Dynamic Power Reduction of Digital Circuits by Clock Gating

Padmini G.Kaushik[1], Sanjay M.Gulhane[2],Athar Ravish Khan[3]

[1,2,3]Department of Electronics & Telecommunication
Jawaharlal Darda Institute of Engineering and Technology, Yavatmal, Maharashtra, India.

Abstract

Clock gating technology can reduce the consumption of clock signals' switching power of flip-flops. The clock gate enable functions can be identified by Boolean analysis of the logic inputs for all adders. However, the enable functions of clock gate can be further simplified, and the average number of adders driven by enable functions can be improved. In this way, the circuit area can be reduced; therefore, the clock gating can be improved and power saving can be achieved.

Keywords: *AND gate, Clock gating, power, power gating.*

1. Introduction

With the decrease of feature sizes and increase of clock frequencies in integrated digital circuits, power consumption has become a major concern for modern integrated circuit designs. Power dissipation has a dynamic component, due to the switching of active devices, and a static component, due to the leakage of inactive devices. Since our work targets dynamic power only, further references to "power" in this we will imply the dynamic power.

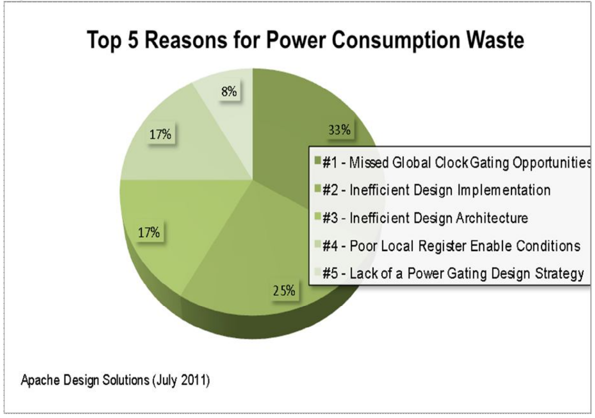
Clock gating is one of the most effective and widely used techniques for saving clock power. The clock net is one of the nets with the highest switching density, resulting in high power dissipation in the adders. A promising technique to reduce the power dissipation of the clock net is selectively stopping the clock in parts of the circuit, called "clock gating". It is very well integrated into semi-custom design flows nowadays. By gating the clock, the switching activity of the adders clock signal is reduced. However, clock gating circuitry itself occupies chip area and consumes additional power; therefore a judicious selection of circuit.

2. Reasons For Power Consumption Waste

Low-power seems to be on everyone's mind these days, and it's not just the chip design teams. One common consumer complaint is that the "battery life is way too short"! And of course, we all know this one, "OMG – that laptop is sure hot"! Even data center facilities managers lament, "We can't supply enough power to the equipment—and when we do, we can't cool it!"

But, it wasn't always like that. We didn't use to design with power in mind. As long as the design met functional specifications and performance targets, it was ready to be shipped. So, either power was not an issue with smaller designs at that time, or else a bigger heat sink or fan could be used. Fast forward to now, where increased design complexity combined with the drive toward mobile applications requires designs to have power methodologies. One key to low-power design, in addition to a host of automatic optimization techniques, is to eliminate or reduce power consumption waste.

While working with a variety of customers on low-power designs, we found at least 20 reasons for wasted power. We listed the top 5 here, including how customers are dealing with these issues in current design flows.



Apache Design Solutions (July 2011)

2.1 Missed Global Clock Gating Opportunities

While local register-level clock gating has been automated with the aid of synthesis tools (see more below), global or "architectural" clock gating has not. To control the clocks at a global level you must understand the design intent, including under what operating conditions the clocks are required to run and when the clocks can stop. Knowing the design intent is not something an EDA tool can easily achieve, but the issue actually goes beyond clocks driving registers in the design. Clocks are also used by synchronous memories, which are the predominant type used today. Redundant memory Read and Write cycles, without the address and data changing cycle-to-cycle, wastes huge amounts of power. It is up to designers to understand the design and seek opportunities to stop the clocks when operation is not required.

