Pushing the Performance Boundaries of ARM Cortex-M Processors for Future Embedded Design

By Ravi Andrew and Madhuparna Datta, Cadence Design Systems

One of the toughest challenges in the implementation of any processors is balancing the need for the highest performance with the conflicting demands for lowest possible power and area. Inevitably, there is a tradeoff between power, performance, and area (PPA). This paper examines two unique challenges for design automation methodologies in the new ARM® Cortex®-M processor: how to get maximum performance while designing for a set power budget and how to get maximum power savings while optimizing for a set target frequency.

Contents

Introduction

The ARM Cortex-M7 processor is the latest embedded processor by ARM specifically developed to address digital signal control markets that demand an efficient, easy-to-use blend of control and signal processing capabilities. The ARM Cortex-M7 processor has been designed with a large variety of highly efficient signal processing features, which demands very power-efficient design.

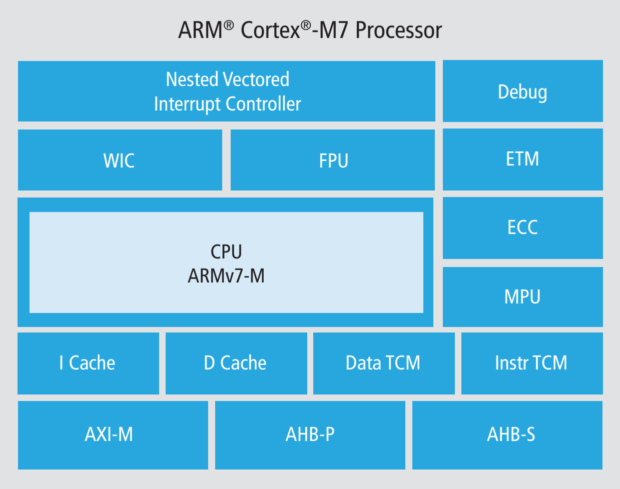

Figure 1: ARM Cortex-M7 Block Diagram

The energy-efficient, easy-to-use microprocessors in the ARM Cortex-M series have received a large amount of attention recently as portable and wireless/embedded applications have gained market share. In high-performance designs, power has become an issue since at those frequencies power dissipation can easily reach several tens of watts. The efficient handling of these power levels requires complex heat dissipation techniques at the system level, ultimately resulting in higher costs and potential reliability issues. In this section, we will isolate the different components of power consumption on a chip to demonstrate why power has become a significant issue. The remaining sections will discuss how we approached this problem and resolved it using Cadence® implementation tools, along with other design techniques.

We began the project with the objective of addressing two simultaneous challenges:

1. Reach, as fast as possible, a performance level with optimal power (AFAP)

2. Reduce power to the minimum for a lower frequency scenario (MinPower)

Before getting into the details of how we achieved the desired frequency and power numbers, let's first examine the components which contribute to dynamic power and the factors which gate the frequency push. This experiment has been conducted on the ARM Cortex-M7 processor. The ARM Cortex-M7 processor has achieved 5 CoreMark/MHz – 2000 CoreMark* in 40LP and typical 2X digital signal processing (DSP) performance of the ARM Cortex-M4 processor.

Dynamic power components

In high-performance microprocessors, there are several key reasons which are causing a rise in power dissipation. First, the presence of a large number of devices and wires integrated on a big chip results in an overall increase in the total capacitance of the design. Second, the drive for higher performance leads to increasing clock frequencies, and dynamic power is directly proportional to the rate of charging capacitances (in other words, the clock frequency). A third reason that may lead to higher power consumption is an inefficient use of gates. The total switching device capacitance consists of gate oxide capacitance, overlap capacitance, and junction capacitance. In addition, we consider the impact of internal nodes of a complex logic gate. For example, the junction capacitance of the series-connected NMOS transistors in a NAND gate contributes to the total switching capacitance, although it does not appear at the output node. Dynamic power is consumed when a gate switches. However, interest has risen in the physical design area, to make better use of the available gates by increasing the ratio of clock cycles when a gate actually switches. This increased device activity would also lead to rising power consumption. Dynamic power is the largest component of total chip power consumption (the other components are short-circuit power and leakage power). It occurs as a result of charging capacitive loads at the output of gates. These capacitive loads are in the form of wiring capacitance, junction capacitance, and the input (gate) capacitance of the fan-out gates. Since leakage is <2% of total power, the focus of this collaboration was only on **dynamic power**.

