How to Manage Power Consumption in Advanced FPGAs

In this article, we discuss various sources of power consumption and techniques that can be used for power management in advanced FPGA devices. Then we describe how various features of Synplicity's Amplify Physical Optimizer software can be used to realize the power management techniques.

by Sanjay Thatte
Technical Marketing Manager
Synplicity Inc.
sanjay@synplicity.com

and John Blaine
Customer Application Engineer
Xilinx Inc.
john.blaine@xilinx.com

Current technology trends point to a rapid growth in FPGA device sizes with designs operating at higher frequencies. These design characteristics give rise to a number of complex issues. As designers, you must simultaneously:

- Address the stringent goals of meeting fast timing performance
- Fit the design into a cost-effective device
- Meet aggressive product schedules.

Additionally, high operating frequency and a high percentage of device utilization can increase design power consumption and junction temperature substantially. As the temperature rises past typical device ratings, performance and reliability are degraded. Device power consumption can reach a level that causes the maximum rated junction temperature to be exceeded, resulting in thermal destruction of the FPGA device.

Factors such as ambient temperature, airflow, and heat sinks, which can prevent device overheating, may be beyond your control. Industrial parts with extended temperature ratings are an option, but these parts are expensive and have limited package selection.

In addition to timing, area, and time to market, it is imperative that you proactively address the issue of power consumption and thermal stability.

An FPGA device, used in battery-powered applications or even high-performance applications where heat dissipation is a concern, can benefit significantly by applying some basic power management techniques.

Power Consumption

Power consumption in digital CMOS circuits arises from:

- Leakage current
- Transient short-circuit current between supply rails during transistor switching
- Charging/discharging of parasitic capacitances during normal internal logic state changes
- Charging/discharging of parasitic capacitances due to variation in input arrival times
- Charging/discharging of external load capacitances.

Neither the leakage nor the transient current can be optimized by design implementation. Therefore, to minimize power consumption, you must focus on optimizing the last three sources of power consumption involving capacitance.

The general formula for calculating power consumption for a design is based on the operating voltage, sum capacitance of all interconnect and logic resources, and the frequency of transition at the nodes.

$$P = \text{sum}\ (C * V^2 * f)$$

P = total power consumption
V = operating voltage
C = net capacitance
f = transition frequency

The operating voltage and external load capacitance are typically determined by system design requirements. To minimize power consumption for an FPGA device, the internal net capacitance and the toggle frequency must be reduced.

Power Management Techniques

Techniques used to minimize power consumption attempt to reduce the number of switching signals and the capacitance on the nets that are switching frequently. Some of these power minimization techniques are to:

- Minimize the number of clock buffers switching and the clock network capacitance
- Minimize capacitance on high frequency logic
- Isolate high activity logic to reduce interconnect length
- Isolate memory with high frequency access
- Minimize unnecessary switching and eliminate glitches.

Power management involves the application of advanced tools at the beginning of the design cycle to address the complex issues of interconnect. Fortunately, minimizing interconnect capacitance on critical nets helps you to reach your timing performance goals as well as reduce power consumption. Standard logic synthesis tools are not equipped to help you proactively manage the interconnect capacitance. As a designer, you must use a more advanced physical synthesis tool to achieve these goals.

Synplicity's Amplify® Physical Optimizer™ tool is the only market-proven FPGA physical synthesis software solution available today. More than 130 companies are already using the Amplify tool to manage the interconnect-related issues effectively and to reach aggressive timing performance goals.

Amplify Physical Synthesis

The Amplify tool provides an intuitive interface for creating regions on an FPGA device and then assigning the desired logic to those regions. The tool then uses physical constraints that incorporate your knowledge of the design's timing and power requirements to perform physical synthesis and to create a highly optimized design netlist.

The following sections briefly describe a number of Amplify's advanced features that can help you effectively manage timing and power goals. [Editor's note: For a more in-depth description of Amplify tool capabilities, go to Xcell Online at *www.xilinx.com/publications/xcellonline/*.]

1. HDL Coding Style

The Amplify tool uses HDL code with design constraints, such as timing and physical constraints, to perform advanced physical synthesis. You can significantly influence device power consumption through careful HDL coding. With proper coding structures, logic can be turned off when not needed.

