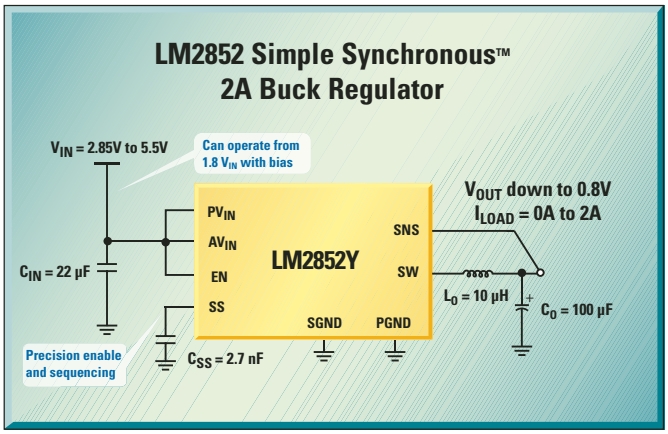
- Output voltage adjustable down to 0.6V
- Switching frequency from 50 kHz to 1 MHz
- Startup with a pre-biased load prevents sagging of output
- Available in a TSSOP-14 package

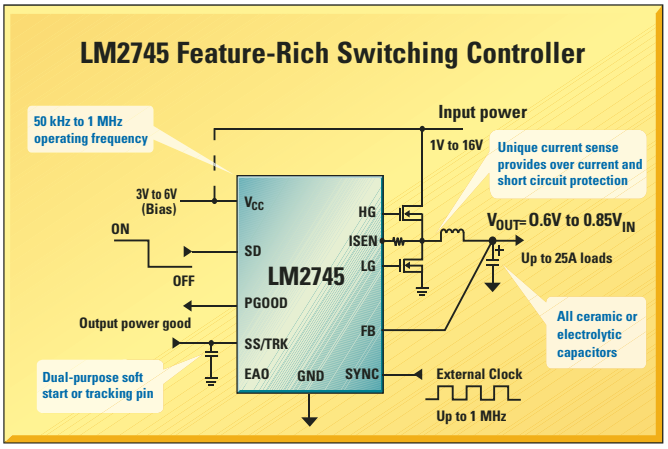
Ideal for FPGA/DSP applications in networking, telecommunications, enterprise, and automated test and measurement applications

For free samples, evaluation boards, datasheets and online design tools, visit us today at:

power.national.com

Or call 1-800-272-9959





Solutions for Xilinx FPGAs

Design guides, app notes, and online tools for Xilinx FPGAs

www.national.com/see/xilinxfpga

National Semiconductor

The Sight & Sound of Information

WP221 (v1.0) March 23, 2005

Static Power and the Importance of Realistic Junction Temperature Analysis

By: Matt Klein

Total power consumption of a board or system is important; each FPGA or ASIC in a system is beginning to be forced to meet a power budget. With this concern and the trend of increasing static power with use of high performance 90 nm FPGAs, Xilinx has put considerable effort into reducing static power in the Virtex™-4 FPGAs. To this end, it is important to consider a realistic operating temperature for the FPGAs, which can easily have junction temperature up to and in excess of 85°C. As junction temperature rises, static power rises exponentially, fueling this concern.

Introduction

Customers expect FPGA and ASIC vendors to reduce cost and increase performance. Typically, this was achieved in the past by reducing transistor sizes, hence increasing the performance and decreasing the die area and cost.

Reducing transistor sizes increases transistor leakage current, and therefore static power. With 90 nm technology in FPGAs or ASICs, there is a particular challenge to reduce static power. A further problem is that leakage rises dramatically with junction temperature. This white paper describes where static power comes from and its variation with temperature, providing insight into how Virtex-4 FPGAs address decreasing static power. Xilinx leverages its vast experience as the industry leader, producing millions of 90 nm FPGAs, to research this issue and consequently reduce power in 90 nm Virtex-4 FPGAs.

Even though system speeds are increasing, core voltage is dropping, which reduces the rate of increase of dynamic power; however, static power is growing exponentially over time as we move to smaller and smaller technology nodes because of increasing transistor leakage. Figure 1 from the International Technology Roadmap for Semiconductors (ITRS) shows a cross-over point as the industry arrives at 90 nm and smaller technology nodes, where static power is beginning to eclipse dynamic power for many applications.

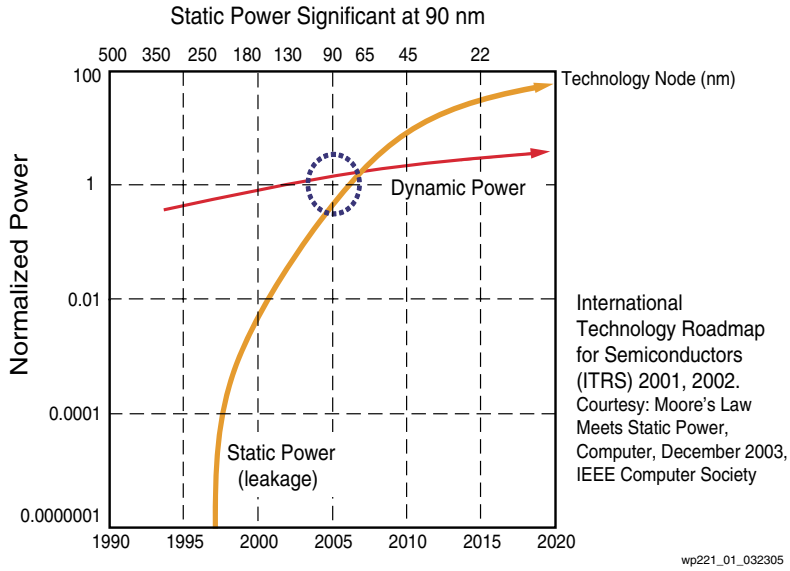


Figure 1: **Static and Dynamic Power vs. Technology Node**

Decreased Power Lessens Other System Design Issues

FPGAs are being used increasingly in many applications, so reducing power consumption in FPGAs provides huge benefits to the system design. Some of the key benefits are shown below:

- Fewer thermal concerns - lower power causes smaller rise in junction temperature, preventing thermal runaway:
  - Use smaller heat sinks or no heat sinks at all.
  - Allow for a simpler system thermal design requiring less airflow and smaller or fewer fans.
- Lower cost power system - reduced power requires smaller power supplies:
  - Power supplies cost from $0.50-$1.00/Watt.

- ♦ Additionally, there are some *total power* system considerations where it is important to stay below a step function jump, such as < 500 W, < 1 KW, etc. If the design is at the border, every watt counts.
- ♦ A smaller and simpler power circuitry means:
  - Fewer components.
  - Smaller PCB.

- Higher system reliability - running with reduced power lowers the junction temperature, and lower junction temperature improves reliability.

Since total power consumption of a board or system is important, each FPGA or ASIC in a system must meet a power budget. Users need to design within the power budget while being pushed to higher performance and lower cost - a rather difficult challenge at 90 nm. With this in mind, Xilinx has put considerable research and development into reducing static power in the Virtex-4 FPGAs and actually reversed the trend for high static power in a high performance 90 nm FPGA.

Where Does Leakage and Static Power Come From?

A well-known rule of semiconductor physics is that when transistor length decreases, leakage current increases. Smaller physical distances make it easier for current to leak. Both source-to-drain leakage and gate leakage are inversely proportional to channel length and gate oxide thickness, respectively, and show a dramatic increases in leakage. Static power is the power consumed due to leakage in the transistors even when the transistors are not switching. Therefore, power is consumed in the FPGA even when it isn't performing a task.

The leakage is dramatically influenced by junction temperature. This is why it is important to look at realistic junction temperatures for the FPGAs in the system to properly account for static power. Figure 2 shows the dramatic increase in I_{CCINTQ}, the leakage or static current for the core transistors in the FPGA.

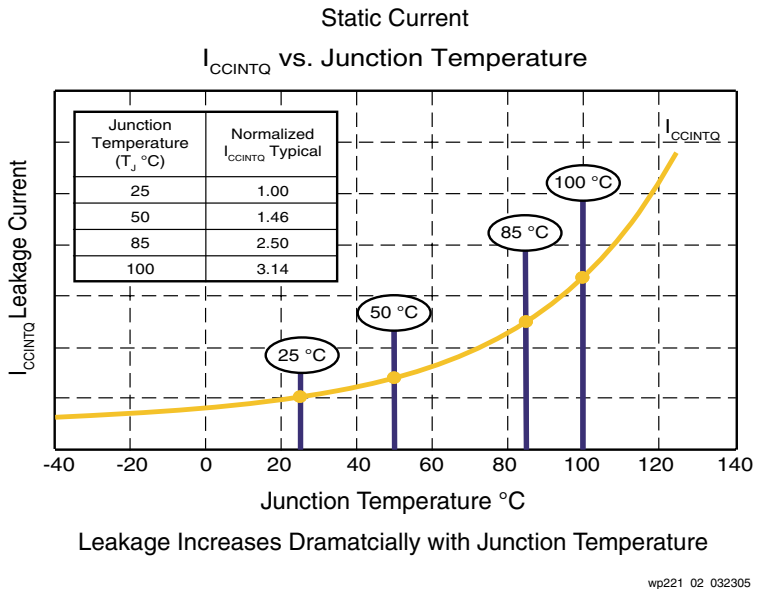


Figure 2: **Leakage Current vs. Temperature for 90 nm Virtex-4 Devices**

Two components contribute to leakage:

- Source-to-Drain Leakage: This current flows from source to drain of the transistor, even when the transistor gate is off. When the transistor gets small, it is harder to prevent this current from flowing, and therefore, 90 nm transistors tend to exhibit

source-to-drainage leakage with much greater magnitude than larger transistors, all parameters being equal. The other problem is the thickness of the gate oxide. A thinner oxide allows the transistor to be switched on and off faster, but it also increases leakage. The amount of leakage is also influenced by the threshold voltage of the transistor. The threshold voltage, or V_T of the transistor, is the voltage between the gate and the source at which the channel conducts current to a certain extent. Small high-speed transistors need a lower threshold voltage (influenced by oxide thickness and doping) to maintain the speed with which the transistor can be turned on and off via gate control, but this also tends to increase the leakage because the channel of the transistor can not be turned off completely. The goal is to make fast transistors, but unfortunately, physics is against this. Another important note is that source-to-drain leakage increases exponentially with increasing temperature; in going from a junction temperature (T_J) of 25°C to 85°C, it goes up by a factor of 5x (see IS → D, Figure 3, page 4).

- Gate Leakage: This current flows from gate to substrate. This component of leakage is now more substantial as transistor gate oxide thickness has decreased at the 90 nm technology node. At room temperature, the leakage from gate to substrate is larger than the source-to-drain leakage in the 90 nm fast thin oxide transistors; however, unlike source-to-drain leakage, this gate leakage only increases marginally with increased temperature (see I_{GATE}, Figure 3).

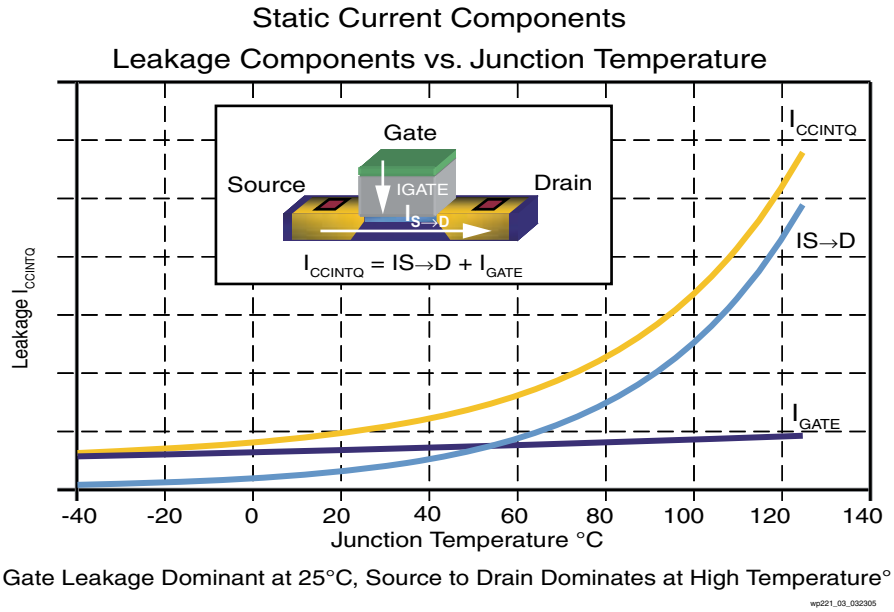


Figure 3: **Leakage Current Components versus Temperature for the 90 nm Virtex-4 FPGA**

The net effect is that total leakage $I_{CCINTQ} = IS \rightarrow D + I_{GATE}$ goes up by 2x - 3x between 25°C and 85°C.

FPGAs and ASICs are getting larger and larger due to requirements of hardware and system designers. This means that with parts the size of the new Virtex-4 LX200, there are upwards of one billion transistors to worry about. A leakage of 10 nA per transistor amounts to a leakage of 10 A!!

There is a silver lining for FPGA manufacturers that is not applicable to ASIC manufacturers: Xilinx has solved the issue of high static power in the Virtex-4 device, a high performance 90 nm FPGA, by using triple-oxide.

What is Triple-Oxide and How Does It Affect Static Power

Triple-oxide simply means that we use a third thickness of gate oxide in making some of the transistors in the FPGA (two oxide thicknesses are used in devices like the Virtex-II Pro FPGA). Most transistors in the past had a thin oxide layer, some with NMOS or PMOS transistors that operate at either low V_T or regular V_T. Thick-oxide transistors are mostly used for I/O drivers.

Oxide deposition thickness is a very stable and controllable process in the semiconductor industry because it depends on temperature, concentration, and exposure time. Figure 4 shows the Virtex-4 transistor with the middle oxide thickness used in the triple-oxide process. The oxide thickness is still very, very thin, but this thicker oxide transistor has much lower leakage than the standard thin-oxide low V_T and regular V_T transistors used in Virtex-II Pro FPGAs and in various parts of Virtex-4 FPGAs.

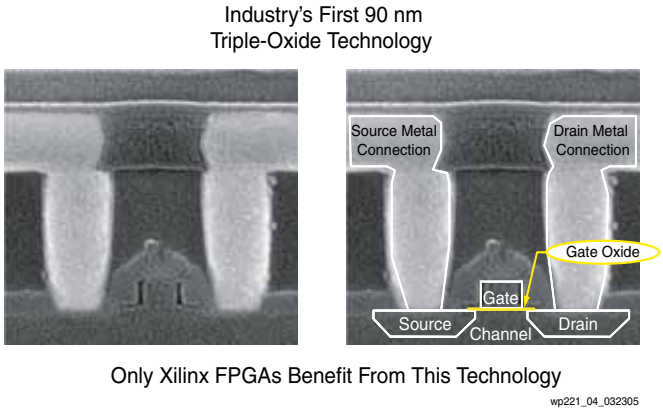


Figure 4: **Middle Oxide Thickness Virtex-4 Transistor Used in Triple-Oxide Process and with Highlighted Portions of the Transistors**

Why Doesn't Everyone Use Triple-Oxide?

If triple-oxide is such a great process, why don't other companies like Intel or IBM use it in their own ASICs? They probably would if it benefited them. The reason they don't is that all of their transistors need to run at speed; hence, they must use the low V_T, leakier transistors for everything.

Unique to SRAM-based FPGAs and in contrast to ASICs, all transistors don't need to be the smallest, fastest, high-leakage transistors. FPGAs use a large number of transistors (up to 1/3 of those in the FPGA) to hold the design-specific settings for the interconnect transistors and the logic functions. These are the configuration memory cells. Additionally, the interconnect pass transistors, while they need to be fast at passing a signal from source to drain, don't need to be switched on and off rapidly (controlled by changing the gate voltage). The connections in the FPGA are known in advance. A transistor that doesn't need to have its gate switched rapidly doesn't need to be a thin oxide leaky transistor at all, low V_T or otherwise.

