International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 04 | Apr 2020 www.irjet.net p-ISSN: 2395-0072

Realization of Low Power N- Bit Kogge Stone Adder

Viraj Bhusari[1]

[1]*M. Tech Scholar, Dept. of Micro and Nano-Electronics, School of Electronics Engineering, Vellore Institute of Technology, Vellore, Tamilnadu-632014, India*

Abstract - *Adder is the most critical component in Arithmetic circuits. The performance of arithmetic circuits enhances the processor efficiency. In this work, Kogge–stone Adder which is of the type Carry Look-ahead is analyzed for its operation and performance. Static power consumes a significant portion of the available power budget in the circuit. In Kogge–Stone Adder [1], static power consumption increases as input bit, N increases [2]. Adaptive power gating is an effective method to reduce idle leakage current. This technique is applied to 8-bit Kogge-Stone adder in order to reduce the static power consumption. For the purpose of understanding, the reported work on Kogge–stone adder without and with power gating technique, for 8-bit is implemented and simulated using Verilog HDL, cadence virtuoso and Synopsys tool.*

Key Words: **Static Power, Dynamic Power, Synopsys and Cadence Tool, Power Gating, Adaptive Power Gating.**

1. INTRODUCTION

The continuous scaling in the CMOS technology leads to a new regime where the efficiency of an IC will be constrained with its power dissipation. Until now the researches were not giving more importance on static power dissipation. But with rapid advancement in the modern VLSI design, where the number of transistors crossing few millions, it has become a major factor while designing. With different leakage currents flowing even when it is not necessary through the huge number of transistors leads to large static power dissipation. This amount of static power dissipation is almost comparable to the normal dynamic power dissipation. Many attempts have been made in order to minimize this static power dissipation. One among it is power gating, where no static path will be present between supply and ground for the flow of leakage currents.

Enhancing the speed and reducing the power consumption of adders improves the efficiency of ALU. As other basic operations like subtraction and multiplication are interpreted in terms of addition itself. As the power dissipation due to switching activity is inevitable, reduction of static power gains more importance as it improvises the functionality of adder. This will be reflected in fast and accurate operation of digital system. The primary objective is to reduce the leakage power in a high speed Kogge-Stone adder by means of adaptive power gating. The different architectures of adders are carry ripple adders, carry skip adder, carry look-ahead adder, parallel prefix tree adders. The distinguishing factor among these adders is the way in which the carry propagates from one stage to next, as number of bits increases. Ripple carry adder is least efficient in this regard and carry look-ahead adder is most. This is the main motivation to choose Kogge-stone adder in our analysis. Also kogge-stone adder implementation is the most straightforward, and also it has one of the shortest critical paths of all tree adders. This adder improves the speed by determining whether the bit pair will generate the carry or propagate the carry.

2. 8-bit KOGGE STONE ADDER

The Kogge-Stone adder is most famous version of carry look-ahead adder. It generates the carry signals in O (log2N) time, and is widely considered as the fastest adder design possible. It has the most common architecture for high-performance adders in industry. The Kogge-Stone adder concept was first developed by Peter M. Kogge and Harold S. Stone. In Kogge-stone adder, carries are generated fast by computing them in parallel at the cost of increased area.

The Kogge Stone Adder (KSA) has regular layout which makes them favored adder in the electronic technology. Another reason the KSA is the favored adder is because of its minimum fan-out or minimum logic depth. As a result of that, the KSA becomes a fast adder but has a large area .The delay of KSA is equal to log2n which is the number of stages for the "o" operator. The KSA has the area (number of "o" operators) of (n*log2n)-n+1 where n is the number of input bits

The complete functioning of KSA can be easily comprehended by analyzing it in terms of three distinct parts:

1. Pre processing

This step involves computation of generate and propagate signals corresponding too each pair of bits in A and B. These signals are given by the logic equations below:

$p_i = A_i$ xor B_i

$g_i = A_i$ and B_i

2. Carry look-ahead network

This block differentiates KSA from other adders and is the main force behind its high performance. This step involves computation of carries corresponding to each bit. It uses group propagate and generate as intermediate signals which are given by the logic equations below:

$P_{i:j} = P_{i:k+1}$ and $P_{k:j}$
$G_{i:j} = G_{i:k+1}$ or $(P_{i:k+1}$ and $G_{k:j})$

3. Post processing

This is the final step and is common to all adders of this family (carry look ahead). It involves computation of sum bits. Sum bits are computed by the logic given below:

