

Power Aware High Level Synthesis with Gated Clock Skew Management

T. Devimeena¹

PG Student

Vivekananda College of Engineering for women
Tiruchengode

V.Saravanan²

Associate Professor

Vivekananda College of Engineering for women
Tiruchengode

ABSTRACT

A new method of achieving the target output with a less number of clock pulses has been introduced. Clock signal is a particular type of signal that oscillates between a high and a low state and is utilized like a metronome to coordinate actions of circuits. Although the word signal has a number of other meanings, the term here is used for "transmitted energy that can carry information". In some cases, more than one clock cycle is required to perform a predictable action. As the circuits become more complex, the problem of supplying accurate and synchronized clocks to all the circuits becomes increasingly difficult. A hierarchical low power module approach is utilized for near optimal results. A clock gating architecture can be added with the clock scheduling scheme to control the unnecessary power flow between the idle sequential circuits. The overall power reduction can be calculated by implementing the clock scheduling and power gating techniques in a SRAM Memory architecture with static and dynamic power calculation.

General Terms

Static Random Access Memory, Register Transfer Logic, Data Flow Graph, Depth First Search Algorithm.

Keywords

Clock Skew Scheduling (CSS), Clock Skew Management, High Level Synthesis (HLS), Clock Gating.

1. INTRODUCTION

A clock-skew management scheme that selects a minimum set of clock phases to meet the timing constraint is developed an HLS methodology combined with clock-skew management is presented to achieve higher performance with lower power consumption. The HLS scheme effectively exploits clock skews so that slower but power-efficient modules can be used extensively while a tight timing constraint can still be achieved. In addition, the reallocation operation that forces variables to be reallocated to other registers is introduced. This modification changes the required clock skews so that more modules can be assigned to slower ones without increasing clock period. In other words, power consumption is reduced under the same circuit speed. Along with a clock gating architecture is added with the clock scheduling scheme to control the unnecessary power flow between the idle sequential circuits.

2. CLOCK SKEW SCHEDULING

In a synchronous circuit clock skew (T_{skew}) is the difference in the arrival time between two sequentially-adjacent registers. Given two sequentially-adjacent registers R_i and

R_j with clock arrival times at register clock pins as T_{Ci} and T_{Cj} respectively, then clock skew can be defined as:

$$T_{skew-i,j} = T_{Ci} - T_{Cj}$$

Clock skew scheduling often denoted as “cycle stealing”, computes a set of individual delays for the clock signals of the registers and latches of synchronous circuits to minimize the clock period. The schedule globally tunes the latching of the state holding elements such that the delays of their incoming and outgoing paths are maximally balanced. The computed intentional differences in the clock arrival times also referred to as “useful skew”, are then implemented by designing dedicated delays into the clock distribution. CSS can be more efficient if it is carried out in the HLS process [8]. It achieves the minimum clock period through register binding. The clock-skew scheduled HLS scheme reduces power consumption by assigning the smallest sleep transistor to each module so as to minimize leakage power in standby mode. A problem of applying the CSS technique is that it may create far too many clock phases. Unfortunately, this problem is hardly considered in previous methods, so it may be difficult to apply this technique to real circuits. To deal with this problem, a clock-skew management scheme that selects a minimum set of clock phases to meet the timing constraint is developed.

3. CLOCK SKEW MANAGEMENT

3.1 High Level Synthesis

The number of clock phases can be reduced by combining clock skew management with HLS as shown in Figure 1. HLS system translates a CDFG or DFG to a RTL structure. The tasks performed by an HLS system include scheduling and resource binding. Low-power design techniques can be applied in HLS. HLS, sometimes referred to as C synthesis, ESL synthesis, algorithmic synthesis, or behavioral synthesis, is an automated design process that interprets an algorithmic description of a desired behavior and creates hardware that implements that behavior. The starting point of a high-level synthesis flow is ANSI C/C++/System C code. The code is analyzed, architecturally constrained, and scheduled to create a register transfer level HDL, which is then in turn commonly synthesized to the gate level by the use of a logic synthesis tool. The goal of HLS is to let hardware designers efficiently build and verify hardware, by giving them better control over optimization of their design architecture, and through the nature of allowing the designer to describe the design at a higher level of tools while the tool does the RTL implementation [12]. Verification of the RTL is an important part of the process. Hardware design can be created at a variety of levels of abstraction.