Triple-Oxide and What It Does to Reduce Static Power

Xilinx already has had experience with 90 nm FPGAs for a few years through the deployment of the Spartan™-3 FPGAs. Consequently, a few important things were learned. The Spartan-3 is a medium performance FPGA, primarily optimized for the lowest possible cost. In contrast, the Virtex-4 FPGAs are primarily optimized for very high performance along with decent cost reduction. When using 90 nm transistors of the highest performance, the leakage described above becomes very large; so what wasn't a problem in the Spartan-3 FPGAs would have been a problem in Virtex-4 devices had it not been for the early realization of this leakage issue by the Xilinx IC design team.

Triple-oxide means that in addition to thin-oxide, small, fast, transistors for the FPGA core and thick-oxide, higher voltage, swing transistors for the I/O, we have introduced a third middle-thickness transistor, which can be used for configuration memory cells and interconnect pass transistors. These transistors operate at a higher threshold voltage than the thin oxide transistors.

So, the bottom line is that if a design uses hundreds of millions of transistors in a high-performance 90 nm FPGA, the transistors can be lower leakage middle-thickness oxide, which greatly reduces leakage without compromising performance in the FPGA. The net effect is that the total FPGA leakage is substantially lower than that of a similar-sized FPGA that doesn't use triple-oxide.

If a high-performance FPGA is made without using triple-oxide to substantially lower the total leakage, not much can be done after the fact; as mentioned earlier, raising threshold voltage via doping changes or increasing oxide thickness can't reduce leakage affecting performance.

Xilinx made smart choices at the begining of the design phase. In addition to triple-oxide, Xilinx IC designers balanced gate lengths and gate widths of transistors, voltage thresholds, and circuit design to create a high-performance 90 nm FPGA with low leakage.

Figure 5 shows leakage amounts for the the Virtex-4 device compared to the 130 nm Virtex-II Pro device and other 90 nm high performance FPGAs at 85°C. This data is based on prediction tools and data sheets from Xilinx and prediction tools from the competing 90 nm FPGA. The Virtex-4 device has greater than 60-70 % less leakage than its competitor, which translates to 60-70 % less static power.

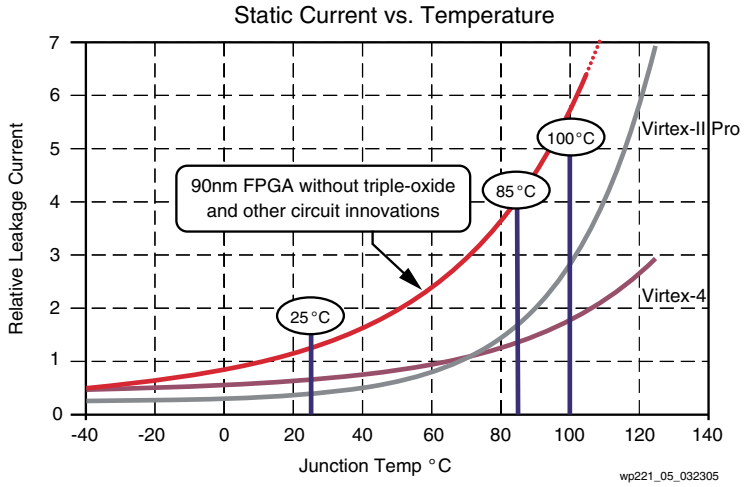


Figure 5: **Relative Leakage Current in the Virtex-4 Device versus the Competitor for Similar Logic Density**

Xilinx has measured thousands of devices; the overall leakage, and hence static power, is low. Figure 6 shows measurements of actual parts from Xilinx and Altera, specifically the Virtex-4 XCV4LX100 and the Stratix II EP2S90.

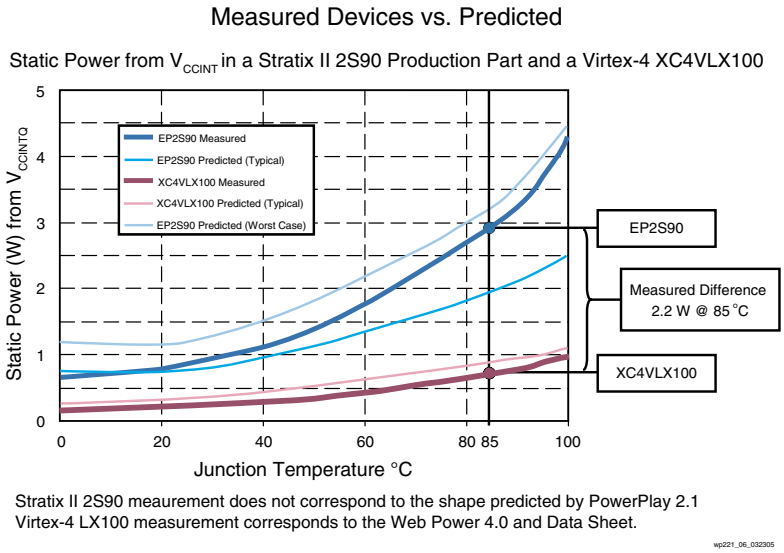


Figure 6: **Measured versus Predicted Static Power from V_{CCINT} in Stratix II EP2S90 and Virtex-4 XC4VLX100**

The measured power savings of 2.2W for the XC4VLX100 versus the EP2S90 at 85°C is significant, especially considering that the XC4VLX100 is a larger FPGA than the EP2S90. Also note that the measured data for the Stratix II EP2S90 compared to the worst case and typical prediction data from Altera's PowerPlay Early Power Estimator for Stratix II, version 2.1 is much closer to the worst case prediction at 85°C junction temperatures and beyond. On the other hand, the measured data for the XC4VLX100 is actually lower than the typical predicted static power versus junction temperature from the Xilinx Virtex-4 device (see the Virtex-4 data sheet http://www.xilinx.com/bvdocs/publications/ds302.pdf and the Web Power Tools versions 4.0 and 4.1).

FPGA Operating Environment and Static Power

Operating environment for the FPGA must be considered carefully when looking at Static Power consumption in the high performance 90 nm FPGAs. Designers look at the normal ambient environment for various equipment that uses FPGAs, but it is the T_J of the FPGA die that is most important when looking at static power. As in Figure 2, there is 2.5 times more leakage at a junction temperature of 85°C than there is for a junction temperature near 25°C, and there is greater than 3 times more leakage at 100°C than at 25 °C.

In Table 1, some of the typical junction temperatures might be from greater than 75°C to beyond 100°C, depending on the type of product.

Table 1: **Environmental Conditions for Various Types of Equipment**

	Operating Temperature Environment		
	Ambient Outside Equipment T_{AO} (°C)	**Ambient Inside Equipment T_{AI} (°C)**	**Device Junction Temperature T_J (°C)**
Type of Equipment	High	High	High
Computing, Storage, Server Racks	45	55	75 - 85

Table 1: **Environmental Conditions for Various Types of Equipment *(Continued)***

	Operating Temperature Environment		
	Ambient Outside Equipment T_{AO} (°C)	**Ambient Inside Equipment T_{AI} (°C)**	**Device Junction Temperature T_J (°C)**
Metro and Access Networking	85	100	> 100
Automotive, Aerospace and Defense	100	> 100	> 100

Typically, designers using FPGAs have power budgets ranging from < 1 W for a small device like a Virtex-4 LX15, to 4-6 W in an LX60, to 10 W in an XC4VLX100 and so forth. Figure 7 shows the static power consumed by the Virtex-4 devices and Stratix II devices at 85°C based on Altera's PowerPlay Early Power Estimator for Stratix II version 2.1 and the Xilinx Web Power Tools versions 4.0 and 4.1. As mentioned earlier, the Xilinx tools and data sheets seem to be more conservative than the competitor's tools. Nonetheless, a large static power savings is gained in the Virtex-4 FPGAs compared to the Stratix II FPGAs across comparable part densities from the two companies.

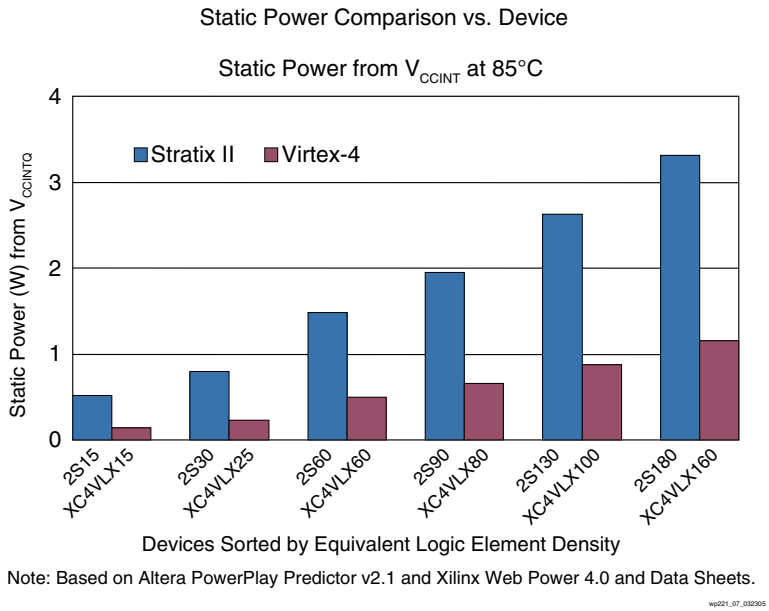


Figure 7: **Static Power at 85°C in Virtex-4 and Stratix II Equivalent Density FPGAs**

Conclusion

When considering thermal operating environments for various industries, many consider junction temperatures from 85°C to 100°C. Even at these junction temperatures, the Xilinx Virtex-4 FPGAs present an acceptable amount of static power, which leaves plenty of design margin for the dynamic components of power consumption in the FPGAs. The closest 90 nm competitive FPGA has 2-3 times the worst case static power consumption and up to 5 times the typical static power consumption. Virtex-4 devices give you the design edge to meet your performance without breaking your power budgets.

XILINX®

XAPP378 (v1.2) June 5, 2005

Using CoolRunner-II Advanced Features

Summary

This application note describes how to implement the CoolRunner™-II advanced features in the Xilinx software. These features include the DualEDGE triggered registers, clock divider, CoolCLOCK, DataGATE, Schmitt trigger inputs, and I/O termination types. HDL code examples are available for download, see **Code Examples Download**, page 9.

CoolRunner-II

Xilinx CoolRunner-II CPLDs combine performance and low power in a single device. For more information on the architecture of CoolRunner-II CPLDs, see **References**, page 10. CoolRunner-II CPLDs feature enhanced clocking flexibility and provide design capabilities that significantly reduce power consumption.

All design features discussed in this application note are supported in Foundation™ ISE™ and WebPACK™ ISE software from Xilinx. For more information on Xilinx software, see **References**, page 10.

Software Attributes

There are two main methods of attribute entry, each with their own advantages. A User Constraint File (UCF) has the advantage of being easily edited while being separate from the design source files. It has the benefit of allowing changes without needing to re-synthesize the source code. For more information on entering constraints with the UCF, see **References**, page 10.

Attribute entry within the source has the advantage of not needing to maintain a separate file for the design constraints. All CoolRunner-II VHDL and Verilog attribute examples are only applicable to the XST synthesis tool.

The attributes that are user definable and specific to the CoolRunner-II CPLD include: CoolCLOCK, DataGATE, Schmitt trigger input, keeper or pullup I/O termination, I/O standards, VREF, and open drain outputs. Some design constraints from this list are software selectable via the ISE Project Navigator GUI. Individual signal constraints must be explicity defined in the methods described above.

Please note that several syntax and help guides for attributes exist within the Xilinx ISE Project Navigator. These help files are updated with each release of software and provide a reliable source of information. The help menu can be located from within ISE under the menu option **Help | ISE Help Contents | CPLD Attributes**.

DualEDGE Registers

CoolRunner-II DualEDGE triggered registers allows designers to reach unprecedented performance levels. CoolRunner-II CPLDs can double system performance by creating DualEDGE triggered (DET) registers. CoolRunner-II DET registers allow data to be registered on both the rising and falling edge of a clock.

CoolRunner-II DET registers can be used for logic functions that include shift registers, counters, comparators, and state machines. Designers must evaluate the desired performance of the CPLD logic to determine use of DET registers.

The DET register can be inferred in any ABEL, HDL, or schematic design. Table 1 lists the inference methods available to create a DET register in CoolRunner-II.

Table 1: **DET Register Inference Summary**

Design Entry	Instantiation Method
ABEL	Use the following syntax: QOUT:=data; QOUT.DEC=clock;
VHDL/Verilog	Infer a dual edge triggered register.
Schematic	Instantiate a FDDn[S][R][E] component.

Examples

A designer can infer a single edge triggered (SET) register in any HDL design. A SET register active on the rising edge of the input clock would require the following VHDL or Verilog syntax.

```
VHDL: if (clock'event) and (clock = '1') then
Verilog: always @ (posedge clock)
```

The required syntax to infer a CoolRunner-II DET register requires the register be active on both the rising and falling edge of the clock. The following VHDL syntax would be used to infer a CoolRunner DET register.

```
process (clock)
begin
  if (clock'event) then
    ...
  end if;
end process;
```

The following Verilog syntax would be used to infer a DET register in CoolRunner-II.

```
always @ (negedge clock or posedge clock)
  ...
```

The DET register is available with all macrocells in all devices of the CoolRunner-II family.

Clock Divider

CoolRunner-II CPLDs provide additional clocking flexibility to the DET register feature with the clock divider. The CoolRunner-II clock divider provides the capability to divide an incoming clock and globally distribute the divided clock to all macrocells. The clock divider provides additional power savings by reducing the toggle frequency of the internal clock network.

The CoolRunner-II clock divider is available on global clock, GCK2, and can divide the incoming clock by 2, 4, 6, 8, 10, 12, 14, and 16. The clock divider creates a 50-50 duty cycle divided clock without affecting T_{CO}. The clock divider output is initialized low by the CPLD power up reset circuitry.

The clock divider circuit includes an active high synchronous reset, referred to as CDRST. When the CDRST signal is asserted, the clock divider output is disabled after the current cycle. When the CDRST signal is de-asserted the clock divider output will become active upon the first edge of GCK2.

Figure 1 illustrates the CoolRunner-II clock divider.

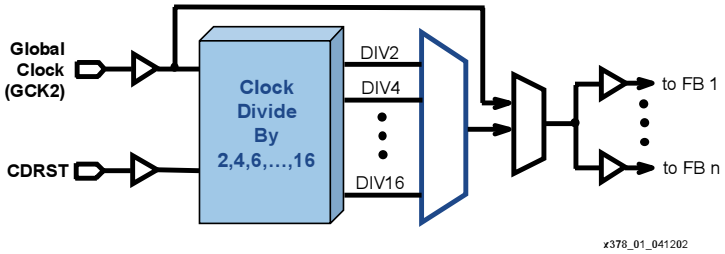


Figure 1: **CoolRunner-II Clock Divider**

The CoolRunner-II clock divider includes a built in delay circuit. With the delay feature enabled, the output of the clock divider will be delayed for one full count cycle. When used, the clock divider does not output a rising clock edge until after the divider reaches the terminal count value. The delay feature is either enabled or disabled upon configuration. The type of clock divider component instantiated will determine if the delay is enabled or disabled. Figure 2 illustrates a timing waveform of the CoolRunner-II clock divider with the delay enabled and disabled.

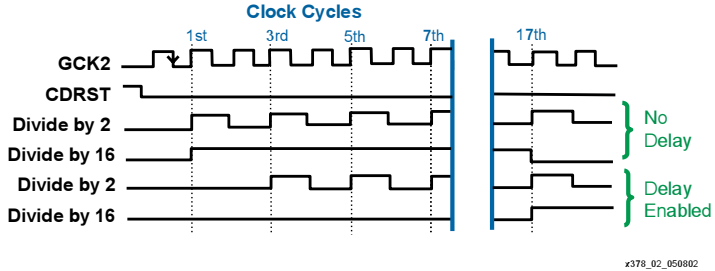


Figure 2: **Clock Divider Waveform**

Xilinx Synthesis Technology (XST) allows a clock divider component to be instantiated directly in the HDL source code. Table 2 lists the available clock divider components that can be instantiated in any ABEL, HDL, or schematic design.

