International Journal of Current Engineering and Technology ISSN 2277 – 4106

Available at http://inpressco.com/category/ijcet

Research Article

Low power VLSI design approach for 16 bit binary counter to reduce power

Shilpa Shrigiri[a*] and Yojna Bellad[a]

[a]E&CE, SGBIT, Belgaum

Abstract

Gating of the clock signal in VLSI chips is nowadays a mainstream design methodology for reducing switching power consumption. As a consequence many techniques have been proposed to reduce power dissipation. This paper gives the circuit level design of a 16-bit binary counter implemented with clock gating at nibble (4-bit) level. It also gives the power comparison between the normal implementation and the one with clock gating in terms of power. Mentor Graphics tool is used to obtain the gate level hardware design and its simulations. This analysis stresses the use of clock gating as an efficient power reduction technique.

Keywords: *Low power design, binary counter, flip-flops, clock gating, power dissipation, clock distribution network, EDA tools, overhead.*

1. Introduction

Low-power techniques are essential in deep submicron VLSI design due to the continuous increase of clock frequency and chip complexity. Several recently proposed techniques yield low-power operation reducing signals switching activity. Such techniques are generally applied to internal nodes with high capacitive load that heavily contribute to total power dissipation. In particular, the clock system, composed of flip-flops and a clock distribution network, is one of the most power consuming sub-systems in a VLSI circuit. As a consequence many techniques have been proposed to reduce clock system power dissipation . Mentor Graphics is the leading EDA (Electronic Design Automation) tool used to obtain the gate level hardware design and its simulations. Several techniques to reduce the dynamic power have been developed, of which clock gating is predominant. Ordinarily, 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. (We will use the terms toggling, switching and activity to mean the same). With clock gating, the clock signals are ANDed with explicitly defined enabling signals. Clock gating is employed at all levels: system architecture, block design, logic design, and gates. Clock enabling signals are usually introduced by designers during the system and block design phases, where the interdependencies of the various functions are well understood. In contrast, it is very difficult to define such signals at the gate level, especially in control logic, since the interdependencies among the states of various flip- flops (FFs) depend on automatically synthesized logic. We claim that a big gap exists between clock disabling that is derived from the HDL definitions and what can be achieved through detailed knowledge regarding the FFs‟ activities and how they are correlated with each other. Clock gating does not come for free. Extra logic and interconnects are required to generate the clock enabling signals and the resulting area and power overheads must be considered. In the extreme case, each clock input of a FF can be disabled individually, yielding maximum clock suppression. This, however, results in a high overhead; thus suggesting the grouping of several FFs to share a common clock disabling circuit in an attempt to reduce the overhead. On the other hand, such grouping may lower the disabling effectiveness since the clock will be disabled only during time periods when the inputs to all the FFs in a group do not change. In the worst case, when the FFs‟ inputs are statistically independent, the clock disabling probability equals the product of the individual probabilities, which rapidly approaches zero when the number of involved FFs increases. It is therefore beneficial to group FFs whose switching activities are highly correlated and derive a joint enabling signal. Assessing the effectiveness of clock gating requires therefore extensive simulations and statistical analysis of FFs activity. Disabling the clock input to a group of FFs (e.g., a register) in data-path circuits is very effective since many bits behave similarly. Unlike data-path, control logic requires far greater design effort for successful clock gating. This stems from the random nature of the control logic. In many cases clock gating is applied only to the first level of gaiters directly driving FFs, since the majority of the load occurs at the leaves of the clock tree

