Actel®
POWER MATTERS

Power-Aware FPGA Design

by

Hichem Belhadj, Vishal Aggrawal, Ajay Pradhan,
and Amal Zerrouki

February 2009

Abstract

Power consumption requirements in new, autonomous, multimedia-savvy consumer products that can store, transmit, and receive information have catapulted system architects and board and chip designers into a new realm. Even when designers attempted to reduce system power consumption, their approaches were not comprehensive and focused enough to achieve optimal results.

The goal of this paper is to examine each design step and component of system power with the purpose of providing techniques to reduce wasteful power consumption. These techniques cover system partitioning, chip design, and board layout. The proposed design techniques cover RTL coding, arithmetic architecture power-profiling, and place-and-route hints. Some of these techniques may seem antiquated, but they have been revisited to fit the profile of applications required by most of the new consumer products targeting low-power FPGAs. In addition, available power modes are exploited to minimize further power consumption, energy, and battery life.

Introduction

The latest process technologies revealed a challenging side of physics: the dramatic increase in static power, which is worse for FPGAs than for ASICs. Today, power comes to play in bill-of-materials (BOM), board design, chip design, testing, and production flow.

ASIC and ASSP vendors dealt with power in various ways; however, FPGA vendors have only recently introduced novel and power-friendly FPGA architectures and features. A key feature is the availability of various power modes and power voltages. Actel IGLOO® and ProASIC®3L FPGA families feature various power modes with state saving (On, Static, Idle, and Flash*Freeze modes), as well as operation at 1.5 V or 1.2 V for both the core and the I/Os. In 2007, Altera® announced a more power-friendly derivative family of the Max® II, called Max IIZ [MaxIIZ2007]. Xilinx® led the charge a while back with the CoolRunner™ CPLD family in 2006 [CoolRunner2006] and, more recently, has offered low-power derivatives of its Virtex® families. All three vendors provide several analysis tools to help users estimate the power consumption at different stages of their design cycles [BADAZ2008]. Efforts to improve the backend tools also help reduce wasted power [Libero2009]. This paper aids power reduction by clearly defining and describing FPGA design techniques and their impact on power and energy consumption.

FPGA Power Components and System Power Profile

FPGA Power Components

Several criteria are used when selecting an FPGA from the abundant offerings available in the market. Cost, capacity, performance, features, and packaging are usually the main drivers in a system architect or designer's choice of one FPGA over the others. With the rise of power-conscious applications in the portable consumer, medical, and even military market segments, power is getting a higher rank in the priority list.

Everyone is familiar with the traditional static and dynamic power present in ASICs or FPGAs. However, not everyone knows that, unlike ASICs and nonvolatile FPGAs, volatile FPGAs have two additional power components: configuration power consumed during the programming at system power-up and inrush power dissipated during the device functional power-up, as depicted in Figure 1 on page 4. FPGA-based board designers must account for the configuration and inrush powers while sizing their power supplies and selecting batteries. Despite the efforts of SRAM-based FPGA vendors to reduce the inrush and programming components, these are still present and have a severe negative impact, especially when several FPGAs are populating a single board or are powered from a common supply on different boards.

This additional power dissipation is even more serious for systems with frequent On/Off cycles and must be considered when estimating battery life.

In addition, volatile FPGAs require an external boot PROM for configuration storage, which adds to the overall power dissipation. Even though some vendors have embedded a large flash memory within the same device, the additional storage power is still present.

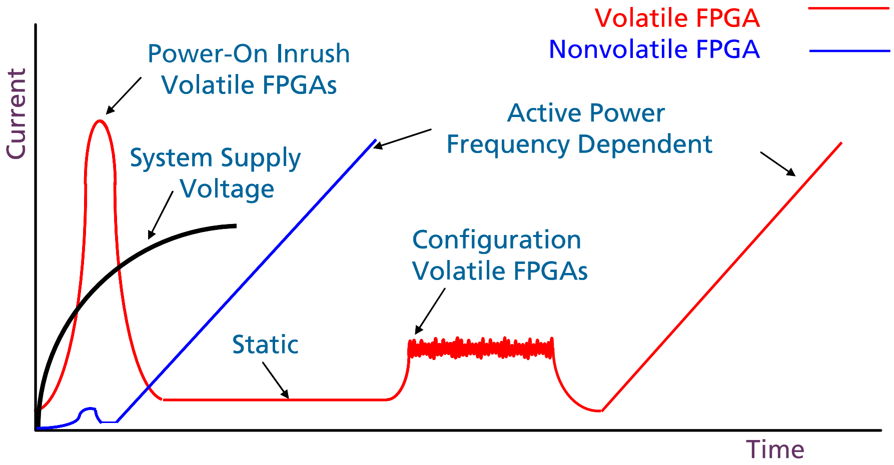


Figure 1: **Volatile vs. Nonvolatile FPGA Power Profiles**

System Power Profile

Before choosing a strategy to improve the power figures, system architects and their design teams must identify the actual system operating modes and the associated power scenarios. The example system power profile in Figure 2 shows that the system will operate at different temperatures and that the up-time or duty cycle is close to 50-50.

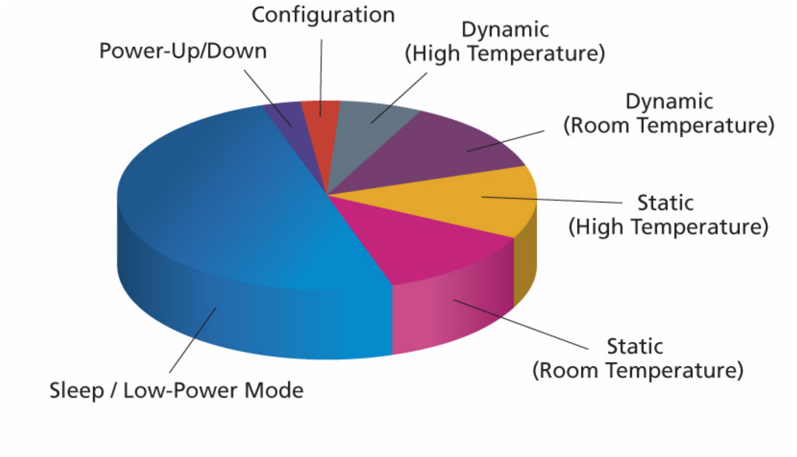


Figure 2: **System Operation Modes and Power Profile over Time**

A quick look at such a graphical representation of the system power profile eases decision-making. In the case of the system illustrated in Figure 2, it is obvious that efforts to reduce temperature, static power, and dynamic power will be worthwhile.

Understanding Power Numbers

As power consumption becomes an important design consideration, understanding the results from power-analysis tools can significantly impact the efficiency of a design. FPGA designers must understand the dynamics of the arriving data patterns and the possibilities of exploiting FPGA power modes. In addition, FPGA designers need to consider architectural options that target not only absolute power numbers for a particular clock cycle, but also the energy per computation. Analyzing specific power component numbers instead of the entire power profile results in a narrow interpretation of power numbers, which can lead to a design that consumes power inefficiently.

For example, while measuring actual silicon power consumption numbers of several designs implemented on IGLOO devices, it becomes evident that there were different implementations of the same basic design. Each of these implementations resulted in a different power number. The analysis of these differences in implementation area, timing, and power attributes revealed that considering simple, single-cycle power numbers was misleading and even erroneous. Some implementations had the worst power consumption if only one clock cycle was considered and would have been abandoned if the data arrival pattern and the potential use of power modes or power-down options were not considered. One design implementation was architected to perform computation on a data packet in one cycle, while two others—apparently more power-friendly implementations—required 6 and 12 cycles, respectively, for the same computation. Based on the input data rate, the apparently power-hungry implementation could completely shut off the computation engine for 5 or 11 cycles and save significantly more energy (increasing battery life) than the two other implementations. The targeted FPGA, an Actel IGLOO device, offers a sleep mode, called Flash*Freeze mode, which allows a drastic reduction in both static and dynamic power while retaining the registers and RAM states. Entering and staying in this mode allows the user to save power and battery life. To illustrate this, two architectures for a simple DES design were created: One serial implementation required 16 cycles to complete data processing, while a parallel implementation processed the same data in a single cycle. Table 1 provides the power results for a 100 MHz clock frequency.

Table 1: **Attributes of Serial vs. Parallel DES Implementations**

Attribute	Parallel	Serial
Total Area (Comb + Seq Modules)	9,404	1,720
Flip-Flops	1,980	68
Clock Power (mW)	42	5
Total Power (mW)	270	167

Reducing Static Power

There are various ways of reducing static power. It begins with technology development engineers, and includes silicon design engineers, package engineers, FPGA designers (end users) and board-level engineers. The following sections illustrate the delicate balance between manufacturability, testability, yield, observability, die area, power, and speed.

Process Development and Static Power Considerations

Static power is dominated by leakage current in various forms:

- Sub-VT leakage
- Junction leakage (i.e., source/drain, well, and triple-well junctions)
- GIDL or gate-induced drain leakage
- Gate leakage

Technology development and process engineers have various goals:

- Engineers address reliability by optimizing cell parameters to improve EMI, hot-carrier injection performance, and all the reliability attributes.
- Engineers strive to reduce leakage by tracking leakage components while adjusting multiple VTs for NMOS and PMOS devices. In Figure 3, the curves illustrate a 130 nm process node, the differences for multiple VTs between a PMOS device, where leakage is proportional to the threshold, and the NMOS, which has junction and GIDL leakages that are dominant.
- Engineers must be aware of potential yield killers and savers. For instance, they must identify and fix any point-defect-related leakage as it could be a yield killer for large-area.
- Technology development engineers must find a balance between leakage and speed over operating conditions for various process corners to ensure lower power without compromising speed (Figure 4 on page 7).