The expression for dynamic power is:

$P_{\text{dynamic}} = \alpha C V_{dd}^2 f$……………………………………(1)

In (1), C denotes the capacitance being charged/discharged, V_{dd} is the supply voltage, f is the frequency of operation, and α is the switching activity factor. This expression assumes that the output load experiences a full voltage swing of V_{dd}. If this is not the case, and there are circuits that take advantage of this fact, (1) becomes proportional to (V_{dd} * V_{swing}). A brief discussion of the switching factor α is in order at this point. The switching factor is defined in this model as the probability of a gate experiencing an output low-to-high transition in an arbitrary clock cycle. For instance, a clock buffer sees both a low-to-high and a high-to-low transition in each clock cycle. Therefore, α for a clock signal is 1, as there is unity probability that the buffer will have an energy-consuming transition in a given cycle. Fortunately, most circuits have activity factors much smaller than 1. Some typical values for logic might be about 0.5 for data path logic and 0.03 to 0.05 for control logic. In most instances we will use a default value of 0.15 for α, which is in keeping with values reported in the literature for static CMOS designs [1,2,3]. Notable exceptions to this assumption will be in cache memories, where read/write operations take place nearly every cycle, and clock-related circuits.

Here are five key components of dynamic power consumption and how we addressed a few of these components:

- Standard cell logic and local wiring
- Global interconnect (mainly busses, inter-modular routing, and other control)
- Global clock distribution (drivers + interconnect + sequential elements)
- Memory (on-chip caches) — this is constant in our case
- I/Os (drivers + off-chip capacitive loads) — this is constant in our case

Timing closure components

One fundamental issue of timing closure is the modeling of physical overcrowding. The problem involves, among other factors, the representation and the handling of layout issues. These issues include placement congestion, overlapping of arbitrary-shaped components, routing congestion due to power/ground, clock distribution, signal interconnect, prefixed wires over components, and forbidden regions of engineering concerns. While a clean and universal mathematical model of physical constraint remains open, we tend to formulate the layout problem using multiple constraints with sophisticated details that complicate the implementation. We need to consider multiple constraints with a unified objective function for a timing-closure design process. This is essential because many constraints are mutually conflicting if we view and handle their effects only on the surface. For example, to ease the routing congestion of a local area, we tend to distribute components out of the area to leave more room for routing. However, for multi-layer routing technology, eliminating components does not save much on routing area. The spreading of components actually increases the wire length and demands more routing space. The resultant effect can have a negative impact on the goals of the original design. In fact, the timing can become much worse. Consequently, we need an intelligent operation that identifies both the component to move out and the component to move in to improve the design.

Accurately predicting the detail routed signal-integrity (SI) effects, before the detail routing happens, and its impact to timing is of key interest. This is because a reasonable misprediction of timing before the detail route would create timing jumps after the routing is done. Historically, designs for which it is tough to close timing have relied solely on post-route optimization to salvage setup/hold timing. With the advent of “in-route optimization”, timing closure has been bridged earlier during the routing step itself using track assignment. In addition, if we can reduce the wire lengths and make good judgment calls based on the timing profiles, we can find opportunities to further reduce power. This paper will walk through the Cadence digital implementation flow and new tool options used to generate performance benefits for the design. The paper will also discuss the flow and tool changes that were done to get the best performance and power efficiency out of the ARM Cortex-M7 processor implementation.

Better Placement and Reduced Wirelength for Better Timing and Lower Power

As discussed in the introduction, wire capacitance and gate capacitance are among the key factors that impact dynamic power, while also affecting wire delays. While evaluating the floorplan and cell placement, it was noticed that the floorplan size was bigger than needed and the cell placement density was uniform. These two aspects could lead to spreading out of cells, resulting in longer wirelength and higher clock latencies. In order to improve the placement densities, certain portions of the design were soft-blocked, and the standard cell densities were kept above 75%.

