

PowerAdviser: An RTL Power Platform for Interactive Sequential Optimizations

Nainala Vyagrheswarudu and Subrangshu Das

Texas Instruments Inc.
Bangalore, India
vyagrhee@ti.com, s-das2@ti.com

Abhishek Ranjan

Calypto Design Systems Inc.
Noida, India
aranjan@calypto.com

Abstract — Power has become the overriding concern for most modern electronic applications today. To reduce clock power, sequential clock gating is increasingly getting used over and above combinational clock gating. Given the complexity of manually identifying sequential clock gating changes, automatic tools are becoming popular. However, since these tools always work within the scope of the design and the constraints provided, they do not provide any insight into additional power savings that might still be possible. In this paper we present an interactive sequential analysis flow, *PowerAdviser*, which besides performing automatic sequential changes also provides information for additional power savings that the user can realize through manual changes. Using this new flow we have achieved dynamic power reduction upto 45% more than a purely automated flow.

Keywords – Sequential Clock Gating, Sequential Analysis, Sequential Optimization, Observability, Stability, PowerAdviser, Power Analysis, Power Optimization.

I. INTRODUCTION

Reducing power consumption in a semiconductor device is becoming one of the most careabouts. It is suggested, that in future, power will be the limiting factor when determining the maximum number of applications that can simultaneously be active [1] and not just the amount of functionality that can be packed in a single die as governed by Moore's law [2].

Clock and register power are considered to be the biggest power consuming components today. To reduce clock power, clock gating is used to gate the clocks when writing into the register is redundant [3]. Sequential clock gating, where design behavior is analyzed across multiple cycles to identify redundant writes into a register, has emerged as a very powerful technique to identify new clock gating conditions in the design [4]. To reduce manual effort, there are solutions [6], which can automatically identify and modify the RTL to insert new sequential clock gating conditions. While automatic sequential clock gating has significant advantages over manually inserted clock gating, it does not provide any information to the designer about additional clock gating that may be possible in the design but was not done by the tool because it could not find a suitable clock gating condition to gate the redundant clock toggles.

In this paper, we introduce an interactive sequential analysis flow called *PowerAdviser*. The flow provides information to the user about redundant clock toggles where

the automatic tool has not been able to identify a suitable clock gating condition to save power. Information provided by PowerAdviser can be used to modify the RTL or the constraints provided to the tool to make the design amenable for automated sequential clock gating optimization. We also report the power savings obtained after using the PowerAdviser flow versus just using the automated flow. The flow has been applied to several designs in a video processing engine, where it has provided up to 45% additional dynamic power savings over conventional automatic sequential clock gating.

II. PRIOR WORK

Clock Gating is one of the most frequently used techniques in RTL to reduce dynamic power consumption [3]. It involves inserting clock gating conditions in the RTL to reduce switching activity on the clock network thereby reducing dynamic power consumption. Since the translation is purely combinational, it is also referred to as *Combinational Clock Gating*. On the other hand, *Sequential Clock Gating* uses multi-cycle analysis of the design to identify writes that are either unobservable down-stream or the same value is written in consecutive cycles. Gating of first type of redundant writes is called *Observability Based Clock Gating* and the second type of redundant writes is called *Stability Based Clock Gating* [5].

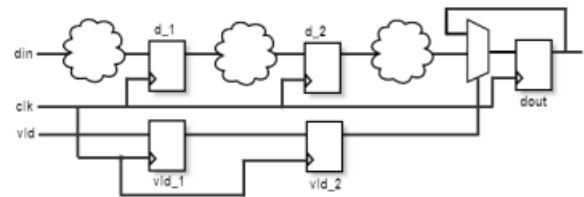


Figure 1: Unobservable Writes

Figure 1 shows an example of writes to the registers d_1 and d_2 which, under some conditions, are never going to be observed at the design output. Similarly, in Figure 2 we have stable writes to registers f_2 and d_{out} across consecutive clock cycles under certain conditions.

In recent years, tools like [6] provide an automated way to implement sequential clock gating logic in the design. These tools are mostly deployed once the RTL has been fully verified and is ready to be taken through RTL synthesis. Figure 3 shows a design flow which makes use of automatic sequential clock gating tool as a final step in the RTL design flow.

