Design Flow for Flip-Flop Grouping in Data-Driven Clock Gating

SK. MAHABOOB BASHA¹, N. VENKATA SATISH²

¹Research Scholar, Dept of ECE, Aditya College of Engineering & Technology, Surampalem, Andhra Pradesh, India,
Email: basha.aec@gmail.com.

²Assistant Professor, Dept of ECE, Aditya College of Engineering & Technology, Surampalem, Andhra Pradesh, India,
Email: satish.nitw13@gmail.com.

Abstract: Data driven clock gating is a popular technique used in many synchronous circuits for reducing dynamic power dissipation. Power optimization plays the important role in the recent years. Clock gating is a popular technique used in many synchronous circuits for reducing dynamic power dissipation. Gating manually inserted into the register transfer level (RTL) design. When a logic unit is clocked, its underlying sequential elements receive the clock signal regardless of whether or not they will toggle in the next cycle. In this flip-flops are grouped so that they share a common clock enabling signal to reduce the hardware overhead. It is observed that the commonly used synthesis based gating still leaves a large amount of redundant clock pulses. In these d-flip fops are used to grouping for reducing the power. Here the Xilinxx software tool will be used for implementing this proposal system.

Keywords: Clock Gating, Clock Networks, Dynamic Power Reduction.

I. INTRODUCTION

One of the major dynamic power consumers in computing and consumer electronics products is the system’s clock signal, where it takes 30%–70% of the total dynamic power consumption. There are many techniques used to reduce the dynamic power are developed, in which clock gating is predominant. Ordinarily, when a logic unit is clocked, it is based on the sequential elements receiving the clock signal, sequentially they will toggle in the next cycle whether it is required or not. With clock gating, the clock signals are ANDed with explicitly predefined enabling signals. Clock gating is employed at all levels: system architecture, block design, logic design, and gates. Several methods to take advantage of this technique are described, with all of them depending on various heuristics in an attempt to increase clock gating opportunities. With the rapid increase in design complexity, computer aided design tools supporting system-level hardware description have become commonly used. Although substantially increasing design productivity, such tools require the employment of a long chain of automatic synthesis algorithms, from register transfer level (RTL) down to gate level and net list. Unfortunately, such automation leads to a large number of unnecessary clock toggling, thus increasing the number of wasted clock pulses at flip-flops (FFs).

In a recent paper, a model for data-driven gating is developed based on the toggling activity of the constituent FFs. The optimal fan out of a clock grater yielding maximal power savings is derived based on the average toggling statistics of the individual FFs, process technology, and cell library in use. In general, the state transitions of FFs in digital systems depend on the data they process. Assessing the effectiveness of data-driven clock gating requires, therefore, extensive simulations and statistical analysis of the FFs’ activity. Another grouping of FFs for clock switching power reduction, called multibit FF (MBFF). MBFF attempts to physically merge FFs into a single cell such that the inverter driving the clock pulse into its master and slave latches are shared among all FFs in a group. MBFF grouping is mainly driven by the physical position proximity of individual FFs, while grouping for data driven clock gating should combine toggling similarity with physical position considerations. While answered the question of what is the group size that maximizes power savings, this paper studies the questions of: 1) which FFs should be placed in a group to maximize the power reduction and 2) how to algorithmically derive those groups.

II. DATA-DRIVEN CLOCK GATING

Clock enabling signals are very well understood at the system level and thus can effectively be defined and capture the periods where functional blocks and modules do not need to be clocked. Those are later being automatically synthesized into clock enabling signals at the gate level. In many cases, clock enabling signals are manually added for every FF as a part of a design methodology. Still, when modules at a high and gate level are clocked, the state transitions of their underlying FFs depend on the data being processed. It is important to note that the entire dynamic power consumed by a system stems from the periods where modules’ clock signals are enabled. Fign shows the FFs’
toggling activity in an arithmetic block comprising 22K FFs, designed in 40-nm technology, taken from Ceva’s X1643 DSP core for multimedia and wireless baseband applications. The statistics is obtained from extensive simulations of typical modes of operation, consisting of 240-K clock cycles. When the FFs clock signal is enabled is only 10%, which is still responsible for the entire dynamic power consumed by that block. The clock enabling signals are obtained by RTL synthesis and manual insertions.

Fig. 1. Toggling statistics of Ceva’s X1643 DSP core over 240-K clock cycles.

A FF finds out that its clock can be disabled in the next cycle by XORing its output with the present data input that will appear at its output in the next cycle. The outputs of k XOR gates are ORed to generate a joint gating signal for k FFs, which is the n latched to avoid glitches. The combination of a latch with AND gate is commonly used by commercial tools and is called integrated clock gate (ICG). Such data driven gating is used for a digital filter in an ultralow-power design. A single ICG is amortized over k FFs. There is a clear tradeoff between the numbers of saved (disabled) clock. Pulses and the hardware overhead. With an increase in k, the hardware overhead decreases but so does the probability of disabling, obtained by OR ing the k enable signals. Let the average toggling probability of a FF (also called activity factor) be denoted by p (0 < p < 1). The latch and gater (AND gate) overheads are amortized over k FFs. It is shown in [9] that the number k of jointly gated FFs for which the power savings are maximized is the solution of

\[ (1 - p)^k \ln(1 - p) (c_{FF} + c_{W}) + c_{latch}/k^2 = 0 \]

Where cFF is the FFs clock input capacitance, cW is the unit-size wire capacitance, and clatch is the latch capacitance including the wire capacitance of its clk input. Such a gating scheme has considerable timing implications, which are discussed. We will return to those when discussing the implementation of data-driven gating as a part of a complete design flow. For the scheme proposed in Fig. 2 to be beneficial, the clock enabling signals of the grouped FFs should preferably be highly correlated. Data-driven clock gating is shown to achieve savings of more than 10% of the total dynamic power consumed by the clock tree it took advantage of the very low dynamic range of the data in a digital filter. The gating logic is tailored to the structure of the filter, whereas the approach discussed in this paper is more general and applies to large scale and a wide range of designs.