Table 2: **Clock Divider Library Components**

Component	Description
CLK_DIVn	Global Clock Divider Component. No support of the synchronous reset or start delay features. Available: CLK_DIV2, 4, 6, 8, 10, 12, 14, 16
CLK_DIVnR	Global Clock Divider with Synchronous Reset. No support of the start delay feature. Available: CLK_DIV2, 4, 6, 8, 10, 12, 14, 16R
CLK_DIVnSD	Global Clock Divider with Start Delay. No support of the synchronous reset. Available: CLK_DIV2, 4, 6, 8, 10, 12, 14, 16SD
CLK_DIVnRSD	Global Clock Divider with Synchronous Reset and Start Delay. Available: CLK_DIV2, 4, 6, 8, 10, 12, 14, 16RSD

VHDL Example

To design with the CoolRunner-II clock divider in VHDL requires both a component declaration and component instantiation. The component declaration declares the name and interface of the clock divider unit. The VHDL component declaration syntax for using a clock divide by 2, the CLK_DIV2 component is shown here.

```
component CLK_DIV2 is
port (
  CLKIN : in STD_LOGIC;
  CLKDV : out STD_LOGIC );
end component;
```

The component instantiation associates signals with the ports of the clock divider component. If a clock divide by 2 is desired, the CLK_DIV2 component must be instantiated. The incoming clock signal, *clk*, is declared on the CLKIN port and the clock divider output signal, *clk_div_by_2*, is declared on the CLKDV output port. The VHDL syntax is shown here for instantiating the CLK_DIV2 component.

```
U1: CLK_DIV2
port map(
  CLKIN => clk,
  CLKDV => clk_div_by_2 );
```

If a clock divide by 16 with a synchronous reset and start delay is desired, the CLK_DIV16RSD component must be declared and instantiated. The VHDL syntax for the component declaration is shown here.

```
component CLK_DIV16RSD is
port (
  CLKIN : in STD_LOGIC;
  CDRST : in STD_LOGIC;
  CLKDV : out STD_LOGIC );
end component;
```

The component instantiation assigns the port signals. The input clock signal, *clk*, is declared on the CLKIN port. The clock divider reset signal, *clk_div_rst*, is declared on the CDRST port. The clock divider output, *clk_div_by_16*, is declared on the CLKDV port. This component will also enable the start delay circuitry in the CoolRunner-II clock divider. The syntax to instantiate the CLK_DIV16RSD component in VHDL is shown here.

```
U1: CLK_DIV16RSD
port map (
  CLKIN => clk,
  CDRST => clk_div_rst,
  CLKDV => clk_div_by_16 );
```

Verilog Example

Verilog design entry with XST does not require a component declaration; only the component instantiation is necessary. The Verilog syntax to instantiate the CLK_DIV16RSD component is

shown here. The input clock signal, *clk*, is assigned to the CLKIN port. The clock divider reset signal, *clk_div_rst*, is assigned to the CDRST port. The clock divider output, *clk_div_by_16*, is assigned to the CLKDV port.

```
CLK_DIV16RSD U1 (
  .CLKIN (clk),
  .CDRST (clk_div_rst),
  .CLKDV (clk_div_by_16) );
```

ABEL Example

Designing with clock dividers in CoolRunner-II requires both the component declaration and component instantiation. The clock divider must be declared as an external component in an ABEL design as shown here.

```
CLK_DIV2R external (CLKIN, CDRST -> CLKDV);
```

Component instantiation in an ABEL design occurs by assigning an identifier to the clock divider component. In this example, U1 is the identier assigned to the clock divider component, CLK_DIV2R, using the functional_block ABEL keyword.

```
U1 functional_block CLK_DIV2R;
```

The following equations illustrate the component port mapping. The input clock, *clk*, is mapped to the CLKIN port. The clock divider reset signal, *clk_div_rst*, is mapped to the CDRST port. The clock divider output signal, *clk_div_by_2*, is assigned to the CLKDV output port.

```
U1.CLKIN = clk;
U1.CDRST = clk_div_rst;
clk_div_by_2 = U1.CLKDV;
```

Notes:

1. The signal assigned to the CDRST port will automatically be mapped to the CDRST/I/O pin.
2. The output of the clock divider circuit is only available as a clock input to registers within the CPLD. The clock divider output can not be used in combinational logic or routed off-chip for external use.
3. The clock divider is available on CoolRunner-II 128 macrocell devices and larger.

CoolCLOCK

The CoolRunner-II CoolCLOCK feature is the technique of combining the global clock divider and DET registers. Power savings are achieved by dividing the global clock by 2, distributing a lower frequency clock on the internal clock network, and then doubling the clock at each macrocell. Zero clock skew can be acheived due to the zero insertion delay of the clock divider and the DET registers. Figure 3 illustrates the CoolRunner-II CoolCLOCK feature.

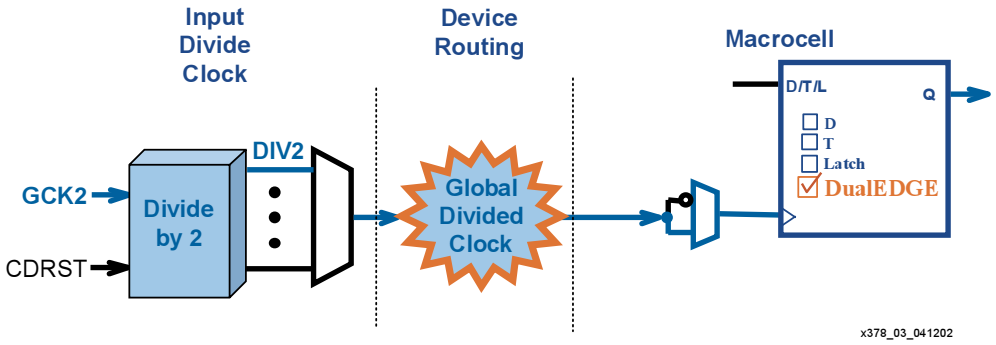


Figure 3: **CoolCLOCK**

Since GCK2 is the only clock network that can be divided, the CoolCLOCK feature is only available on GCK2. The CoolCLOCK feature can be implemented by assigning an attribute to an input

clock. The CoolCLOCK attribute replaces the need to instantiate the clock divider and infer DET registers. Table 3 lists the methods available to use the CoolCLOCK attribute..

Table 3: **CoolCLOCK Attribute**

Attribute Format	Syntax	Example
UCF	NET *<clock name>* COOL_CLK;	NET clk COOL_CLK;
ABEL	XILINX PROPERTY 'COOL_CLK *<clock name>*';	XILINX PROPERTY 'COOL_CLK clk';
VHDL	attribute COOL_CLK : string; attribute COOL_CLK of *<clock name>*: signal is "TRUE";	attribute COOL_CLK : string; attribute COOL_CLK of clk : signal is "TRUE";
Verilog	//SYNTHESIS attribute COOL_CLK of *<clock name>*: signal is "TRUE"; Note: The comment delimiters are intentional and necessary for XST.	//SYNTHESIS attribute COOL_CLK of clk : signal is "TRUE";

Note: The CoolCLOCK feature is available on CoolRunner-II 128 macrocell devices and larger.

DataGATE

CoolRunner-II designers can block specified inputs under the control of the DataGATE function. By blocking inputs, switching signals do not drive internal chip capacitance and thereby reduce overall power consumption. The last value on the input pin prior to the assertion of the DataGATE rail is latched and used by the CPLD internally. Figure 4 illustrates the DataGATE feature in CoolRunner-II.

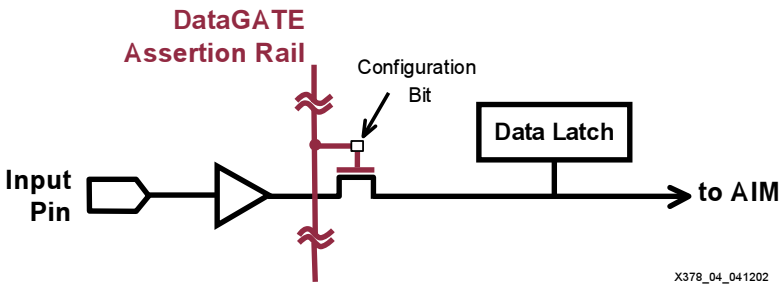


Figure 4: **DataGATE Block Diagram**

There are two attributes associated with the DataGATE feature in CoolRunner-II. The first attribute specifies if an input will be affected by DataGATE and the second designates the DataGATE control signal.

The DataGATE feature is selectable on a per pin basis. Each input pin that uses DataGATE must be assigned a DATA_GATE attribute. Table 4 illustrates the syntax for enabling DataGATE on an input signal.

Table 4: **DataGate Attribute**

Attribute Format	Syntax	Example
UCF	NET *<signal name>* DATA_GATE;	NET data_in DATA_GATE;
ABEL	XILINX PROPERTY 'DATA_GATE *<signal name>*';	XILINX PROPERTY 'DATA_GATE data_in';
VHDL	attribute DATA_GATE : STRING; attribute DATA_GATE of *<signal name>*: signal is "TRUE"; Note: The string attribute need only be declared once for all DATA_GATE attributes.	attribute DATA_GATE : STRING; attribute DATA_GATE of data_in : signal is "TRUE";
Verilog	//SYNTHESIS attribute DATA_GATE of *<signal name>*: signal is "TRUE"; Note: The comment delimiters are intentional and necessary for XST.	//SYNTHESIS attribute DATA_GATE of data_in : signal is "TRUE";

The DataGATE assertion rail can be driven from either an I/O pin or internal logic. The DataGATE enable signal is a dedicated DGE/I/O pin for each package in CoolRunner-II. Upon implementation, the software recongnizes a design using DataGATE and automatically assigns this I/O pin to the DataGATE enable control function, DGE. Internally generated DataGATE contol

logic can be assigned to this I/O pin with the BUFG=DATA_GATE attribute. The methods of assigning the DataGATE enable signal are shown in Table 5.

Table 5: **DataGate Control Attribute**

Attribute Format	Syntax	Example
UCF	NET *<signal name>* BUFG=DATA_GATE;	NET dg_en BUFG=DATA_GATE;
ABEL	XILINX PROPERTY 'BUFG=DATA_GATE *<signal name>*';	XILINX PROPERTY 'BUFG=DATA_GATE dg_en';
VHDL	attribute BUFG : STRING; attribute BUFG of *<signal name>*: signal is "DATA_GATE"; Note: The string attribute need only be declared once for all BUFG attributes.	attribute BUFG : STRING; attribute BUFG of dg_en : signal is "DATA_GATE";
Verilog	//SYNTHESIS attribute BUFG of *<signal name>*: signal is "DATA_GATE"; Note: The comment delimiters are intentional and necessary for XST.	//SYNTHESIS attribute BUFG of dg_en : signal is "DATA_GATE";

Schmitt Trigger

Each CoolRunner-II I/O has multiple input buffers used for various I/O standard configurations. One of these input buffers behaves as a Schmitt trigger input and is enabled upon CPLD configuration. The Schmitt trigger input allows the board designer the flexibility to utilize the CoolRunner-II with both high speed signals as well as slow switching signals on the same device. Slowly switching signals can cause havoc on digital systems by causing double clocking or glitches on a CMOS input, however this can be virtually eliminated by using the Schmitt trigger input. The Schmitt trigger input use is only at the cost of a few nanosecond delay (refer to the CoolRunner-II datasheet, see **References**, page 10).

Table 6 illustrates the attribute syntax to assign the Schmitt trigger input buffer to a specific signal..

Table 6: **Schmitt Trigger Attribute**

Attribute Format	Syntax	Example
UCF	NET *<signal name>* SCHMITT_TRIGGER;	NET data_in SCHMITT_TRIGGER; NET clock SCHMITT_TRIGGER;
ABEL	XILINX PROPERTY 'SCHMITT_TRIGGER *<signal name>*';	XILINX PROPERTY 'SCHMITT_TRIGGER data_in'; XILINX PROPERTY 'SCHMITT_TRIGGER clock';
VHDL	attribute SCHMITT_TRIGGER : STRING; attribute SCHMITT_TRIGGER of *<signal name>*: signal is "TRUE"; Note: The string attribute need only be declared once for all SCHMITT_TRIGGER attributes.	attribute SCHMITT_TRIGGER : STRING; attribute SCHMITT_TRIGGER of data_in: signal is "TRUE"; attribute SCHMITT_TRIGGER of clock: signal is "TRUE";
Verilog	//SYNTHESIS attribute SCHMITT_TRIGGER of *<signal name>*; Note: The comment delimiters are intentional and necessary for XST.	//SYNTHESIS attribute SCHMITT_TRIGGER of data_in; //SYNTHESIS attribute SCHMITT_TRIGGER of clock;

I/O Termination

CoolRunner-II pins may be terminated in the following ways: keeper (also referred to as bushold) and pullup. Usage of the keeper and the pullup circuitry is exclusive on a global basis. When one of these two (keeper and pullup) termination modes is selected for any number of signals, the other termination mode is no longer available to any other signal.

Keeper

The keeper circuitry provides the ability to hold the last known value on an I/O pin using weak pullup/down resistors. If an unterminated I/O pin was in high-impedence and floating, this would cause excessive leakage current. The keeper circuitry eliminates the need for external termination

that would resolve this. Table 7 illustrates the attribute syntax for specifying the keeper termination on any I/O pin.

Table 7: **Keeper Attribute**

Attribute Format	Syntax	Example
UCF	NET *<signal name>* KEEPER;	NET data_in KEEPER; NET clock KEEPER;
ABEL	XILINX PROPERTY 'KEEPER *<signal name>*';	XILINX PROPERTY 'KEEPER data_in'; XILINX PROPERTY 'KEEPER clock';
VHDL	attribute KEEPER : STRING; attribute KEEPER of *<signal name>*: signal is "TRUE"; Note: The string attribute need only be declared once for all KEEPER attributes.	attribute KEEPER : STRING; attribute KEEPER of data_in: signal is "TRUE"; attribute KEEPER of clock: signal is "TRUE";
Verilog	//SYNTHESIS attribute KEEPER of *<signal name>*; Note: The comment delimiters are intentional and necessary for XST.	//SYNTHESIS attribute KEEPER of data_in; //SYNTHESIS attribute KEEPER of clock;

Pullup

The internal pullup allows the designer to eliminate external pullup resistors on the board, thereby reducing cost and simplifying board layout. Table 8 illustrates the attribute syntax for specifying the pullup I/O termination.

Table 8: **Pullup Attribute**

Attribute Format	Syntax	Example
UCF	NET *<signal name>* PULLUP;	NET data_in PULLUP; NET clock PULLUP;
ABEL	XILINX PROPERTY 'PULLUP *<signal name>*';	XILINX PROPERTY 'PULLUP data_in'; XILINX PROPERTY 'PULLUP clock';
VHDL	attribute PULLUP : STRING; attribute PULLUP of *<signal name>*: signal is "TRUE"; Note: The string attribute need only be declared once for all PULLUP attributes.	attribute PULLUP : STRING; attribute PULLUP of data_in: signal is "TRUE"; attribute PULLUP of clock: signal is "TRUE";
Verilog	//SYNTHESIS attribute PULLUP of *<signal name>*; Note: The comment delimiters are intentional and necessary for XST.	//SYNTHESIS attribute PULLUP of data_in; //SYNTHESIS attribute PULLUP of clock;

I/O Configuration

CoolRunner-II devices support multiple I/O banks in a single device, allowing for easy interfacing to different voltage standards on one chip. A device can support one I/O standard per bank (i.e., XC2C128 has two banks and can therefore support up to two I/O standards). Regardless of which I/O voltage standard is selected, any pin may be configured as an open-drain output.

I/O Standards

Table 9 lists the supported I/O standards on CoolRunner-II devices. Note that all standards are not supported in every density.

Table 9: **CoolRunner-II Supported I/O Standards**

	XC2C32A	XC2C64A	XC2C128	XC2C256	XC2C384	XC2C512
I/O Banks	2	2	2	2	4	4
LVTTL	Yes	Yes	Yes	Yes	Yes	Yes
LVCMOS33, LVCMOS25, & LVCMOS18	Yes	Yes	Yes	Yes	Yes	Yes

Table 9: **CoolRunner-II Supported I/O Standards**

	XC2C32A	XC2C64A	XC2C128	XC2C256	XC2C384	XC2C512
1.5V I/Os	Yes	Yes	Yes	Yes	Yes	Yes
SSTL2-1 & SSTL3-1	No	No	Yes	Yes	Yes	Yes
HSTL-1	No	No	Yes	Yes	Yes	Yes

Figure 5 illustrates how to specify the default I/O standard for all pins in a design. The I/O standard can be selected in the Implement Design Process Properties window under the Basic Tab.

