WP370 (v1.3) March 1, 2011

# Reducing Switching Power with Intelligent Clock Gating

By: Frederic Rivoallon

Xilinx delivers the first automated, fine-grain clock-gating solution that can reduce dynamic power by up to 30% in Virtex®-6, Spartan®-6, Kintex™-7, and Virtex-7 FPGA designs.

Xilinx intelligent clock-gating optimizations are automatically performed on the entire design, introduce no new tools or steps to the flow, and generate no changes to the existing logic or to the clocks that alter the behavior of the design. And, in most cases, the timing is also preserved.

# Intelligent Clock Gating Overview

Clock gating is a well understood power optimization technique employed in both ASIC and FPGA designs to eliminate unnecessary switching activity. This method usually requires the designers to add a small amount of logic to their RTL code to disable or deselect unnecessarily active sequential elements—registers, for example. Despite the obvious value of reduced dynamic power afforded by this method, the designer faces significant challenges when attempting to perform these optimizations manually:

- Truly reducing activity in the design requires intimate knowledge of the design itself and typically requires numerous changes to the RTL.
- Most ASIC and FPGA designs today comprise some combination of new, legacy, and third-party IP circuit designs, but typically only the new designs are candidates for clock-gating optimizations. Designers rarely if ever attempt these optimizations on legacy and IP design. They usually do not have sufficient depth of knowledge about the design and operation of the legacy RTL code, and it requires too much time to manually develop meaningful clock-gating optimizations.
- Applying clock-gating optimizations usually requires the addition of more tools and more steps to the design flow and can precipitate the creation of an intricate set of new clocks requiring complex timing analyses (as is often the case for ASIC optimization). Unless the gains in power efficiency are sufficient and essential to the success of the design, the additional complexity and time can be prohibitive and add risk.

Xilinx has an automated capability linked to the place and route portion of the standard FPGA design flow that uses a set of innovative algorithms to perform an analysis on all portions of the design (including legacy and third-party IP blocks). Having analyzed the logic equations to detect sourcing registers that do not contribute to the result for each clock cycle, the software utilizes the abundant supply of clock enables (CEs) available in the logic to create fine-grain clock-gating or logic-gating signals that neutralize superfluous switching activity, as shown in Figure 1.

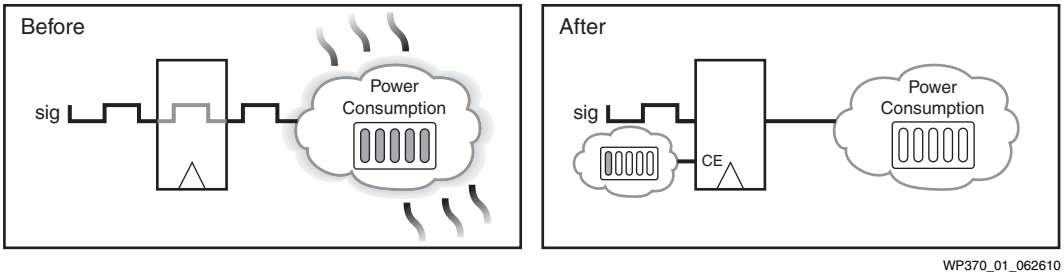


*Figure 1:* **Intelligent Clock Gating Dramatically Reduces Switching Power Consumption**

Each CE is ideally suited for power optimization because it connects to the basic cluster of the FPGA logic (the slice). The CE controls a small number of registers (only eight), providing the level of granularity that matches the minimum size of buses used by the vast majority of designs (see Figure 2).