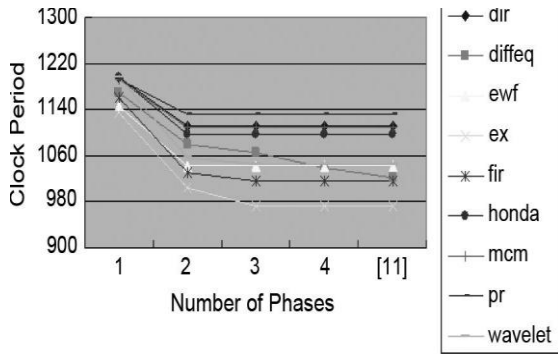


Figure 1 Clock skew management with HLS

3.2 Module binding

Module binding is nothing but register binding. Module binding can be executed by heuristic algorithm. Heuristic refers to experience-based techniques for problem solving, learning, and discovery. Where an exhaustive search is impractical, heuristic methods are used to speed up the process of finding a satisfactory solution. The main aim of using this algorithm is for time consumption [11]. This algorithm includes several modes.

3.3 Reallocation

Reallocation can be done with the register in the circuits. We can assume the values in the register as variables. One can move the variables from one register to another register. Thus the variables are shifted only when those variables are not required for computation. A variable that can be moved during its lifetime is referred to as transportable, and the process of moving a variable is called reallocation.

3.4 Low power synthesis flow

The following conditions are assumed in the synthesis process.

- 1) An SDFG is known with given RCs.
- 2) Each type of function (e.g., multiplier, adder) has several implementations.
- 3) The parameters of every implementation, including delay and power, are known.
- 4) A sufficient delay is reserved for multiplexers, as the interconnect structure is not known until the circuit is finalized.

If reallocation is allowed, it will be executed first. Module binding can be determined by either MILP formulation or heuristic algorithm. Once module binding is finished, the next step is to minimize the number of required clock phases under given timing constraints. The procedure shown in Fig. 3 is used for this purpose. The data path synthesis is finished after register binding and interconnection analysis. Since multiple clock phases are used, the controller design has to consider these clock skews to generate proper control signals. The last step finds the minimum clock period with the known number of clock phases. Details of the flow will be given here.

4. PROPOSED CLOCK GATING

Analysis of the circuits generated by logic synthesizers dissolves that this functionality is implemented by providing a conditional loop back from the output of the flip-flop to its input.

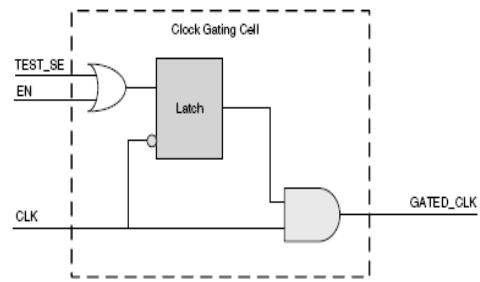


Figure 2 Standard clock gating cell

If such a loop back is active, the flip-flop needs not to be clocked, because the value of the flip-flop will not change; the flip-flop is in the so called hold mode. A promising technique to reduce the power dissipation of the clock net is selectively stopping the clock in parts of the circuit, called clock gating.

4.1 Clock Gating with CSS

One of the frequently used power reduction techniques for synchronous circuits is clock gating. Clock gating includes additional logic to the circuit to reduce the clock tree. Clock gating disables a portion of the synchronous circuit so that the switching power of the flip-flop goes to zero and only leakage currents are incurred. Clock gating works only by using the enable conditions in the flip-flops to gate the clock signal. If there is no enable signal in the design, implementing clock gating will not be possible. When the enable signal goes high, the flip-flops are clocked, whereas when the enable signal goes low, the flip-flops maintain their previous state. There are two types of clock gating styles: the latch-based or latch-free clock gating styles. In the latch-free clock gating style, a requirement needs to be imposed on the circuit that all enable signals should be held constant from the rising edge until the falling edge of the clock to avoid truncating the generated clock pulse prematurely or generating multiple clock pulses unnecessarily. Where as in the case of the latch-based clock gating style, a level-sensitive latch is added in the design to hold the enable signal.

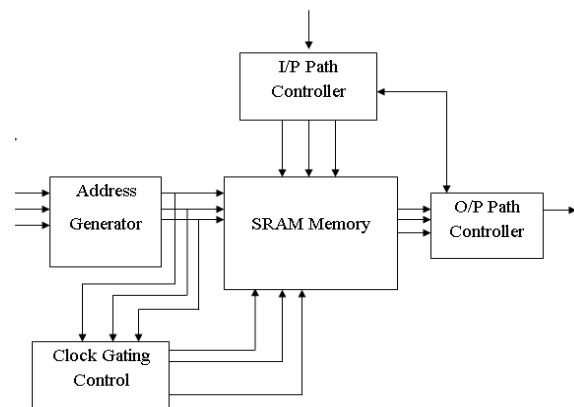


Figure 3 Clock gating architecture

In this the input can be given in form of any pattern. The bits are given separate address by using the address generator. These inputs are stores in the SRAM memory as in figure 4. Because the data stored will not be lost in the SRAM other than DRAM. Then the clock gating application is applied to particular bit, we needed. So that power can be consumed by keeping the clock for the particular bit in idle state. Then the output is sent through the output data path controller. Then the

operation for any circuit is verified and the power is reduced when compared to circuit without gating. In the previous method the clock skews are eliminated. Along with the clock skew scheduling the clock gating is added in order to reduce the power consumption further.