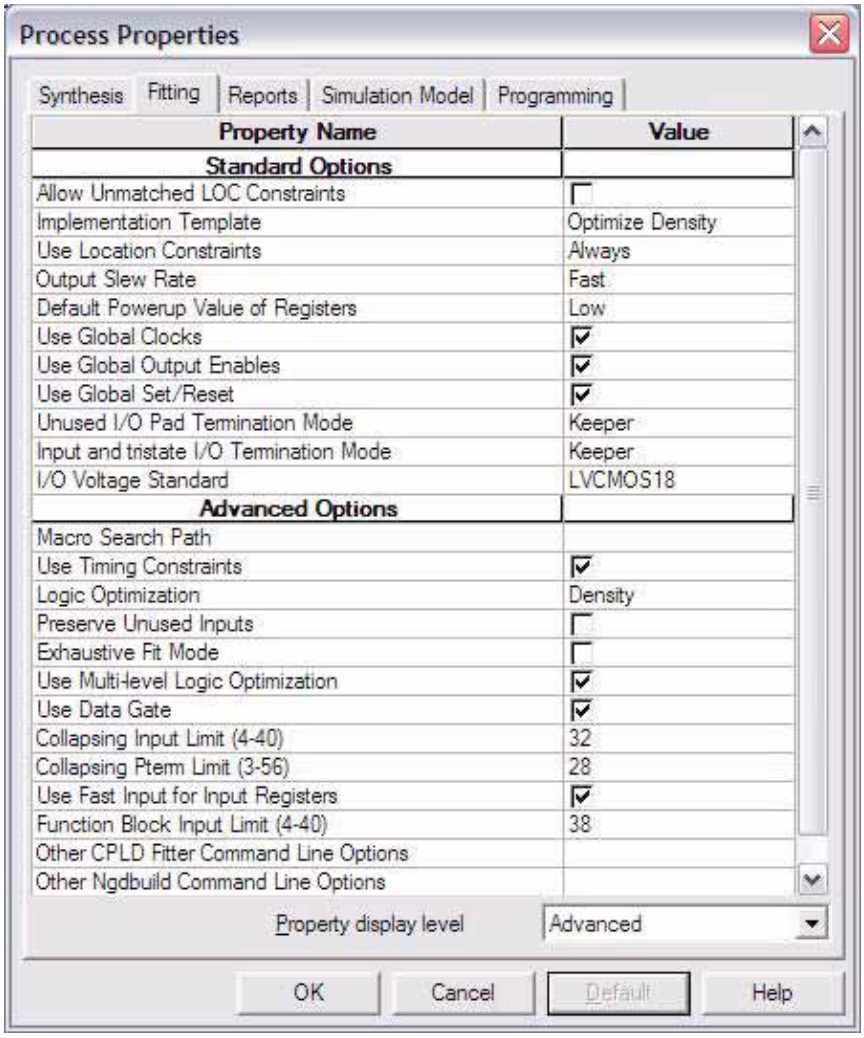


Figure 5: **Global I/O Standard Selection**

If a design requires multiple I/O standards on the same device, each pin must be manually declared with the appropriate I/O standard attribute. Table 10 illustrates the available I/O standard attributes that can be declared.

Table 10: **I/O Standard Attributes**

I/O Standard	Attribute Name
LVTTL	LVTTL
LVCMOS 3.3V	LVCMOS33
LVCMOS 2.5V	LVCMOS25
LVCMOS 1.8V	LVCMOS18
1.5V I/O	LVCMOS15

Table 10: **I/O Standard Attributes**

I/O Standard	Attribute Name
SSTL 2-1	SSTL2_I
SSTL 3-1	SSTL3_I
HSTL-1	HSTL_I

Table 11 illustrates the syntax for specifying an I/O standard attribute. The examples shown in Table 11 are for specifying the LVCMOS18 I/O standard. For other standards, LVCMOS18 can be replaced with the appropriate attribute name shown in Table 10.

Table 11: **I/O Standard Attribute Syntax**

Attribute Format	Syntax	Example
UCF	NET *<signal name>* *<I/O standard attribute name>*;	NET data_in IOSTANDARD=LVCMOS18; NET clock IOSTANDARD=LVCMOS18;
ABEL	XILINX PROPERTY 'IOSTANDARD=LVCMOS18 *<signal name>*';	XILINX PROPERTY 'IOSTANDARD=LVCMOS18 data_in'; XILINX PROPERTY 'IOSTANDARD=LVCMOS18 clock';
VHDL	attribute IOSTANDARD : STRING; attribute IOSTANDARD of *<signal name>*: signal is "*<I/O standard attribute name>*"; Note: The string attribute need only be declared once for all IOSTANDARD attributes.	attribute IOSTANDARD : STRING; attribute IOSTANDARD of data_in: signal is "LVCMOS18"; attribute IOSTANDARD of clock: signal is "LVCMOS18";
Verilog	//SYNTHESIS attribute IOSTANDARD of *<signal name>* is "*<I/O standard attribute name>*"; Note: The comment delimiters are intentional and necessary for XST.	//SYNTHESIS attribute IOSTANDARD of data_in is "LVCMOS18"; //SYNTHESIS attribute IOSTANDARD of clock is "LVCMOS18";

Open Drain

A signal that is grounded when "false" and is in high-impedence when "true" is considered an open-drain signal. An output on CoolRunner-II can be configured as open drain by simply declaring the OPEN_DRAIN attribute in the Xilinx software. Table 12 illustrates the syntax for specifing an open drain output with the OPEN_DRAIN attribute.

Table 12: **Open Drain Attribute**

Attribute Format	Syntax	Example
UCF	NET *<signal name>* OPEN_DRAIN;	NET data_out OPEN_DRAIN;
ABEL	XILINX PROPERTY 'OPEN_DRAIN *<signal name>*';	XILINX PROPERTY 'OPEN_DRAIN data_out';
VHDL	attribute OPEN_DRAIN : STRING; attribute OPEN_DRAIN of *<signal name>*: signal is "TRUE"; Note: The string attribute need only be declared once for all OPEN_DRAIN attributes.	attribute OPEN_DRAIN : STRING; attribute OPEN_DRAIN of data_out: signal is "TRUE";
Verilog	//SYNTHESIS attribute OPEN_DRAIN of *<signal name>*; Note: The comment delimiters are intentional and necessary for XST.	//SYNTHESIS attribute OPEN_DRAIN of data_out;

Code Examples Download

Example source code is available for download. Both VHDL and Verilog code with test benches are available for using the CoolRunner-II advanced features.

THE DESIGNS ARE PROVIDED TO YOU "AS IS". XILINX MAKES AND YOU RECEIVE NO WARRANTIES OR CONDITIONS, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, AND XILINX SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. These are only example designs, not fully functional cores. XILINX does not warrant the performance, functionality, or operation of these designs will meet your requirements, or that the operation of the designs will be uninterrupted or error free, or that defects in the designs will be corrected. Furthermore, XILINX

does not warrant or make any representations regarding use or the results of the use of the designs in terms of correctness, accuracy, reliability or otherwise.
XAPP378 - http://www.xilinx.com/products/xaw/coolvhdlq.htm

Conclusion

Performance and low power have finally come together with CoolRunner-II CPLDs. The available advanced features furthur reduce power consumption and provide advanced clocking management options. For additional assistance with the CoolRunner-II advanced features, please contact the Xilinx hotline or refer to the Xilinx web support (http://support.xilinx.com/).

References

1. CoolRunner-II family data sheet
2. CoolRunner-II Application Notes
3. Xilinx ISE design entry software
4. Application Note: XAPP352: Utilizing a UCF for CoolRunner XPLA3 CPLDs

XAPP377 (v1.0) May 8, 2002

Low Power Design with CoolRunner-II CPLDs

Summary

CoolRunner™-II CPLDs are the only CPLD to combine both high performance and low power to form the next generation CPLD. This application note describes the design methodologies that can be employed to obtain the lowest power possible using the CoolRunner-II CPLD by utilizing its unique power saving features.

Introduction

CoolRunner-II CPLDs continue to employ the features of Fast Zero Power (FZP) technology as found in the earlier CoolRunner CPLD generations. But due to unique Xilinx design techniques, smaller geometries, and state of the art process technology, FZP technology has further advanced CoolRunner-II CPLDs as the low power standard. Other CPLD manufacturers have attempted to reproduce the FZP true CMOS technology of the CoolRunner CPLDs, but have not been able to meet the CoolRunner benchmark. For the first time in the CPLD industry, CoolRunner-II devices deliver both true high performance and low power at the same time, along with the lowest standby current in the industry without the use of power down modes. In addition, these devices offer unique power saving features such as CoolCLOCK, DataGATE, and DualEDGE flip-flops.

Traditional CPLDs use sense amp type product terms to provide fast propagation delay times. Sense amp product terms are biased in a manner to detect a small change in voltage levels on the word line which indicates a change in the logic state of the product term. The transistor biasing constantly draws current, even at standby. With this in mind, these types of CPLDs cannot provide a low power solution, as can CoolRunner-II CPLDs. As sense amp type device sizes become larger in macrocell count, power grows significantly since there are many more product terms to consume power.

As process technology shrinks, sense amp type CPLDs will inherently consume more power to maintain performance. Smaller process technology demands lower power supply voltages thereby reducing the gain of the sense amp. Further, product term transistors leak more with the smaller geometries. To maintain performance, a sense amp CPLD will need to be designed such that its biasing compensates for the leakage and boosts gain to detect the smaller voltage swing of the word line. Higher biasing will cause more current to flow through the bias network, thereby increasing total power consumption. Other CPLD manufacturers that use sense amp based technology will inevitably go through a learning process to re-design their current products to compensate for the effect of ever shrinking process technology.

CMOS product terms, as used in the FZP technology, inherently consume less power when not switching states. Since CMOS logic exhibits low standby current, CoolRunner-II CPLDs use this technology to reap the benefits of low power. Additionally, CMOS technology benefits from smaller geometries where the device consumes less power and becomes faster.

Power Saving Features

New architectural features have been added to the CoolRunner-II product line to enhance the power saving capabilities of the FZP technology. This section describes these features and how they can be used to save power.

All new features described below are available with the XC2C128 and larger devices. The smaller devices, XC2C32 and XC2C64, do not include some features, specifically DataGATE, Clock Divider, and CoolCLOCK.

DataGATE

Many times devices are connected to a data bus which are not being addressed by the master device. When a CPLD is in this situation, it will use more power than necessary since the data on the bus is not useful to the CPLD, but the data lines continue to toggle, which subsequently toggles the internal logic of the CPLD. Any logic that changes state within the CPLD will consume power. Therefore, it follows that disconnecting the CPLD from the data bus when the device is not addressed conserves power, as highlighted in this example.

DataGATE solves this issue by disconnecting external signal activity from the internal logic, consequently reducing power consumption of the device. This is achieved by using a unique, software enabled, CoolRunner-II pass gate at the I/O pin (when configured as an input) which is controlled by the DataGATE global net. This control net originates from a specific pin/macrocell and can be driven by either an external signal or an internally developed signal using logic elements. When the pass gate on the input pin is disabled by the DataGATE control net, an internal latch drives the CPLD logic network maintaining the same logic level that was present on the input pin just prior to asserting the DataGATE control net. This preserves the current logic state internal to the CPLD while external data changes states.

For example, a CoolRunner-II CPLD that shares a data bus with other devices will most likely have its own address and typically will not be addressed continuously. Two options exist to disconnect the data bus from the CoolRunner-II CPLD when not addressed. First, if the device is addressed using a chip select signal, this signal can be assigned to the DataGATE control pin and used to isolate the inputs from the data bus. Second, if using an address bus, the address of the device can be decoded internally to the CPLD which can then be used to enable, via DataGATE, the data bus inputs to receive data when addressed. When the CPLD is not addressed, DataGATE would disconnect the external data bus signals using address decoding internally to the CPLD. In this case, the result of the internal decoding is routed to the DataGATE pin to disconnect the toggling bus.

DataGATE can be configured to affect any or all I/O pins, with the exception of JTAG pins and the DataGATE pin itself. The above example discusses DataGATE configured to affect a few macrocell I/O pins. It may be beneficial in other applications to disconnect the system clock or clocks from the CPLD. The two elements that consume the most power are output buffers and the clock tree. If the CPLD is not needed for a specific amount of time, the DataGATE feature could be used to gate the clock. Doing so will dramatically reduce power consumption. However, caution must be exercised when gating a clock since undesired logic transitions may occur.

There are other cases where the unique CoolRunner-II DataGATE feature is useful to reduce power. Also, DataGATE is flexible such that the entire device or only parts of the device can be isolated from external signals, depending on the application. For example, it can be used in CoolRunner-II CPLDs to enhance board troubleshooting procedures.

Schmitt Trigger

CoolRunner-II Schmitt trigger inputs are useful for applications that require hysteresis on the inputs. A useful application might be where noise is an issue on specific pins. Hysteresis provides added noise immunity. Another application could implement an oscillator circuit which is constructed external to the CPLD, but CPLD logic is used to implement portions of the oscillator circuit.

To ensure the lowest power consumption in the CoolRunner-II CPLDs, disable the Schmitt trigger inputs since these types of buffers consume more power than the regular input buffer. It is important to design and operate the system in a low noise environment when Schmitt triggers are disabled to prevent inadvertent transitions on the CPLD inputs.

Clock Divider

Global clock networks tend to be the largest power consuming elements in CPLDs. Any effort to reduce the frequency of the global clock network greatly benefits the system with respect to power consumption. Therefore, CoolRunner-II devices have been designed to include a clock divider network on global clock, GCK2. Without introducing additional clock delays, the clock

divider has the capability of dividing the system clock by even integers ranging from 2 to 16, as shown in Figure 1.

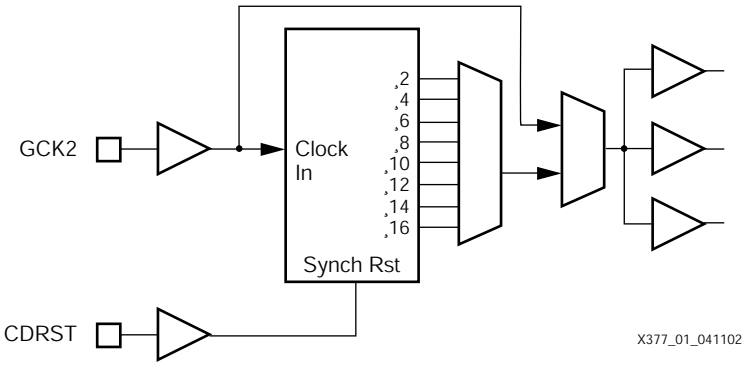


Figure 1: **Clock Division Circuitry for GCK2**

Some systems use state machines, for example, that do not require the full speed of the external system clock. A clock divider is the perfect tool to reduce system power in this case. The clock divider provides an excellent alternative to adding a user defined clock divider built from logic, which would waste logic otherwise usable for more features in the design. A lower frequency on the global clock network, provided by the clock divider, will reduce the power consumed by the CoolRunner-II CPLD.

Generally speaking, designing with the slowest system clock possible will reduce power consumption. To this end, the clock divider provided in the CoolRunner-II architecture will greatly assist the designer.

DualEDGE Registers

By utilizing both edges of the clock signal, the macrocell can do twice the work when configured as a DualEDGE flip-flop. Figure 2 displays the macrocell configured as a DualEDGE flip-flop. A system without the aid of the DualEDGE flip-flop would need to provide a clock at twice the frequency to obtain the same work output at the macrocell. Since the macrocells with DualEDGE flip-flops operate on both the rising and falling edges of the clock, the clock network is used more efficiently. Consequently, power consumption is reduced when the global clock net is operating at a lower frequency.