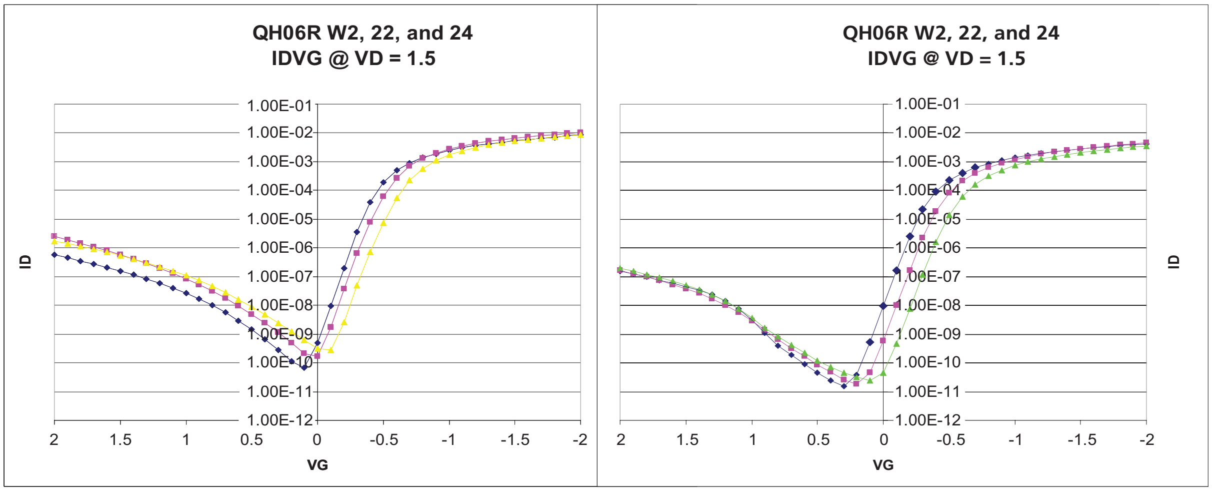


Figure 3: **NMOS and PMOS Leakage – Trade-Off between GIDL and VT**

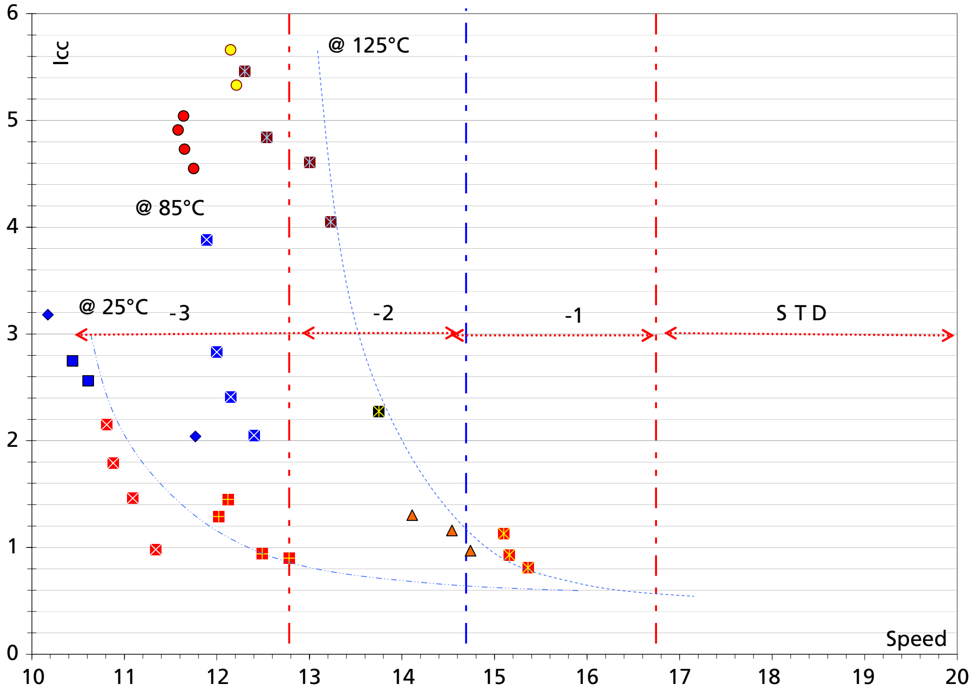


Figure 4: **I_{CC} vs. Speed over Process Splits**

FPGA Silicon Design Static Power Considerations

In the architecture definition and implementation phases, architects and silicon design engineers strive to eliminate any source of leaky circuitry, while watching for area, speed, signal integrity, reliability, and inexpensive but complete testability. Techniques such as multi-supply and multi-VT, usually used in ASICs and ASSPs, are also adopted in FPGA silicon design. Multi-supply decorrelates power grids of some blocks in the FPGA in order to allow a shut-down of some of these blocks. Multi-VT aims at reducing the overall leakage in the design by introducing a minimum of leaky or standard-VT cells and a maximum of low-leakage or high-VT cells.

For example, an FPGA I/O designer must build a configurable block that can support multiple single-ended and differential standards, with different slew rates, drive strengths, pull-up/-down, Schmitt trigger, and other advanced features. One of the main goals is to minimize the static power for each of these standards by adjusting the dimensions and attributes of each PMOS and NMOS transistor and the configurable switches. The embedded SRAM block designer also has a responsibility to eliminate any potential leakage. This is one example, but the point could be made as well with designers of analog blocks and other architectural features that must be considered. *However, the common goal for all designers is the implementation of means to partially or completely power-down these blocks (SRAM, PLL, I/O bank, RC oscillators, RTC, etc.) when not used [Fusion 2006, IGLOO2007].*

FPGA User and Static Power-Aware Decisions

FPGA designers can contribute to lower static power by following three simple principles:

1. Smallest die
2. Least resources
3. Master your FPGA architecture

FPGA families usually include a set of dies with different capacities and, in some cases, different feature sets. The "smallest die" principle is based on the fact that smaller dies in a family have lower static power compared to larger dies. FPGA designers should select the smallest possible die in the family, while ensuring that they still meet the performance targets.

The "least resources" principle pushes the designer towards reducing the number of resources, such as RAMs, PLLs, and I/Os. To reduce I/O count, for example, the designer must use time multiplexing and minimal-I/O-count design partitioning, which are techniques that can help switch off an I/O bank, or reduce the number of I/O standards used within a bank.

The "know your FPGA architecture" principle results from understanding the various power-down modes of dynamic resources such as PLLs, RC oscillators, and I/O banks as well as the capabilities offered by resources. Some FPGAs, such as Actel IGLOO FPGAs, offer the same I/O standards with various voltages. Using lower reference voltages for one or more banks will improve the savings even more[1]. A costly mistake FPGA designers can make is to use internal I/O pull-up/pull-down resistors as a precaution against the driving bus going tristate. In active mode, if the bus is driven with a value opposite to the pull ("0" for pull-up and "1" for pull-down), there is a drastic increase in static power per pin. Unless the device up-time is lower than 5%, the designer needs to communicate with the board designer to ensure continuous driving of the input, in particular for clock buffers.

Static-Power-Conscious Board Design

Board designers play a critical role when it comes to thermal management, voltage levels, and resistive loads.

Static power is adversely affected by increasing temperature. The nonlinear increase of static current could lead not only to an increase of static current but also to more serious issues such as thermal runaway. Keeping the ambient temperature as low as possible with cooling tricks is not an easy task—especially when board space and budgets are tight.

Driving inputs to the full voltage level, avoiding resistive loads, and grounding unused pins are all good board design practices for decreasing static power.

Reducing Dynamic Power

Dynamic power correlates with various parameters:

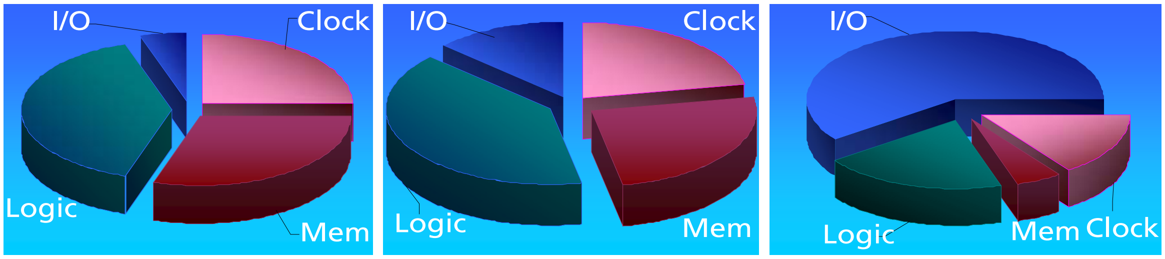
- Used FPGA resources (logic blocks, clock trees, embedded RAM, PLLs, etc.)
- Loads and resistive terminations on I/Os
- Data patterns and their arrival dynamics or signal activity or toggle rates
- Signal static probabilities

Designers must be more selective in fighting dynamic power than in reducing static power, channeling their efforts through analysis. Post layout, simulation-based power estimators are the basis for performing this analysis and must lead to a clear design power profile.

[1] The FPGA designer needs to perform a timing analysis to guarantee no interface timing violations.

Design Dynamic Power Profile

The design dynamic power profile provides a clear picture of the consumption of each of the used FPGA resources. Because FPGAs are flexible and allow many types of applications to be mapped on the same device, there is no formula for efficiently tackling the dynamic power without a deep understanding of the actual design power profile. Figure 5 provides examples of power profiles for different designs. Analyzing the power profile of the MPEG design helps the designer avoid wasting effort to lower the I/O dynamic power. However, when using a system controller, the focus of the power optimization and thermal management should be dedicated exclusively to the I/Os.



MPEG4 Codec ATM Switch Bone System Controller

Figure 5: **Designs Dynamic Power Profile**

The following sections are dedicated to power-aware design techniques that can be used to eliminate wasted power or radically reduce the contribution of each of the design blocks. Each section identifies where power is dissipated and presents the techniques along with an illustration of the effect of these techniques on actual silicon.

RAM Power Dissipation

RAM blocks use power during read and write operations. To lower this power usage, the circuits involved and addresses accessed have been analyzed in order to offer effective power reduction techniques.

Where Does the RAM Power Go?

For the read operation, the following circuits are involved:

1. Address and control latches
2. Row predecoder
3. Read column decoder
4. Row final decoder
5. Read column decoder control
6. Sense amplifier
7. Data output MUXes and latches
8. Sense enable logic
9. Read control logic
10. Bit-line precharge

Most of the current will be dissipated by the sense amplifier, bit-line precharge, and data output MUX/latches.