Fig. 2. Practical data-driven clock gating.

III. OPTIMAL FFS GROUPING FOR JOINT CLOCK GATING

Knowing the optimal group size k, the next step is to partition the FFs of a system into k-size sets such that the power savings will be maximized. Such tools are focusing on skew, power, and area minimization, but they are not aware of the toggling correlations of the underlying FFs, which this paper is focusing on. The optimal value of k is obtained from under toggling independence assumption, but in reality the toggling may be correlated, so in practice one can expect higher saving than the theoretical lower bound obtained under independence assumption. A practical design methodology should preserve the integrity of system clock enabling signals. This means that the FFs of a k-size set must all belong to the same enabled clock (called hereafter pre-enabled). A bottom-up process for a coarse, block-level gating is proposed by repeating the MCPM algorithm. We have adapted this idea to FF-level gating. Starting with n individual FFs and constructing the associated n-vertex FF pair wise activity graph, an MCPM algorithm then finds the best FFs pairing. A new n/2-vertex pair wise activity graph is then defined where its vertices correspond to the matching (n/2 edges) found in the former step. The process repeats K times until groups of size k = 2K are determined. For k = 2(K = 1), MCPM indeed solves the problem of minimizing the
number of redundant clock pulses, but its repetitive application for \( k > 2 \) \((K > 1)\) may not find the minimum, as otherwise this would contradict the NP-hardness. Still, the iterative MCPM algorithm is practical and has acceptable run time.

IV. IMPLEMENTATION AND INTEGRATION IN A DESIGN FLOW

In the following, we describe the implementation of data driven clock gating as a part of a standard backend design flow. It consists of the following steps.

1. Estimating the FFs toggling probabilities involves running an extensive test bench representing typical operation modes of the system to determine the size \( k \) of a gated FF group by solving.

2. Running the placement tool in hand to get preliminary preferred locations of FFs in the layout.

3. Employing a FFs grouping tool to implement the model and algorithms using the toggling correlation data obtained in Step 1 and FF locations’ data obtained in Step 2. The outcome of this step is \( k \)-size FF sets (with manual overrides if required), where the FFs in each set will be jointly clocked by a common gater.

4. Introducing the data-driven clock gating logic into the hardware description (we use Verilog HDL). This is done automatically by a software tool, adding appropriate Verilog code to implement the logic described in Fig. 2. The FFs are connected according to the grouping obtained in Step 3. A delicate practical question is whether to introduce the gating logic into RTL or gate level description. This depends on design methodology in use and its discussion is beyond the scope of this paper. We have introduced the gating logic into the RTL description.

5. Re-running the test bench of Step 1 to verify the full identity of FFs’ outputs before and after the introduction of gating logic. Although data-driven gating, by its very definition, should not change the logic of signals, and hence FFs toggling should stay identical, a robust design flow must implement this step.

6. Ordinary backend flow completion. From this point, the backend design flow proceeds by applying ordinary place and route tools. This is followed by running clock tree synthesis. Few timing-related comments are in order. The extra gating delay introduced by the feedback loop in Fig. 2 should not exceed the delay margins of paths from the clock input clk. \( g \) of FF1 to the data input D2 of FF2. In ordinary designs, notably in automatically synthesized blocks, most of the delay margins are large enough to absorb the introduction of the gating logic. If at a later stage timing violations due to the gating are found, one can simply drop the data-driven gating from the troublesome FFs. We found very few of those in our designs, less than 5% of the FFs. Relaxation of the clock cycle can also overcome this problem, but it must be considered in a wider context of power delay trade off and product specifications, which is beyond the scope of this paper.
VI. CONCLUSION

Look-ahead clock gating has been shown to be very useful in reducing the clock switching power. The computation of the clock enabling signals one cycle ahead of time avoids the tight timing constraints existing in other gating methods. A closed-form model characterizing the power saving was presented and used in the implementation of the gating logic. The gating logic can be further optimized by matching target FFs for joint gating which may significantly reduce the hardware overheads. While this paper discussed the case of merging two target FFs for joint gating, clustering target FFs in larger groups may yield higher power savings. This is a matter of a further research.

VII. REFERENCES

Design Flow for Flip-Flop Grouping in Data-Driven Clock Gating


Author's Profile:

Sk Mahaboob Basha, is an M.Tech student at Aditya College of Engineering & Technology, Surampalem. Areas of interest are low power VLSI, Optical Networks and Network Security.

N. Venkata Satish, is an Assistant Professor at Department of ECE at Aditya College of Engineering & Technology, Surampalem and he is having more than 5 years of Academic Teaching Experience. Areas of interest are VLSI System Design, Signal Processing.