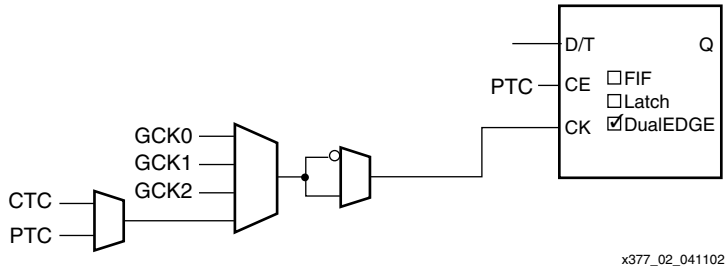


Figure 2: **Macrocell Clock Chain with DualEDGE Option Shown**

DualEDGE flip-flops further enhance the functional possibilities of the clock divider and therefore improve power savings. The global clock can be effectively divided by odd integers of 3, 5, and 7 if used with the DualEDGE flip-flop. For example, if a divide by 3 clock is desired for the design, the Clock Divider can be set to a divide by 6 and then effectively doubled by the DualEDGE flip-flop resulting in a divide by 3 characteristic. This is important to note since, again, lower clock frequencies always save power. The DualEDGE flip-flop effectively adds more functionality to the clock divider network.

Perhaps the design contains two state machines. One state machine can efficiently operate at 1/4th the system clock frequency, yet the other state machine can much more efficiently operate at 1/8th the system clock frequency. DualEDGE flip-flops can be employed in conjunction with the clock divider to obtain such a scenario. The state machine running at 1/8th

the system clock frequency can simply use the clock divider configured as divide by 8. Enabling the DualEDGE flip-flop on macrocells assigned to the other state machine and assigning the divide by 8 clock divider to those macrocells as well will yield an effective clock frequency that is 1/4th the system clock frequency. This means that the single clock divider can obtain virtual dual functionality of a divide by 8 and a divide by 4 counter. Notice that both clocks are synchronized with each other and the system clock so that the two state machines operate concurrently.

Summarizing the previous discussion, when utilizing the CoolRunner-II clock divider and/or the DualEDGE flip-flops, it is possible to obtain clock divisors of 2, 3, 4, 5, 6, 7, 8, 10, 12, 14 and 16. Additionally, when a group of macrocells use the clock divider and some of those macrocells use DualEDGE flip-flops, the clock divider network can effectively deliver two clock frequencies of divisors 1-2 (using CoolCLOCK described later), 2-4, 3-6, 4-8, 5-10, 6-12, 7-14, and 8-16. When DualEDGE flip-flops are used with the clock divider, lower power is achieved while more efficiently operating the design.

CoolCLOCK

CoolRunner-II CPLDs are equipped with a unique feature, CoolCLOCK, to further reduce the power consumption of the global clock network without affecting the speed of the clock. Note that CoolCLOCK does not impose additional clock delays. As explained earlier, any efforts to reduce the clock frequency of the global clock net will significantly reduce power consumption.

CoolCLOCK reduces power consumed by the global clock net by dividing the external clock frequency by 2 before it is applied to the global clock network. This clock division occurs early in the clock tree near the clock input buffer so that the divided clock affects the majority of the clock network. Since the global clock network contains a relatively large amount of internal capacitance, a slower frequency will significantly reduce power consumed by this net, GCK2. This divided clock signal is then effectively doubled at the macrocell using the DualEDGE clocking feature of the flip-flops in the macrocell, shown in Figure 3. This ensures that the original clock frequency is applied to the macrocell as intended by the external system. Only macrocells that require the original clock frequency will be configured to utilize the DualEDGE flip-flop feature. Other macrocells can be configured to use the divided clock frequency to further reduce power when those macrocells don't require clocking at full speed.

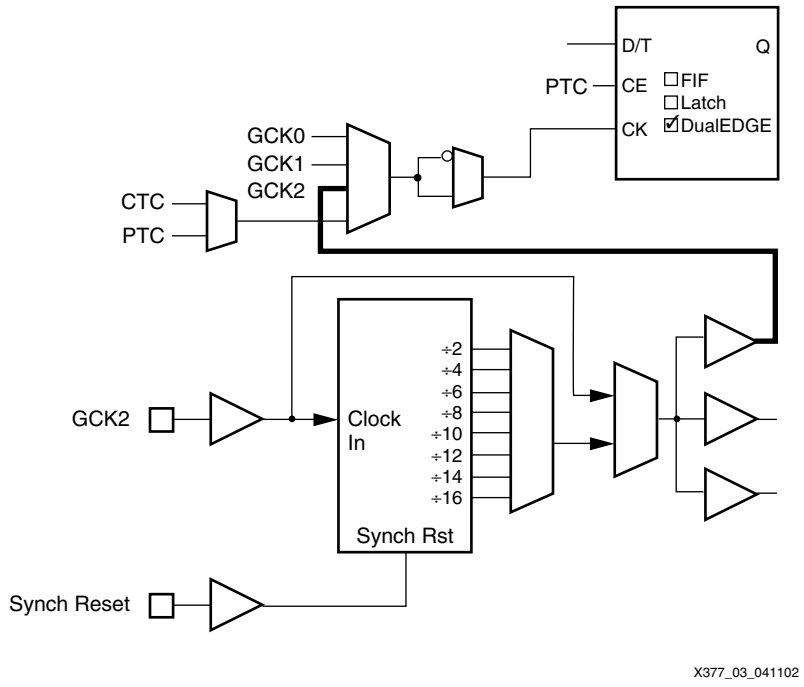


Figure 3: **CoolCLOCK Created by Cascading Clock Divider and DualEDGE Option**

Weak Pull-up and Bus Hold

All CoolRunner-II CPLDs include I/O termination options to reduce power consumption of the I/O due to externally 3-stated busses. The I/O termination circuitry can be configured in three ways: weak pull-up, bus hold, and no termination. Pull-up and bus hold are selected on a global basis and are mutually exclusive. Subsequently, the use or absence of the selected termination is specified on a per pin basis. Weak pull-up connects a high-impedance resistive load onto the I/O pin to prevent a floating situation on the pin. Bus hold is essentially a full latch on the I/O pin which drives the last state present on the I/O, either High or Low, prior to the bus going to the high impedance state. Bus hold is referred to in the software as "Keeper".

Floating inputs, an input which is not driven to a High or Low logic state, can use excessive power since the voltage on the gate of the input buffer may wander to a voltage level between standard logic levels. In this case, the input buffer is driven into the linear region where the P and N channel transistors are both switched on. Therefore, to avoid this situation, I/Os can be configured to use internal pull-ups or bus hold circuitry. I/Os configured as inputs or bidirectional should utilize this feature if it is known that the bus to which the I/O is attached will float at some point in its regular operation.

There are some cases where weak pull-up is undesirable. If the bus is pulled down the majority of time, current will flow through the weak pull-up to ground via the buffer, pulling the bus Low. A design such as this would benefit by using the bus hold circuitry.

A rule of thumb for any CMOS device is to not allow inputs to float. These two features of CoolRunner-II CPLDs, weak pull-up and bus hold, will minimize the chance of consuming excessive power on input pins.

I/O termination should be considered for each application and each I/O to determine the best combination to reduce power consumption. Again, avoid floating inputs whenever possible.

Slew Rate

This is a feature of the I/O structure that regulates the rate at which the output buffer changes states. There are two modes: FAST and SLOW. Designers concerned with reducing reflections on circuit board traces, minimizing RFI or minimizing EMI should specify a SLOW slew rate. Most design engineers considering low power designs will usually be designing slower speed systems and therefore will not be as concerned with reflections. In a case such as this, the system will benefit from a lower power perspective when slew rate is specified to be FAST.

Although an I/O is configured as an output, the input buffer is still connected to the pin and therefore senses changes in voltage on that pin. If the output is configured with a SLOW slew rate, the output voltage switches states less quickly, thereby lingering longer between standard logic levels. The input buffer will therefore be driven in the linear region (the input voltage between standard logic levels) for a longer period of time than if the output was configured in the FAST slew rate mode. Current will flow through the input buffer P and N channel transistors for a longer period of time resulting in higher power consumption. It is therefore recommended to configure the output with a FAST slew rate whenever reflections are not a concern.

Power Saving Techniques

Several rules of thumb should be followed when designing any circuit with CMOS devices to reduce power consumption. A basic understanding of power consumption must be reached prior to discussing these rules of thumb. Therefore, a derivation of the current equation for CMOS devices is necessary. Once the mathematical model of current is understood, it becomes easy to follow the theory of the rules of thumb. To maximize power savings, the designer should apply these concepts when implementing any CMOS device.

Derivation of Current Equation in CMOS Devices

Since the CMOS device is constructed of PMOS and NMOS transistors, the dynamic model is simply a capacitive structure for each transistor. Of course, the basic structure of a capacitor is the dielectric between two plates. In this case, the dielectric of the capacitor is the oxide layer on the silicon wafer and the plates are the poly or metal gate together with the inversion layer in the channel. The interconnecting metal/poly routing is also modeled as a capacitor where the plates are the routing itself together with any underlying routing or conductive material and the dielectric is any non-conductive structure between the plates. With these capacitive structures,

the CMOS device is largely modeled as a collection of capacitors. Recall the basic equation for current through a capacitor:

$$I = C \cdot \frac{dV}{dt}$$

Breaking the derivative into its components we can extract the basic equation for frequency, which is the inverted value of a change in time, and simplify the equation:

$$I = C \cdot dV \cdot \frac{1}{dt}$$

$$I = C \cdot dV \cdot f$$

Since the voltage for capacitive structures in CMOS devices changes as a square wave with discontinuities between logic HighHigh and Low and is ideally rail to rail, we can further simplify the equation. For example, in a system of supply voltage, V, the change in voltage, dV, is V to 0V as shown and reduced here:

$$dV = V_2 - V_1$$

$$dV = V - 0$$

$$dV = V$$

Therefore the current equation for each capacitive structure becomes:

$$I = C \cdot V \cdot f$$

where:

- **I** = current in Amperes
- **C** = the capacitance of the capacitive structure in Farads
- **V** = the system voltage in Volts
- **f** = the toggle frequency of the capacitive structure in Hz

Dynamic device current is the summation of all capacitive structures toggling over time. Voltage remains the same for all equations and can be assumed to be a constant. A device of n capacitive structures can be represented as follows:

$$I_{Dynamic} = V \cdot \sum_{i=1}^{n} C_i \cdot f_i$$

To obtain total device current, static current must be added to the dynamic current:

$$I_{Total} = I_{Static} + V \cdot \sum_{i=1}^{n} C_i \cdot f_i$$

For illustrational purposes, it may be easier to discuss total current using a simplified version of this equation:

$$I_{CC} = I_{CCQ} + C \cdot V \cdot f$$

where:

- **I_{CC}** = *total* device current in Amperes
- **I_{CCQ}** = *quiescent* device current in Amperes
- **C** = the *lumped* capacitance of the device in Farads
- **V** = the system voltage in Volts
- **f** = the *average* device toggle frequency in Hz

Recall that to obtain power, multiply current by voltage to yield Watts.

Reduce System Speed

It becomes obvious from the equation that for a fixed device capacitance and voltage, reducing the average device toggle rate will reduce power consumption.

Limiting the system clock speed as well as the data bus speed to slower values will reduce power consumption since average device toggle rate will become smaller. Careful analysis of the required CPLD clock speed is essential for low power design. Over-clocking the CPLD design beyond the needs of the logic unnecessarily consumes extra power. Evaluating the minimum required speed of the CPLD logic will ensure that the system clock will have a minimal effect on power consumption.

Drive Inputs to Standard Logic Levels

Power consumption will rise dramatically when a CMOS input buffer is not driven to known, standard logic levels, otherwise known as allowing the input to "float". A voltage level between standard logic levels causes the input buffer transistors, typically P and N channel, to be biased in a manner where both are in the ON state. When biased in this way, a large amount of current will flow between the power and ground supplies of the device via this channel. It is therefore imperative to drive the input buffer to a known logic level High or Low state to turn off one of these two transistors and avoid this situation.

The beauty of CMOS logic is that power consumption is nearly zero when logic gates are held at a known logic input level. FZP technology found in CoolRunner-II CPLDs uses this principle to provide dramatic power savings over other CPLDs.

Further, it is advised to drive the input to the full voltage rail on a High logic level and fully to ground on a logic Low level. Even though the voltage level is within the acceptable logic levels, i.e., V_{IH} and V_{IL}, the closer the voltage is to the absolute voltage rails, the less current will be consumed in the input buffer. This effect is much less than the scenario where the input voltage floats between logic levels, but nevertheless can have a significant impact on total power consumption when summed across several I/Os.

Increase Input Edge Speed

CMOS device inputs that are driven by a slowly switching source will consume more power since the input spends more time biased in the linear region. Again, the linear region is found when an input is biased at a voltage between standard logic High or Low levels. When the CMOS input buffer is biased in this manner, both the P and N channel transistors are turned on, allowing a relatively large amount of current to pass to ground. The longer the buffer is biased in this fashion, the more power will be consumed by the device. Therefore, it is recommended to quickly switch the signal as it is applied to the inputs of the CMOS device. This applies to all clock pins, dedicated input pins, or I/Os configured as inputs or bidirectional.

Eliminate Bus Conflicts

Occasionally, bus conflicts occur where two output buffers are driving a line at the same time in opposite directions. This adversely affects logic performance as well as power consumption. Two drivers attempting to swing the bus at opposite voltage levels will draw excessive current.

A similar situation occurs when a bus is pulled High via a pull-up resistor, for example, and the output buffer is driving the bus Low. In this example, current will flow from the power source, through the pull-up resistor, the N channel transistor in the output buffer, and to ground. Current, in this case, is a function of the value of the pull-up resistor and the N channel impedance. A weak pull-up resistor, on the order of 10k ohms or more, is a good place to start if it is desired to use such a component. The larger the resistor, the less power will be consumed, but this will slow down the bus response time when the resistor is required to charge the bus to the High level if the bus is in the 3-state condition.

It is recommended to continuously drive a bus line High or Low with one device at a time and remove pull-up resistors whenever possible to obtain the lowest power condition. In some situations, this may not be possible. For example, an SMBus or I2C SDA line is required to be released by all components but be at a High level when released. In this case, a pull-up resistor is required.

Bus Terminations

A different case where pull down resistors are necessary is when shunt bus termination is required to reduce reflections in high speed designs. These terminations are usually designed to be pull down resistors whose value equals the equivalent impedance of the data bus transmission line and are positioned as close to the load pin as possible. Whenever the output buffer is driving a transmission line with shunt resistors, the P channel transistor will source current into the load and therefore will raise power consumption.

It may be possible to avoid using shunt termination resistors by using a very short transmission line. The rule of thumb for a short transmission line is one whose length is less than one-sixth the electrical length of the rise time, and is described using the following equation:

$$\text{Length} \ll \frac{1}{6} \cdot \frac{T_{rise}}{\sqrt{LC}}$$

where:

- **Length** = maximum line length in inches
- $\mathbf{T_{rise}}$ = rise time in seconds
- **L** = the line inductance in Henries/inch
- **C** = the line capacitance in Farads/inch

By using the CoolRunner-II CPLD slew rate feature configured to SLOW, the rise time becomes longer. Using the above equation, it can be determined if the length of the transmission line can be effectively long enough to avoid reflections altogether. If by using the SLOW slew rate feature, the length of the transmission line is short by the above rule of thumb, shunt bus termination resistors can be avoided thereby saving power.

If either method is not an option, insert a series termination resistor positioned at the source pin with the same value as the transmission line impedance. This option will avoid excessive power consumption (since there is no shunt load) and eliminate reflections at the source. However, a series termination resistor will allow one reflection at the load (since the load impedance does not match the transmission line impedance) which implies that any component mid way between the source and the load will see two transitions: the incident wave and the single reflected wave created at the load.

Reduce System Voltage

Using the total current equation with fixed capacitance and average frequency, it becomes readily apparent that reducing system voltage will reduce power consumption, provided the system voltage remains within recommended operating specifications. Therefore, it makes sense to use a 1.8V device in lieu of a 3.3V or 2.5V device. CoolRunner-II CPLDs are 1.8V devices and therefore utilize this concept. Further, a device made with a smaller process technology, such as CoolRunner-II CPLDs, will generally have lower lumped internal capacitance values thereby reaching additional power savings. Combining these two factors with reducing average system frequency will also cut power consumption.

With any voltage device, there is a recommended operating range, and it may be advantageous to operate low in that voltage range to further reduce power consumption. Voltages that are outside the recommended operating range may cause excessive power consumption including adverse functional performance.

Reduce Bus Loading

Connecting an external bus to a CMOS device will increase power consumption due to the loading effects of the bus on the device. The primary factor from loading is given by capacitive or resistive bus components as seen by the output buffer looking into the bus.