For the RAM write operation, the following circuits are involved:

1. Address and control latches
2. Row predecoder
3. Write column decoder
4. Row final decoder
5. Write column decoder control
6. Write driver
7. Bit-line precharge

Most of the current will be used by the write driver and bit-line precharge.

Figure 6 provides silicon power measurements for the reads and writes for 12, 16, and 24 cascaded RAM blocks (with binary and Gray address schemes). As expected from the description above, the write operation consumes slightly less power than the read access.

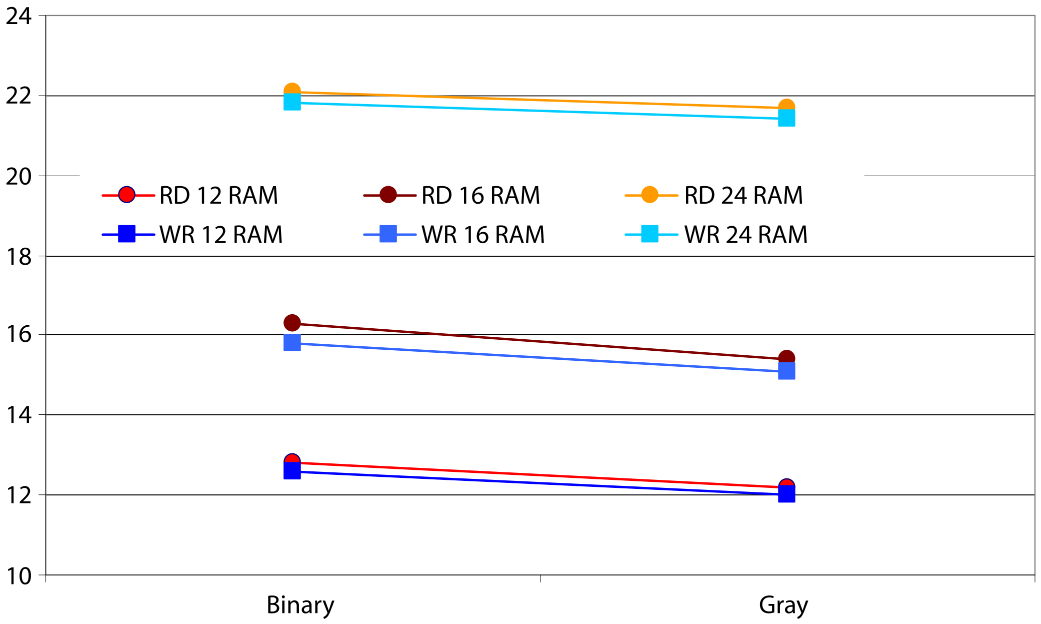


Figure 6: **Read vs. Write Dissipation for Various RAM Sizes and Address Encoding**

Impact of Address and Enable Signals on RAM Power

The most important signals that affect the power dissipation are the address lines, Read Enable (RE), and Write Enable (WE). Some other signals, such as Block Select, also play a role, but their impact is not covered in this paper.

Address Impact on RAM Power

While evaluating the impact of the address changes for successive read or write accesses on the RAM power dissipation, address randomness—based on the Hamming distance between successive addresses—has significant impact on power. Figure 7 provides silicon power measurements for different RAM sizes, and illustrates the increase of Read and Write RAM power with larger Hamming distance between successive addresses.

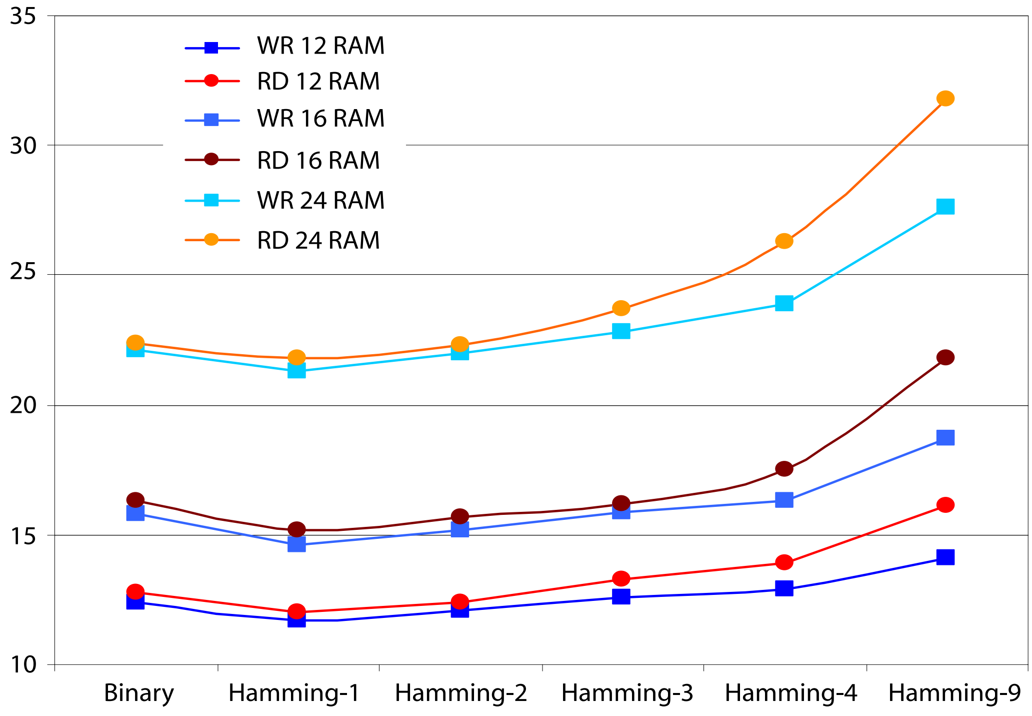


Figure 7: **Successive Addresses Hamming Distance – Read and Write RAM Power**

This experiment shows the significance of mastering the address changes and reducing the Hamming distances between successive addresses in order to minimize the RAM power waste.

Enable Signal Impact on RAM Power

Designers often write their RTL code without regard for the sequence of reads and writes, but investigation shows that efficient sequencing results in power savings. Figure 8 shows the difference between the power consumed when 1) a write is followed by a read and 2) three successive writes are followed by three successive read operations.

The conclusion is that more power is saved by performing as many write operations as possible before enabling the RE signal, and reading the memory for as many required data items as possible before switching back to a write operation. Figure 8 shows the impact of the address change strategy discussed above.

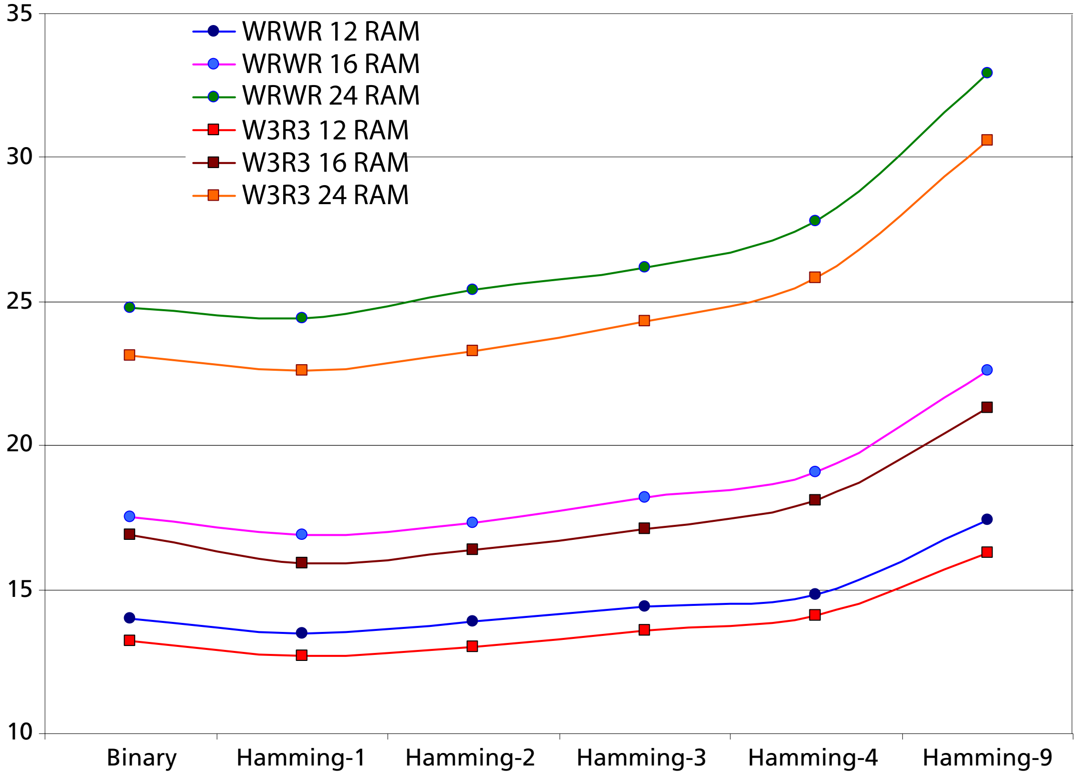


Figure 8: **Impact of Write and Read Sequencing on RAM Power**

Performing as many write operations with minimal-address Hamming distance as soon as possible, followed by as many read operations with least-address Hamming distance as late as possible, would harness the savings generally.

Even greater savings would result from disabling the Block Select signal between the last write operation and the first read access.

Other Techniques to Reduce RAM Power

There are more opportunities to reduce wasted power; in particular, when cascading multiple blocks to build a large RAM, or when the data and/or the address bits are not changing systematically every clock cycle.

RAM Cascading

FPGAs offer several embedded RAM blocks with unique sizes but variable aspect ratios. This feature opens the door for different cascading schemes. Figure 9 is an illustration of two alternatives that have different timing and power attributes.

In one case, all the RAM blocks toggle at each clock cycle as their outputs are concatenated to build the output. In the second case, only one RAM block is active at a time. However, there is overhead logic that not only could consume extra power, but also definitely affects timing. Designers should check whether the address-generation logic addresses one RAM a large number of times before moving on to another one. If the address locality is guaranteed, then cascading schemes where only one RAM is active at a time are viable.