2.2 Inefficient Design Implementation

This large area encompasses all steps in the flow that follow RTL design and functional verification, such as synthesis, placement, clock-tree synthesis, routing, timing optimization and closure. There are several instances for uncontrolled implementation tools to introduce power inefficiencies into the design, including synthesis that may oversize logic gates, or pick a power-inefficient micro-architecture for an arithmetic component—but of course meeting timing constraints. During placement, some cells that will be connected by high-activity nets may be placed far apart, resulting in high capacitance and wasted power. Aggressive clock skew constraints will result in excessive clock buffering plus a large number of buffers, and clock tree balancing may also result in additional buffers. Routing constraints may result in long wires for high-activity nets, similar to issues seen with inefficient placement. So, the way to achieve efficient implementation is to provide proper constraints to the automatic implementation tools, and not be too aggressive, especially in timing optimization and closure.

2.3 In Efficient Design Architecture

Although much larger opportunities for power reduction exist at higher levels of abstraction, it is no big surprise that inefficient design architecture does not appear at the very top of this list. Even though it is no designer’s intent to create an inefficient architecture, there are several aspects that must be considered. One aspect that is related to missed global clock gating opportunities is an architectural issue. However, true architectural considerations must go beyond that to regard how fast the clocks must be for any given functional or performance requirement; how many pipeline stages are needed to meet latency requirements; and how much work can or should be done per cycle. Another aspect is the memory sub-system organization. Once the amount of memory required is known, how should it be partitioned? What types of memories should be used? How often do they need to be accessed? All of these issues greatly affect power consumption, so designers must make power-performance-area tradeoffs for various alternative architectures in order to make informed decisions.

2.4 Poor Local Register Enable Conditions

As mentioned above, register clock gating is well automated in modern logic synthesis tools. Given an existing enable condition for a register, a synthesis tool will insert a clock gating cell controlled by the enable signal instead of implementing a recirculating mux. Synthesis tools do this while meeting timing constraints, as well as ensuring testability. So, where is the potential for wasted power here? It all has to do with just how efficient the enable condition is gating the register clock when it is not required. High clock gating coverage, i.e. the percentage of registers with enable conditions, while a useful metric, does not always translate into high clock gating efficiency. By studying clock enable conditions, and understanding how much clock power is consumed downstream of clock enables, designers can focus on areas that have the most inefficiency and represent the most power savings opportunities.

2.5 Lack Of A Power Gating Strategy

Leakage power is now a large proportion of total power, starting with 65nm designs, and is even more dominant in 40nm and below designs. Although automatic techniques can be used downstream in the design flow to reduce leakage power, such as multi Vt cell optimization, power gating (or power shut-off) is by far the most effective practical technique for reducing leakage power consumption. However, power gating can’t simply be left up to implementation tools. The design must be partitioned into power domains up front, and control signals must be designed to ensure proper operation, complete with state retention circuitry if required.

3. Gating

The process of selecting only that portion of wave between specified time intervals or between specified amplitude limits. It is also used to control the signal by means of combinational logic elements.

It is a process in which a predetermined set of conditions, when establish permits a second process to occur. The powers saving technique are

- Data Gating
- Power Gating
- Clock Gating

3.1 Clock Gating

Clock gating is a technique that reduces the switching power dissipation of the clock signals. By inserting a clock gate circuitry unnecessary clock switching of adders can be avoided during clock cycles when stored data remains unchanged. The condition under which a clock transition is passed through the clock gate is known as enable function, which is the input of the EN port of clock gate circuitry. The internal node that can replace an original enable function is called enable function.