Capacitive loading comes in two forms: lumped and distributed. Lumped capacitance is typically found from the gate of the input buffer of other connected CMOS devices. It is also developed from any capacitive element that is attached to the bus. Distributed capacitance is present due to the routing of the trace on the PCB. Both types will charge and discharge during logic transitions based on the previous logic state of the bus. Capacitive loading will draw current from the CMOS device whose output buffer is driving the bus, thereby increasing apparent power consumption of that device as seen at its power pins. To minimize power

consumption based on this capacitive loading effect, it is necessary to reduce the size of the lumped capacitance found in attached devices, and to shorten the PCB traces. Doing so will also increase potential system speed since reflections are less likely.

Resistive loading is usually found when devices with a resistive element to their impedance is attached to the bus. For example, a pull down resistor attached to a PCB trace will allow for current to flow from the CMOS device power rail, through the P channel of the output buffer transistor, through the resistor, and to ground, increasing observed power consumption at the CMOS device power pins. Reducing resistive loads will reduce power consumption.

Keep in mind that many components are comprised of more than one type of impedance element. For example, capacitors also have resistive and inductive elements, albeit small.

Further Power Saving Techniques

For further discussions regarding power saving techniques, particularly those involving logic design, review XAPP346 - Low Power Tips for CoolRunner Design found at http://www.xilinx.com/xapp/xapp346.pdf. Although the referenced application note discusses CoolRunner XPLA3 CPLDs, the basic principles apply to CoolRunner-II CPLDs.

Conclusion

For the first time in the CPLD industry, CoolRunner-II products combine true high speed logic with ultra low power. Unique features of the CoolRunner-II CPLD, such as CoolCLOCK and DataGATE, allow the designer to further reduce power consumption. By using these features combined with good design practice, as outlined in this document, designers can be assured that their design will experience optimal low power benefits without sacrificing high speed.

References

1. Johnson H. and Martin G. (1993): High-Speed Digital Design: A Handbook of Black Magic Prentice Hall PTR

XAPP395 (v1.2) September 22, 2003

Using DataGATE in CoolRunner-II CPLDs

Summary

This application note outlines the various ways designers can utilize the DataGATE feature of CoolRunner™-II CPLDs.

Introduction

CoolRunner-II CPLDs deliver the lowest power consumption in today's CPLD marketplace. Built from standard CMOS structures, they achieve the classic I_{CC} = CVF relationship where "C" is driven capacitance, "V" is voltage swing and "F" is the switching frequency of gates driven within the part. The capacitance and voltage values in the equation are essentially demanded by the 0.18 micron process, but the "F" part of the expression is due to the design characteristics of the customer design inside. The desire to lower the "F" aspect resulted in the DataGATE feature in CoolRunner-II CPLDs.

As the name implies, the feature "gates" data. This application note shows how to get as near to the origin of the I_{CC} versus "F" curve as we can with today's technology. It describes the practical aspects of how DataGATE works, the timing model, and what you will need to do in the software to make it work. Further, it shows four different ways the circuitry can be used for power reduction, as well as for circuit debugging, printed circuit board "hot plugging," and device security. These are just some ideas on how to use this feature, and others may come to mind as users learn to operate DataGATE.

The final section of this application note lists additional application notes which cover other aspects of Xilinx CoolRunner-II CPLDs.

Operation

Figure 1 details the delivery of the DataGATE Assertion Rail throughout the I/O structure of a CoolRunner-II CPLD. This image doesn't convey the generation of the DataGATE activation method, but simply the delivery of the control signal to each input pin, with independent control of whether that pin will participate in the gating action. Specifically, two items should be noted. First, each pin can be programmed to participate in the DataGATE operation or not—the standard default is not to participate. The second aspect is that the individual pass transistors permit signal entry into the CPLD, or blocking of the input pin. Figure 2 expands the detail for a single input pin. If the input pin becomes blocked, the last driven value into the CPLD will automatically be latched and held so a solid binary value is delivered into the CPLD core. It should be noted that the Assertion Rail is driven from a specific macrocell within the CPLD, and that when that macrocell drives high, input data will be blocked at participating pins. If the Assertion Rail is low, data passes freely into the chip. The condition of the DataGATE Assertion Rail is manifested at a specific pin (designated DGE), which will vary from chip to chip within the CoolRunner-II family.

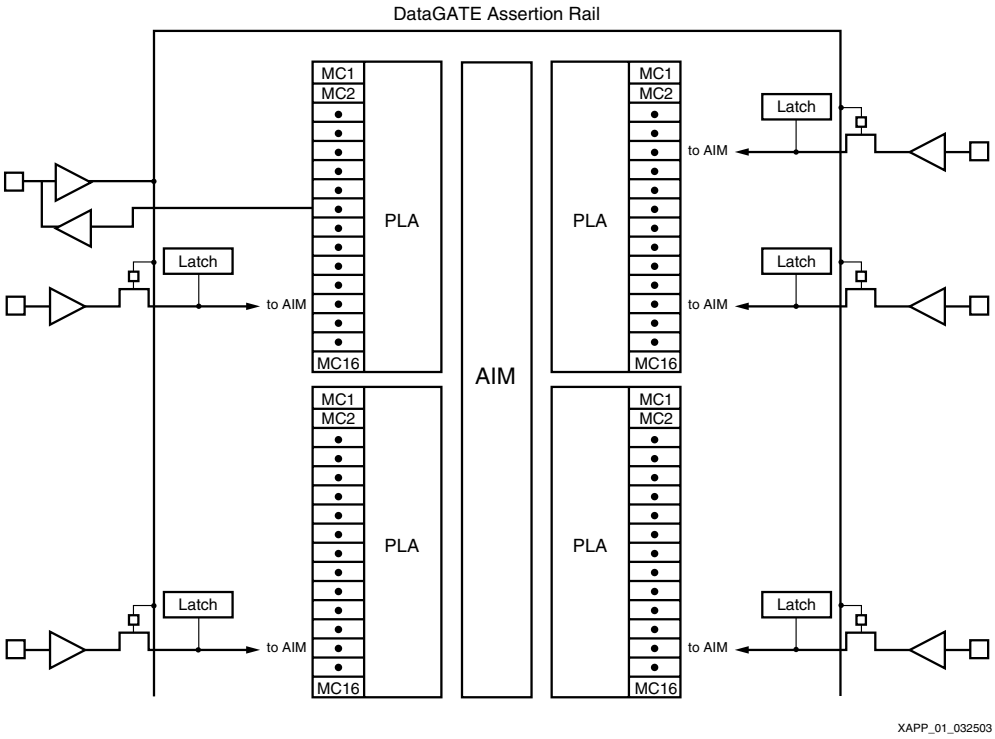


Figure 1: **DataGATE Assertion Rail**

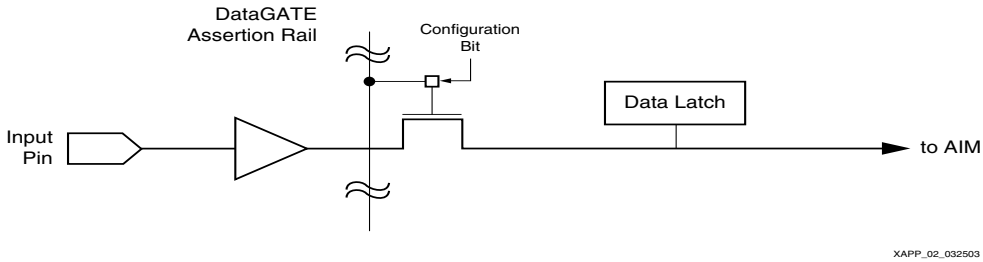


Figure 2: **DataGATE Assertion Rail Input Pin, Latching Old State**

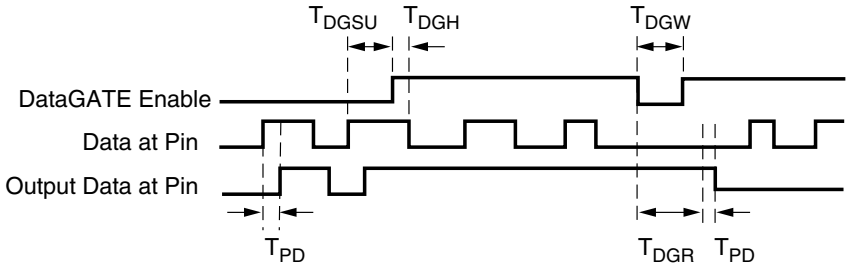


Figure 3: **DataGATE Timing Parameters**

Figure 3 details the timing action of the DataGATE operation. Specifically, the DataGATE Enable pin, if driven from an external pin, can be viewed as a clocking type function. Given that, it has standard clock type timing parameters, a setup time (T_{DGSU}), hold time (T_{DGH}), a minimum width (T_{DGW}) and DataGATE recovery time (T_{DGR}). For clarity, Figure 3 also shows the propagation time (T_{PD}) from input to output of a simple signal. DataGATE operation is described in detail in **XAPP378**, which describes how to declare the appropriate conditions in ABEL, VHDL and Verilog. DataGATE pin participation is also described, so you can easily dictate which pins will participate in the operation.

Applications

Simple Power saving

Figure 4 shows the simplest DataGATE configuration, and possibly the most frequently used method. Here, a simple external signal is delivered on a pin to the CPLD, which connects that signal through the Advanced Interconnect Matrix (AIM) to the DataGATE macrocell. The DataGATE macrocell is a specific macrocell in each part, which can drive the assertion rail. The logic created at the macrocell is arbitrary, so for this example, we can assume it is a simple connection. Hence, when the external pin drives high, that signal drives the Assertion Rail high. When the external pin drives low, so goes the Assertion Rail. As discussed earlier, when driven low, participating and nonparticipating input pins will have their signals forwarded into the CPLD. When driven high, participating input pins will be blocked and latched, while nonparticipating pins will still pass their signals into the CPLD. It is possible for all input pins to be "gated," so care should be used in choosing signals. Blocking clocks typically has the highest payoff, but also the highest risk. Figure 4 shows the case where an external condition decides when to save power.

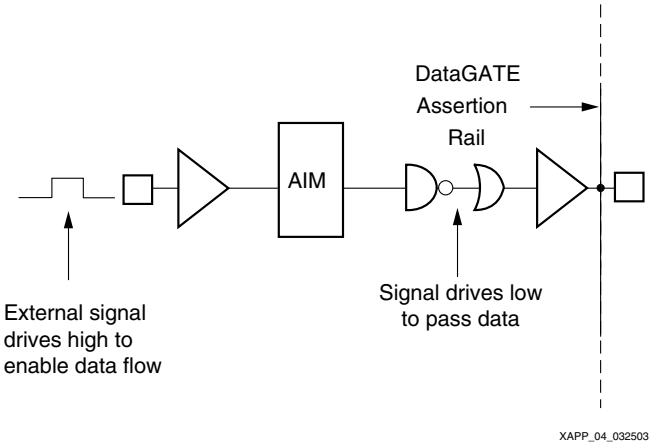


Figure 4: **Simple External DataGATE Control Signal**

Timer

Figure 5 expands on the idea of Figure 4, including a timeout circuit (timer) that is employed to drive the Assertion Rail whenever the timeout signal asserts. We can assume that the timer delivers a "0" when the timer is counting clocks, but when it hits a previously chosen timeout period, it asserts a logical "1" and drives the blocking signal to the Assertion Rail. Because most timers are fairly simple counters, and frequently, it is not important to be too fussy about exactly which count value is chosen (ie, plus or minus a clock or two won't matter), it may be smart to pick the simplest logic structure that can solve the timer problem. In all likelihood, this will be a Linear Feedback Shift Register (LFSR), which will produce most of the values of a binary counter, but in a non-consecutive order and with minimum logic beyond the macrocell flip flops. By either "And"ing the appropriate state, or choosing a state variable for direct connection to the Assertion Rail, the timer can drive the Assertion Rail as needed.

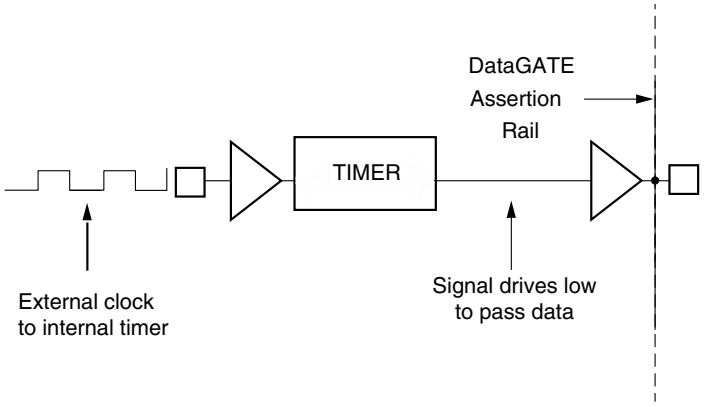


Figure 5: **DataGATE Under Control of Internal Timer**

It would also be possible to use the timer circuit to lock out signals that are deemed to be infrequently used, but periodically "wake up" to sample the pins for whatever reason that might be needed. Alternately, selected signals simply become permanently blocked at the completion of a timeout period. Another view would not have the counter counting "time," but rather some other number of events that drive the "timer" clock input site—say, count 400 interrupt signals then block this set of pins.

Controller

As suggested earlier, any logic can be constructed to drive the Assertion Rail, so state machines can be created, too. In this sense, external variables can be combined with state variables to create the condition that drives the rail. As described with the timer, external signals can create a tally that is used to power down, but also, combinations of events and states can create the blocking event. Figure 6 shows the insertion of an arbitrary controller module, which includes flip flops and logic to interact with the external signals and clocks.

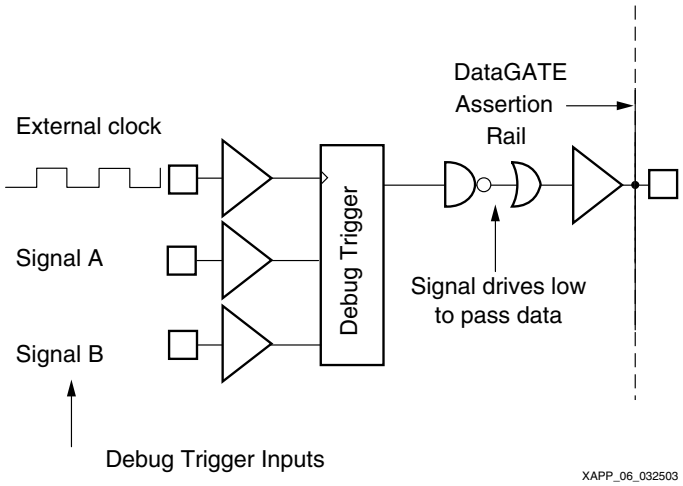


Figure 6: **DataGATE Rail Driven by Internal State Machine Controller**

Another important point to consider when using DataGATE is the overall time operation of a system. At initial power-on, certain actions occur in many systems—processor bootstrap, memory initialization, memory space and I/O space definition, etc. After these initial activities, many items remain in their initial condition and stay there throughout the operation. If certain registers or logic are only used at the beginning, but free running strobe signals, clocks, or data continue unnecessary switching, then those logic sections might benefit from "DataGating."

Debugging with DataGATE

Latching the input pins when the Assertion Rail blocks inputs creates an additional capability: an on-chip debugger. Logic analyzers typically capture the contents of binary signals—the "state"—by triggering on an event, then snapshotting the data into a memory buffer for later readback. Combining the previously described controller with the JTAG circuitry inherent in a CoolRunner-II permits a simpler, but effective capability. For instance, assume that most of the pins are participating in the DataGATE operation. Assume next that we wish to isolate the state of the CPLD when a certain combination of inputs and internal CPLD states occurs—say an "error" state. By creating a state machine out of scrap gates and flops within the CPLD, the inputs can be blocked and held until the JTAG circuit reads back the entire state of the CPLD for dissection and analysis with a PC and JTAG cable. Alternately, the triggering event of a logic

analyzer can be taken over to the CPLD and applied just as in the first, simple “power saving” model described above. Figure 7 gives some detail on how this might occur.

