Simple and Effective Ideas to Reduce Dynamic Power Consumption in FPGA Design

By

Ofer Hofman

ofer@sitaltech.com

Sital Technology Ltd., 17 Atir Yeda St., Kfar-Saba 44643, Israel
Tel.: +972-9-7633300 Fax: +972-9-7663394 Email: info@sitaltech.com Web: http://www.sitaltech.com

Introduction

As FPGA devices include more logic, memory and special functions such as memory interfaces and DSP blocks, and as they move to newer and more comprehensive technologies, they make increasing demands on power.

There are numerous benefits in reducing power consumption such as better reliability, lower cooling cost, simpler power supply and delivery and longer battery life. Designing for low power consumption without compromising performance requires a power-efficient FPGA architecture and good design practices.

Types of Power Consumption

There are two types of power consumption in an FPGA: static power and dynamic power. Static power consumption is the power used when the design is idle. It is independent of the activity of the circuit and is dissipated either as transistor leakage or as bias current. Total static power is the combined total of each transistor's leakage power and all bias currents in the FPGA. As process technologies shrink, from 130nm to 90nm to 65nm and onward, and as transistor dimensions shrink, transistors become inherently more leaky and static power consumption rises.

Dynamic power is the power consumed by activity in the part, i.e., the charging and discharging of capacitance within the part as it manipulates or moves data. For example, the faster a gate is toggled the more dynamic power it consumes.

Reducing Dynamic Power Consumption

Sometimes, because of a particular functional implementation, some signals are toggled very fast, but unnecessarily. As a result, extraneous power is consumed and wasted. Power consumption could be reduced if we could find these points and shut down the unnecessary toggling.

The responsibility for locating these points in the design and solving them is in the hands of the electronic engineer/designer. Below, we provide helpful tips to the designer for solving common, but often overlooked, dynamic power-consumption problems.

Finding the Toggle Stoppers

The starting point is to identify *toggle stoppers*. Toggle stoppers are gates, or flip-flops, in a design that have an intensive input toggle rate and a very slow or nonexistent output toggle rate. The input toggling is wasted as it does not contribute anything to the function of the product. These gates are perfect candidates for dynamic power optimization.

A good method for optimization would be to tunnel back in the design from the toggle stopper in order to reach a subsystem that is doing work for no useful purpose. Stopping this unit could reduce a great deal of dynamic power consumption.

Examples

Below are three practical examples of how to identify and solve dynamic power wastage.

Example 1 - Set-Top Box

The first example is a general explanation of a common type of problem: the lack of communication between devices or blocks where the transmitter does not know that the receiver is unavailable and continues to transmit. There is no performance problem in the repeated extraneous transmission, but it results in a tremendous waste of power.

A set-top box for cable TV reception decodes the selected channel, decodes the MPEG compression, produces HD video and sends it to the TV. But what if the TV is off?

This is a classic case of an intensive input toggle rate and no output toggle. The toggle stopper is in the TV. The toggle input rate is video while the first turner block's output is null! To solve this problem, one would tunnel all the way back to the cable plug and shut down all set-top box activity, all of which is wasted.

Is this simple solution implemented today? No! There is no feedback from the TV to the set-top box. Thus millions of set-top boxes in the world work in vain and expend energy for nothing most of the day!

In FPGA design, there can be cases like this one. The designer should look for them in order to reduce unnecessary power consumption.

Example 2 - AND Gate Design

From the standard AND gate, all sorts of circuits can be built. In this example, we show how a very simple and common oversight can result in a surfeit of unnecessary toggles and waste of power. It is not hard to imagine how far this oversight can multiply power wastage over an entire device.

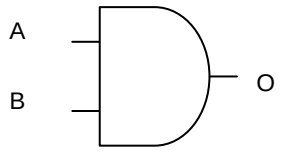
Assume a simple 2-input AND gate in a major design as displayed.

A and B are inputs. O is the output.

The number of toggle events of A is saved in Tg(A).

The number of toggle events of B is saved in Tg(B).

The number of toggle events of O is saved in Tg(O).



For a given period of simulation time, if (Tg(O) is much smaller than Tg(A) or Tg(B), then this could indicate a situation where A or B are toggling for no purpose. Energy is lost for nothing. For example, if this AND gate had one of its inputs disabled, the other input is essentially toggling for nothing as the output will always be 0. This is a complete waste of power.

The designer who does not pay attention to this situation is not aware of the power loss he is causing in his design.

