



## Design and Verification Techniques for Clock Gating

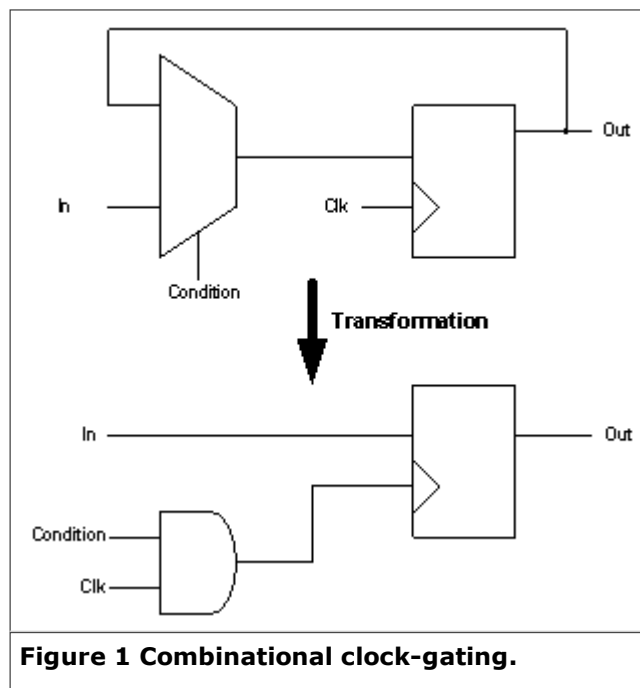
May 21, 2009 -- The demand for mobile consumer device has made power management the number one consideration in today's system design. To increase battery life, system designers adopt aggressive power management techniques which includes multi voltage islands, power gating, dynamic voltage and frequency scaling, clock-gating, etc. in the system. Advance power-management techniques greatly complicate verification. The tradeoff between power reduction and verification cost is not always clear, so designers tend to be cautious, leaving power savings on the table.

### Clock-gating in all shapes and sizes

Power has become a primary consideration during hardware design. Dynamic power can contribute up to 50% of the total power dissipation. Clock-gating is the most common RTL optimization for reducing dynamic power. Effective clock-gating implementation requires skillful application and comprehensive verification. There is a vast array of clock-gating techniques available to designers. Most clock-gating is done at the Register Transfer Level (RTL). RTL clock-gating can be grouped into three categories: system-level, sequential and combinational.

System-level clock-gating stops the clock for an entire block, effectively disabling all functionality. On the contrary, combinational and sequential selectively suspend clocking while the block continues to produce an output.

Combinational clock-gating is a straightforward substitution to the RTL code. It reduces power by disabling the clock on registers when the output is not changing. Clock-gating logic is substituted when code such as "if (condition) out <= in" is present.

