Low Power Design Strategies For SoCs

Anjali Gupta¹, Atikant Shrivastava², Parag Parandkar³ and Sumant Katiyal⁴

¹,³ChameliDevi Institute of Technology and Management, ²IES,IPS Academay, Indore, ⁴School of Electronics, DAVV, Indore

E-Mail:¹anjaligupta1128@gmail.com, ²atikant@gmail.com, ³p_paragp@yahoo.com, sumant578@yahoo.com

ABSTRACT
In every application, power management must take precedence, whether to reduce energy use or to minimize heat dissipation to lower cooling and packaging costs. Power management is becoming an increasingly urgent problem for almost every category of design, as power density, measured in watts per square millimeter, rises at an alarming rate. From a chip-engineering perspective, effective energy management for a SoC (System-on-a-chip) must be built into the design starting at the architecture stage; and low-power techniques need to be employed at every stage of the design, from RTL (Register Transfer Level) to GDSII. Power needs to be considered at the very early stages of a design, when the opportunity to save power is at a maximum. At the same time, making a design extremely power efficient results in trading off area and/or timing. A balanced approach is warranted when it comes to power optimization that spans across the entire RTL-to-GDS flow. Today, with the dramatic increase in power requirements, power management is becoming an important aspect for almost all categories of design. The popularity of cell phones, MP3 players, PDAs and other handheld devices has made extended battery life a major selling point for portable systems. Increasing awareness of global warming and green technology is also driving the demand for low-power designs. The rising energy costs and geopolitical forces are causing everyone to rethink how they consume power. As a result, semiconductor and EDA vendors are developing new methodologies and tools that allow them squeeze as much as they can from tightening power budgets. This paper survey recent research on CMOS low voltage and low power ICs to increase the performance. Higher performance was achieved through the use of new architectures (e.g. super scalar and super pipelined); integrating more and more system functions on the SoC increasing the processor’s speed and increasing the processor's ability to perform computations. The trend in the large and cost sensitive, embedded processor market segment is toward higher frequencies. Today one in four IC designs fail due to power-related issues. Power management techniques are considered a critical part of the SoC design flow.

2. LOW POWER PARAMETERS
Power consumption is the calculated as product of operating voltage (Vcc) and the current consumption (Icc). But, generally current consumption is the only measure considered when describing the power characteristics of a chip. This may be treated as inaccurate as decreasing the operating voltage directly reduces the current consumption and thus the overall power. Current consumption increases directly with the system clock frequency so keeping the system clock as low as possible is vital for keeping power consumption down. The clock frequency is affected by a number of factors that include the microprocessor/ microcontroller’s peripheral set as well as the architecture and the instruction set. RISC microcontrollers typically execute in a single clock cycle but some architecture divide the clock down in the same way as CISC architectures do before feeding it to the CPU.

This situation leads to confusion about what clock frequency is really required to execute the target application. Temperature is another factor. Since higher temperatures lead to higher power consumption, designers should always consider the power level for worst-case temperatures. The parameters can be explained as-

- Reducing chip capacitance through process scaling: This approach to reduce the power is very expensive and has a very low return on investment (ROI).
- Reducing voltage: This approach provides the maximum relative reduction in power but it is complex, difficult to achieve and requires moving the memory as well as other systems related design industries to a new voltage standard. The ROI is low.
- Employing better architectural and circuit design techniques: This approach promises to be the most successful because the investment to reduce power by design is relatively small in comparison to the other two approaches.
3. WHERE POWER IS CONSUMED

Power dissipation within a device can be broken down into two basic types -- dynamic power consumption based on the switching activity and the static power consumption based on leakage.

The dynamic power consumption can be broken down into the switch power due to the charging and discharging of capacitive loads driven by the circuit (including wire capacitance and input loads), and short circuit power occurring momentarily during switching when pairs of PMOS and NMOS transistors are conducting simultaneously.

