International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2020): 7.803

Impact of Low Dynamic Logic Power SoCs on Energy Optimizing Techniques

Apoorva Reddy Proddutoori

San Diego
Email: *apoorvaproddutoori[at]gmail.com*

Abstract: ***Larger capacitive loads procure the need for high - speed dynamic logic design implementations leading to higher power consumption. In search of new dynamic gate logic has been of utter importance for design and power optimization. One of the most useful utilizing has been observed through inductor - capacitor (LC) resonation reducing the switching power drastically, by transferring the energy to inductor during operations. This gate, implemented in a CMOS process, demonstrates on - chip integration feasibility with low dynamic power consumption. Collaboration across disciplines is essential for evolving device co - design and low - power technology in SoCs. Designing low - power mobile SoCs involves hardware development, integrated signal devices, circuits, and nanoscale CMOS technology. Reducing power consumption in low - voltage circuits is critical to minimize leakage in active and standby modes. Circuit layout and physical design techniques should be utilized to achieve current reduction. The paper discusses circuit techniques, physical design, voltage island technology, custom design methods, power management, and silicon implementation platform design for efficient SoCs.***

Keywords: *switching power, energy, leakage, gate logic, inductor - capacitor resonance, System - on - chips (SOCs)*

1. Introduction

The re - utilization approach of the logic gates plays vital role in re - creating the RTL logic by allowing flexibility of the rapidly growing design logic, saving both cost of computational time and area. One example of this approach is the reuse of Open cores compatible IP cores in SoC design. The advantage of using these IP addresses is mainly the reduction of project overhead, since these IP addresses are freely available. Another vital methodology is clock gating, the oldest and most commonly used power optimization technique to reduce switching or dynamic power in digital circuits. Two advanced methods have been developed to implement clock gating; software controlled clock gating also referred to as programmable clock gating, and hardware clock gating.

The ever growing for longer battery life in mobile devices continues to drive the need to implement higher levels of efficient power optimization techniques in SoCs. Power optimization techniques consist of a series of power saving methodologies implemented across various clocks and power levels. These types use a combination of hardware - style engines and software input and output applications. When tasks change, some parts of the SoC quickly are pushed into sleep mode. In order to achieve the lowest possible power, the design of the SOCs strive to keep power consumption as minimal as possible, thus reducing the latency to leverage these modes. Multiple synchronous clock domains and IP blocks are built into the SoC, adding to the complexity of the design. This environment presents unique challenges for power optimization and SoC debugging.

2. Design and Power management

a) System Clock Gating

Firstly, clocking has always been the most important and flexible parameter to rely on for power optimizations as the clock tree branches referred to as clock dividers are glitch free and help result maximum operating frequency by adjusting the ratio of the dividers (also known as switching factor).

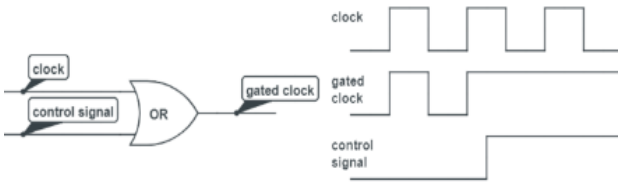

Figure 1: Gate Based Clock Gating Logic

Many clock gating implementations have been developed. That is, zero - order logic, logical logic (with AND or OR gates), turnkey logic, and a smart clock gate. The default setting is a free setting on the lock. One problem with this implementation is non - rotation (slips) of the gate clock. Unwanted errors are removed using a clock - based clock trimming setting (this is a commonly used feature). In this case the trigger value remains constant before the clock expires.

A clock gate can be integrated into a design in several stages, including logic implementation, logic assembly (RTL to gate), physical layout of the design, and clock tree assembly. At the RTL level, the clock tick is specified as a trigger function and added to the RTL code. This enable state is then converted to clock logic by the synthesizer. Nowadays, modern EDA tools support an automatic clock mechanism called smart clock. These devices identify the parts of the circuit to include the finish of the clock. In this case, CG logic is added to the design by building a special Integrated Clock Cell Gating library to enable the clock of a specific component or register.