It is a most popular method for power reduction of clock signals and functional units. A significant fraction of the dynamic power in a chip is in the distribution network of the clock. Up to 50% or even more of the dynamic power can be spent in the clock buffers. The reason is:

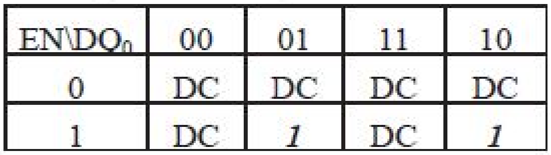
– Clock buffers have the highest toggle rate in the system
– Typically there are lots of clock buffers in a design
– Clock buffers often have a high drive strength to minimize clock delay

Additionally receiving the clock dissipate some dynamic power even if the input and output remain the same.

3.2 Identification Of Enable Sub- Functions

The condition under which a clock transition is passed through the clock gate is known as enable function. The internal node that can replace an original enable function is called enable sub-function. It is assumed that the clock gate enable functions have been identified by Boolean analysis of the feedback loop for all flip flops and the logic cones that implement these enable functions are available. The original enable combinational logic is represented by AIG, and each PO in the AIG represents the enable function for a adder. The first step is to identify enable sub-functions for each. In this, random simulation and SAT Solver are deployed to identify enable functions within the logic cone. Random simulation can identify enable function candidates for each PO; SAT Solver engine can verify whether the enable function candidate is a real enable function. Random simulation can filter a large number of invalid nodes.

Table1. True table of the enable sub function, where “DC” represents “don’t care”



EN\DQ$_0$	00	01	11	10
0	DC	DC	DC	DC
1	DC	*1*	DC	*1*

In addition, random pattern simulation is fast and it can save much time for identification of enable functions. And getting the switching activity of each node is useful for the following accurate power estimation. It is known that exhaustive simulation is a prevalent form of functional verification but time-consuming. So in this paper bit-parallel random simulation is applied, although it can just identify enable sub-function candidates. Random simulation is used to find a counterexample by random search. For example, simulating one thousand times (each time the random input vector is 32-bit) is likely to identify enable function candidates for each PO. At the same time, the switching activity (signal probability) of each node is estimated. Use the simulation signature as a filter.

3.3 Power-Optimal Enable Function Selection

Power saving may be optimized considering the enable functions. After identifying the enable functions, further power evaluation and analysis of each enable function is performed and an optimized clock gating structure is selected for implementation. The optimized clock gating structure is a set of nodes that are the enable functions of POs and maximize power savings. The optimal set of enable functions is found with partition method and DFS.

3.4 Power Estimation

When selecting an enable function node as part of the final solution, the node’s power consumption is estimated. Since each AI node is an AND gate, its capacitance is significantly smaller than the Adder and clock gate’s; according to the statistics, generally, clock gate’s capacitance is four times of that of AI node’s;

CCG = 4 CAI

where CAI is AI node capacitance, and adders capacitance is six times of that of node’s,

CFF = 6 CAI.

So for simplicity, we set the capacitance associated with each AI node’s net as unit 1, clock gate’s capacitance as 4, and primary output’s capacitance. Switching power (*PSW*) is usually modeled by the following formula:

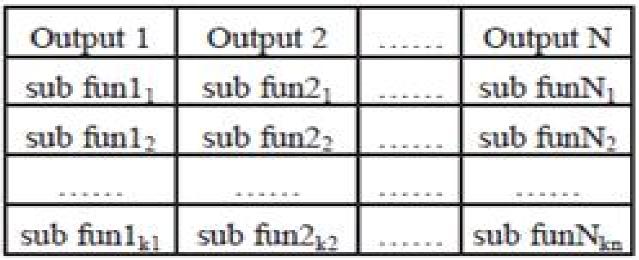
$$P_{sw} = 0.5 * f * V_{dd}^{2} * C_{AI} * SA_{AI}$$

Where *f* is the clock frequency, Vdd is the supply voltage, *CAI* is the load capacitance for AI node, and *SAAI* is the switching activity for AI node. Because the clock frequency *f* and supply voltage Vdd remain unchanged for the entire circuit.