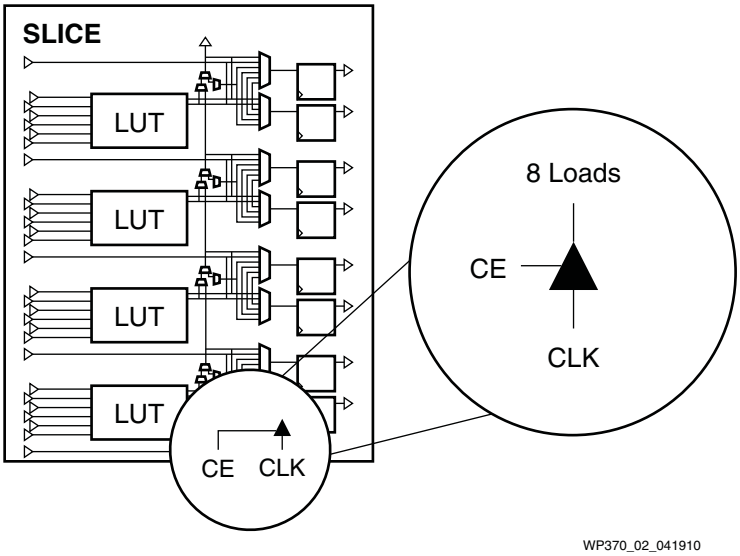


*Figure 2:* **Clock Enables in the Slice**

It is important to note that these optimizations do *not* alter the pre-existing logic or clock placement, nor do they create new clocks. The resulting design is logically equivalent to the original and the additional logic created is separate from previous logic, adding only 2% more LUTs (on average) to the original design. As a result, the optimization does not affect timing in the vast majority of cases because it does not add levels of logic to the original logic paths. These optimizations were first introduced in ISE software in version 12.1, supporting Virtex-6 FPGAs. Support was added for Spartan-6 FPGAs in ISE 12.3, and ISE 13.1 adds support for both Kintex-7 and Virtex-7 FPGAs.

## Additional Optimizations

Intelligent clock gating optimization also reduces power for dedicated block RAM in either simple or dual-port mode. These blocks provide several enables: an array enable, a write enable, and an output register clock enable. Most of the power savings comes from using the array enable, as shown in Figure 3.

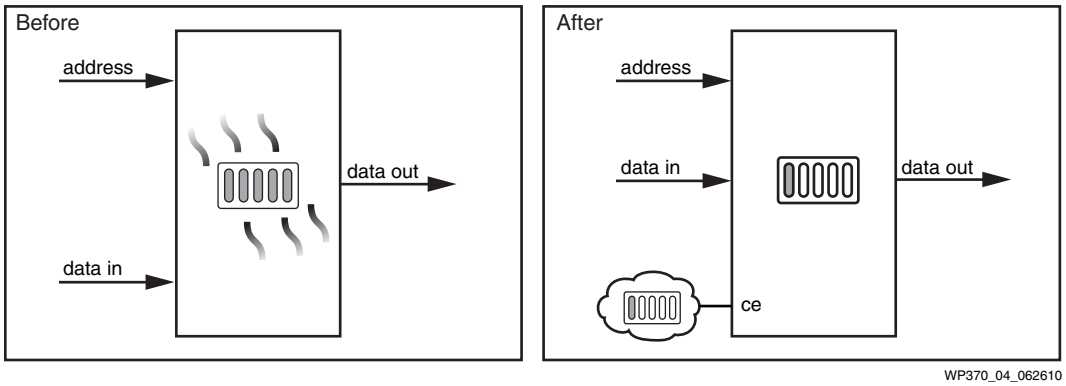


*Figure 3:* **Intelligent Clock Gating Optimizations Can Leverage Block RAM Enables**

For example, in a block RAM followed by a 2-to-1 multiplexer, the optimization implements an OR function in a LUT with the write enable (weR) and the select (preselectR) and connects them to the ENARDEN of the block RAM. The OR function ensures that the block dissipates less power when no data is being written and when its output is not used (i.e., not selected in the multiplexer). Assuming a 50% toggle rate on the write enable of the block RAM, this optimization shows a 26% reduction in dynamic power. Here is an example of the Verilog code:

```
module ram_mux ( input        select, clk, we,
                 input [7:0]  bypass,
                 input [10:0] addr,
                 input [7:0]  data_in,
                 output reg [7:0] result_out );

   reg          preselectR, selectR, weR;
   reg [7:0]    data_out, mem [2047:0];

   always @(posedge clk)
     begin
        // RAM block 2048x8 (inferred)
        if (weR) mem[addr] = data_in;
        data_out <= mem[addr];
        // Registering inputs
        weR <= we;
        preselectR <= select;
        selectR <= preselectR;
        // Mux: RAM output and input data
        result_out <= selectR ? data_out : bypass;
     end
endmodule // ram_mux
```

In that same version of software, these optimizations can detect a clock enable implemented as logic and replace it with a dedicated CE, as shown in Figure 4.