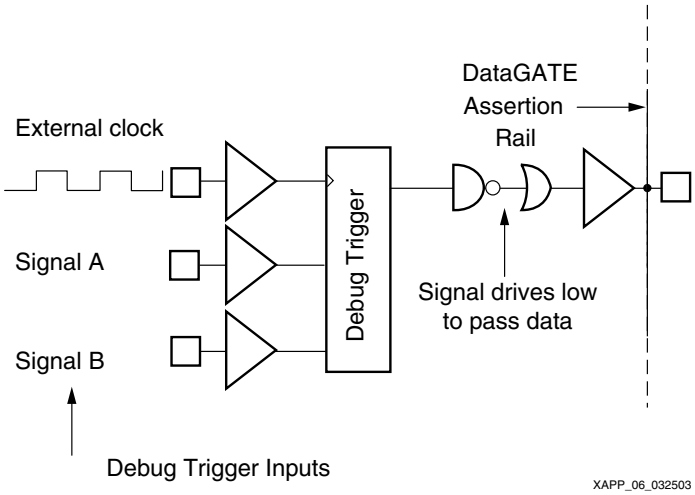


Figure 7: **Debugging with DataGATE**

This model of debugging is one way to use the input latches on the CPLD, although others view this as a very large set of input latches that can be controlled for simple data capture at the pins. The only drawback to this is that the timing for the data capture tends to increase on the larger parts, so users should plan for a little extra time delay.

Hot Plugging with DataGATE

Hot Plugging, Hot Socketing and Live Insertion are all terms that are used interchangeably. The standard situation is this: there is a rack of equipment that is powered up, and a user needs to insert a board into that rack without turning the power off. The methodology for this is somewhat nebulous because expectations vary widely between engineers: depending on how much they know about the situation, engineers all expect different behavior . Some expect the board insertion to go perfectly, without any control signals being disturbed or data bits being dropped. Others expect the board insertion to result in possible data corruption, but not in any catastrophic crashing of the systems. Experienced designers cross their fingers and are satisfied if the event doesn’t result in the rack catching on fire.

A user’s expectation should be that there will be no actual damage, but that extra care must be taken to prevent corrupt behavior. For instance, all the devices on the “cold board” are uncharged where most of the elements within the “hot” rack are charged up (bus lines, decoupling capacitors, whatever). Inserting a discharged board presents a substantial capacitive load to a system that has finite charge on it. This will require charging the “cold” board to the appropriate level before operation can occur. One solution to this has been the “extended finger” method whereby little extension tabs are placed on the cold board, so electricity is delivered to the board before the logic enters the rack connectors; however, this

requires advanced planning to make the little tabs. We present an alternative method which uses the CoolRunner-II DataGATE feature to eliminate corrupt data.

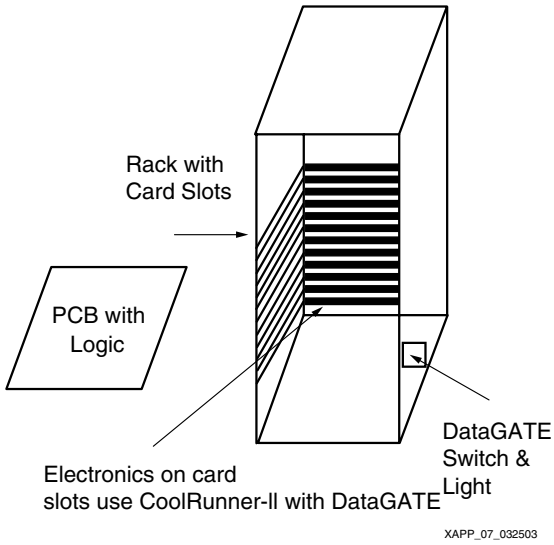


Figure 8: **Hot Plugging with DataGATE**

For this description, we assume all cards in the system, including the cold one to be inserted, have CoolRunner-II CPLDs attached to the slot connector pins. Also, all of the rack boards are connected to a control signal, which is either located on the rack or controlled by software on a card within the rack. The control signal can come from an internal processor signal, or from a switch on the rack (Figure 8). Before the hotplug event occurs, the control signal asserts to make all inputs within the rack latch their current state until the plug completes and the control signal releases. The release of the control signal should occur after the cold board is warmed up and initialized. The system then proceeds to operate. Depending on the system requirements, additional buffering or protocol actions should be included to account for the brief pause in the operation.

Security

Figure 9 shows another DataGATE application. In this case, the CPLD is involved in securing some aspect of the system operation. For simplicity, let's assume a password is involved in the operation. The password is delivered into the CoolRunner-II CPLD, which—if the password is correct—permits other signals to continue entering the device. If the password is incorrect, the DataGATE blocks all the signals coming into the CPLD to forbid future trials. Designing the rest of the CPLD to perform some "mission critical" aspect of the whole system is critical here, so the system won't work while the chip is being gated. To attempt more passwords, the whole system must be power cycled. A more permanent action would be to erase the CPLD if the password doesn't match, but doing so would use the On the Fly Reconfiguration capability instead of only DataGATE.

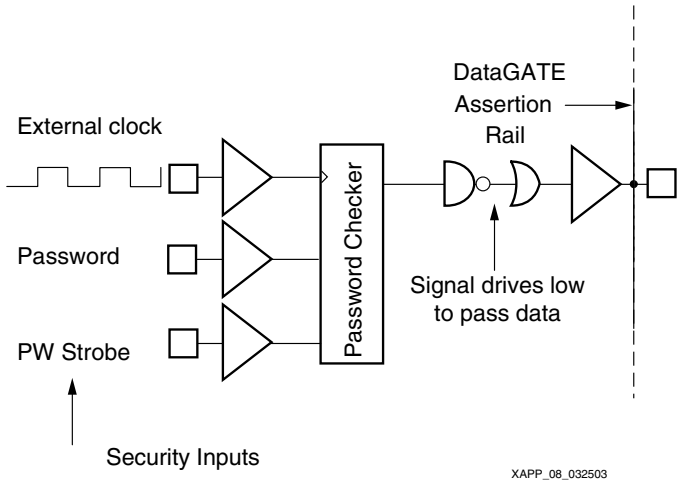


Figure 9: **DataGATE Dynamic Security**

Conclusions

DataGATE is a versatile and helpful feature designers can utilize when designing with CoolRunner-II CPLDs, although they are not required to use it and it can be used only over chosen pins, as needed.

A simple strategy for using DataGATE is to design with no initial regard for it, except in saving the macrocell that drives the assertion rail. Once the design is complete, evaluate the current drawn against the target current budget. Next, consider implementing CoolClock or other power savings options. Again, evaluate consumed current against target current. Review **XAPP 377** for other things to try.
Next, if the design does not hit its target, or you simply wish to see how low the current can be taken, *then* consider using DataGATE. Consider sections of your design that initially "wake up" and perform active behavior, then go "quiet" for the rest of the operation. Assuming they are driven from active pins, can you identify some time or event that would permit them to be blocked by the DataGATE circuit? If so, define the time or event, create a logic circuit to manifest at that time and have it drive the DataGATE rail. Evaluate the current again. If you need less current, can you include more pins? Can you slightly change the block point (in time) to include more pins? If so, alter your design and re-evaluate. Should you encounter a "tweak" that adversely affects your design, you can back up a revision to the last working one and know that it is better than your original solution.

Further Reading

Application Notes

http://www.xilinx.com/xapp/xapp375.pdf (Timing Model)
http://www.xilinx.com/xapp/xapp376.pdf (Logic Engine)
http://www.xilinx.com/xapp/xapp377.pdf (Low Power Design)
http://www.xilinx.com/xapp/xapp378.pdf (Advanced Features)
http://www.xilinx.com/xapp/xapp379.pdf (High Speed Design)
http://www.xilinx.com/xapp/xapp380.pdf (Cross Point Switch)
http://www.xilinx.com/xapp/xapp381.pdf (Demo Board)
http://www.xilinx.com/xapp/xapp382.pdf (I/O Characteristics)
http://www.xilinx.com/xapp/xapp383.pdf (Single Error Correction Double Error Detection)
http://www.xilinx.com/xapp/xapp384.pdf (DDR SDRAM Interface)
http://www.xilinx.com/xapp/xapp387.pdf (PicoBlaze Microcontroller)
http://www.xilinx.com/xapp/xapp388.pdf (On the Fly Reconfiguration)
http://www.xilinx.com/xapp/xapp389.pdf (Powering CoolRunner-II CPLDs)
http://www.xilinx.com/xapp/xapp393.pdf (8051 Microcontroller Interface)
http://www.xilinx.com/xapp/xapp394.pdf (Interfacing with Mobile SDRAM)

CoolRunner-II Data Sheets

http://direct.xilinx.com/bvdocs/publications/ds090.pdf (CoolRunner-II Family Datasheet)
http://direct.xilinx.com/bvdocs/publications/ds091.pdf (XC2C32 Datasheet)
http://direct.xilinx.com/bvdocs/publications/ds092.pdf (XC2C64 Datasheet)
http://direct.xilinx.com/bvdocs/publications/ds093.pdf (XC2C128 Datasheet)
http://direct.xilinx.com/bvdocs/publications/ds094.pdf (XC2C256 Datasheet)
http://direct.xilinx.com/bvdocs/publications/ds095.pdf (XC2C384 Datasheet)
http://direct.xilinx.com/bvdocs/publications/ds096.pdf (XC2C512 Datasheet)

CoolRunner-II White Papers

http://www.xilinx.com/publications/products/cool2/wp_pdf/wp165.pdf (Chip Scale Packaging)
http://www.xilinx.com/publications/whitepapers/wp_pdf/wp170.pdf (Security)

XAPP436 (v1.1) March 30, 2005

Managing Power with CoolRunner-II CPLDs

Summary

This application note demonstrates how multiple devices, including Virtex™-II, Spartan™-3, and Spartan-3L FPGAs, can be effectively power managed by a single CoolRunner™-II CPLD. It is written with battery powered applications in mind.

Introduction

All chips draw power, but some applications are more sensitive than others to the amount drawn. Portable applications are sensitive simply because they draw from a battery. Most digital chips are designed to operate at 5V, 3.3V, 2.5V, 1.8V, and so forth. This does not match well with today's battery voltages. Hence, there will be a regulator or two on most boards. Managing power will involve managing those regulators.

CoolRunner-II CPLDs were designed to operate with a core voltage of 1.8V, well suited to its 0.18 micron core, but its I/O structure supports 3.3V, 2.5V, 1.8V, and 1.5V operation. Being standard low power CMOS, the I/Os also operate within that range, but are only speed specified at those voltages.

CoolRunner-II CPLDs have been successfully used as voltage translators, logic collection sites and even power management solutions. Their "early on" behavior makes them ideal for managing other chips' power. This application note focuses on using CoolRunner-II to manage power for Xilinx FPGAs, providing greater FPGA utility in a portable products. Additional CoolRunner-II qualities are shown to add value to reducing the portable product power budget in many cases.

The Power Equation

As expected, the starting point is physics, but CMOS power consumption can be deceptive. Simplistically, it follows an equation relating the switching speed of a logic gate output, times the voltage range over which it swings, times the load capacitance being driven. Adding in the static leakage component says it all.

Equation 1: $$Power = V(VCF) + static\ power$$

where:

V = output voltage swing

C = load capacitance

F = gate output switching frequency

$Static\ power$ = somewhat constant value for many parts (often negligible)

This equation is a guideline. Originally developed for a simple CMOS inverter, it shows a trend, but only suggests how to reduce power. Equation 1 drives most CMOS power estimation tools. Xilinx provides several ways to estimate power, including spreadsheets, application notes and the XPower software, available with the ISE™ design suite.

The CoolRunner-II Solution

The CoolRunner-II approach to power management attacks both static and dynamic current. A battery is a charge reservoir with varying capabilities on voltage regulation, and in some cases an ability to be recharged. Each battery has its own qualities for delivering charge. Many batteries can deliver high currents for brief time periods, or smaller currents for longer times. Trade-offs can and must be made.

To get a feeling for power activity, let's look at a real world example, that was documented by Portelligent in their Report #116.02- 031023-1d. Figure 1 is a reconstruction of their measurements made on an LG cell phone that was able to record video or take a picture. Figure 1 shows power use while taking a photograph, with the cell phone camera.

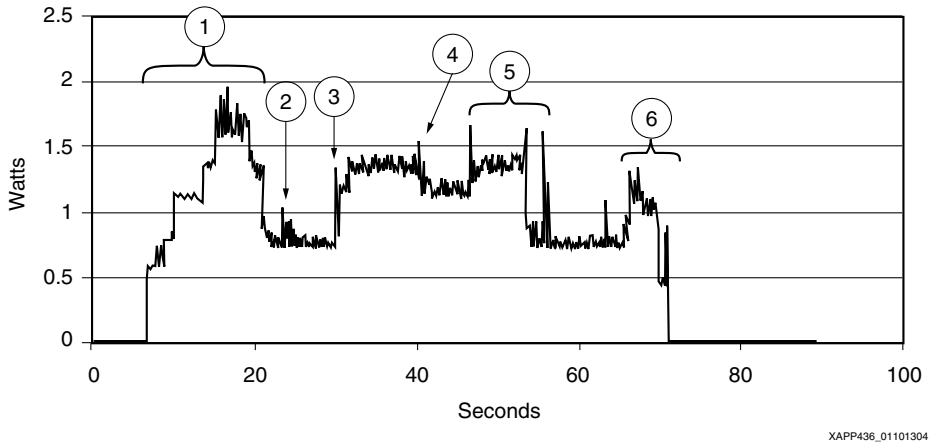


Figure 1: **Power Profile of LG Cell Phone Taking Still Photograph**

The steps in taking a still photo are summarized as:

1. Power up and turn on the LCD backlight
2. Select the camera from menu
3. Select taking a still photo
4. Take the photo
5. Return to the main menu
6. Power down

The numbers in the diagram correspond to the activity list above. The Portelligent report explains the activity of the various chips within the camera, but Figure 1 shows the composite power. There are many variables at play. Several chips within the camera receive power at different times. The display, camera chip and memories are involved here, but none of the standard cell phone capabilities - like making a call. That requires a different profile. Nonetheless, these actions are typical of a large number of today's portable systems ranging from, walkie-talkies to PDAs, to video cameras, to the emerging software defined radios (SDR). 3G cell phones, in particular, with their added on applications and multi-band operation take this type of power management to new heights. In fact, these systems may rival some of the very power conscious projects like the Mars rover or other deep space probes in complexity. Power management is vital to all of these products.

Now, let's shift our attention to Figure 2 which shows a small system using two FPGAs, an ASSP and a CoolRunner-II CPLD. Note, there are also multiple regulator chips shown (LDO), deriving their voltages by regulating down 5 volts. Another approach would be to regulate down from a battery, which most regulators can manage.

This particular example has the FPGAs being powered through the LDOs, but the ASSP simply connects to the distributed supply through a power FET. The particular FET will depend on the channel drop that can be tolerated, the turn on voltage and the power requirements of the ASSP. All power sources are controlled from logic signals coming off CoolRunner-II pins. The idea here is simple. When CoolRunner-II remains turned on, its power consumption is only in microamps (standby), whereas the FPGA current ranges into milliamps, depending on family and density.

In this example, we have combined a 150 nanometer Virtex-II FPGA and a 90 nanometer Spartan-3 FPGA. Each has different power needs, and illustrates choices and trade-offs to be made. As we will see later, the CoolRunner-II will be able to introduce an extra power down mode to the FPGAs, to reduce dynamic power. That will give a choice on whether to turn the part off, and pay for reconfiguration again, or just reduce dynamic power during standard operation. A reasonable regulator for the LDO 2 module is the TPS5003 from Texas Instruments™.