$S_i = p_i$ xor C_{i-1}

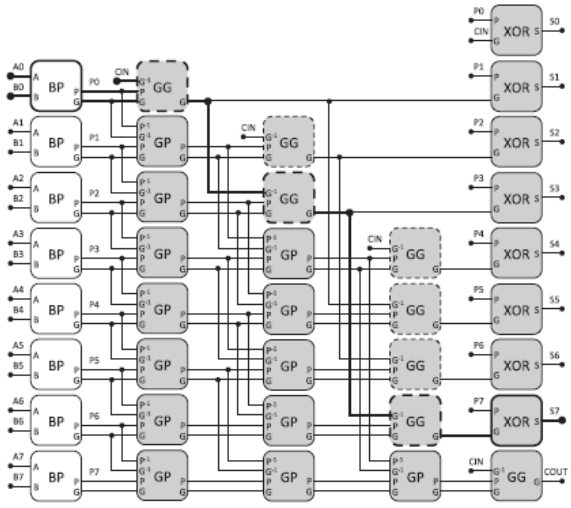

Fig1. Schematic representation of the 8 bit adder

3. POWER DISSIPATION IN DIGITAL IC

Sources of Power Dissipation:
Power dissipation in CMOS circuits comes from two components:

Dynamic dissipation due to
- charging and discharging load capacitances as gates switch
- "short-circuit" current while both pMOS and nMOS stacks are partially ON

Static dissipation due to
- Sub-threshold leakage through OFF transistors
- Gate leakage through gate dielectric
- Junction leakage from source/drain diffusions

Power can also be considered in active, standby, and sleep modes. *Active power* is the power consumed while the chip is doing useful work. It is usually dominated by $P_{switching}$. *Standby power* is the power consumed while the chip is idle. If clocks are stopped and ratioed circuits are disabled, the standby power is set by leakage. In sleep mode, the supplies to unneeded circuits are turned off to eliminate leakage. This drastically reduces the *sleep power* required, but the chip requires time and energy to wake up so sleeping is only viable if the chip will idle for long enough.

3.1 DYANAMIC POWER DISSIPATION

Dynamic power consists mostly because of the switching power. The supply voltage *VDD* and frequency *f* are readily known by the designer. To estimate this power, one can consider each node of the circuit. The capacitance of the node is the sum of the gate, diffusion, and wire capacitances on the node. The activity factor can be measured from logic simulations. The *effective capacitance* of the node is its true capacitance multiplied by the activity factor. The switching power depends on the sum of the effective capacitances of all the nodes. Activity factors can be heavily dependent on the particular task being executed.

For example, a processor in a cell phone will use more power while running video games than while displaying a calendar. CAD tools do a fine job of power estimation when given a realistic workload. Low power design involves considering and reducing each of the terms in switching power.

3.2 STATIC POWER DISSIPATION

Static power is consumed even when a chip is not switching. CMOS has replaced n-MOS processes because contention current inherent to n-MOS logic limited the number of transistors that could be integrated on one chip. Static CMOS gates have no contention current. Prior to the 90 nm node, leakage power was of concern primarily during sleep mode because it was negligible compared to dynamic power. In nanometer processes with low threshold voltages and thin gate oxides, leakage can account for as much as a third of total active power. Because sub-threshold leakage is usually the dominant source of static power, other techniques for leakage reduction are explored, including multiple threshold voltages, variable threshold voltages, and stack forcing.

4. POWER GATING

The easiest way to reduce static current during sleep mode is to turn off the power supply to the sleeping blocks. This technique is called *power gating.* The logic block receives its power from a virtual *VDD* rail, *VDDV*. When the block is active, the header switch transistors are ON, connecting *VDDV* to *VDD*. When the block goes to sleep, the header switch turns OFF, allowing *VDDV* to float and gradually sink toward 0. As this occurs, the outputs of the block may take on voltage levels in the forbidden zone. The output isolation gates force the outputs to a valid level during sleep so that they do not cause problems in downstream logic. Power gating introduces a number of design issues. The header switch requires careful sizing. It should add minimal delay to the circuit during active operation, and should have low leakage during sleep. The transition between active and sleep modes takes some time and energy, so power gating is only effective when a block is turned off long enough. When a block is gated, the state must either be saved or reset upon power-up.