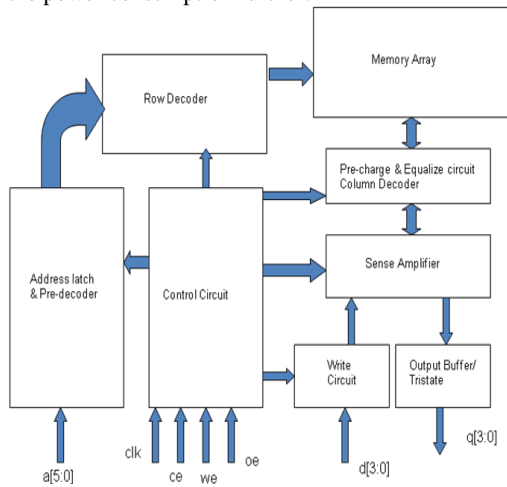


Figure 4 SRAM architecture

5. EXPERIMENTAL RESULTS

The output shows that the clock skews are eliminated along clock gating. By this method the power consumption is reduced less than 48%. In this figure 5 the clock skews are eliminated. Along with the clock skew scheduling the clock gating is added in order to reduce the power consumption. The clock gating is achieved by making enable as zero for the particular module in the circuit. By combining the proposed clock-skew management and low-power module binding, one can achieve significant power reduction with two to four clock phases in most cases. The accurate timing model is developed and used in the proposed low-power module binding scheme.

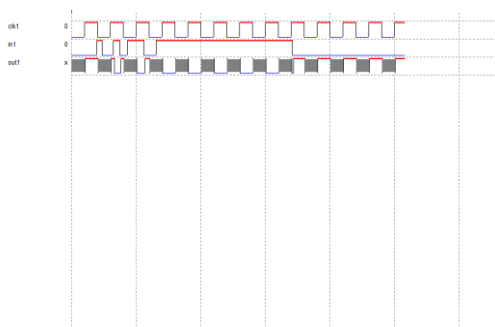


Figure 5 Simulation result for eliminating skews

6. CONCLUSION

In this project the clock gating architecture is added with any type of sequential circuits in order to reduce the power consumption. This clock gating architecture is added with the clock skew scheduling in which the clock skews are managed by the bypass transistors and capacitors. It is used to control the unnecessary power flow between the idle sequential circuits. The overall power reduction can be calculated by implementing the clock scheduling and power gating techniques in a SRAM memory architecture with static and dynamic power calculation.

7. REFERENCES

- [1] Tung-Hua Yeh and Syng Jyan Wang 2012 Power Aware High Level Synthesis with Gated Clock Skew Management.
- [2] Frans Theeuwen, Eric Seelen Power Reduction Through Clock gating by Symbolic Manipulation.
- [3] Li Li1, Jian Sun2, Yinghai Lu1, Hai Zhou1, Xuan Zeng2 "Low Power Discrete Voltage Assignment under Clock Skew Scheduling", Journal of Information Science and Engineering, 2009.
- [4] Shih-Hsu Huang, Chun-Hua Cheng and Dachen Tzeng Simultaneous Clock Skew Scheduling and Power Gated Module Selection for Standby Leakage Minimization.
- [5] Hariyama, M., Ayoma, T., and Kameyama, M. 2005. Genetic Approach to minimizing power consumption of VLSI Processors Using Multiple Supply Voltages.
- [6] Huang, S. H., and Cheng, C. H. 2009 Timing Driven Power Gating in High Level Synthesis." in Prof. Int. Conf. Comput.- Aided Des. Autom. Conf., pp. 173-178.
- [7] Ni, M., memik, S. O., 2009. A Fast Heuristic Algorithm for Multidomain Clock Skew scheduling.
- [8] Obata, T., Kaneko, M., 2004 Clock Signal Skew Scheduling in RT Level Data path Synthesis.
- [9] Deokar, R.B., Sapatnekar, S.S. 2007 A Graph Theoric Approach to Clock Skew Optimization.