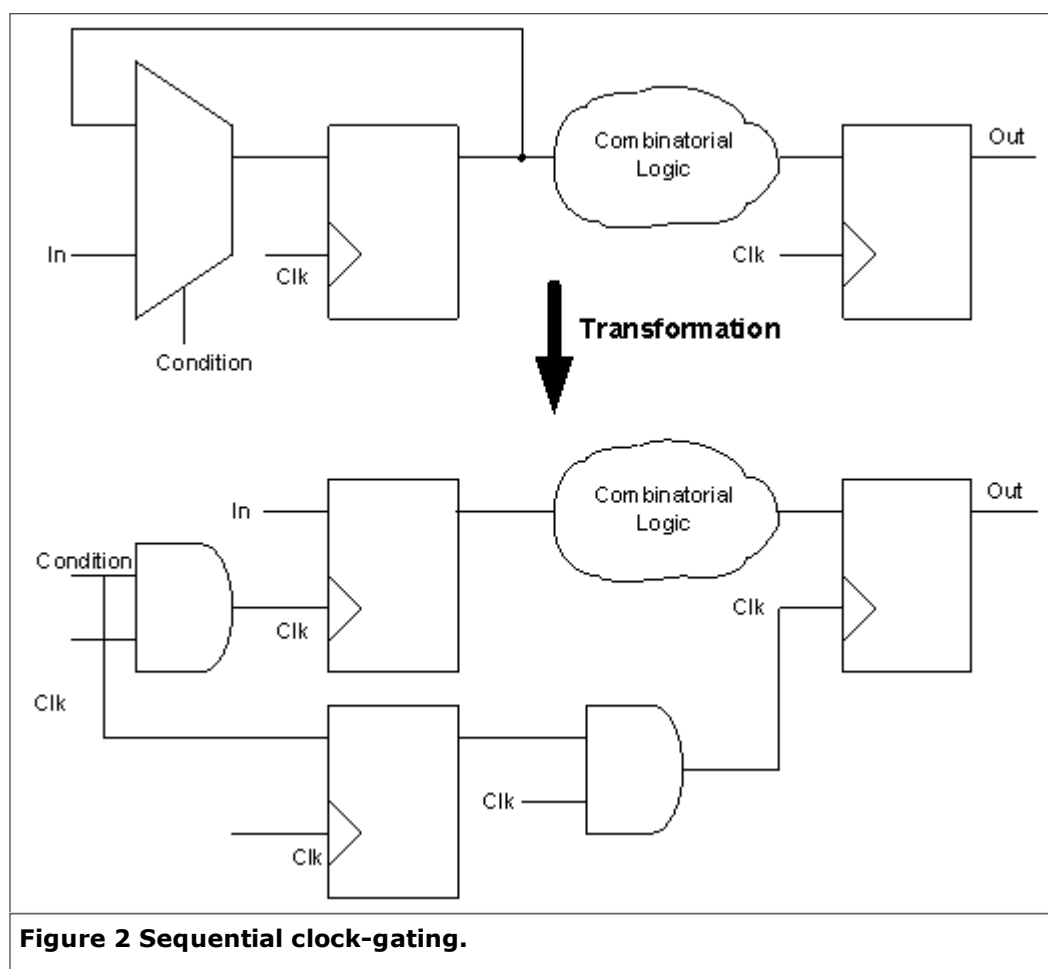


Combinational clock-gating is now a feature in the RTL compilers. Power aware synthesis tools identify RTL coding patterns and make the appropriate substitution.

Since combinational clock gated flops maintain a one to one state mapping with the original RTL, Combinational Equivalence Checking Tools can be used for functional verification. This makes verification comprehensive and simple to setup. In typical designs, combinational clock-gating can reduce dynamic power by 5 to 10%.

Sequential clock-gating alters the RTL micro-architecture without affecting design functionality. Power is optimized by identifying unused computations, data dependent functions and "don't-care" cycles in the original code. Identifying opportunities for sequential clock-gating is difficult, requiring sequential analysis. One example of a sequential optimization is turning off subsequent pipeline stages based on a propagated valid condition (see Figure 2). Because of the additional logic, this transformation makes sense only if the data path

is multiple bits wide.



**Figure 2 Sequential clock-gating.**

Sequential clock-gating is a multi-cycle optimization with multiple implementation tradeoffs and RTL modifications. Consequently there is a greater demand on functional verification resources. On the other hand sequential clock-gating can save significant power, typically reducing switching activity by 15-to-25% on a given block. Since sequential optimizations change the state of the design, Sequential Equivalence Checking (SEC) tools need to use to verify comprehensive changes to RTL like clock-gating.

### System-level and combinational clock-gating in the design flow

System-level clock-gating is designed into the original hardware architecture and coded as part of the RTL functionality. For example, sleep modes in a cell phone may strategically disable the display, keyboard or radio depending on the phones current operational mode. System-level clock-gating shuts off entire RTL blocks. Because large sections of logic are not switching for many cycles, it has the most potential to save power. On the other hand, these modifications are invasive to the design function. The enable logic is part of an overall power management strategy and sometime includes consideration for software control. Verification of system-level power optimizations must be thought through in the system-level test plan.

Most hardware engineers understand how to write RTL in such a way that synthesis tools can recognize and automate combinational clock-gating. Likewise, hardware architects recognize and build in system-level clock-gating opportunities. Even with these optimizations in place, there is substantial dynamic power-saving opportunities in the RTL if designers understand the cost/ reward tradeoffs of sequential clock-gating. Today, there are no tools to automate sequential clock-gating. However, with SEC now providing verification solution, designers can focus on finding and implementing efficient sequential optimizations.

### Sequential clock-gating in the design flow

A standard practice for design teams is to create a block-wise power budget at the beginning of a project. As blocks are implemented, designers optimize the blocks that are over-budget. Accurate power analysis for technologies at 90nm and below depends on physical place-and-route information. Unfortunately, this information is not available until late in the design flow. This means sequential clock-gating is done late in the project, further highlighting the importance of comprehensive verification.