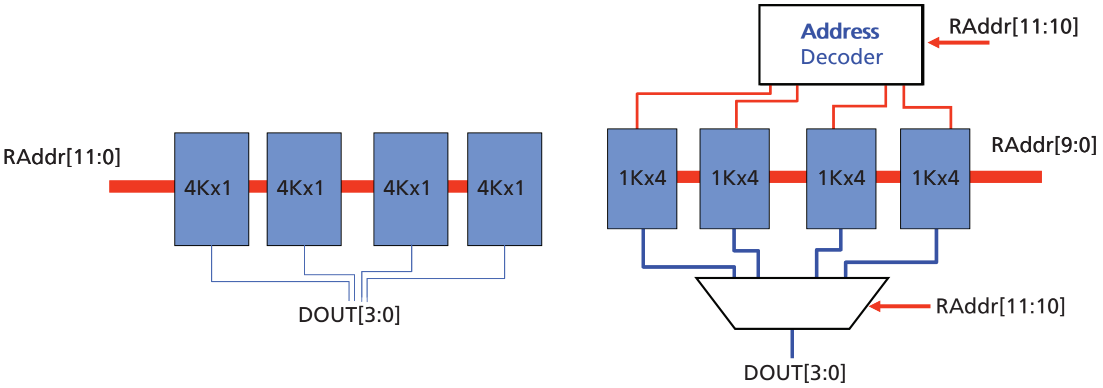


Figure 9: **Potential Embedded RAM Cascading Schemes for 4Kx4 RAM**

Root and Leaf Clock and Enable Gating

Several experiments have been developed to further reduce the RAM power dissipation in a design that uses a CAM. These experiments considered the RAM cascading options and the root and leaf gating of the read and write clocks, as well as the RE and WE with the address decoding. The silicon power measurement results are summarized in Figure 10.

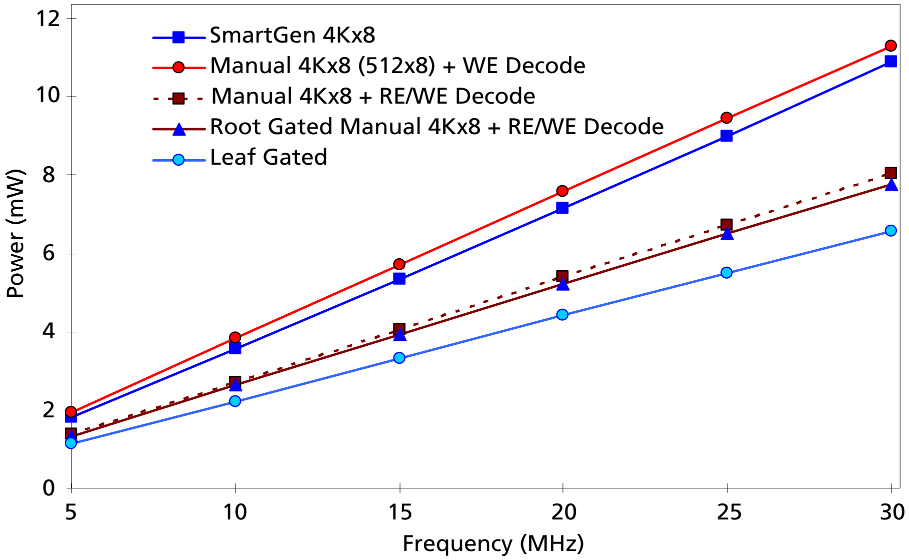


Figure 10: **RAM Cascading and Clock and WE/RE Gating Effect on RAM Power**

A New Technique to Mitigate Peak RAM Power

Usually, because of timing constraints, FPGA designers do not re-investigate the read and write clocks and their relationship. The issue of peak power is even worse when the read and write clocks are driven from the same source. This leads to potential simultaneous accesses on dual-port or two-port RAMs. The proposed technique, inspired by the DDR concept, is to use clocks with opposite edges for these ports. This method guarantees accesses staggered in time and thus a spread over time of the power dissipated by each access. This has been proven on multiple designs, provided that the timing constraints at the input and the output of one of the RAM access operations are met when inverting the clock.

I/O Power Dissipation

The highest power usage in the power profile for a large number of customer designs was in the I/O banks. In addition to the techniques for mitigating static power, described in the "Reducing Static Power" section on page 5, FPGA designers need to work very closely with the system architect and system board designers to challenge such decisions as I/O standards selection, interface timing requirements, electrical requirements and pinout constraints.

Power-Aware I/O Standards Selection

Differential I/Os (LVDS, LVPECL) and resistively-terminated I/Os (HSTL, SSTLs, etc.) have relatively high static power but the lowest dynamic power because of the limited voltage swing. The rule is to use these for the highest toggling frequencies.

For low frequencies and relaxed timing, using single-ended I/Os such as LVCMOS has the advantage of lowering the dynamic power—especially when the FPGA offers I/Os supporting voltages as low as 1.2 V.

Other Techniques for I/O Power Reduction

The first technique is to reduce the I/O number by reconsidering the design/function partitioning over several devices or eliminating I/Os that can be time-multiplexed.

To reduce the activity or toggling rates of the I/Os, designers must eliminate unnecessary glitches at the output of the I/O drivers. In case it is very complex to do so, the alternative is to use a tristate output buffer instead of a simple output and monitor the logic that generates the enable signal of the tristate. Another technique, which has been widely adopted as a good design practice, is selecting bus encoding that helps reduce the number of toggling bits and correlates successive values on the bus.

"Know your FPGA” includes knowing that for some I/O-terminated standards, the power dissipated when driving low is slightly different than when driving high. Using a bus encoding that favors the least drawn current or even inverting the output will lead to less total current for these I/Os. Study the static probabilities of the signals—the fraction of time that the signal will be logic ‘1’ during the period of device up-time.

FPGA designers should also determine the speed and waveform requirements in order to use the lowest possible drive strength that meets them. Working closely with the board layout team in defining pin assignment to I/O banks, designers may be able to reduce the number of compatible I/O standards and voltages required.

New Techniques to Mitigate I/O Power Consumption

Another design technique usually used to mitigate simultaneously switching outputs (SSOs) could also apply to reducing I/O peak power, provided the interface timing allows. This technique staggers the I/O transitions over time. The simplest method of staggering an I/O bus, driven internally by sequential elements, is to divide the active outputs into two groups, some at the positive edge of the clock and others at the opposite edge.

Board designers must pay attention to the design of a trace that implies the lowest capacitance without compromising of the signal integrity for high-speed signals.

Power-Conscious Clock Tree Exploitation

FPGAs have a fixed number of clock networks that can be used to map clocks, reset, and control other signals such as enables. Some FPGA architectures offer die global networks as well as quadrant clock networks.

Clock tree power depends heavily on the frequency and the fanout of each clock domain. At the frontend, designers can reduce fanout of the clock trees by revisiting their RTL and by reducing unnecessary pipeline stages, or applying local instead of global register balancing. As an example of RTL change, consider the truly pipelined processing depicted in Figure 11. If timing allows and there is no feedback loop, the designer can consider running the clock at half the original frequency and clocking each stage with the opposite edge of the preceding one. This will lead to a reduction of the clock tree power by half, without reducing its load. Pipeline stages including feedback loops can also be considered, but it is a more involved process because it impacts the functionality and requires nontrivial analysis.

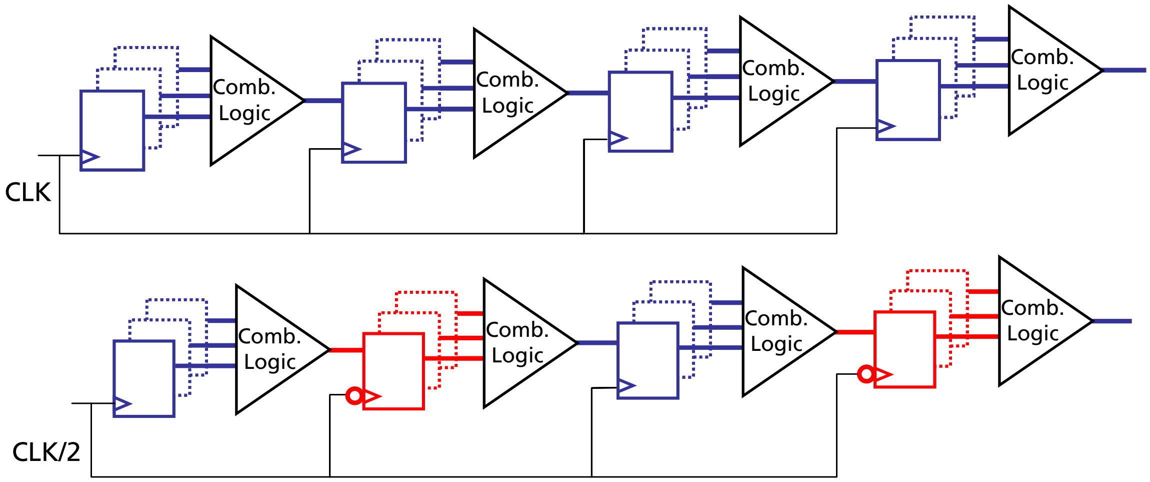


Figure 11: **Clock Power Reduction through RTL Changes of Pure Pipelined Structure**

Moreover, consideration must be given to timing constraints and their propagation. Using less stringent timing constraints prevents or minimizes sequential replication. In this context, users need to appropriately use one of the dangerous synthesis settings of the maximum allowed fanout limits. *This command leads to systematic replication or buffering whether the timing is met or not.*

In the backend, recent versions of vendor place-and-route tools switch off unused segments of clock trees, but experience has shown that this effort is sub-optimal. FPGA users can improve this during floorplanning of their designs by considering clock domains and the interactions among major functional blocks.

Figure 12 on page 16 shows examples of floorplans that allow the optimal exploitation of two clock networks. While this technique and these floorplan recommendations definitely reduce the clock tree power consumption, designers need to watch for the induced congestion that can offset this gain or make the power contribution of regular routing resources increase to an unacceptable level.

One other important feature of a clock network is that it enables the aggregation of the same clock tree to map several clock signals, control signals, or high-fanout nets. This feature also means a straightforward implementation of widely-known power reduction techniques such as clock and data gating.