2. Gated Clock Support

The Virtex™-II family of FPGAs makes advanced clocking schemes available to designers. Using Amplify software, you can take advantage of primitives like BUFGMUX for switching from a high-frequency clock to a low-frequency clock,

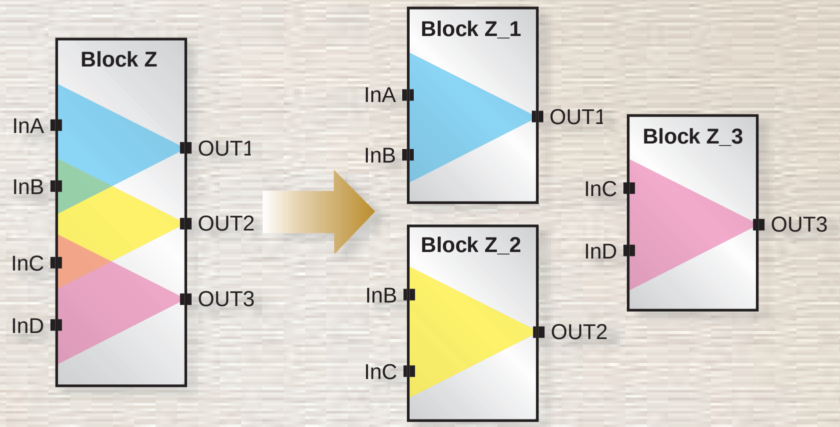

Figure 1 - Zippering

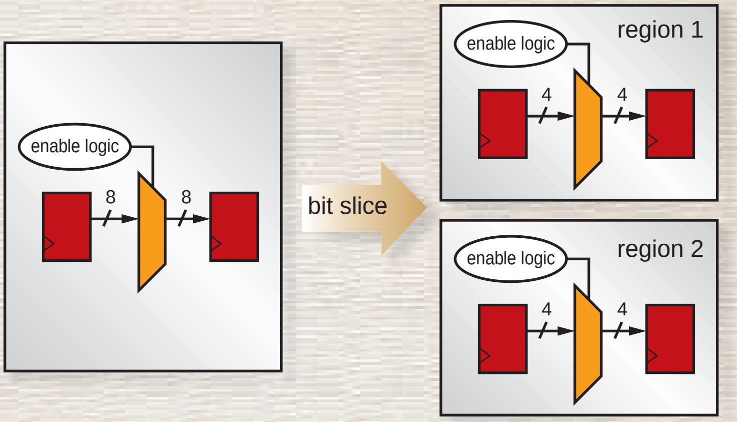

Figure 2 - Bit-slicing critical paths

and BUFGCE for dynamically driving a clock tree only when the corresponding logic is used.

3. Retiming and Pipelining

Typically, large logic blocks have long and active critical paths The Amplify tool automatically rebalances long critical paths by moving registers across logic boundaries to reduce the path length, net capacitances, and variance in delay paths to minimize glitch power.

4. FSM Encoding

The Amplify tool performs powerful state machine encoding and optimizations automatically. The FSM Explorer function can make use of user-specified constraints to choose the optimal encoding for state machines in the design.

5. Pad Type Selection Support

A large output load can increase power consumption. The Amplify software allows you to specify the pad type used when driving these loads. You can select a slower pad with the xc_pad_type attribute to reduce power consumption.

6. Zippering

Large functional blocks typically tend to be spread out over the FPGA device during placement and routing, causing some nets to have large routing capacitances and unnecessary power consumption. The Amplify tool provides a powerful netlist restructuring capability (Figure 1) to manage such large functional blocks in a design. You don't have to make any changes to the RTL code, and all netlist restructuring is done by the Amplify tool.

7. Bit-Slicing

When a large bus is routed, groups of bus bits must be clustered together to ensure similar timing. The Amplify tool's easy to use graphical interface (Figure 2) allows you to specify both the number of slices and the number of bits per slice. Combining bit-slicing with RTL floorplanning helps you get more uniform delays on the inputs to logic blocks and minimize glitch power. Also, by gaining finer control over bus placement and routing, you can control the capacitances of associated nets and minimize power consumption.

8. RTL Floorplanning

The Amplify Physical Optimizer software provides a user-friendly graphical interface (Figure 3) for creating physical constraints interactively. Working on a display of the device footprint, you can create rectangular regions of desired size at selected locations on the device. You can then assign entire modules or logic on selected critical paths. Through this process, you can easily localize critical paths to restrict the length of critical nets. Controlling the net length prevents incurring large routing capacitances and in turn, excessive delays and power consumption on these nets.

Because the floorplan is created before the design is synthesized, the Amplify tool makes use of the physical constraint information to better optimize the netlist. This netlist, created through Amplify's physical synthesis function, can be tailored specifically to your timing and physical constraints. The Amplify software also synthesizes the gate-level floorplanning it derives from the specified physical constraints, and it forward-annotates the information to the place-and-route tools.