*Corresponding author: **Shilpa Shrigiri**

where the FFs are connected. Even if we could ideally stop the clock from driving all the FFs when it is not required, the rest of the network will continue pumping clock signals and wasting energy. We consider therefore gating higher levels of the clock tree (closer to root). These portions of the tree may also consume considerable power since they are using long and wide wires plus intermediate drivers to avail robust clock signals for far end FFs. The proposed gating will dynamically prune large portions of the clock tree if it becomes clear that none of the driven FFs is subject to a change in the next cycle . Clock gating can be applied to different hierarchical levels. It is possible to disable the clock signal that drives a big functional unit reducing power dissipation on both its internal nodes and its clock line. Recently it has been shown that clock gating can be successfully applied when a different activation function is generated for each flip-flop. The sensible reduction of power consumption is achieved if flip flop input signal switching activity is sufficiently low. In such cases each flip-flop includes its own gating logic and hence the introduced overhead must be limited as much as possible. There are two techniques for clock gating. The first technique, named as Double Gating in the following, applies the gating technique separately to the master latch and to the slave latch of a flip- flop. Although, in this way, the introduced overhead is doubled, it will be shown that significant power dissipation reduction is obtained if input signal switching activity is low. The second technique, named as NC2MOS Gating in the following, uses only one gating logic for the whole flip-flop. The gating logic is sequential, as opposed to the combinatorial approach and has a reduced overhead. The clock distribution network (or clock tree, when this network forms a tree) distributes the clock signal(s) from a common point to all the elements that need it. Since this function is vital to the operation of a synchronous system, much attention has been given to the characteristics of these clock signals. The most effective way to get the clock signal to every part of a chip that needs it, with the lowest skew, is a metal grid. In a large microprocessor, the power used to drive the clock signal can be over 30% of the total power used by the entire chip. Actually the whole design with the gates and all amplifiers in between has to be loaded and unloaded in every cycle. To save energy, clock gating temporarily shuts off part of the tree, but comes at a cost of increased complexity in timing analysis. Clock tree consume more than 50 % of dynamic power. The components of this power are:

1) Power consumed by combinatorial logic whose values are changing on each clock edge
2) Power consumed by flip-flops and
3) The power consumed by the clock buffer tree in the design.

RTL clock gating works by identifying groups of flip-flops which share a common enable control signal. Traditional methodologies use this enable term to control the select on a multiplexer connected to the D port of the flip-flop or to control the clock enable pin on a flip-flop with clock enable capabilities. RTL clock gating uses this enable signal to control a clock gating circuit which is connected to the clock ports of all of the flip-flops with the common enable term. Therefore, if a bank of flip-flops which share a common enable term have RTL clock gating implemented, the flip-flops will consume zero dynamic power as long as this enable signal is false.

2. Existing Methodologies

In the traditional synchronous design style, the system clock is connected to the clock pin on every flip-flop in the design. This results in three major components of power consumption:

1. Power consumed by combinatorial logic whose values are changing on each clock edge (due to flops driving those combo cells).
2. Power consumed by flip-flops (this has non-zero value even if the inputs to the flip-flops, and therefore, the internal state of the flip-flops, is not changing).
3. Power consumed by the clock tree buffers in the design.

Gating the clock path substantially reduces the power consumed by a Flip Flop. Clock Gating can be done at the root of the clock tree, at the leaves, or somewhere in between. Since the clock tree constitutes almost 50% of the whole chip power, it is always a good idea to generate and gate the clock at the root so that entire clock tree can be shut down instead of implementing the gating along the clock tree at the leaves.

3. Implementation

3.1. 16-bit counter without clock gating.

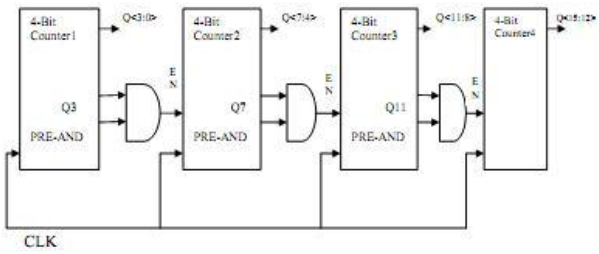

Figure 1. 16-bit counter without clock gating.

Here 16-bit counter is implemented using four 4-bit counters as shown in Figure 1. As we are doing 16-bit counter with clock gating at nibble level so in order to make comparison easy we have implemented without clock gating also at nibble level. These counters are connected through AND gate, where input to the AND gate are last bit i.e, fourth bit of previous counter(Q3) and ANDed output EN.Q2.Q1.Q0 which is considered as PRE_AND in the schematic of four bit counter. Whenever Q3 and PRE_END are HIGH that time a pulse is given to EN of next 4-bit counter by which it will start counting, as already shown in Table 4.2, that whenever EN is HIGH that time only the counter must work similarly here counters are working only when EN of corresponding

counter is HIGH. And that is provided by positive output of previous AND gate which we can see in Fig. 4.16.As it is synchronous counter a clock pulse (CLK) is provided to all the counter at the same time by using a
PULSE of 5V with '0' delay, rise and fall time as 10ps.