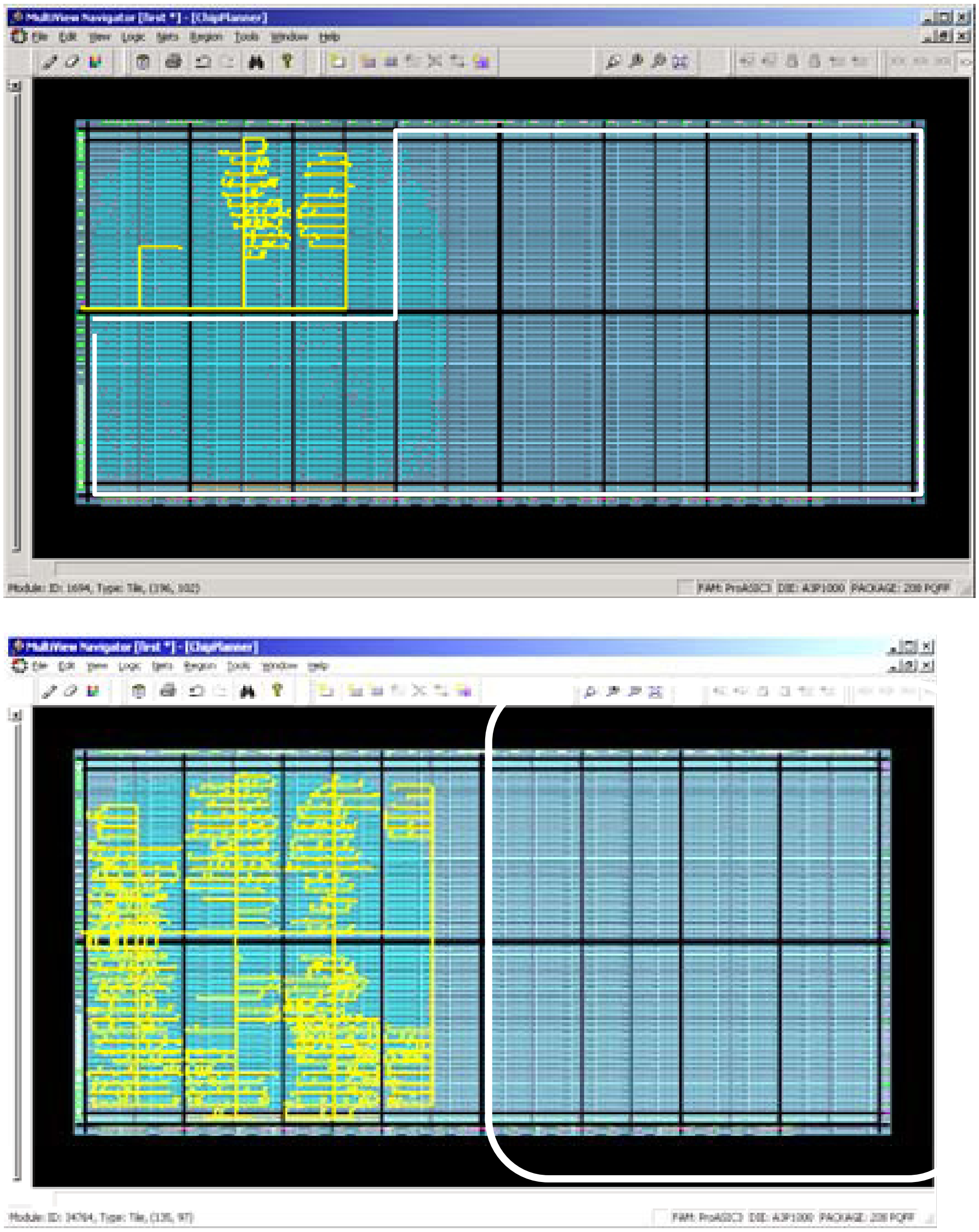

MultiView Navigator [first *] - [ChipPlanner]
File Edit View Logic Nets Region Tools Window Help
Module: ID: 1694, Type: Tile, (196, 102)
FAM: ProASIC3 DIE: A3P1000 PACKAGE: 208 PQFP
MultiView Navigator [first *] - [ChipPlanner]
File Edit View Logic Nets Region Tools Window Help
Module: ID: 34764, Type: Tile, (135, 97)
FAM: ProASIC3 DIE: A3P1000 PACKAGE: 208 PQFP

Figure 12: **Clock Domains Power-Aware Floorplan**

Figure 13 shows Actel's IGLOO clock tree architectures with dedicated MUXes to enable not only simple clock and data gating, but also lower power for regular nets by assigning clock segments to regular high-fanout nets [IGLOO2007]. Designers must notice that this routing assignment to high-fanout nets implies a placement constraint of the destination cells (the load) within the physical reach or scope of these routing resources. This implies the need for care and analysis to make sure this implicit placement constraint does not deteriorate timing or increase congestion.

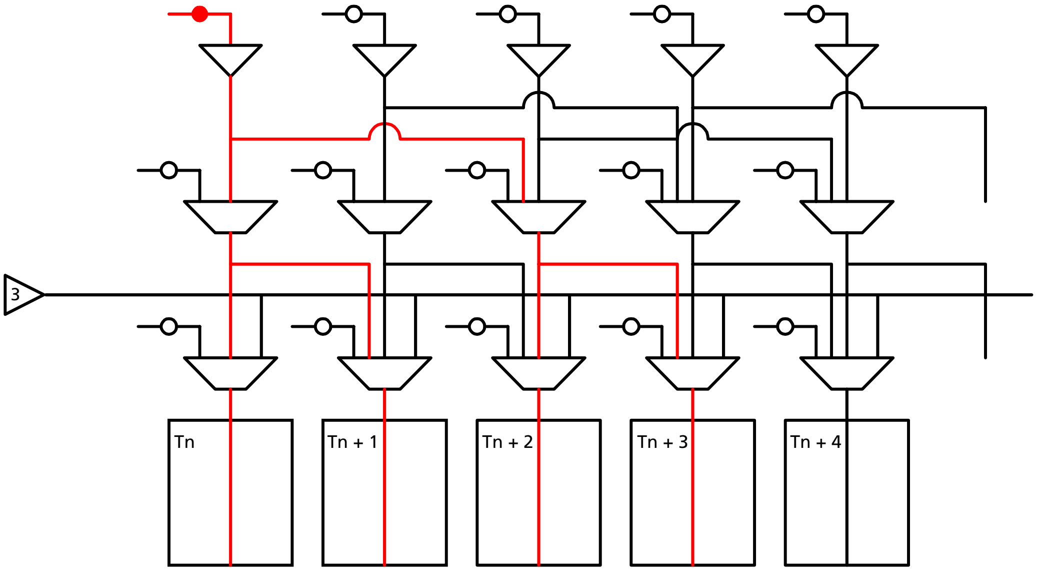


Figure 13: **Dedicated MUX Tree For Clock Network Aggregation**

Power-Oriented PLL Configuration

Currently, almost all FPGA architectures include several special clock conditioning circuits (CCCs) based on a PLL core for clock synthesis with potential negative and positive delays. FPGA designers usually care most about the input and output frequency ranges and the jitter attributes. Other important details that can help mitigate the power consumption of these CCCs may be missed. For example, some FPGA architectures allow dynamic reconfiguration of the PLL or even a power-down input pin to the PLL core [Fusion2006, IGLOO2007]. Users running their systems at different frequencies, depending on the operating mode, could use the dynamic reconfiguration feature, if they can afford the lock time when changing frequencies. This method eliminates the need for MUXing of clocks, which could introduce glitches, and allows the system to run at lower frequencies when appropriate.

Using the power-down mode of the PLLs prevents power dissipation by the CCCs, but also requires the tolerance of the lock time.

Finally, the various multipliers and dividers around the PLL core usually allow more than one combination to synthesize one or several output clocks. The choice of combinations reducing the maximum PLL output helps reduce wasted power to a minimum. Users are advised to check the results of the vendor's automatic CCC generators.

Reducing Logic and Nets Dynamic Power

Several papers have been dedicated to this subject alone [BASZ2000, Belhadj 2002]. The following sections focus on the power profiling of some arithmetic DesignWare (DW) elements, and a brief discussion of techniques to eliminate or reduce the propagation of unnecessary glitches.

Power-Aware Synthesis Options/Constraints Setting

Because of the pressure to meet timing requirements, few FPGA designers give sufficient attention to the synthesis process and use stringent global timing constraints. This usually leads to larger utilization of the FPGA logic elements and the associated routing resources.

Logic resources could be reduced by studying the slack distribution for each clock domain, which is the number of paths violating the required timing specification and the overall severity of these violations. For the blocks that have relaxed timing or sufficient margin in all internal paths, careful area-oriented synthesis will lead to a sizeable reduction of the needed logic resources and thus lower power dissipation for these blocks. The caveat is the need to analyze the result of the area-oriented mapping by performing a timing analysis and being alert for glitch propagation paths due to an artificially higher number of logic levels.

Setting synthesis options and constraints should be the focus, applied appropriately to timing-critical blocks or sub-blocks. Analysis of slack distribution and avoidance of global synthesis settings are keys to improving logic dynamic power dissipation.