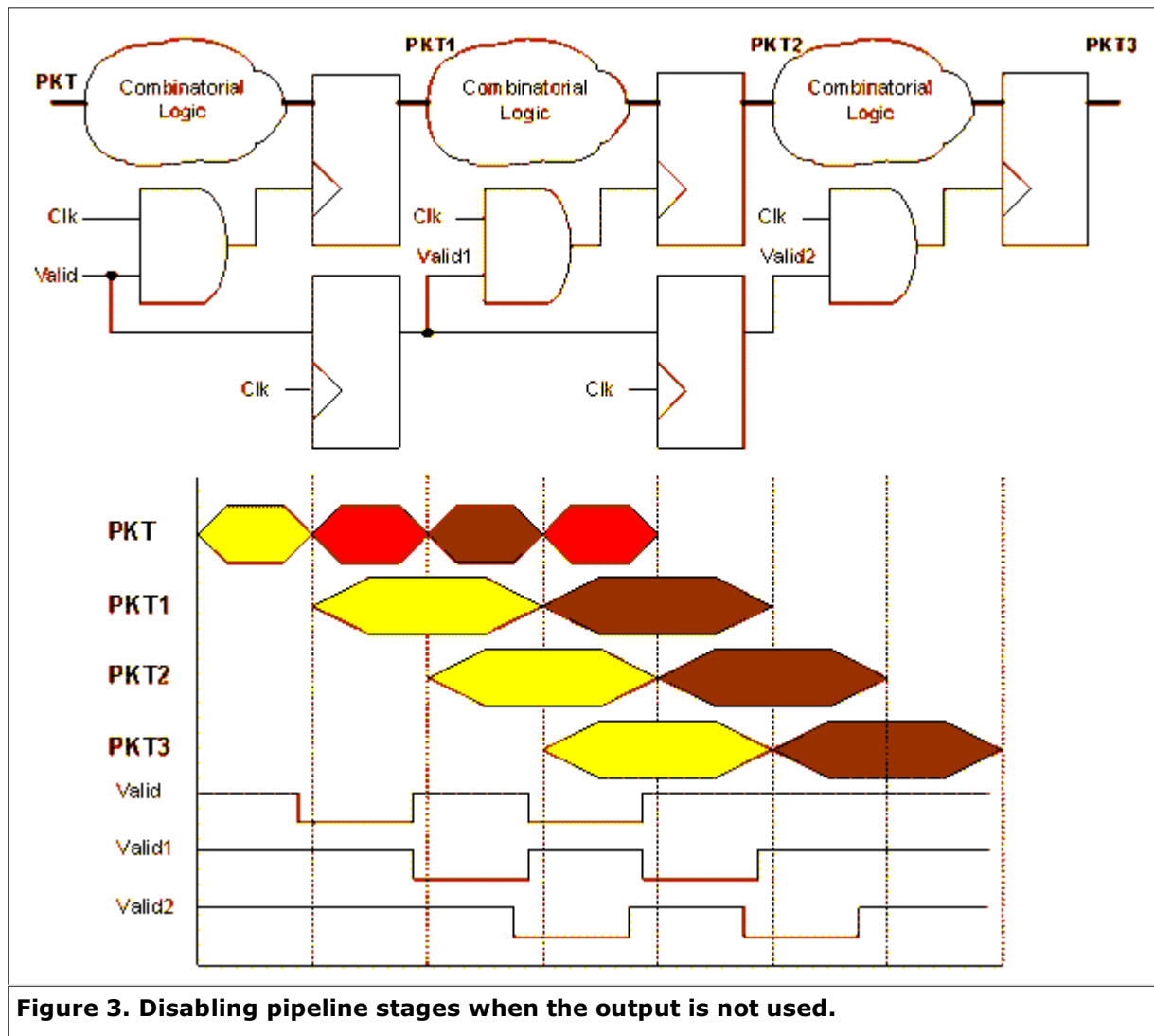
Identifying the enable condition is difficult for sequential clock-gating. The enable logic can become very complex. Multi-cycle analysis of the design is needed, making it nearly impossible to ensure correctness by construction. To adequately verify sequential clock-gating with simulation, testbenches must be monitored and modified to cover all enable/disable conditions. Further complicating the verification are cross hierarchies and interaction with neighboring blocks that run on different frequencies. Because of the cost of verification, only

mandatory sequential optimizations were allowed. This scenario has changed with the availability of SEC tools.

SEC functionally verifies sequential optimizations by comparing the clock-gated RTL to the corresponding original design. SEC uses formal sequential analysis to verify all possible input sequences that enable and disable clocks without testbenches or assertions. This has the advantage of saving the time of modifying testbenches and running regressions. In addition, SEC efficiently verifies clock-gating schemes that cross hierarchies and block boundaries without having to conceive specific testbench sequences.

### Sequential clock-gating example

A sequential clock gating opportunity can be found at instruction pipeline block. The first challenge is to identifying the "don't care" status in the pipeline. Using this information and analyzing backwards across three to four pipeline stages, an opportunity for reducing power can be identified. In this specific optimization, previous pipeline stages are disabled in earlier cycles when it is determined that the output is not used in the current cycle. Figure 3 shows a simplified diagram of this logic optimization.



**Figure 3. Disabling pipeline stages when the output is not used.**

The difficulty in implementing this optimization is keeping track of the signals that crossed pipeline stages and contribute to the enable condition. There are no automated tools that can provide this information so this analysis needs to be done manually by reading the RTL source code and waveforms. An SEC tool can be used for functional/ formal verification. An SEC tool can identify designs differences as short (4 cycles or less) counter examples which can help to debug quickly and locate errors.

### Conclusion

RTL clock-gating is a common technique for reducing dynamic power. Today, there are no automated tools to identify or make sequential RTL clock-gating optimizations. Such optimizations require experienced engineers who know when and how to apply the appropriate sequential change. Since this is a manually transformation, verification is critical. SEC tools can verify clock-gating, giving designers the confidence to make aggressive power optimizations late in the design process. The result is a lower-power, higher-quality design.

**Reprinted from SOCcentral.com, your first stop for ASIC, FPGA, EDA, and IP news and design information.  
Copyright 2002 - 2009 Tech Pro Communications, P.O. Box 148, Jamison, PA 18929**