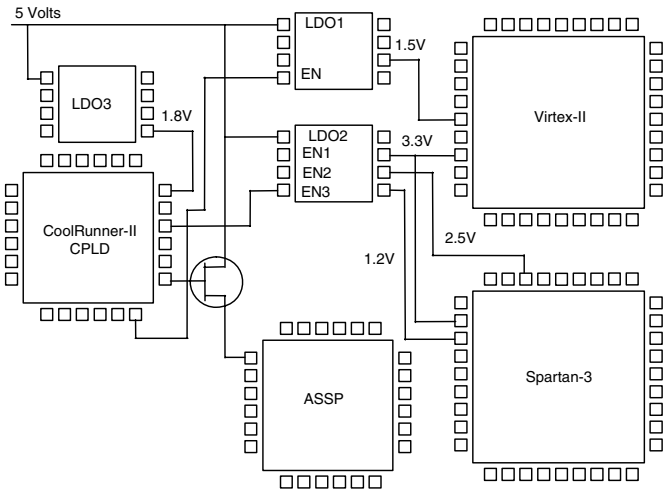


Figure 2: **Multiple Chips "Power Managed" by a CoolRunner-II CPLD**

The Virtex-II FPGA has reduced power up surge, but larger density parts may draw more current than a portable budget allows. Each FPGA might be a candidate for shutdown, except when needed. Table 1 summarizes quiescent internal current drawn by various Virtex-II family members. The dynamic current is a function of signal switching rates within the parts, and is design dependent. To estimate the power, you can use any of the methods mentioned earlier, or simply build up the design and measure the requirement. Additional estimation resources are listed at the end. Table 2 gives similar data for Spartan-3 FPGAs.

Table 1: **Virtex-II FPGA Typical Internal Quiescent Currents**

Virtex-II Device	Typical Internal Quiescent Supply Current(1)
XC2V40	3 mA
XC2V80	5 mA
XC2V250	8 mA
XC2V500	10 mA
XC2V1000	12 mA

1. Refer to the data sheet for the most accurate and up-to-date information

Table 2: **Spartan-3/L FPGA Typical Internal Quiescent Currents**

Spartan-3/L Device	Spartan-3 Typical Internal Quiescent Supply Current(1)	Spartan-3L Typical Internal Quiescent Supply Current(1)	Spartan-3L Quiescent(1) Max (Hibernate Mode)(2)
XC3S50	10 mA		
XC3S200	20 mA		
XC3S400	35 mA		
XC3S1000/L	65 mA	30 mA	6 mA
XC3S1500/L	65 mA	50 mA	8 mA

1. Refer to the data sheet for the most accurate and up-to-date information
2. Hibernate Mode available for Spartan-3L. **See DS313**.

Figure 3 shows how a current profile might look (averaged) over time. Usually, there is an initial surge as all on board capacitance becomes charged, various parts undergo configuration, initialization, bootstrapping, and so forth. Then things settle down, and various chips can be turned off, placed into low power, or whatever, as dictated by the application. The average current draw of this profile is substantially less than the initial surge value, so there may be a payoff for turning chips off all together, or placing them in a low power mode. Table 3 and Table 4 give configuration times in byte wide mode at maximum speed.

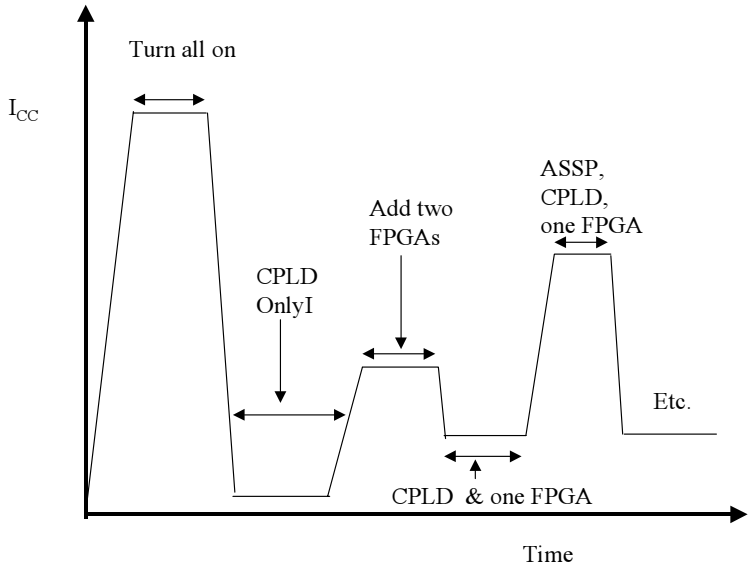


Figure 3: **Current Profile of a System with Various Active Devices Over Time**

Table 3: **Virtex-II Configuration Parameters**

Virtex-II Device	Configuration Bits	Configuration Time in Microseconds (at 50 MHz)
XC2V40	338,976	42.372
XC2V80	598,816	74.852
XC2V250	1,593,632	199.204
XC2V500	2,560,544	320.068
XC2V1000	4,082,592	510.324

Table 4: **Spartan-3 Configuration Parameters**

Spartan-3 Device	Configuration Bits	Configuration Time in Microseconds (at 50 MHz)
XC3S50	439,264	54.9
XC3S200	1,047,616	130.95
XC3S400	1,699,136	212.392
XC3S1000	3,223,488	402.936
XC3S1500	5,214,784	651.848

By lowering the average power, CoolRunner-II CPLDs can dramatically extend the battery life of a system in a way that brings the high flexibility and value of FPGAs into the portable world. Let's show how more value is gained using CoolRunner-II DataGATE.

DataGATE

DataGATE was designed to stop unwanted input switching from continuously draining power in CoolRunner-II CPLDs. Additional applications evolved from testing to security, and are documented in the Advanced Features and DataGATE application notes. However, one additional application is simply to “DataGATE“ other chips.

Figure 4 shows how the DataGATE feature works. A metal rail (DataGATE Assertion Rail) circles the whole chip inside, near the pins. Each input site provides a place where the received signal can be blocked by a pass transistor, depending on two conditions. The first condition is an enable bit, selecting that pin to participate in the DataGATE decision. The second condition is simply whether the DataGATE Rail is asserted. If the rail is asserted and that input’s participation selected, the input signal is blocked from penetrating the chip, until the rail releases assertion. It’s that simple. When the rail asserts, blocking follows immediately. The previous input level automatically latches, so static CMOS logic signals forward into the CPLD core. The signal freezes until released. When the rail releases, switching action resumes.

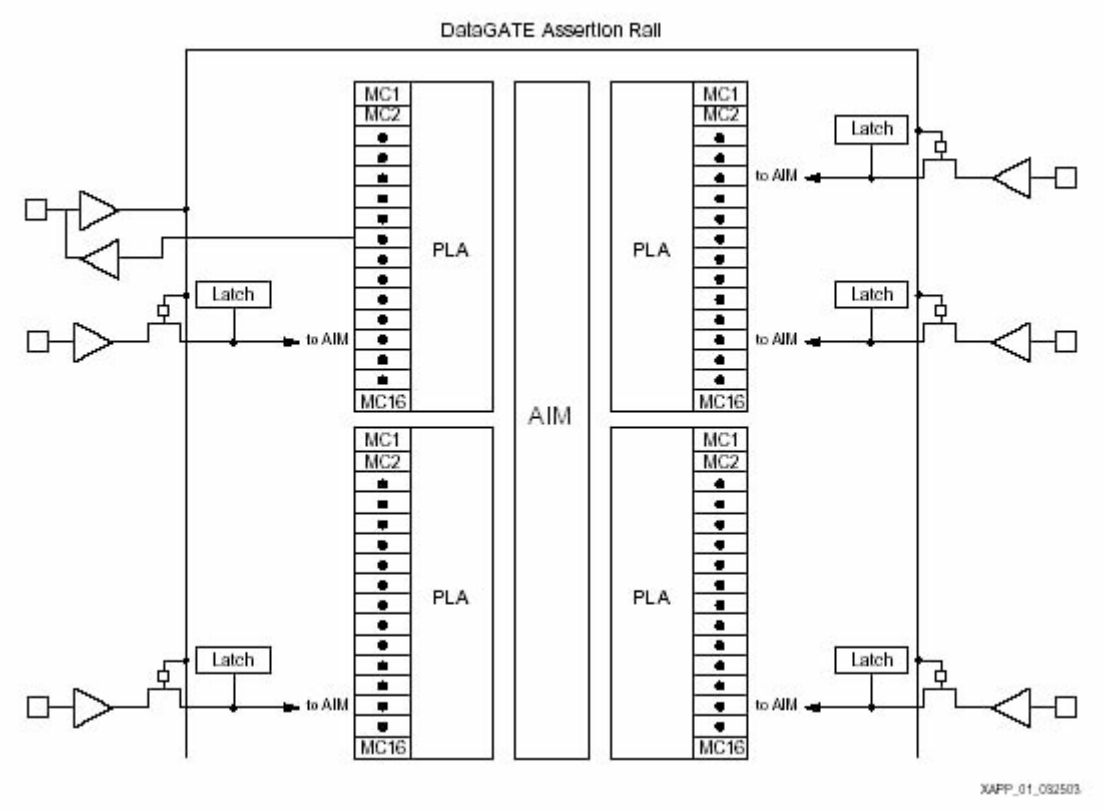


Figure 4: **DataGATE Architecture**

Figure 5 shows a close-up of how the pass transistor, enable cell and latch all connect to automatically block and freeze input signals that forward through the CoolRunner-II core.

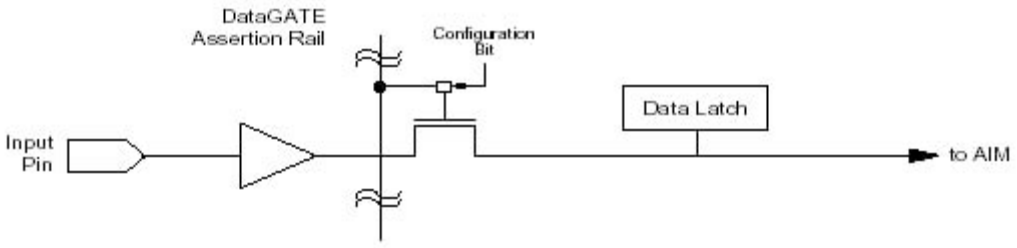


Figure 5: **Close-up of DataGATE Switch Mechanism**

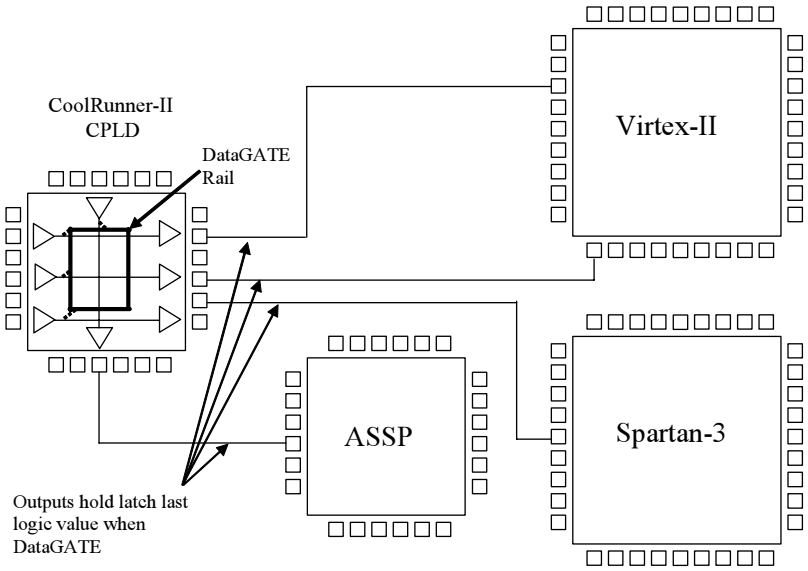


Figure 6: **DataGATE Blocking Switching Activity to Other Chips**

Figure 6 shows how signals passing through the CoolRunner-II DataGATE freeze signals to other chips. In this situation, we passed signals through the CoolRunner-II part, directly to the outside, where they drive to other chips. These will be held at the logic level that was last on the input pad, when DataGATE asserts. Should a 3-state signal be forwarded through the CoolRunner-II output pins, it will naturally be pulled to a high or low value, by the weak keeper that is on them, thereby covering the case when a “frozen” output gets 3-stated by another condition. An interesting proposition of passing signals through the CoolRunner-II is that it might be done as a natural side product of simply translating voltage levels on signals as they interface through the CPLD. The DataGATE action, is really for free under those conditions, which occur frequently.

DataGATE is offered on CoolRunner-II CPLDS that have 128, 256, 384 and 512 macrocells. This extends into the hundreds of I/O pins that can be blocked as needed by the DataGATE facility. If multiple sets of pins need to be blocked, under different circumstances, then multiples of the smaller parts can be used to cover the number and condition needs of any given design.

At this point, we have not described what drives the DataGATE rail. It is very simple. One macrocell within a particular CoolRunner-II CPLD is designated as the “DataGATE” macrocell. It is identical to all other macrocells, which means any logic situation a designer sets up to drive that macrocell, asserts the DataGATE signal, if enabled in the design software. The DataGATE signal releases whenever the logic driving that macrocell dictates. An event as simple as a switching input can trigger the DataGATE macrocell, and an event as complex as a conditional state machine driving a timer can trigger the macrocell. Designers are free to dream up whatever they want. It is possible to block any input chosen, including clocks, but extreme care must be used when designing that way! All, none or any subset of the input pins can be blocked. Being a reprogrammable CPLD, this facility can be used experimentally to determine the best set of signals to “freeze” and the best set of circumstances to assert and release the rail.

So, how do you know what to “freeze?” That will vary, from system to system. Here’s an example. When a microprocessor first bootstraps, it frequently sends values from its databus into a CPLD. These might be address values being loaded into comparators to select ranges of memory and I/O devices. Once those registers are initialized they need only compare against the address lines to operate. The databus connections are never needed again, but are still attached. DataGATE lets designers identify the time when the connection is no longer needed, and eliminate the extraneous switching that draws unneeded current. To learn more about other things you can do with DataGATE, check the references at the end.

Conclusion

We have omitted some details. What is the power impact of unpowered I/O pins attached to powered up termination resistors? How much leakage per pin occurs if an unpowered pin is driven by a powered one? Many questions can only be answered by assessing the specific situations with the particular device's data sheet in hand. We hope the methods described here will have some value by simply increasing your choices on power reduction methods.

CoolRunner-II CPLDs are designed to be inherently low power devices. Additional features within them – including DataGATE, can help other chips also reduce their overall power, when properly applied.

References

Portelligent Report #116.02-031023-1d

Estimation equation: http://www.xilinx.com/bvdocs/appnotes/xapp317.pdf
Low power design methods: http://www.xilinx.com/bvdocs/appnotes/xapp346.pdf
Decreasing power: http://www.xilinx.com/bvdocs/appnotes/xapp347.pdf
Accurate XPLA3 estimation: http://www.xilinx.com/bvdocs/appnotes/xapp360.pdf
Accurate CoolRunner-II Design estimation: http://www.xilinx.com/bvdocs/appnotes/xapp377.pdf
Powering CoolRunner-II: http://www.xilinx.com/bvdocs/appnotes/xapp389.pdf
DataGATE: http://www.xilinx.com/bvdocs/appnotes/xapp395.pdf
XPower http://www.xilinx.com/xlnx/xebiz/designResources/
Power Estimator (Spartan) http://www.xilinx.com/cgi-bin/power_tool/power_Spartan3
Power Estimator (Virtex-II) http://www.xilinx.com/cgi-bin/power_tool/power_Virtex2

Additional Information

CoolRunner-II Data Sheets, Application Notes, and White Papers
Device Packages
Online Store
Spartan-3 Data Sheets, Application Notes, and White Papers

Power Management Solution Guide

"As chip technology progresses to 90 nm and below, power becomes a burning issue in system design."

www.xilinx.com/xcell/power1/

Published by

Xcell Publications

XILINX®
The Programmable Logic Company™

Corporate Headquarters
Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
Tel: (408) 559-7778
Fax: (408) 559-7114
Web: www.xilinx.com

European Headquarters
Xilinx
Citywest Business Campus
Saggart,
Co. Dublin
Ireland
Tel: +353-1-464-0311
Fax: +353-1-464-0324
Web: www.xilinx.com

Japan
Xilinx, K.K.
Shinjuku Square Tower 18F
6-22-1 Nishi-Shinjuku
Shinjuku-ku, Tokyo
163-1118, Japan
Tel: 81-3-5321-7711
Fax: 81-3-5321-7765
Web: www.xilinx.co.jp

Asia Pacific
Xilinx, Asia Pacific Pte. Ltd.
No. 3 Changi Business Park Vista, #04-01
Singapore 486051
Tel: (65) 6544-8999
Fax: (65) 6789-8886
RCB no: 20-0312557-M
Web: www.xilinx.com

Distributed By:

FORTUNE 2005 100 BEST COMPANIES TO WORK FOR

Printed in U.S.A.

PN 0010865