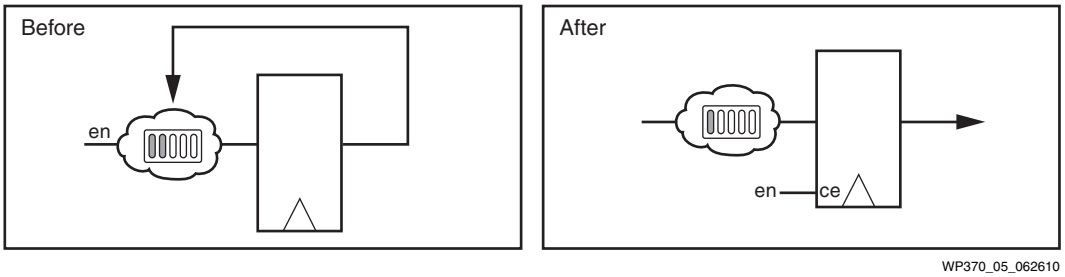


*Figure 4:* **Rewiring a Clock Enable as a Native CE to Reduce Power**

# Intelligent Clock Gating in the Design Implementation Flow

ISE Design Suite (v12.x and later) integrates the intelligent clock-gating optimization software in the design implementation flow that the user runs after synthesizing a design. Design implementation comprises four main steps:

1. Translate merges the incoming netlists and constraints into a Xilinx design file.
2. Pack and place design elements into the FPGA slices accounting for the physical and timing constraints.
3. Route the design according to the timing constraints.
4. Generate a programming file to create a bitstream file that can be downloaded to the device.

During translation, all IP, netlists, and blocks from various design tools (which might or might not be processed RTL) are funneled into a design database builder to create the Xilinx *native generic database* (NGD) file (ngdbuild). With the **`-power`** option selected (see Figure 5), the MAP process performs an analysis of the logic and creates and applies the optimizations before placement.

As a result, power optimizations are automatically created for the entire design, removing unnecessary switching activity from the designer's new code, as well as from any third-party IP and/or any legacy design blocks. The MAP process then runs a design rule check on the NGD file and maps the logic design onto the Xilinx FPGA. The results are output to a *native circuit description* (NCD) file, which is used for placing and routing.

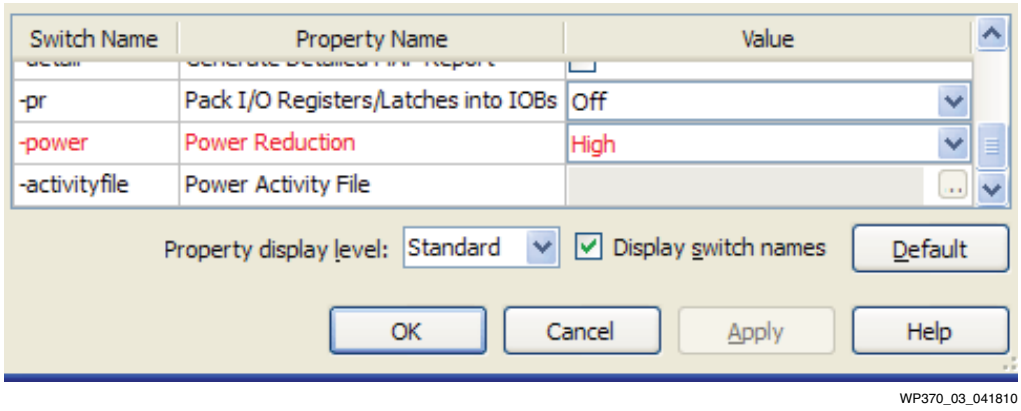


*Figure 5:* **Turning On the Intelligent Clock-Gating Optimizations in ISE Software**

# Benchmark Results

The benefits of intelligent clock gating vary depending on the design; several designs will not benefit while others will. Based on a suite of customer designs, many designs have shown a significant reduction in dynamic power switching of up to 30%. See Figure 6.