3.5 Power Optimization

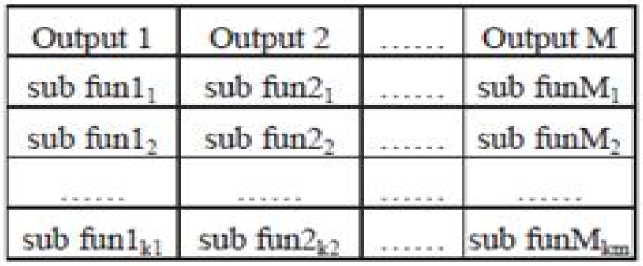
A PO may be gated by one or more enable sub functions. Likewise, each enable function may logically gate one or more POs. A matrix can be derived in order to further represent the logical functionality of the enable functions. As shown in following table2;

Table2. Function--Output matrix



Output 1	Output 2	......	Output N
sub fun1_1	sub fun2_1	......	sub funN_1
sub fun1_2	sub fun2_2	......	sub funN_2
....	....	....	....
sub fun1_{k1}	sub fun2_{k2}	......	sub funN_{kn}

Table3. Sub matrix



Output 1	Output 2	......	Output M
sub fun1_1	sub fun2_1	......	sub funM_1
sub fun1_2	sub fun2_2	......	sub funM_2
....	....	....	....
sub fun1_{k1}	sub fun2_{k2}	......	sub funM_{km}

The rows of the matrix represent enable functions, and the columns represent the gated sequential functions. The complexity of traversing all the solution space is extremely high. So partition and clustering are deployed to speed up. Our algorithm will select some rows to cover the gated sequential signals to improve the original clock gating and to get more power saving. As shown in table 3 , we get several sequential functions in one cluster and traverse all the solution space to obtain the optimal solution in which the total power saving is best for these sequential functions. It is assumed that there are *N* POs and each node has *M* enable functions. The complexity of traversing all the solution space is *O* (*MN*) without clustering. If we partition the global *N* POs into *K* clusters in which there are *N/K* PO nodes, the complexity in each cluster is *O*(*M N/K*) and the total complexity is only *O*(*K*M N/K*).

4. Techniques Of Clock Gating

There are four different techniques for Clock Gating. Here we have discussed using AND gate technique below:

4.1 *And Gates*

In sequential circuit one two-input AND gate is inserted in logic for clock gating. One input to AND gate is clock and while the second input is a signal used to control the output (means it will control the sequential circuit's clock). For experimental purpose we are taking a simple counter shown in Fig 1(A) as a sequential circuit application. Figure 1(B) shows the waveform of the output of regular counter, initially at reset = '0', counter initialized to "0" and after that when reset='1' counter increments at each negative edge of the clock. Figure 2(A) shows the clock gating technique for the counter by inserting one AND Gate. Figure 2(B) shows the output of counter when counter is negative edge triggered and enable ('en') changes from clock cycle starting from negative edge to the next negative edge, in this case output of the counter changes after one clock cycle of being en='1'. From Figure 2(C) we have observed that when counter is positive edge triggered and enable is changing starting from positive edge to the next positive edge, counter increments one extra time, due to tiny "Glitch", when it goes down due to more falling time of the enable, and the output in this case is wrong.

Like AND gate based clock gating, NOR gate, Latch based AND gate and Latch based NOR Gated Clock circuits are also used for clock gating.

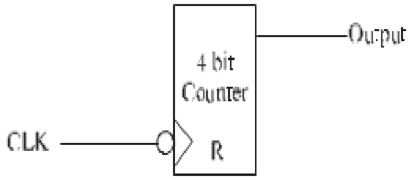


Fig. 1(A): Basic Counter (negative edge triggered).

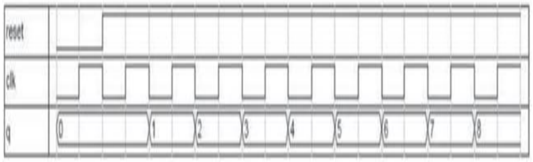


Fig.1(B): Normal output of the counter without Clock gating

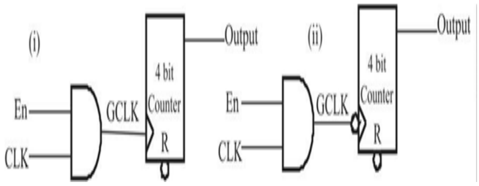


Fig. 2(A): Clock gating using AND gate Circuit.