The leakage power can also be broken down into the following factors. Current flowing through the reverse biased diode formed between the diffusion regions and the substrate (I_{diode}). Another is the current flowing through transistors that are not conducting, tunneling through the gate oxide (I_{subthreshold}). Leakage of the device is dramatically impacted by the operating temperature. Therefore, as the chip heats up, the static power dissipation increases exponentially.

![Figure 1: Static and dynamic power consumption](image)

Leakage current within a 130nm process with a 0.7V threshold gives approximately 10-20 pA per transistor. If threshold voltage is reduced to 0.3V in the same process, the leakage current raises very high to 10-20 nA per transistor, increasing exponentially in smaller geometries.

The need to reduce power consumption has become more critical as larger, faster ICs move into portable applications. As a result techniques for managing power throughout the design flow are evolving to assure that all parts of the product receive power properly and efficiently and that the product is reliable. However, the design complexity has increased in such a way that the design problem has essentially changed from an optimization of the design in two dimensions, i.e. performance and area, to optimizing a three-dimensional problem, i.e., performance, area, and power as shown in the Fig. 2.

![Figure 2: Three dimensional optimization](image)

The best power management decisions are made at the system level during system design. At this point in the development cycle, system designers make initial decisions about the end-product power requirements and distribution. That involves determining the overall power budget, making critical hardware/software trade-offs, and working with designers to select the proper process technology for semiconductor fabrication, such as gate length and multi-voltage/multi-threshold. Electronic Design Automation (EDA) gives system designers the design and verification environment they need to achieve power/performance estimation at the system level [1]. Such system-level decisions drive all subsequent software and semiconductor requirements.

Now, the requirements that fully define the low power budget for an SoC architecture are described. These may be derived from some form of standards-based requirements limiting current draw under certain conditions, or alternately to extend the life of the battery in the case of a mobile application. The solution for the target applications will differ in how the device is controlled and architected. Once the requirements are clearly defined, various architectures are explored and potential trade-offs are determined.

By starting at the highest level of abstraction, where the potential for maximum savings are, and further refining this through the levels of design abstraction, power savings can be continually driven towards the target budget as shown in Fig.3.

![Figure 3: Prospects for Low power operability at different levels of abstraction](image)

In finalizing the SoC architecture, a number of considerations and decisions will need to be made at various stages of the design abstraction to reach the optimal solution. These includes requirements as system performance, processor and other IP selection, new modules to be designed, target technology, the number of power domains to be considered, target clock frequencies, clock distribution and structure, I/O requirements, memory requirements, analog features and voltage regulation[2]. All of these are contributors to the power budget and therefore can be targeted for power minimization to achieve the low power goal.

For bringing all the pieces of the architecture together, the global control and clock features will be considered next, that
can be used to reduce the overall power of the system. A design is likely to have many modes of operation for various application demands, such as startup, active, standby, idle, and power down. In some cases multiple levels of these modes will be used to achieve the best overall power management strategy. These modes tend to be generally controlled by a combination of software and hardware features, and need to be planned into the system development from a very early stage of the design process.

One of the best ways to save as much power as possible is to scale the voltage to the optimal levels for the required performance. The impact of reducing voltage levels, however, is to increase the gate delay, and beyond certain level that becomes impractical. The ideal solution is to have varying modes of operation, with the target to power down as much of the design as possible for the given application, reducing both dynamic and leakage power [4]. In standby mode, for example, the minimum amount of logic required should be maintained on a low voltage domain to bring the device out of this state on demand from some external event, then moving through the modes of operation to the required performance level.

This solution provides the maximum saving, but it also carries the largest overhead in terms of complexity. These range through the considerations for on or off chip switching regulation, power domain isolation, performance impact of delays associated with the switching and resumption of stable power, and potential loss of state for flip-flops and memory requiring save and restore routines, along with all the additional associated test and verification requirements.

In developing this type of implementation, consideration needs to be given to all of the above items and the feasibility of the management of the periods of time where this can be realistically achieved.