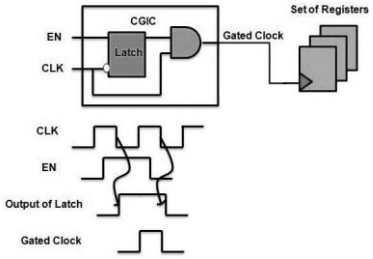

Figure 2: Latch Based Clock Gating Logic

b) Idle State Retention

Many recent times SoCs use IP blocks that require very low wake - up latency times. As part of the design logic architecture, these SoCs implemented two power lines for their IP blocks to help achieve output delay goals. Dual domain rails are made as a standard rail and a space - saving rail. Both power rails receive power in normal scenario use. If the IP block is not required, a shutdown occurs and power is removed from the normal operating rail, but power remains on the space saving rail. This architecture enables very low output latency because there is no need to load the state from a stored memory buffer address. In this implementation, the debug logic can be created to use the save state track. Using this method saves the state of the debug logic, keeping it available both during and after sleep. Debug logic should be kept to a minimum, and a power mode clock or power gate should be used to minimize power consumption using the power saver logic.

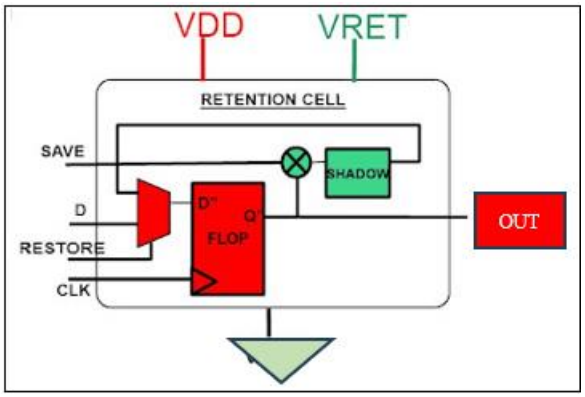

Figure 3: State Retention Logic Cell

c) Power Gating

The commonly used methodology that is very useful in verifying and debugging power control operation and shutdown conditions is decoy clocks and power clippings, also known as logic potentials. This method is particularly useful in designs where the main debugging logic is affected by power management events. A setting that controls the logic for applying power to IP blocks. This configuration setting has no effect, the power relays are removed when the control logic commands power. However, when configured correctly, the logic gate configuration prevents it from taking advantage of power. This function allows the entire input/output sequence to be followed without loss of integrity, but no loss of performance. Debug logic remains enabled and debug logic data is available during startup and shutdown sequences. If implemented, the rollback - rollback sequence is either ignored or kept as part of the flow, depending on the requirements of the debugging scenario. This mode affects the behavior of the entire system, and can be responsible for hiding problems related to power consumption. However, this is very useful for validating startup and shutdown sequences, debugging and debugging recovery and recovery sequences. In clock gate mode, the same configuration bits can be used to prevent clock gate events. You can also use a trigger logic circuit in your debug logic to catch key events and maintain the integrity of suspend events.

3. SOC Implementation

a) Inductor - Capacitor Resonation Logic

Figure 4 shows the basic switching pattern of a conventional dynamic gate with a charge capacitor C. The capacitor is first applied to V_{dd} by closing switch S1 in the previous stage. When the output variation is reached, it opens switch S1 and clamps the output to V_{dd}. If the output is high, nothing else is done. The total energy supplied by the electric motor is $CVdd^2$. Half of this is lost as resistance when charging the capacitor. Half of the energy is stored in output capacitor C. When the output is low, switch S2 must be closed to send the node to ground during evaluation. The energy in C is dissipated as heat as it flows. The power loss is the frequency f times $CVdd^2$. So with a 1 Gp/s logic switch, you need 1 mW to achieve a 1 V swing across a 1 pF capacitor. For SoCs with nanofarad switching capabilities, power is in watts.

b) Dynamic Operation

The dynamic strategy of power optimization using LC circuit logic includes the well - known load scaling (LS) strategy, also known as the slot method. These strategies attempt to run the system at its optimal operating point with minimal downtime without balancing current or pending processing requests against the operating point of the system. The scaling strategy is simple and effective for workloads. In a test configuration that checks load scaling, each voltage/ frequency level is assigned a voltage level symbolizing a predetermined operating state. The scaling logic uses simple heuristics to set policies based on system load over time. All policies also include delayed extensions. This helps reduce inefficiencies caused by selecting lower control points during load expansion.