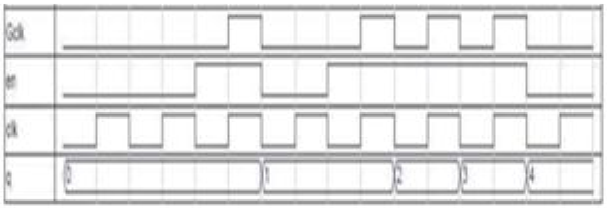


Fig. 2(B): Output of Counter when counter is Negative edge triggered.

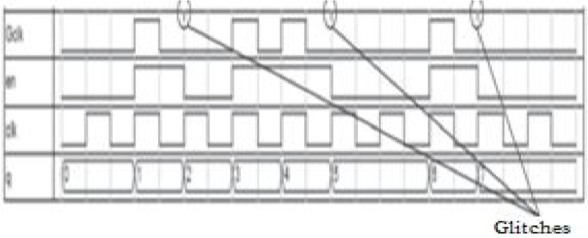


Fig. 2(C): Wrong Output due to Glitch, when counter is Positive edge triggered.

5. Implementation

The AND gate based clock gating circuit is applied for 3 bit full adder. The figure 3(A) shows the circuit diagram of simple 3 bit full adder without clock gating. The figure 3(B) shows the circuit diagram 3 bit full adder with clock gating circuit.

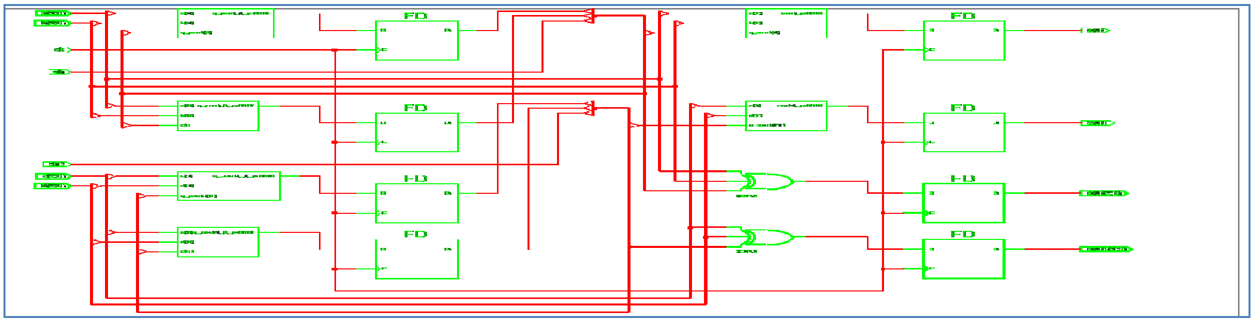


Fig. 3(A): 3 bit Full adder without clock gating circuit.