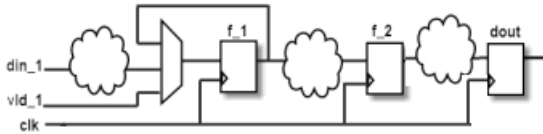


Figure 2: Stable Writes

While **automatic sequential clock gating** can save significant power [5], these tools have their limitations:

- Optimization by automatic tools is bound by the scope of the design and constraints provided
- No feedback is provided on whether an RTL construct or user-provided constraint is inhibiting the creation of a clock gating expression
- Tools often make extensive changes to the RTL and hence cannot be used in the early stages of RTL development
- Designers are often wary of making major changes in the design due to sequential clock gating
- Tools do not provide any early indication of how much

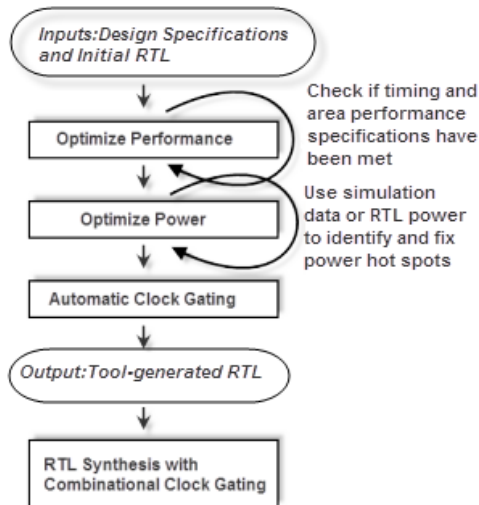


Figure 3: Traditional RTL Design Flow

power saving is actually possible

Clearly, there is a need for an interactive sequential clock gating flow that not only overcomes the pitfalls of the fully automated sequential clock gating tools but also can work concurrently with the RTL refinement phase very early in the design process. To provide maximum benefit, the flow should:

1. Provide **very early** feedback (on the initial RTL) about the total power saving possible in the design
2. Provide **complete** clock gating expressions wherever possible and also provide potential clock gating opportunities with reasons as to why they are not being automatically found (**incomplete**)
3. Provide fast and accurate estimates of power savings and area cost for both complete as well as incomplete clock gating conditions

The following section describes an interactive sequential analysis based clock gating flow called *PowerAdviser*, which overcomes the disadvantages of automated sequential clock gating to provide additional power savings.

III. POWER ADVISER

There are several reasons behind why an automatic tool may not be able to identify a complete clock gating expression to gate redundant writes (including the power cost of adding the new enable logic may be more than the savings). Most of these can be fixed by making simple changes in the RTL and/or modifying the constraints provided to the tool.

A. Interactive Sequential Analysis Flow

PowerAdviser has been built for use in conjunction with the RTL design phase. In the early stages of the RTL design, it predicts the power saving possible in the design by creating **complete** and **incomplete** sequential clock gating expressions. Figure 4 shows the *PowerAdviser* flow.

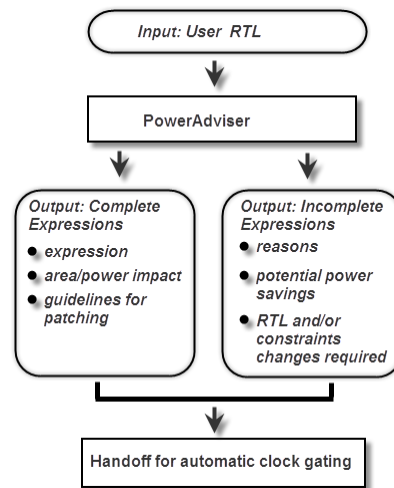


Figure 4: Power Adviser Flow

The most significant aspect of the PowerAdviser flow is its ability to present **incomplete** clock gating expressions, their power saving potential and the exact reasons as to why these expressions are incomplete. Estimating power savings of incomplete expressions has its own challenges:

- Some of the signals used to create incomplete expressions might not even be present in the RTL
- Evaluating the impact of using an incomplete expression in the context of actual design without altering the design state is difficult
- Traditional techniques [8][9] are not sufficient to estimate power savings of sequential changes

PowerAdviser solves these issues by creating temporary hierarchies which contain clock gating expressions and the portions of the design impacted by it. Power saving is evaluated by estimating the power (using sequential techniques [10]) of such a hierarchy with and without the

clock gating expression. PowerAdviser evaluates all expressions in one go, removing the need to invoke lengthy simulations or external power analysis tools [7], thereby providing significant runtime gains.

B. Types of Incomplete Expressions

The reasons that make an expression **incomplete** as well as the actions required from the user to make them **complete** are:

Case 1: Previous cycle value of a signal is not available

To create observability based clock gating expression, the one cycle early value of a signal is required (Figure 1). In case such a signal is not available, PowerAdviser creates a clock gating expression assuming that it is available but marks the expression incomplete. Figure 5 shows a design with two sub-blocks ‘Data path’ and ‘Control path’. Here, when the designer is working on the sub-block ‘Datapath’, a complete clock gating expression does not exist for register *d_2*. In this situation, PowerAdviser creates an expression assuming that the one cycle prior value of signal *vld_2* is available.