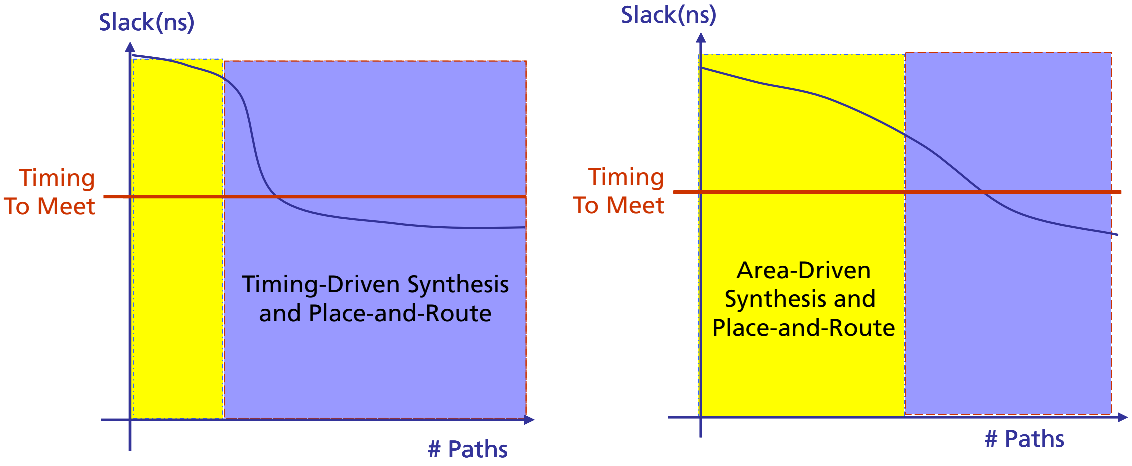


Figure 14: **Slack Distribution Analysis and Synthesis Goals Setting**

Power Profiling of DesignWare Arithmetic Blocks

Designs often perform heavy processing and require efficient implementations of arithmetic blocks. The DesignWare library offers a wide variety of arithmetic blocks with several architectures to better fit the area and performance needs of designs, considering their particular conditions and context. While the area and speed attributes are fully mastered for fine-grain FPGA architectures, the power profiles of these DW blocks need to be studied when power dissipation is a high priority. Figure 15 on page 19 through Figure 17 on page 21 provide the power figures for the various options of multipliers and adders. An analysis of the root causes of the differences in power numbers is covered in "Power Characterization of DesignWare and ModuleCompiler Elements on ProASIC" [Belhadj 2002]. The primary conclusion in the paper [Belhadj 2002] is that the arithmetic block power numbers are dominated by routing and correlate to a wide spread of the wire lengths.

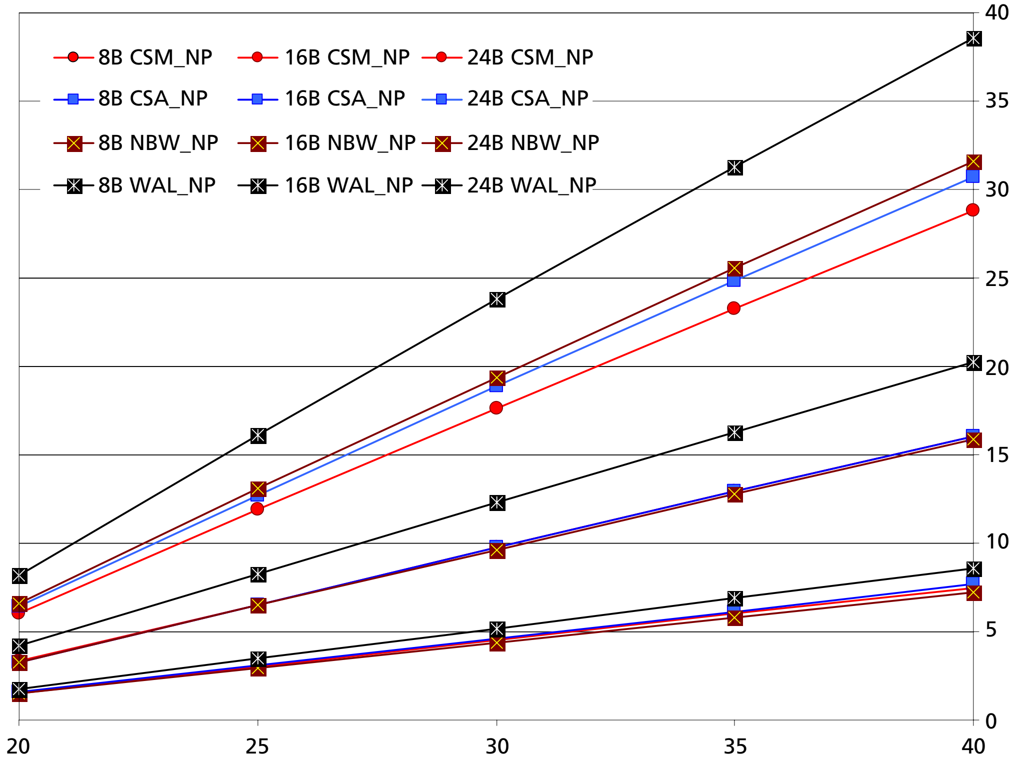


Figure 15: **8-Bit, 16-Bit, and 24-Bit DW Multipliers Power Profiles**

Architectural investigation uncovered a new adder architecture that mitigates the dependency of power arithmetic block power numbers on routing. The new architecture, subject of another publication [Belhadj 2008b], has one dominant feature: all internal nets have the same fanout.

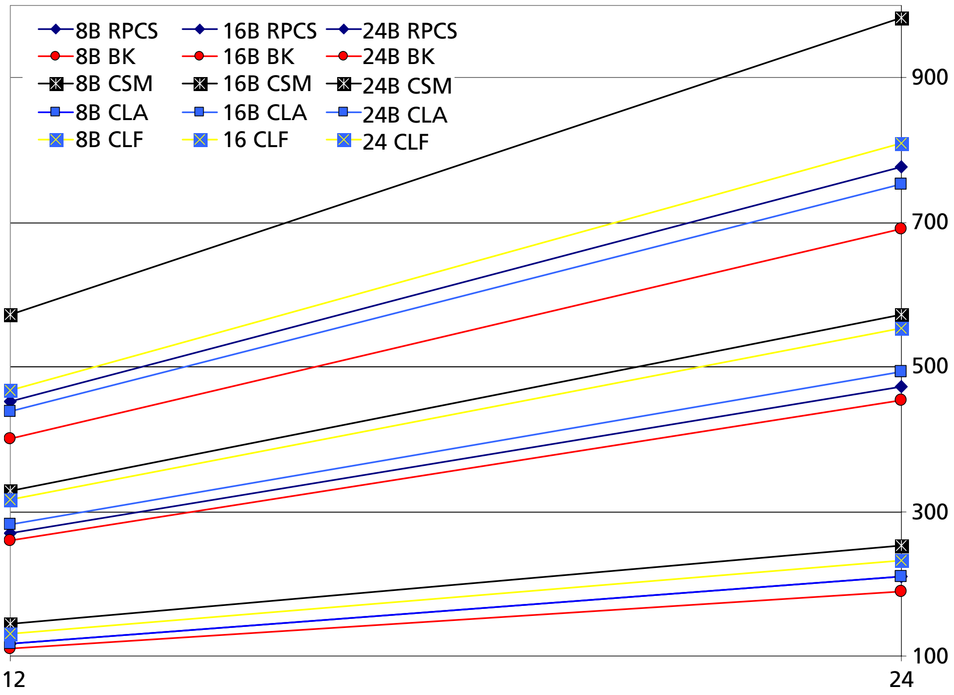


Figure 16: **Power Profile of 8-Bit, 16-Bit, and 24-Bit DesignWare Adders**

The results in Figure 17 compare the power figures of DesignWare Brent-Kung to this new architecture. The lead of the proposed adder architecture in terms of reduced power dissipation widens with the data width manipulated.

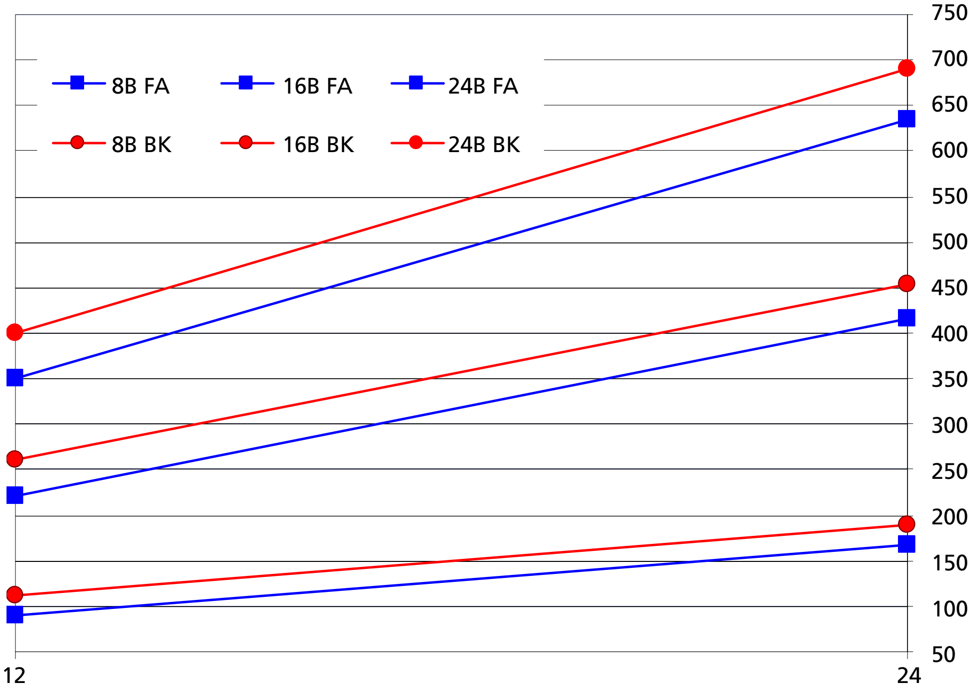


Figure 17: **Fast Adder Architecture vs. Brent and Kung DW Adder – Power Profile**

The recommendation is to consider not only the power figures associated with these arithmetic blocks, but also the area and speed attributes as well.

Power Profiling of Counters

Very often designs include several types of up and/or down counters with various lengths. There are three aspects of counters:

- Counters are used to account for a sequence of events, an elapsed time, etc.
- Counters also drive a load such as RAM address or data busses, a state machine's next state or output logic, etc.
- Counter outputs are used to perform certain processing when they reach various decoded values.

Figure 18 on page 22 provides the power profiles for Gray, binary, and ring counters. It reveals that the binary counters offer the lowest power, followed by the Gray, while the composition of ring counters has the worst power figure, mainly because of the large load on the clock. Systematically using sequential binary counters to reduce power is not a complete solution. Gray counters driving a large load bus allow better power saving than binary counters.

Moreover, both Gray and binary counters require more logic and will burn more power if a large number of counter values need to be decoded. Ring counters could be the best power-friendly solution if a uniform composition is used and when several counter values are decoded.