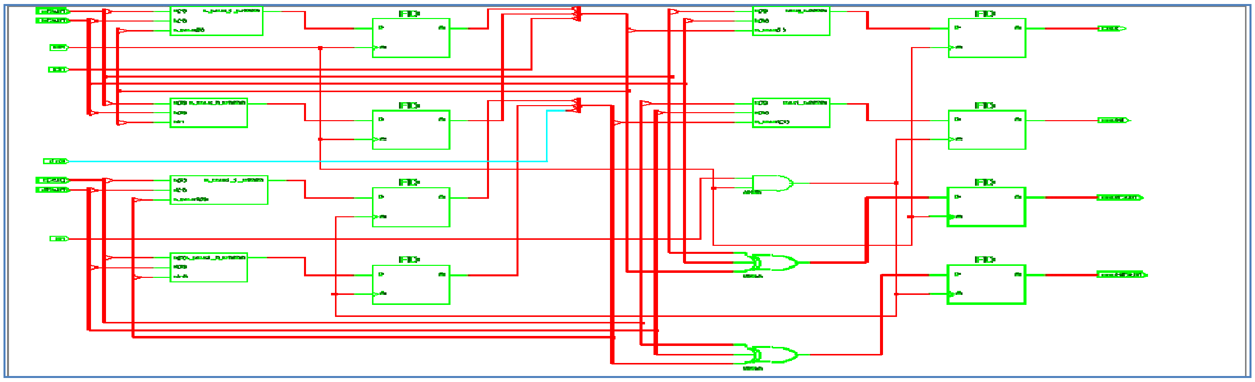
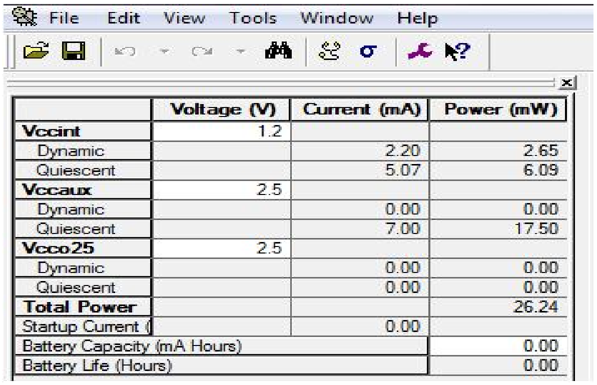


Fig. 3(B): 3 bit Full adder with clock gating circuit.

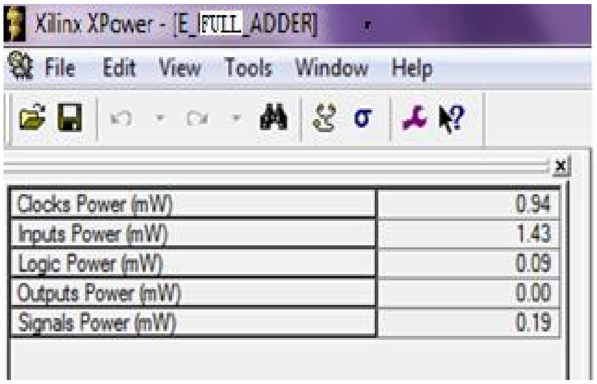
6. Result

The dynamic power reduction by clock gating technique using AND gate is verified for 3 bit full adder. The results shows that a significant change in dynamic power. Figure 4(A) shows the input voltage, current drawn, dynamic and static power drawn and total power drawn for 3 bit full adder without clock gating applied. Figure 4(B) and figure 4(C) shows the power and current subtotals of different



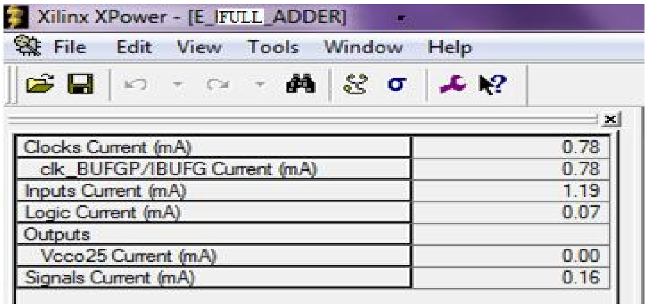
	Voltage (V)	Current (mA)	Power (mW)
Vccint	1.2		
Dynamic		2.20	2.65
Quiescent		5.07	6.09
Vccaux	2.5		
Dynamic		0.00	0.00
Quiescent		7.00	17.50
Vcco25	2.5		
Dynamic		0.00	0.00
Quiescent		0.00	0.00
Total Power			26.24
Startup Current (		0.00	
Battery Capacity (mA Hours)			0.00
Battery Life (Hours)			0.00

Fig. 4(A): Power and current summery for 3 bit.