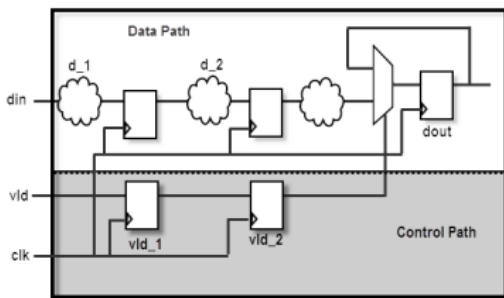


Figure 5: Incomplete Clock Gating Due to Design Scope

Based on this feedback, the designer can take remedial measures. The designer can either bring register *vld_2* within the scope of the sub-block that is being optimized or choose to work at a higher scope which includes both the sub-blocks.

Case 2: The size of the expression is very big

Clock gating expressions need to be expressed in terms of signals present in the RTL. In Figure 1, to gate register *d_2*, access to the signal driving the input of register *vld_2* is required. In this case, the signal *vld_1* is already available in the RTL. However, it is possible that there is no RTL signal corresponding to the signal driving the input of register *vld_2*:

```
always@(posedge clock)
begin
case (select):
4'b0000 : vld_2 <= in1;
4'b0001 : vld_2 <= in2;
....
end
```

One option for the tool is to duplicate the logic driving the input of register *vld_2* but that is clearly sub-optimal from a power savings stand-point. Instead, PowerAdviser creates an incomplete clock gating expression. To ensure that an input RTL signal to *vld_2* becomes available for the clock gating expression to be created with zero power overhead, the user

can rewrite the code as follows.

```
always@(in1 or in2 or ...)
begin
case (select):
4'b0000 : vld_2_in <= in1;
4'b0001 : vld_2_in <= in2;
...
end
always@(posedge clock)
vld_2 <= vld_2_in;
```

Case 3: Signal is in a different clock domain from the register being gated

To ensure the functional correctness of a clock gating expression, the register being gated and the signals forming its gating expression must be clocked by the same clock signal. However, often clocks that appear to be different in the context of a sub-block could actually be coming from the same root clock at a higher level or they could be gated versions of the same clock, as shown by Figure 6 (notice that *d_2* and *vld_2* are clocked by gated versions of the same root clock).

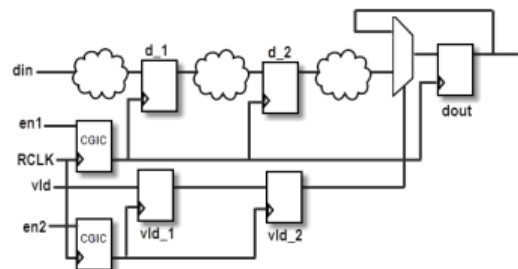


Figure 6: Difference in clocks leading to incomplete expressions

PowerAdviser creates a clock gating expression assuming that signals are in the same clock domain but marks those expressions incomplete. Using feedback from PowerAdviser, the designer can modify clock constraints so that the automatic tool or a subsequent run of PowerAdviser is aware of the relationship between the gated clocks — making the clock gating expressions realizable.

Case 4: Design is over-constrained

An automatic Sequential Clock Gating tool requires i) setting up constraints, ii) blackboxing incomplete modules, iii), excluding timing critical signals from the clock gating expressions, and iv) possibly, excluding registers or design portions from clock gating.

Most often, the designer is oblivious of the impact these constraints have on power when specifying them. An automatic tool uses these settings to find clock gating expressions but does not reveal expressions and the associated potential power savings it had to drop because of these constraints. PowerAdviser presents such expressions along with the constraints that make them incomplete. The designer can choose to relax constraints or complete portions (which were marked as blackboxes) to realize additional power savings.

IV. DESIGN FLOW WITH POWER ADVISER

Figure 7 shows the modified design flow where PowerAdviser works concurrently with RTL refinement.