On-chip power gating can use p-MOS header switch transistors or n-MOS footer switch transistors. N-MOS transistors deliver more current per unit width so they can be smaller. On the other hand, if both internal and external power gating is used, it is more consistent for both methods to cut off *VDD*. P-MOS power gating also is simpler when multiple power supplies are employed. As a practical matter, ensuring that GND is always constant reduces confusion among designers and CAD tools; this alone is enough for many projects to choose p-MOS power gating. Theoretically, it is possible to use *fine-grained power gating* applied to individual logic gates, but placing a switch in every cell has enormous area overhead.

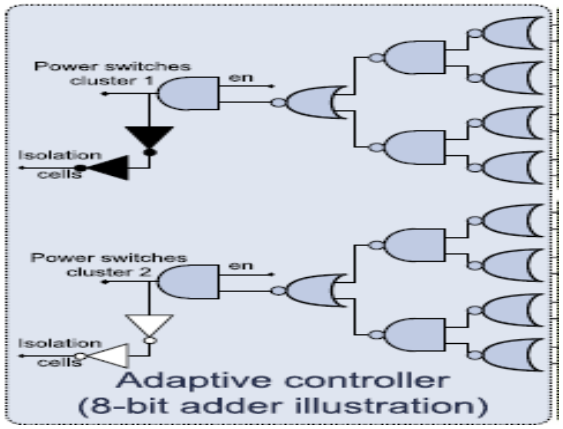

Fig.2. Adaptive power gating for 8 bit KSA, using basic logic gates and enable input.

5. RESULTS

5.1 SIMULATION RESULTS OF 4-BIT KOGGE STONE ADDER WITH AND WITHOUT POWER GATING

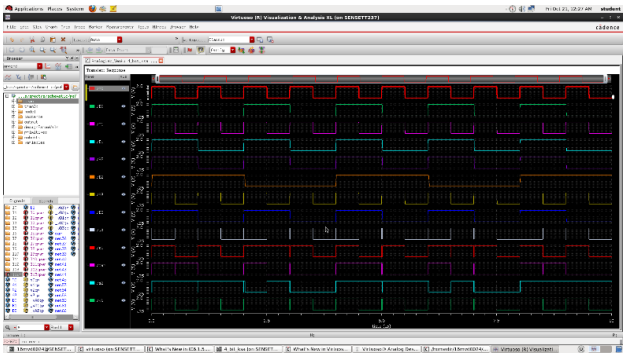

Fig-5: Simulation results without power gating

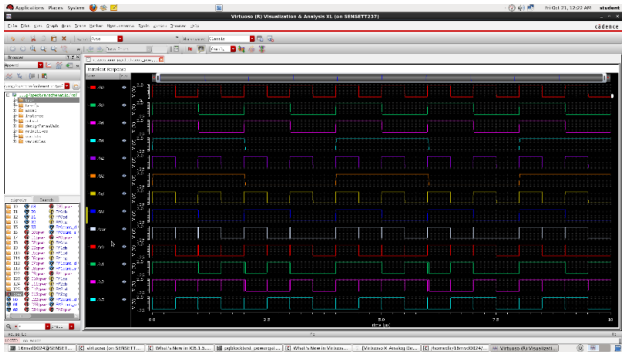

Fig-5: Simulation results with power gating

5.2 STATIC POWER CALCULATION WHEN ALL THE INPUTS ARE ZERO

From the report the static Power dissipation without adaptive power gating and with power gating is 43.33 µw and 33.33 µw. The total power reduction because of power gating = 29%.

5.3 8-BIT KOGGE STONE ADDER AND POWER GATING TECHNIQUE

RTL generation using QUARTUS software:

