SoC Power Management with a NoC

Jonah Probell, Arteris

Introduction

As the importance of power minimization continues to grow, new techniques are being added to industry standard chip design methodologies. A network-on-chip (NoC) touches all stages of the chip design process. Below I will describe the techniques with the most significant effect on power consumption, and how they can be accommodated in a NoC design.

Data path optimization and performance modeling

A NoC is made of links between various network units. The data path width of each link can be independently configured by using serialization adapters. As a result, all buses and logic within a NoC can be as small as possible within the constraint of their bandwidth requirements.

A scalable network avoids the need for building separate switches and bridges to scale interconnectivity. A simple protocol minimizes the need for context management buffers within interface units. Furthermore, a configurable NoC allows the use of register files, SRAMs, or flip-flops for buffering, as is optimal, for each buffer location. These flexibilities minimize area and therefore leakage power consumption.

Overdesign wastes power. Accurate and insightful tools to analyze performance and pinpoint limitations allows thorough exploration of possible design trade-offs and a near-optimal result.

Voltage/power/clock domain partitioning

This is key to power management in any mobile or heat-constrained system. A NoC is what connects IPs. Therefore, it is the sensible mechanism to separate IPs into different power, clock, and voltage domains. A NoC enable all blocks to be power-controlled individually. This makes software-controlled fine granularity block partitioning easy and safe.

Finer granularity partitioning enables a greater portion of the chip to be turned off in different use cases. For example, when rendering scenes with few polygons or at low resolution, only a portion of an array of GPUs need be powered up at full voltage and clock frequency. In some chips the sub-components of GPU engines and other IPs are powered down using hierarchical domains.

Power disconnect protocol

To implement power-down or clock-off of domains a standard protocol is used for communication between a SoC level power management unit (PMU) and local power disconnect units within the NoC. The protocol safely handles fencing (responding safely to incoming request to powered-down targets) and draining (the completion of pending transactions to and from IPs in the domain to be powered down) while leaving software no opportunity to create a deadlock. A NoC configuration tool simplifies and automates placing power and clock disconnects within the internal transport topology of a NoC.

A NoC supports multiple alternate behaviors for accesses to powered down or clocked off domains at socket or transport power/clock domain crossings. These include null responses, error responses that generate exceptions, and wakeup on demand. Wakeup on demand allows simplification of power management software since the system PMU and NoC power disconnect units work together to automatically wake-up the required domains only when they are accessed.

Asynchronous crossing

A NoC configuration tool simplifies and automates the placement of asynchronous clock domain crossings within a NoC topology. This allows all IPs and units of the NoC to run at their optimal frequency and voltage, no matter how many there are.

Clock gating

The clock tree toggles more than any logic in a chip. It is one of the largest power sinks. This is particularly true for a NoC because, since it is physically distributed throughout the chip, a NoC tends to have long clock nets and relatively high wire capacitance.

A NoC supports independent unit-level clock gating. If there is no traffic through the unit then its clock is gated off, cycle-by-cycle. Even in a heavily loaded system, most of a NoC is clock gated off most of the time. Compared to full switches, which must have a clock if any traffic might be present, a NoC consumes an order of magnitude less clock tree power in real systems.

Unit level clock gating in a NoC is complementary to the per-flop clock gating implemented by synthesis tools.

Automation

For a specified architecture, a NoC configuration tool automatically generates power description files (CPF, UPF, UPF 2.0) so that synthesis tools can automatically insert level shifters, isolation cells, and retention flops. This helps designers with automation of power management physical design.

Partial retention

Partial retention is valuable for saving power. Retention flops are big and require an always-on power supply, so it is good to minimize their number. A NoC supports partial retention to minimize the number of retention flops needed.

Late stage changes

Last minute changes to a chip design, such as to fix unanticipated inter-block timing violations, often require forgoing opportunities for power optimization in the back end process. Since a NoC touches all IPs in a chip, a small change could require resynthesizing large numbers of already-optimized blocks. A modular NoC uses a simple physical interface protocol between all units. As a result, timing violations can be fixed by the insertion of small pipeline stage units between blocks. This can be done without the need to resynthesize any blocks.

Furthermore, due to the simplicity of the internal protocols of modular NoC units, late stage ECOs are possible that otherwise would not be.

Performance probe used as feedback for DVFS

In advanced SoC implementations, probes within a NoC are used to monitor traffic statistics. These are used by software as feedback to adjust dynamic voltage frequency scaling (DVFS). For example, NoC traffic from a wireless interface IP can be monitored and the probe can assert an interrupt based on programmable thresholds. When a high threshold is reached, the power management software goes to a higher power operating mode. When a low threshold is reached, the power management software goes to a lower power operating mode.

Power aware task scheduling

In other advanced SoC implementations, statistics probes are used to gauge the amount of traffic generated by different schedule-flexible tasks. Those tasks are then scheduled based on the phase of cyclical schedule-invariant tasks with. Latency meters are used to monitor the effectiveness of the scheduling, providing feedback for real-time algorithm tuning.