Clocks Power (mW)	0.94
Inputs Power (mW)	1.43
Logic Power (mW)	0.09
Outputs Power (mW)	0.00
Signals Power (mW)	0.19

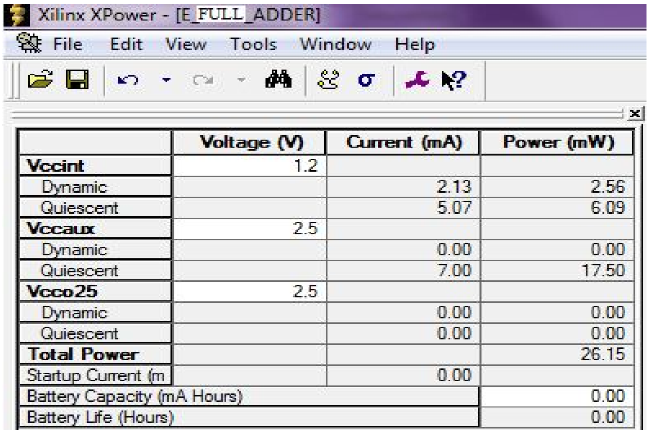
Fig. 4(B): Power subtotal for 3 bit Full adder without clock gating.



Clocks Current (mA)	0.78
clk_BUFGP/IBUFG Current (mA)	0.78
Inputs Current (mA)	1.19
Logic Current (mA)	0.07
Outputs	
Vcco25 Current (mA)	0.00
Signals Current (mA)	0.16

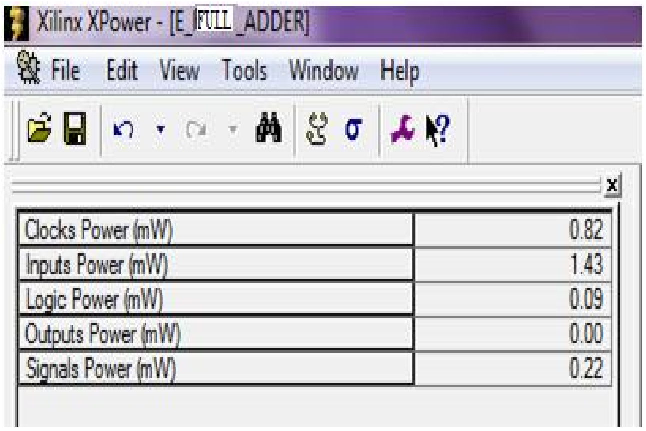
Fig. 4(C): Current subtotal for 3 bit Full adder without clock gating.

Figure 5(A) shows the input voltage, current drawn, dynamic and static power drawn and total power drawn for 3 bit full adder with clock gating applied. Figure 5(B) and figure 5(C) shows the power and current subtotals of different elements.


Xilinx XPower - [E_FULL_ADDER]
File Edit View Tools Window Help

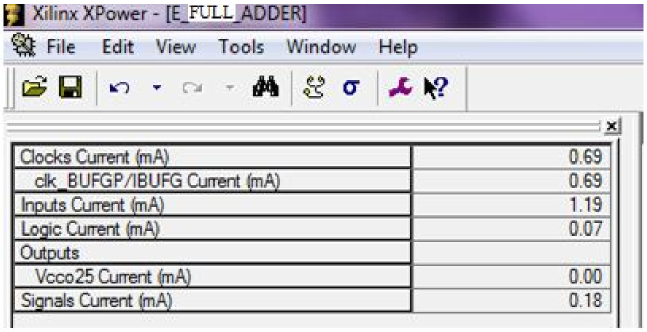
	Voltage (V)	Current (mA)	Power (mW)
Vccint	1.2		
Dynamic		2.13	2.56
Quiescent		5.07	6.09
Vccaux	2.5		
Dynamic		0.00	0.00
Quiescent		7.00	17.50
Vcco25	2.5		
Dynamic		0.00	0.00
Quiescent		0.00	0.00
Total Power			26.15
Startup Current (m		0.00	
Battery Capacity (mA Hours)			0.00
Battery Life (Hours)			0.00