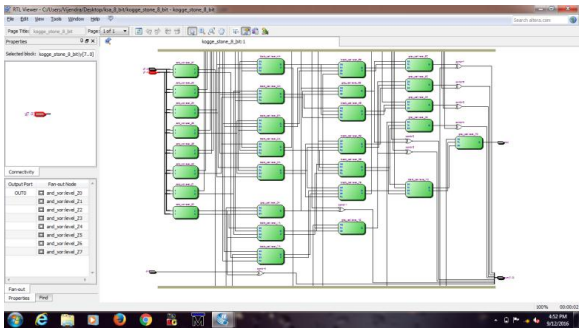

Fig-5: 8-Bit Kogge-Stone adder

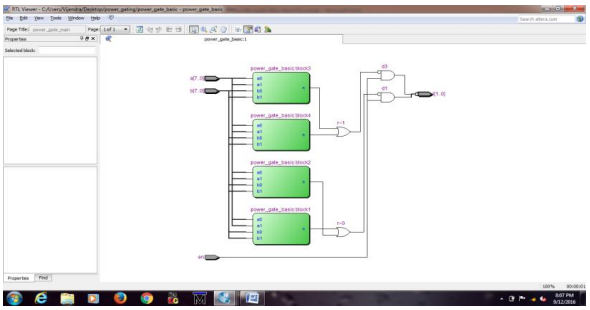

Fig-5: Power gating for 8-bits

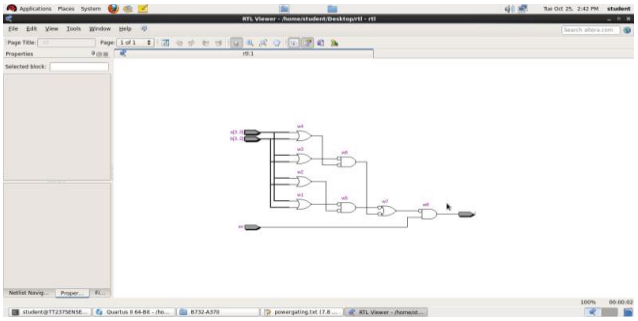

Fig-5: Power gating for 4-bits power gating circuit

5.4 STATIC POWER CALCULATION

Dynamic power and static power dissipation comparison between 4 bit and 8 bit KSA without and with power gating circuit are given below in the form for chart.

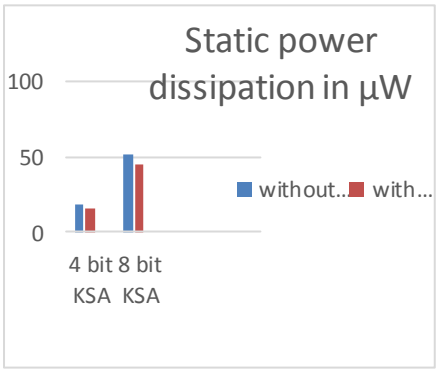

Chart-1: Static power dissipation comparison with and without power gating

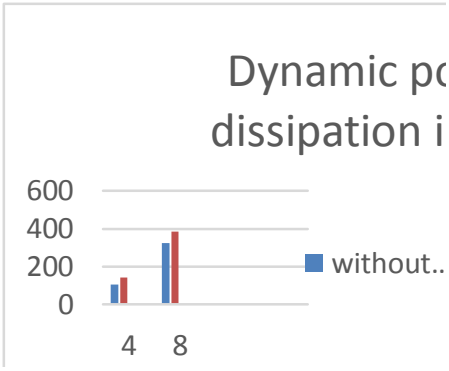

Chart-2: Dynamic power dissipation comparison with and without power gating

From above two charts we can clearly observe that as we go for higher bits, static power is reduced with the help of power gating technique, but dynamic power increases in small amount due to power gating.

6. CONCLUSION

In this work we have implemented kogge-stone adder which is of type carry look-ahead adder. Also we have implemented adaptive power gating technique, which is helpful to reduce the static power of adder. From above report the static power reduction is ((51.42-45.50)* 100) / 51.42 = 11%.

For future work, we can work on extension to this work for higher bits such as 1024-bit adder, 2048-bit adder. Similarly, we can design the layout for these higher bit adder and we can extract the parasitics for more precise results. Also we can optimize the design for higher speed adders.

REFERENCES

[1] P.M. Kogge and H.S. Stone, "Parallel Algorithm for The Efficient Solution of a General Class of Recurrence Equations," IEEE Trans. on Computers, Vol. C-22, No 8, August 1973.
[2] R.P. Brent and H.T. Kung, "A Regular Layout for Parallel Adders," in IEEE Trans. on Computers, Vol C-31, No 3, March 1982.
[3] D. Harris, "A Taxonomy of Parallel Prefix Networks," Proc. 37th Asilomar Conf. Signals Systems and Computers, pp. 2213–7, 2003.

BIOGRAPHIES

Viraj Bhusari, received his masters in VLSI design (2017-19), from Vellore Institute of Technology, Vellore, Tamilnadu. He completed several projects in Low power IC and Memory design. The author has completed his B.E in Electronics and Communications from Dr. Babasaheb Ambedkar Marathwada University, Maharashtra in 2016.