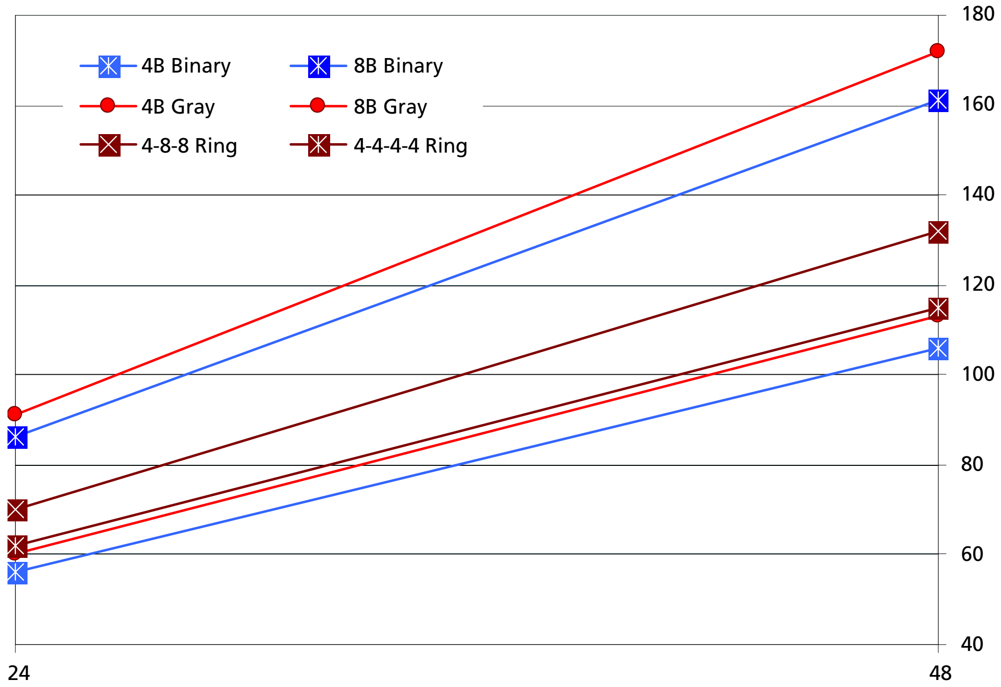


Figure 18: **Binary, Gray, and Ring Counters Power Profiles**

Logic Toggling Activity Spread and Peak Power

Most coding styles use one edge of the clock to drive all registers in a design without an underlying timing need. Designers prefer easy timing verification and often do not separate friendly analysis from actual need. While considering peak power, there is room for improving the peak consumption by revisiting the clocking. Figure 19 on page 23 illustrates the concept of changing the clocking schemes of a portion of the registers while remaining functionally equivalent. This change yields a different logic toggling profile and a lower static power. Notice, however, that the overall energy dissipated is the same in both the schemes.

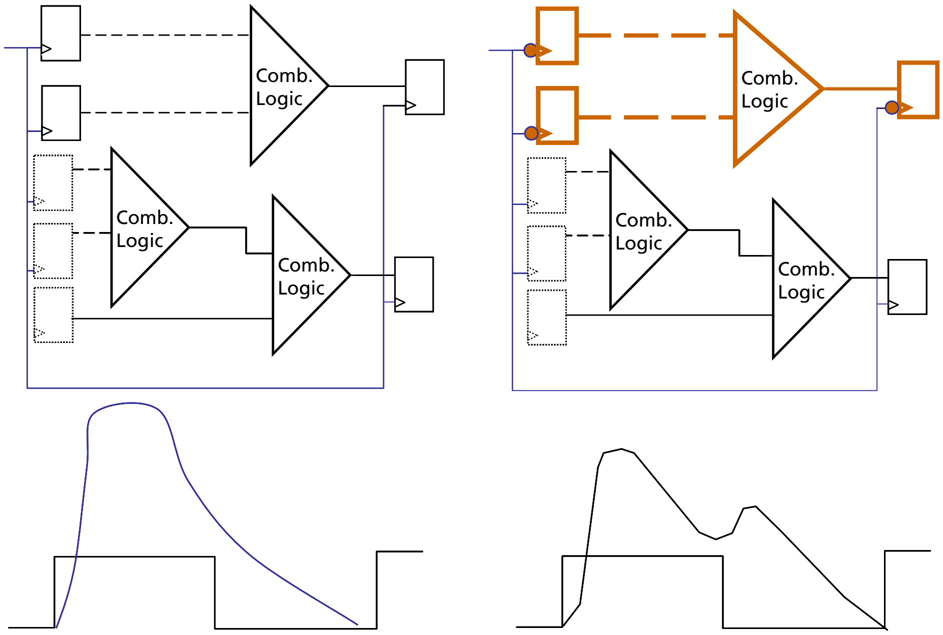


Figure 19: **Clocking Schemes Changes and Logic Activity Spread**

Glitch Reduction Techniques

This section covers the basics of glitch reducing techniques and provides hints for eliminating the wasted power due to unnecessary glitches in FPGA designs.

A signal is considered “glitchy” if it changes its value several times within a period of time and only the last value is actually exploited. Unstable logic expressions, unbalanced sets of paths driving a sensitive combinatorial cell, or a MUX select line that can toggle several times within a clock period are examples of sources of glitches. Some of the glitches are absorbed by the cell delays, as they are spurious and do not propagate. Some others, however, do propagate and can affect the power dissipation if they propagate through deep combinatorial cones and if their associated fanout is high.

There are two goals of glitch reduction techniques:

- Prevention—keeping the glitches from occurring
- Cure—reducing the propagation of glitches, or staggering them over time

Many publications have been dedicated to glitch reduction, and various approaches have been proposed for their prevention and cure [Raghunathan1996, Belhadj 2000, KKH1999, WYDFH2000, Wilson2004, Lim2005, SK2005]. Some of the proposals rely on pipelining, retiming, or insertion of a register with a phase-shifted clock in paths with large numbers of logic levels. Others introduce path-balancing techniques for ASIC designs. These are very difficult to implement in FPGAs because of the non-deterministic estimation of the routing delays in the FPGA world. Other efforts have been focused on multiplexer trees and have dealt with highly active inputs or select lines.

Actel’s more comprehensive approach takes into consideration both the timing attributes and the toggling activity of internal nets. In this context, the problems are classified into two main categories: glitches in single-cycle paths vs. glitches in multi-cycle and false paths.

The sketch of glitch reduction in the case of multi-cycle and false paths follows the steps listed below:

1. Identify highly-toggling nets and their drivers.
2. Estimate the worst-case timing for the inputs of the driver cells.

3. Push the source of the glitching up or down in the logic.
4. Re-estimate the worst-case timing for the inputs of the driver cells.
5. After the driver, insert a register clocked with opposite or direct edge of the clock, depending on the previous timing analysis.

Notice that this technique requires a change in the PrimeTime static timing analysis scripts in order to account for the inserted extra registers in the exceptions.

For single-cycle paths, even inserting registers with a phase-shifted clock as proposed in “Flip-Flop Insertion with Shifted-Phase Clocks for FPGA Power Reduction” [Lim2005] is very complex and may lead to drastic changes in behavior and tedious static timing analysis. The above approach is much more straightforward and requires enough timing slack to allow the insertion of an AND gate driven by the “glitchy” net driver and the clock opposite edge, as depicted in Figure 20.

The only caveat is that each time the clock signal goes high, the output of the inserted AND gate will transition to low. This extra toggle is worthwhile because it replaces a larger number of unnecessary toggles. There is no effect on functionality as long as the fanout of the AND gate gets into a logic cone that lands at a register input. Inserting a transparent latch will also do the job, but timing analysis gets a bit more complex.

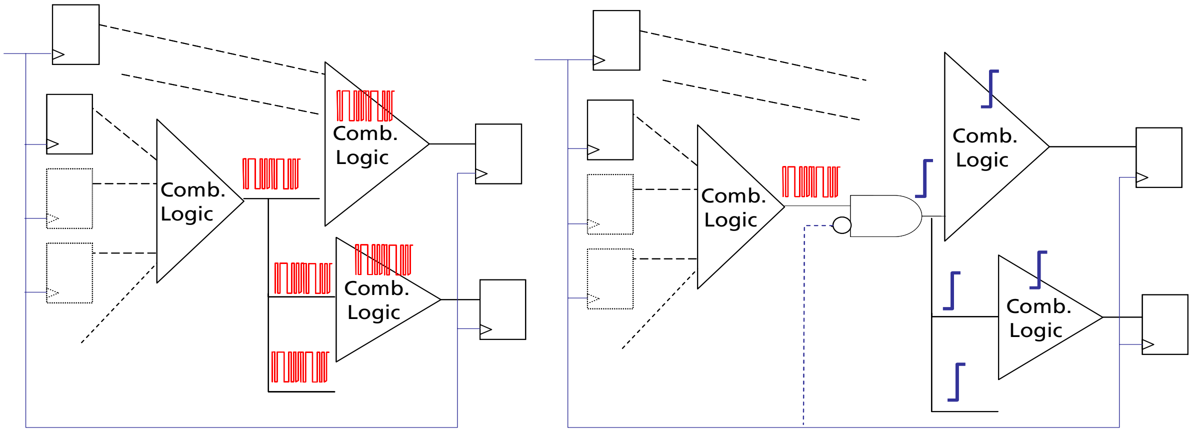


Figure 20: **Glitch Propagation Reduction**

Proposed Power Reduction Methodology

When dealing with stringent power budgets and targeting FPGAs, system architects and their design team need to know the end system operating modes and power profile. Armed with knowledge about the FPGA’s architecture, embedded functional blocks, and power-oriented features (power modes, various operating voltages, power-down of some blocks), the team has a great likelihood of meeting the power budget and channeling design efforts.

For the system architect, the proposed methodology is as follows:

1. Elaborate all the scenarios of end system operating modes. Include the thermal profile for each mode and the system power profile. Analyze the percentage of time in Idle, Sleep, and Shutdown modes, frequency of ON/OFFs, and the duty cycle of operation.
2. If the frequency of ON/OFFs is high, be alert for the inrush and programming currents for volatile FPGAs.