3.2. 16-bit Counter with Clock gating.

Figure 2. shows the 16-bit level where 4 such 4-bit blocks are stitched together using the clock gating logic. The AND function signal (called NXT_AND) from each of these 4 stages, feed the clock gating logic for the subsequent stage. The clock gating logic is already shown. We can see that the NXT_AND which in a normal counter would act as an enable signal for the subsequent stage is acting as a gating signal for clock for the next stage. It implies that the gating signal is directly derived from the existing logic without any need for additional gating function logic in this case. Hence gating signal is given as input to clock gating logic along with clock to generate the gated clock signal for the next 4-bit stage. For the present design, i.e. a binary sequence, we can see that the first 4 bit counter (Q[3:0]) has a continuous clock since its gating function is Vdd (logic 1). For the second 4 bit stage, the clock would be a function of the NXT_AND signal of the first 4-bit stage. i.e. one clock pulse every 16 clock cycles, Similarly 3rdand 4th 4-bit stage clock will pulse once every 256 and 4096 clock cycles respectively. This comes with an additional area overhead for the clock gating logic. Also the AND gate delay on the clock line will subsequently cause a bit of delay on the counter output. But the reduction in switching power is substantial.

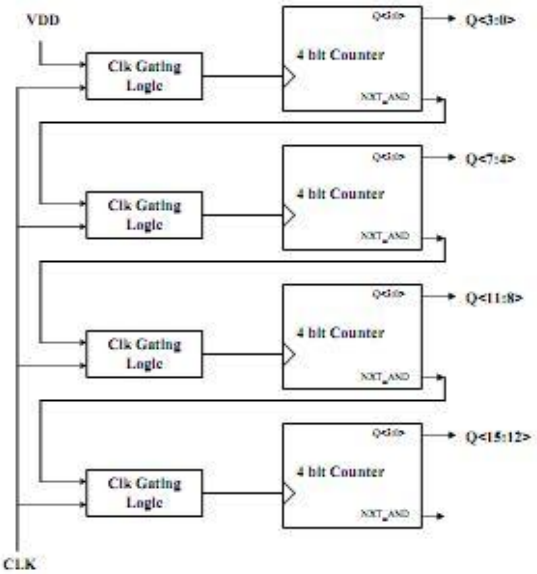

Figure 2. 16-bit clock gating based counter.

4. Results

The Comparisons are made with a normal 16-bit binary counter as reference. Calculations are shown below.
Equation for Average Power Consumption (Pmax) is:
Pavg = V Iavg …… Eq. 1

Where, V= Supply voltage (V=1.2V) Iavg = Average supply current (uA)
16-bit counter without Clock Gating: Iavg = -251.0082uA
Hence, Pavg1 = -301.209uW
16-bit counter with Clock Gating: Iavg = -97.58uA
Hence, Pavg2 = -117.096uW

$$\% \text{ decrease} = \frac{Pavg1 - Pavg2}{Pavg1} \times 100 \qquad (6.2)$$

Table 1. Comparison table.

Parameters	Normal Counter	Optimized Counter	% decrease
I_{avg}	-251.0082uA	-97.58uA	61.12
P_{avg}	-301.209uW	-117.096uW	61.12

The analyses is carried out for different Process Corners (V=1.2V, T=27 ºC). The following table shows the results.

Table 2. Process corners comparison table

Process Corners	I_{avg}	P_{avg}
SS	-88.2129uA	-105.855uW
TT	-97.58uA	-117.096uW
FF	-109.9455uA	-131.934uW

The analyses is carried out for different voltage supplies (P=TT, T=27 ºC). The graph of Power v/s Voltage is shown below,

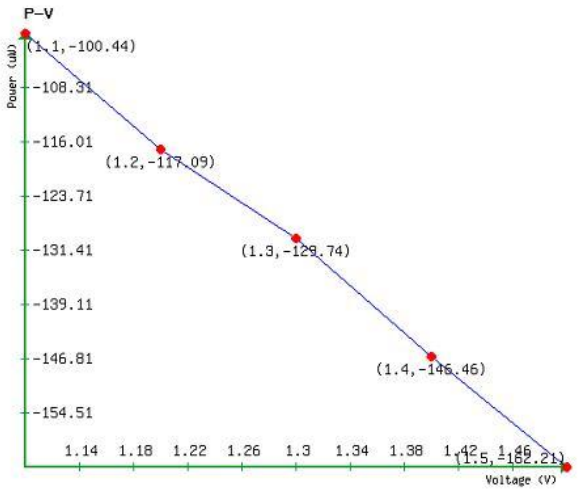

Figure Power v/s voltage graph for different values.