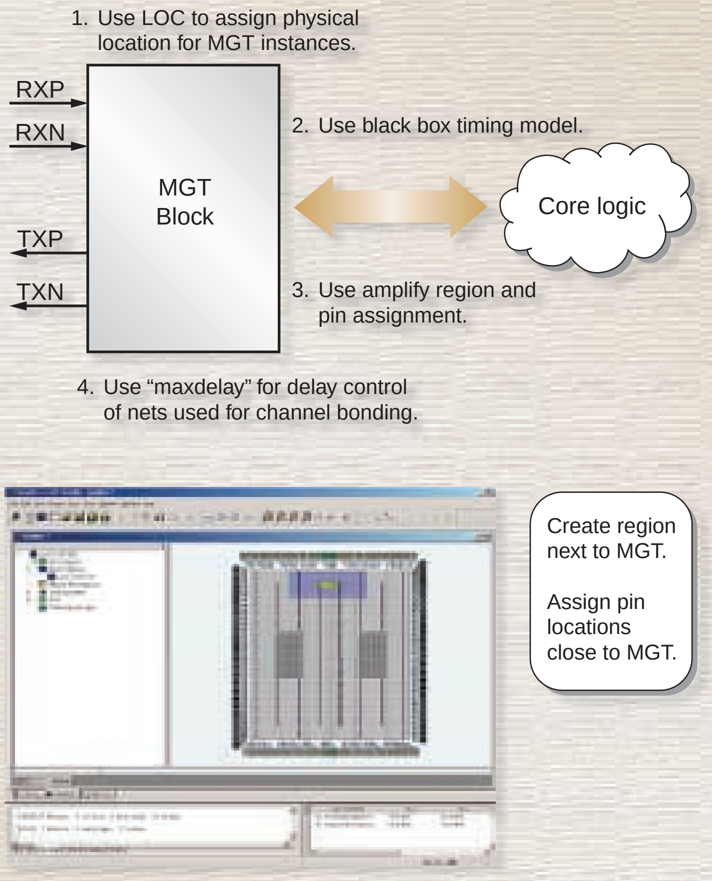
1. Use LOC to assign physical location for MGT instances.
RXP
RXN
TXP
TXN
MGT Block
2. Use black box timing model.
Core logic
3. Use amplify region and pin assignment.
4. Use "maxdelay" for delay control of nets used for channel bonding.
Create region next to MGT.
Assign pin locations close to MGT.

Figure 3 - Amplify RTL floorplanning with MGTs (multi-gigabit transceivers) in Virtex-II Pro™ FPGAs

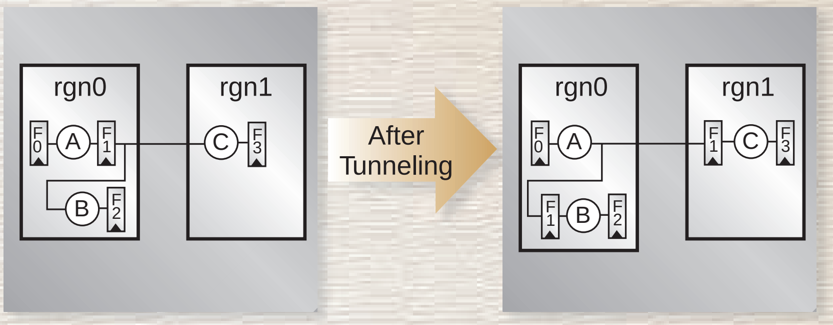

Figure 4 - Tunneling

Amplify allows you to perform clock domain floorplanning easily. You can select all the registers driven by a high-frequency clock net and assign them to a region that follows clock tree boundaries. Xilinx P&R software disconnects unused clock tree branches. Reduction in number of switching clock buffers and clock net capacitance reduces power consumption. The Amplify interface allows selection and assignment of RAM resources to isolate high-access memory, BlockMULT to isolate high-activity logic, and I/O pins to minimize external loading.

9. Tunneling

After you create regions and assign logic to those regions, the Amplify tool performs some intelligent optimizations to make sure there is no excessive region-to-region routing. Whenever a register drives logic into another region, an unnecessary routing penalty is incurred. The Amplify tool automatically replicates and moves a copy of the register to the region where the logic is being driven (Figure 4), keeping the net capacitance low and minimizing power consumption.

10. Replication

The Amplify tool performs various physical optimizations to control net capacitances. For nets with large fanouts, the Amplify software automatically replicates the driving cells to reduce the fanout and power consumption. Reduction in net capacitance can help control power consumption. The Amplify tool also provides the ability to perform manual replication.

Conclusion

Amplify Physical Optimizer software from Synplicity is becoming a must-have tool for FPGA design. The Amplify tool helps you resolve interconnect related issues early in the design cycle and enables you to manage power consumption without sacrificing timing performance in advanced FPGA designs.

Synplicity offers various channels to help and support you in using Amplify software. The Amplify software installation includes extensive help documentation and tutorials. Amplify training is also available through an online, self-paced course – and through a one-day laboratory session that will give you a detailed, hands-on understanding of the full potential of Amplify software. To request Amplify training, send an e-mail message to *training@synplicity.com.*

To get the link for downloading the latest Amplify installation, contact your local Synplicity sales office. You must also send an e-mail request to *license@synplicity.com* to obtain an evaluation license.

For help with any questions about Amplify product usage, please contact Synplicity at 408-215-6000 or send an e-mail to *support@synplicity.com.*