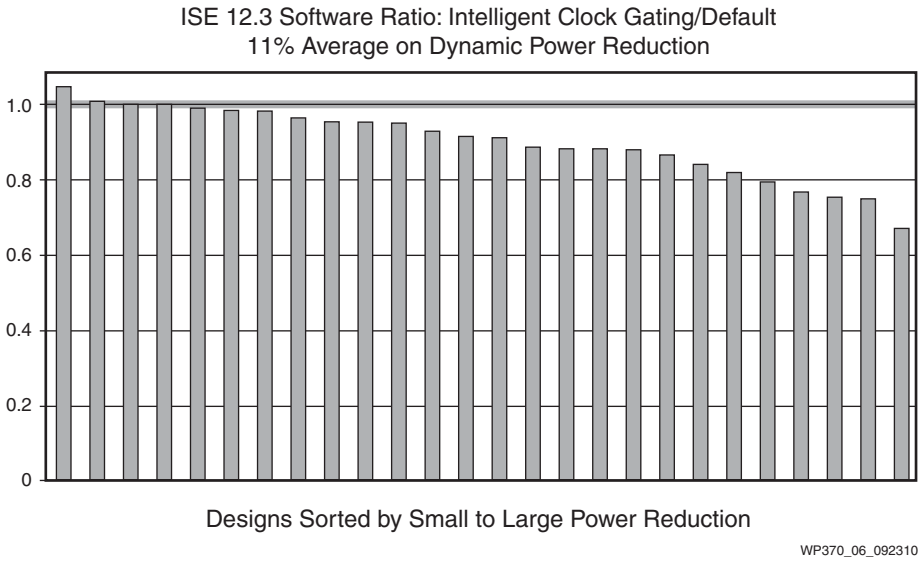


*Figure 6:* **Dynamic Power Reduction**

In addition to the customer suite of designs, intelligent clock gating was also used on a test design based on the Virtex-6 FPGA Connectivity Kit. This particular design, which includes a power hungry module of encryption/decryption, saw a 41% reduction in dynamic power. The design was implemented in the kit's Virtex-6 XC6VLX240T FPGA and included a PCIe® block, a DMA unit, a XAUI module, and the encryption/decryption block. The optimizations reduced dissipated dynamic power on $V_{CCINT}$ from 6.41 watts to 3.75 watts while maintaining the system bandwidth on all links (including PCIe and XAUI) without any loss in throughput.

# Summary

The intelligent clock-gating optimization feature provided in the ISE Design Suite greatly simplifies the effort to reduce dynamic power in FPGA designs. The traditional approach to clock-gating optimization used in ASIC design presupposes an intimate knowledge of the design, thereby virtually precluding optimization of legacy and third-party IP blocks. New tools, new steps, and complex timing analyses are typically required to compensate for the inevitable new "gated clocks" and the changes in logic that are produced.

In contrast, Xilinx intelligent clock-gating optimizations are automatically performed on the entire design, introduce no new tools or steps to the flow (compared to the default flow), and generate no changes to the existing logic or clocks that would alter the behavior or timing of the original design version.

# Revision History

The following table shows the revision history for this document:

| Date | Version | Description of Revisions |
|---|---|---|
| 05/03/10 | 1.0 | Initial Xilinx release. |
| 07/23/10 | 1.1 | Added Additional Optimizations, Figure 3, and Figure 4. |
| 10/05/10 | 1.2 | Added Spartan-6 FPGA information. Updated Additional Optimizations. Added Figure 6. |
| 03/01/11 | 1.3 | Added Kintex-7 and Virtex-7 FPGA information. Updated ISE version information. Updated Intelligent Clock Gating Overview. |

# Notice of Disclaimer