Conclusion

The Comparisons are made with a normal 16-bit binary counter as reference. As per the readings in the table of results, it can be seen that there is a considerable amount of power reduction as in case of 16-bit counter with clock gating technique. The clock gating cells contribute to additional leakage. But in spite of these, the savings in active power is substantial, i.e., 61.12%.

This paper demonstrates the strength of clock gating on a simple 16 bit binary counter, implemented with a nibble level gating logic. Although it adds a bit of area and delay to the existing logic, clock gating is a default choice considering the active power reduction it gives. Since the counter output depends largely on the frequency of Clock provided also the transient period and the sizing of the inverters and NAND used in the schematic, by trying various other sizing for these but with added delay, the output till the last bit can be obtained provided delay is not a major issue. Also, the nibble level partition may not be the most optimal choice in terms of Area, Power and Speed. We can try different combinations like Byte level splits or Word level splits or completely random and non unique splits depending on the final application and the best savings. This Idea can also be extended to other sequential logic blocks. In fact Clock Gating is a default choice in today's sequential design involving millions of gates and a variety of sub blocks. A particular block inactive at a given time can be shut off by blocking clock to the block. Even present day tools support clock gating as one of the power reduction feature which can be evaluated and added onto our design at a fairly later stage of the design.

References

A.G.M. Strollo, E. Napoli, D. De Caro, Low-power flip-flops with reliable clock gating, Microelectronics Journal, Vol. 32 , 2001, pp. 21-28.

Shmuel Wimer and Israel Koren, Fellow, IEEE, The Optimal Fan-Out of Clock Network for Power Minimization by Adaptive Gating, IEEE transactions on Very Large Scale Integration (VLSI) Systems, Vol. 20, No. 10, October 2012, pp.1772-1780.

Anantha P. Chandrakasan, S. Sheng, R.W. Brodersen, Low-power CMOS digital design, IEEE Journal ofSolid-State Circuits, Vol. 27, No. 4, 1992, pp. 473-483.

Bishwajeet Pandey and Manisha Pattanaik, Clock Gating Aware Low Power ALU Design and Implementation on FPGA, International Journal of Future Computer and Communication, Vol. 2, No. 5, October 2013, pp. 461-465.

Jagrit. K, M. Ayoubkhan and Arti Noor, A Review of Clock Gating Techniques, MIT International Journal of Electronics and Communication Engineering, Vol. 1, No. 2, August 2011, pp. 106-114.

Padmini G.Kaushik, Sanjay M.Gulhane, Athar Ravish Khan, A Survey on Power Management Techniques, International Journal of Scientific & Engineering Research, Vol. 3, Issue 8, August-2012, pp. 1-5.

Dushyant Kumar Sharma, Effects of Different Clock Gating Techinques on Design, International Journal of Scientific & Engineering Research Volume 3, Issue 5, May-2012, pp. 1-4.

Neil .E. Weste, Principles of Cmos Vlsi Design, Pearson Education Publications, 2004.

Noureddine Chabini, Ismaïl Chabini, El Mostapha Aboulhamid, and Yvon Savaria, Methods for Minimizing Dynamic Power Consumption in Synchronous Designs with Multiple Supply Voltages, IEEE Transactions on Computer-Aided design of Integrated Circuits and Systems, Vol. 22, No. 3, March 2003, pp. 346-351.

Chitambara Rao.K, Nagendra.K, Sreenivasa Rao.Ijjada, Implementation of Low-Power and Low-energy synchronous SAPT Logic, International Journal of Distributed and Parallel Systems (IJDPS), Vol.3, No.1, January 2012, pp. 185-194.

Y. Shin, J. Seomun, Kyu-Myung Choi and T. Sakurai, Power gating: Circuits, design methodologies, and best practice for standard-cell VLSI designs, ACM Transactions on Design Automation of Electronic Systems, Vol. 15, No.4, Article 28, September 2010.