5(A): Power and current summery for 3 bit Full adder


Xilinx XPower - [E_FULL_ADDER]
File Edit View Tools Window Help

Clocks Power (mW)	0.82
Inputs Power (mW)	1.43
Logic Power (mW)	0.09
Outputs Power (mW)	0.00
Signals Power (mW)	0.22

Fig. 5(B): Power subtotal for 3 bit Full Adder with clock gating


Xilinx XPower - [E_FULL_ADDER]
File Edit View Tools Window Help

Clocks Current (mA)	0.69
clk_BUFGP/IBUFG Current (mA)	0.69
Inputs Current (mA)	1.19
Logic Current (mA)	0.07
Outputs	
Vcco25 Current (mA)	0.00
Signals Current (mA)	0.18

Fig. 5(C): Current subtotal for 3 bit Full adder with clock gating.

7. Conclusion

In this .paper, we considered clock gating for power reduction and propose that clock gating capability be in corporate at various points in the clock distribution network. We implemented AND gate based clock gating circuit for 3 bit full adder as system. Depending on the architecture and clock activity, clock power can be reduced by over 50%.

Acknowledgments

The authors would like to thank firstly, our GOD, and all friends who gave us any help related to this work. Finally, the most thank is to our families and to our country INDIA which born us.

References

[1] R. Mehra Synopsys Inc., ”Improve Clock Gating through Power-Optimal Enable Function Selection” DAC 2009. 45th ACM/IEEE, 8-13 June pages(4).

[2] Jagrit kathuria centre of development computing noida, “A Review of Clock Gating Techniques ” 2009 page(1-9).

[3] Benini, L.; De Micheli, G.; IEEE Trans. CAD Volume 15, Issue 6, June 1996 pages(1-29)

[4] Safeen Huda, Muntasir Mallick, Jason H. Anderson“Clock Gating Architectures For FPGA Power Reduction” Dept. of ECE, Univ. of Toronto pages(1-7)

[5] Shmuel Wimer and Israel Koren, Fellow, IEEE “ The Optimal Fan-Out of Clock Network for Power Minimization by Adaptive Gating” 2011 pages(1-9)

[6] V. G. Oklobdzija, Digital System Clocking—High-Performance and Low-Power Aspects. NewYork:Wiley*, 2008.*

[7] N. H. E. Weste and D. Harris*, CMOS VLSI* Design—A Circuit and System Perspective. Boston, MA: Addison-Wesley*, 2005,* ch. *7, 12.*

[8] Narayana Koduri, Kiran Vittal, Atrenta Inc. (San Jose, Calif.) “Power analysis of clock gating at RTL ”

[9] Yu-Min Kuo Shih-Hung Weng Shih-Chieh Chang, *“A Novel Sequential Circuit Optimization with Clock Gating Logic*”, IEEE/ACM International Conference on Computer-Aided Design, ICCAD 2008. , pp-230 – 233, 2008.

[10] Changjun Kang and Chunhong Chen, *“Activity-sensitive clock design for low power consumption”,* Canadian Journal of Electrical and Computer Engineering, Vol. 32, No. 4, pp-221 - 226 ,Fall 2007.

[11] Seungwhun Paik, Inhak Han, Sangmin Kim, And Youngsoo Shin, *Senior Member, IEEE* “Clock Gating Synthesis Of Pulsed-Latch Circuits ” Ieee Transactions On Computer-Aided Design Of Integrated Circuits And Systems, Vol. 31, No. 7, July 2012

[12] 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

[13] Alberto Bonanno, Alberto Bocca, Alberto Macii, Enrico Macii, and Massimo Poncino “Data-Driven Clock Gating for Digital Filters ” J. Monteiro and R. van Leuken (Eds.): PATMOS 2009, LNCS 5953, pp. 96–105, 2010,_Springer-Verlag Berlin Heidelberg 2010.