Example 3 - Arbiter Design

In this more complicated example, we show how a typical arbiter design can result in unnecessary power consumption.

In a certain implementation:

Eight units read or write data to and from shared memory.

Each unit has a 64-bit address and a 64-bit data bus.

In order to communicate, each unit sends a request signal and waits for a grant from the arbiter.

The arbiter works in a round-robin fashion. It loops and checks these eight request signals in eight consecutive clocks.

When a request goes high, the arbiter parks on this unit, provides a grant, and muxes out the data and the address buses of this unit to the shared resource. At the same time, the arbiter instructs a memory controller to manage the transaction for this unit.

This is certainly a common communication design and it works well. But is it power-efficient? Let's check.

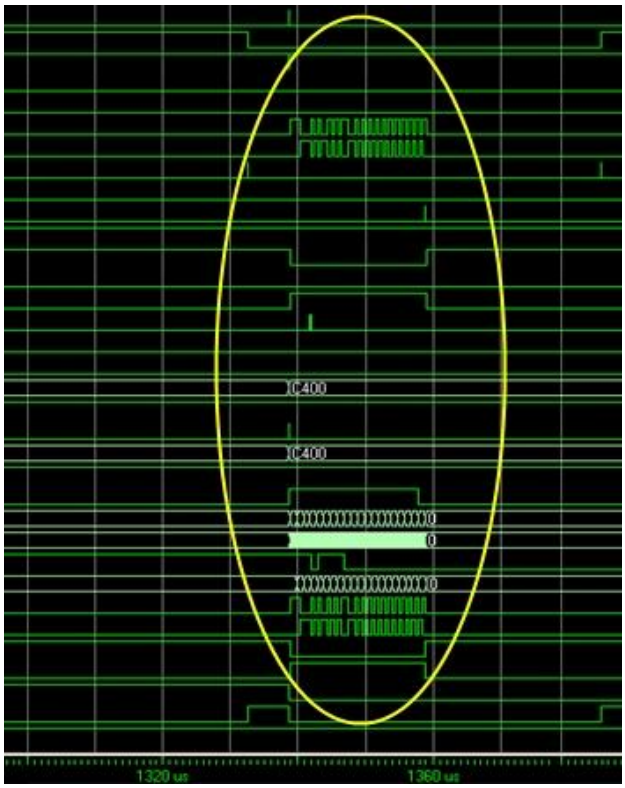
When no unit requests the memory, the arbiter loops. Every time the arbiter changes, it points to a new unit and this unit's data and address buses are multiplexed to the memory controller. This muxing of the address and the data buses when there was no request is not an error; the design will work perfectly. Yet, there is a huge waste of power: 128 bits were toggled every clock, but they were not used! In the memory controller, there is a register that samples the multiplexed address and data bus when it is instructed to work. When there is no request, this register does not sample these 128 bits. So this register is a toggle stopper.

Now the designer has to think how to solve this waste of power. What should he do?

A possibility is to park the arbiter until there is an active request from one of the eight units. This is a very simple register transfer level (RTL) change without disturbing the functionality. And a lot of power is saved!

Sital Solves Power Consumption Problems Automatically

Sital Technology is well acquainted with all sorts of dynamic power consumption problems and has designed into its FPGAs numerous automatic power-saving techniques. In practical application after practical application, in project after project, Sital FPGAs successfully employ solutions to the types of power problems described above. For example, gates are monitored for situations where toggles are unnecessary and feedback loops are implemented to make communications more efficient. In fact, for a comparable design, Sital's FPGAs approach the low-power characteristics of ASICS!



The display on the right shows the toggling of a Sital FPGA. As can be seen, toggling occurs only when necessary, thus saving considerable power.

Sital designs are used by many customers in space and avionics applications always meeting customer requirements for performance along with efficient power usage. For example, for a large customer in India, Sital designed, developed and delivered a VME board. In a controlled test at 70º C, the temperature increase from the Sital board was measured to be a mere 3º C!

Conclusion

An engineer who is keen to reduce power consumption can use design techniques to achieve his goal. Since the final product is often replicated many times, like the chips in a car or airplane, the power savings can become quite significant and can certainly contribute greatly to energy economies.

As an industry leader, Sital designs into its products important techniques for dynamic power reduction. Customers who use Sital products have a head start in obtaining significant energy reductions and are pleased with the results.

If you distribute this document to your engineers, you will contribute to energy savings!

-- Ofer Hofman