3. If the duty cycles show a small percentage of operation and are dominated by idle or sleep states, priority should be given to the reduction of static power, and then reduction of dynamic power (if needed).
4. If the duty cycle is balanced among all these modes, then efforts should be directed equally to both the static and the dynamic power components.
5. If the system is operating most of the time, work with the design team to develop a rough dynamic power profile for the FPGA design.

For the design team, the approach is summarized below:

1. Starting from the dynamic power profile, identify the largest power contributor (logic, clock trees, RAMs, I/Os, etc.)
2. Depending on the identified power-bottlenecks, use the summary tables (Table 2 on page 25 through Table 5 on page 27) and perform the following steps in order of priority:

   i. Make the RTL changes as far as timing allows.

   ii. Perform functional and timing validation of changes.

   iii. Apply the synthesis hints.

   iv. Use power-driven place-and-route.

   v. Apply the floorplan and place-and-route hints.

   vi. Validate.

Table 2: **Summary of Main RAM Power-Aware Techniques**
Refer to the "RAM Power Dissipation" section on page 9 for more information.

Design Option	Premises and Alternatives	Estimated Saving (depends on RAM Size)
Cascading Scheme	If overhead logic (MUXing and decode) is timing critical, and if addresses are not local, adopt cascading with all blocks active, elsewhere use the proposed alternative (Refer to "Other Techniques to Reduce RAM Power" section on page 12 for more information).	"-2%" .. 5%
Root and Leaf Gating of the Enable and Clock	If not possible, adopt root gating of the clock.	5% .. 10%
Write/Read Access Sequencing	As many Write operations before as many Reads as possible. If Read/Write are simultaneous, spread them over 2 edges of the clock (if timing allows).	2% .. 5%
Successive Address Changes	Least Hamming distance	1% .. 4%

Table 3: **Summary of Main I/O Power-Aware Techniques**
Refer to the "I/O Power Dissipation" section on page 14 for more information.

Design Option	Premises and Alternatives	Estimated Saving
Reduce the Number of I/Os	Partitioning and/or time multiplexing	The number of I/Os Contribution per I/O (I/O standard, voltage swing, and toggle rate dependent)
Use Tristate Out Buffer Instead of Plain Output Buffers	Drive wisely the enable signals for tristate buffers.	Proportional to enable toggle rate
I/O Toggle Rate	Investigate bus encoding	Depends on bus encoding quality
I/Os Toggle Timing	Stagger I/O toggling in time is STA allows	No gain but reduction of peak power
I/O Standards Selection and I/O Bank Assignment	For high-frequency signals, use differentials or voltage- referenced standards (mem. interface, etc.). For lower frequency signals, use single-ended with least voltage (1.2 V instead of 1.5 V or 1.5 V instead of 1.8 V, etc.), provided timing and waveform requirements allow.	The higher the frequency and lower the voltage swing, the higher the saving

Table 4: **Summary of Main Clock Tree Power-Aware Techniques**
Refer to the "Power-Oriented PLL Configuration" section on page 17 for more information.

Design Option	Premises and Alternatives	Estimated Saving
Clock Gating	STA and functional validation required.	Could be substantial, but design-dependent
RTL Changes to Clock Groups of Registers with Opposite Clock Edge	Timing must allow for the change.	Peak power reduction depends on the size and routing in the logics driven by registers (could be substantial)
	Easy: No feedback loops	Up to 50%
RTL Changes for Pipelined Logic	Challenging: If feedback loops exist between stages, will require detailed STA.	10%+
RTL Elimination of Unnecessary Pipeline stages	Requires functional and timing validation.	Depends on the size of the eliminated registers
Timing-Driven "set_max_fanout" Setting	Apply if large number of high fanout nets is important and relaxed timing.	Depends on the number of eliminated register replications
Area Oriented Synthesis	Enough positive slack margin and sizeable blocks	Depends on the size of the timing relaxed blocks
Power Driven Place-and- Route	Need to combine it with timing constraints and timing-driven place-and-route.	Up to 10%
Clock Tree Floorplan	Analysis of implicit placement constraints and potential artificial congestion	Depends on die size and span of clock tree

Table 5: **Summary of Main Logic Power-Aware Techniques**
Refer to the "Power-Oriented PLL Configuration" section on page 17 and "Reducing Logic and Nets Dynamic Power" section on page 17 for more information.

Design Option	Premises and Alternatives	Estimated Saving
Area-Oriented Synthesis	STA shows enough positive slack.	Could be substantial but design-dependent
Power-Friendly Arithmetic Blocks	Timing and area attributes of blocks required.	5% to 15%
Power-Friendly Counters	If only final count is used, binary is best. If counter used to drive bus, Gray is best. If several counter values need decoding, ring counters are best.	Depends on number and size of counters as well as use model
RTL Changes for Glitch Reduction	Insertion of registers in multi-cycle and false paths	Depends on logic size in false and multi-cycle paths
	Insertion of AND gate driven by opposite clock edge; requires minor positive slack in "glitchy" paths.	Depends on size of downstream logic and glitching activity
	Insertion of latches driven by opposite clock edge requires STA.	Depends on size of downstream logic and glitching activity
Power-Driven Place-and- Route	Need to combine it with timing constraints and timing-driven place-and-route	5% to 10%

Two disclaimers are necessary:

- *The estimated power-saving figures listed in Table 2 on page 25 through Table 5 apply to the block or the logic to which the hint has been applied. They do not necessarily translate in overall design power saving percentage, and they do not accumulate systematically.*
- *The accuracy of the savings figures will be validated with a large number of real-life designs that provide a large enough sample of various dynamic power profiles.*

Conclusion

This paper has shown that power reduction can and should begin with initial design considerations. Using the smallest die and fewest resources necessary to achieve performance goals is a vital first step. Selecting an FPGA should take into account whether it requires external resources and the overall power profile of the system. When designing the system, techniques such as sequencing read and write accesses for optimum speed, RAM cascading, and using opposite clock edges can improve power savings. I/O power dissipation can be reduced with several techniques as well, and may offer the greatest savings because in many designs the I/O banks were found to consume the most power.

In addition to explaining specific techniques, this paper presented a methodology for design teams and system architects that will help them approach a new design task with the coordination and analysis necessary to meet stringent power budgets. With this information and the FPGA technology available, low power is now a much more realistic goal when designing with FPGAs.

Acknowledgment

Actel thanks Jim Joseph for his contributions to this paper.

References

[BASZ2000] Hichem Belhadj, Vishal Aggrawal, Nelson Soria, Behrooz Zahiri, “Power- Conscious Design on Fine Grain FPGAs,” SNUG Europe, 2000.

[Belhadj2002] Hichem Belhadj et al, “Power Characterization of DesignWare and ModuleCompiler Elements on ProASIC,” SNUG Boston, 2002.

[BADAZ2008a] Hichem Belhadj, Vishal Aggrawal, Julien Dunoyer, Nizar Abdallah, and Amal Zerrouki, “Power Analysis and Thermal Management: The 10 Commandments”, submitted to SNUG Europe, 2008.

[Belhadj2008b] Hichem Belhadj, “Adders: Area-Timing-Power Space Exploration”, submitted to a Conference.

[CoolRunner2006] CoolRunner-II CPLDs, http://www.xilinx.com/products/silicon_solutions/cplds/coolrunner_series/coolrunner_ii_cplds/index.htm

[Fusion 2006] Fusion PSC, http://www.actel.com/products/fusion/default.aspx.

[IGLOO2007] IGLOO FPGA, http://www.actel.com/products/IGLOO/default.aspx.

[KKH1999] Sungjae Kim, Juho Kim, and Sun-Young Hwang, "Efficient Algorithm for Glitch Power Reduction," IEE ELECTRONICS LETTERS, 24th June 1999, Vol. 35 No. 13, pp. 1040–1041.

[Lebas2007] Pierre Lebas, “Low-power synthesis flow for wireless digital circuits,” SNUG Europe, 2007.

[Libero2009] Actel Libero® Integrated Design Environment (IDE) v8.5 Release Notes.

[Lim2005] Hyeonmin Lim et al, “Flip-Flop Insertion with Shifted-Phase Clocks for FPGA Power Reduction,” Proceedings of IEEE/ACM Intl. Conference on Computer Aided Design (ICCAD ’05), San Jose, California, November 2005.

[MaxIIZ2007] Max II-Z CPLD, http://www.altera.com/b/maxiiz.html.

[SK2005] George Selimis and Odysseas Koufopavlou, “Multi-Operation cryptographic engine: VLSI design and implementation,” Journal of Physics: Conference Series 10 (2005), 377–380.

[Wilson2004] S.J.E. Wilson et al, “The Impact of Pipelining on Energy per Operation,” FPL and Applications, Belgium, August/September 2004, pp. 721-725.

[WYDFH2000] N. Weng, J. S. Yuan, R. F. DeMara, D. Ferguson, and M. Hagedorn, “Glitch Power Reduction for Low Power IC Design,” in Proceedings of the Ninth Annual NASA Symposium on VLSI Design, pp. 7.5.1 – 7.5.7, Albuquerque, New Mexico, U.S.A., November 8 – 9, 2000.

[Raghunathan1996] Anand Raghunathan et al, “Glitch Analysis and Reduction in Register Transfer Level Power Optimization”, DAC, 1996.

[KKH1999] Sungjae Kim, Juho Kim, and Sun-Young Hwang, "Efficient Algorithm for Glitch Power Reduction," IEE ELECTRONICS LETTERS, 24th June 1999, Vol. 35 No. 13, pp. 1040-1041.

Actel is the leader in low-power and mixed-signal FPGAs and offers the most comprehensive portfolio of system and power management solutions. Power Matters. Learn more at www.actel.com.

Actel®
POWER MATTERS

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

Actel Europe Ltd.
River Court, Meadows Business Park
Station Approach, Blackwatery
Camberley Surrey GU17 9AB
United Kingdom
Phone +44 (0) 1276 609 300
Fax +44 (0) 1276 607 540

Actel Japan
EXOS Ebisu Building 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150 Japan
Phone +81.03.3445.7671
Fax +81.03.3445.7668
http://jp.actel.com

Actel Hong Kong
Room 2107, China Resources Building
26 Harbour Road
Wanchai, Hong Kong
Phone +852 2185 6460
Fax +852 2185 6488
www.actel.com.cn