The next level of consideration, after defining the voltage partitioning and scaling, should be the system level clock architecture and methods of controlling frequency and associated switching levels. It doesn't address leakage (static) power consumption, but this method goes a long way towards reducing the dynamic power consumption of the device.

It is not uncommon for a design to have the clock distribution and clocked elements consume over 50% of the total power consumption of the device [5]. The scaling of frequency may be directly proportional to any voltage scaling if implemented to meet the required system level performance.

In a given idle or sleep mode, all the non-dependent modules can be gated off completely from the route of the tree, eliminating the switching in both the clock distribution and logic within these parts of the design. The use of multiple clock domains, frequency scaling and frequency phasing to reduce peak power can all be managed from the central level of distribution.

In all the above aspects of implementation for the defined system clock architecture, detailed consideration is required to avoid all forms of clock glitching, the additional overheads associated with multiple domains in terms of functional test, skew control, design for test considerations and timing closure implications.

Once the design architecture has been captured, the RTL code can now be targeted towards a low power synthesis flow, automatically trading power alongside the generally accepted performance and area constraints [3]. The main features targeted by the tools include multiple threshold leakage optimization, multiple supply voltage domains, local latch based clock gating, de-clone and re-clone restructuring, operand isolation, and gate level power optimization.

For multiple threshold leakage optimizations, generally up to three versions of the targeted library are used: Low $V_{th}$ (fast, high leakage), Standard $V_{th}$, and High $V_{th}$ (slower, low leakage). The tool will target to use as many of the high threshold cells as possible, while maintaining the timing constraints, only utilizing the low threshold cells for critical paths. Selecting and targeting the appropriate library and characterization for the application performance requirements are a key consideration that should be addressed early on in the design process.

To support multiple voltage domains, additional characterized libraries for the targeted voltages are required [5]. These may also include multiple threshold variants within them. The savings in costs in terms of power will obviously relate to the quadratic voltage scaling effect. Along with a consistent and supporting tool flow, managing the domain partitioning requires careful design consideration in the early stages of development, and close integration between front end design and layout processing to support all of the above methodology. The user can generally define the range of flops to be driven from a single clock gate to avoid any unnecessary imbalance in the clock distribution network [6].

The clock gating implementation is shown in Fig. 5.
The operand isolation step automatically identifies and shuts down data path elements and hierarchical combinatorial modules with a common control signal. The tool only partially commits to the restructuring, to allow optimal timing and power tradeoffs.

Classical gate level optimization resizes cells, performs pin swapping, removes unnecessary buffering, merges gates, adds buffers to reduce slew and restructures logic to provide the best possible power optimization. However, the majority of these steps are also repeated in the physical domain with real placement and wire length constraints.

Comparative numbers between a base line flow and that of a low power synthesis flow employing the above techniques show that an embedded processor device in a 90 nm technology of approximately 650K gates can achieve savings of greater than 40% for both dynamic switching power and leakage power [7].

4. CONCLUSION
The trends outlined in this paper demonstrate that the on-chip power related challenges associated with designing at the latest technology nodes will remain premier to the designers. The available data indicates that the factors impacting low power design considerations will become an intervening concern for architects and designers of the next generation of SoCs. This will require a proportional incremental step in the design tools and methodologies needed to exploit the full potential of the future technology. Alongside with proposed silicon technology advances, engineering productivity needs to increase to further resolve and reduce the on-chip power challenge. This can only be achieved through new methodologies, tools and design concepts. These may involve working towards more power conservative clock structures, such as low swing flip-flops, double edge triggered flip-flops or conditional flip-flops, giving clock on demand where the internal clock is only activated when the input data will have the effect of changing the output data. Alternately, moves to more fundamental approaches such as complete asynchronous design techniques, or more intelligent clocking structures, will be required to support the more traditional concepts. Tools to support these new methodologies will be required, fully integrating the process from high to low level abstractions of design and analysis.

5. REFERENCES