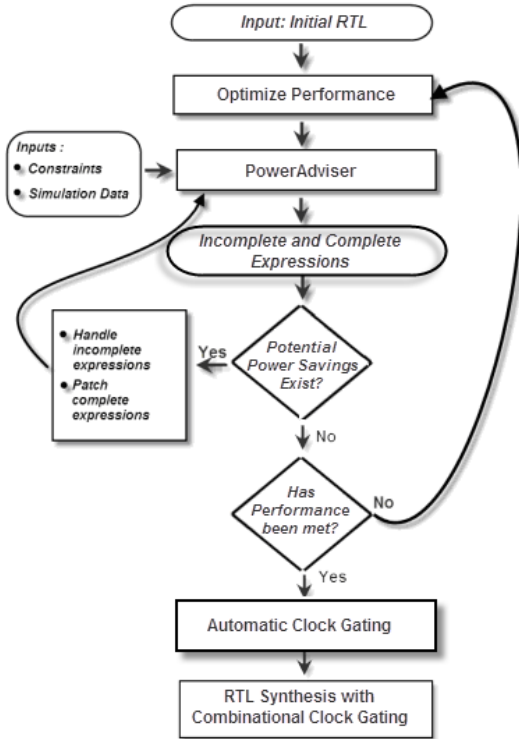


Figure 7: Power Aware Design Flow

Here, the designer finds out, early in the design cycle, the scope of power optimization. If the scope is very less or most of the clock gating expressions are **complete**, the designer can focus on improving the performance targets of the design. However, if there are a significant number of incomplete clock gating expressions, the designer might choose to modify the RTL, making it amenable for automatic clock gating and/or spend more effort on the incomplete expressions that provide maximum power savings. Once the RTL has been modified the designer can iterate until performance targets are met.

V. RESULTS

The PowerAdviser flow has been applied to four different modules of a video processing engine. Details are provided in Table 1. Design1 and Design2 are datapath dominated, Design3 has, more or less, a balance of datapath and control logic while Design4 has a large number of registers. These designs were coded to meet functionality and performance with little planning for power. The designs were taken through automated Sequential Clock Gating. Table 1 shows the power savings obtained for each module. Power was measured by taking the modified designs through RTL synthesis in Design Compiler followed by power analysis in PrimeTime-PX.

Design	Size (KG)	Sequential power saving
Design1	51	7%
Design2	114	10%

Design3	184	18%
Design4	137	12%

Table 1 : Power saving in traditional flow

Using PowerAdviser feedback, the RTL was modified to: (i) bring registers from the outer scope to the design scope, (ii) replace large unviable clock gating expressions using simpler logic derived from existing control signals of state-machines.

After iteratively modifying the RTL, the designs were put through the automatic Sequential Clock Gating flow. Table 2 shows the final sequential power savings obtained using the PowerAdviser flow. The last column in Table 2 shows extra power savings obtained by the PowerAdviser flow.

Design	Sequential power Saving in Power Adviser flow	Additional power saving compared to traditional flow
Design1	52%	45%
Design2	36%	26%
Design3	38%	20%
Design4	26%	14%

Table 2 : Power saving through PowerAdviser flow

VI. CONCLUSION AND FUTURE WORK

In this paper, we presented an interactive sequential analysis and flow (PowerAdviser) that provides information about redundant writes which are not automatically realizable and their potential power savings. This data helps users prioritize manual effort in identifying clock gating conditions for which the power savings are maximum. For real applications, significant power savings were obtained with no impact to the rest of the design flow including schedule. The PowerAdviser flow currently does not assess the impact of its suggestions on the performance of the design and this enhancement will be taken up as future work.

REFERENCES

- [1] J. Markoff, "Progress Hits Snag: Tiny Chips Use Outside Power," http://www.nytimes.com/2011/08/01/science/01chips.html?_r=2.
- [2] Gordon Moore, "Cramming More Components onto Integrated Circuits," *Electronics*, Volume 38, Number 8, April 19, 1965.
- [3] L. Benini and G. De Micheli, "Automatic synthesis of low power gated clock finite state machine," *IEEE Trans. Computer-Aided Design*, vol. 15, pp. 630--643, June 1996.
- [4] J. Sukumar, et al, "Clock gating for power optimization in ASIC design cycle theory & practice," *ISLPED*, pp. 307--308, 2008.
- [5] J. Sukumar, S. Das, A. Ranjan et al, "RTL Power Optimization in Sequential Analysis Platforms," poster presentation in DAC'2010.
- [6] PowerPro CG, Calypto Design Systems Inc. (<http://www.calypto.com/>).
- [7] PowerTheater, Sequence Design Inc. (<http://www.sequencedesign.com>).
- [8] F. Najm, "Low-pass filter for computing the transition density in digital circuits," *IEEE Transactions on Computer-Aided Design*, vol. 13, no. 9, pp. 1123--1131, September 1994.
- [9] Michael G. Xakellis, Farid N. Najm, "Statistical estimation of the switching activity in digital circuits," *Proceedings of the 31st annual Conference on Design Automation*, pp.728--733, June 06-10, 1994
- [10] S. Gupta and F. N. Najm, "Analytical models for RTL power estimation of combinational and sequential circuits," *IEEE Trans. Computer-Aided Design*, vol. 19, pp. 808--814, July 2000.