c) Software - Firmware Debug

Below depicts the study of different SW/FW debugging opportunities;

- Initial Boot SW/FW Debugging: This part of SW/FW FW Debugging is related to platform initialization, it can be a hardware image engine, ROM code, etc. other microsystems. Depending on the implementation, this part of the startup process maybe invisible to the end user.
- Runtime SW/FW Debugging: This part of SW/FW debugging occurs while it is still running under the control of the OS.
- Low Power SW/FW Debugging: This part of the SW/FW debugger is actively switching to low power mode. It also includes verification and debugging in low power situations.

The first boot phase is called the "early boot" phase because it is the phase in which the system is turned on. The next step is to boot the system and run the OS. If the OS is running, the system is idle, or does not require an IP to run, it may be

suppressed. The process of putting the system into low power mode allows it to go into low power mode and back to normal operation. Debug mode with low power during this startup/shutdown is called the SW/FW shutdown phase.

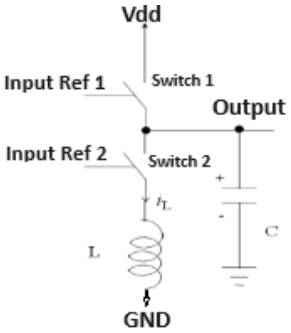

Figure 4: LC Resonation Logic

d) Power vs Area

The idle state under retention as known consumes very minimal current from the main voltage supply Vdd, this form of energy is supplied through the inductor from LC resonant logic. At higher operating speeds, the inductor values drop to as low as square of the clock frequency. Hence, the inductance scales inversely with the capacitance load. To retain the margin of inductor - capacitor components, the resonance frequency is adjusted to be as high as twice the workload data rate.

4. Conclusions

A dynamic logic gate, termed LC resonator, has been shown to consume less than half of the traditional switching or dynamic power, with energy recovery and adiabatic charging and discharging saving over 30% of energy. Simulations indicate that such a gateway can be achieved with current technology, especially in SoCs handling on - chip or off - chip loads in nano - farads capacitance. The use of a resonant inductor in the gate increases operating time with smaller values, and optimizations of system - level features are being developed for future work. The convergence of digital, analog, power management, and RF design technologies is leading to integrators with resonant behavior in high - speed digital gates. Low - power mobile electronics advancements have enabled general - purpose computing, with power optimization techniques focusing on sleep modes and hardware power management techniques. The silicon manufacturing industry is under pressure to meet consumer demands for longer battery life, leading to the development of aggressive SoC and Power Management techniques.

References

[1] Faroudja ABID and Nouma Izeboudjen, " Low Power Methodology for Wishbone Compatible IP cores based SoC design", 7th SEMINAR ON DETECTION SYSTEMS: ARCHITECTURES AND TECHNOLOGIES, Feb 2017
[2] Sankaran Menon1, Chinna Prudvi1, Rolf Kuehnis1, Sukhbinder Singh Takhar1, Spencer Millican2, Eric Rentschler3, Pandy Kalimuthu4, Preeti Ranjan Panda5, Priyadarsan Patra6, Ashish Gupta7, " Techniques for Debug of Low Power SoCs", 20th International Workshop on Microprocessor/SoC Test, Security and Verification (MTV), 2019
[3] Bishop Brock and Karthick Rajamani, "Dynamic Power Management for Embedded Systems", IEEE, March 2003
[4] Yun - Tae Lee, "Low Power SOC in Deep - Submicron Era", IEEE, 2003
[5] Wei Hwang, "New Trends in Low Power SoC Design Technologies", IEEE, 2003
[6] Ignatius Bezzam, C. Mathiazhagan and Shoba Krishnan, "Low power SoCs with Resonant Dynamic Logic using Inductors for Energy Recovery", IEEE, 2012
[7] https: //astep - to - lowpower. blogspot. com/2008/05/retention - register. html