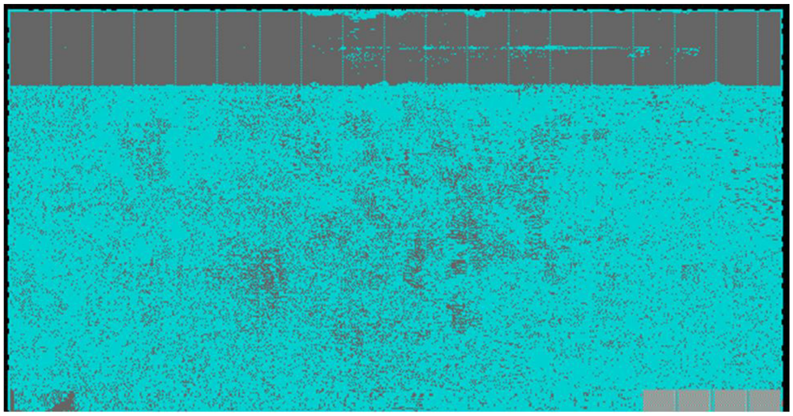

Figure 2: Soft-Blocked Floorplan

Standard cell placement plays a vital role. If the placement is done right, it will eventually pay off in terms of better Quality of Results (QoR) and wirelength reduction. If the placement algorithms can take into account some of the power dissipation-related issues, like reducing the wirelength and considering overall slack profile of the design, and also make the right moves during placement, this would tremendously improve the above mentioned aspect. This is the core principle behind the “Giga Place” placement engine. The Giga Place engine, available in Cadence Encounter® Digital Implementation System 14.1, helps place the cells in a timing-driven mode by building up the slack profile of the paths and performing the placement adjustments based on these timing slacks. We have introduced this new placement engine on the ARM Cortex-M7 design and seen good improvements on the overall wirelength and Total Negative Slack (TNS).

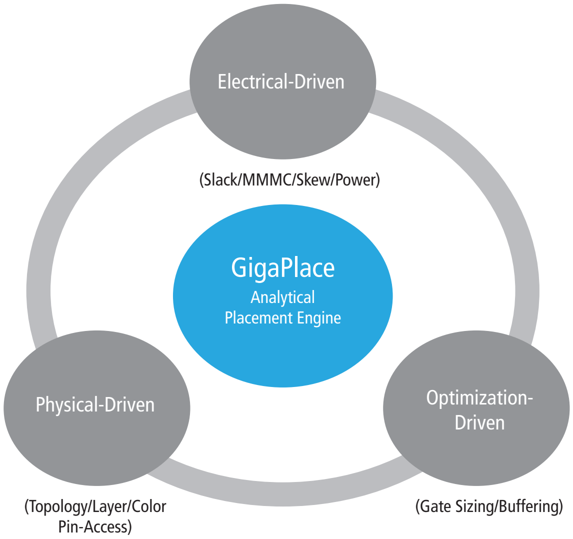

Figure 3: "GigaPlace" Placement Engine

With a reduced floorplan and by removing the uniform placement and utilizing the new GigaPlace technologies, we were able to reduce the wirelength significantly. This helped push the frequency as well as reduce the power. But, there were still more opportunities available to further benefit the frequency and dynamic power targets.

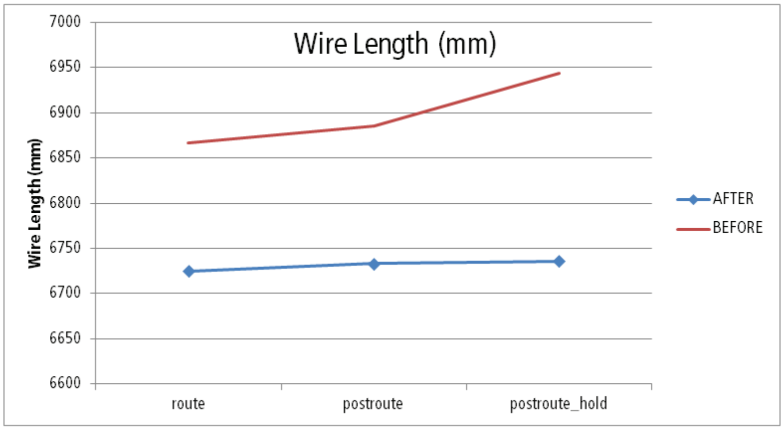

Figure 4: Wirelength Reduced with "GigaPlace and Soft-Blocked" Placement

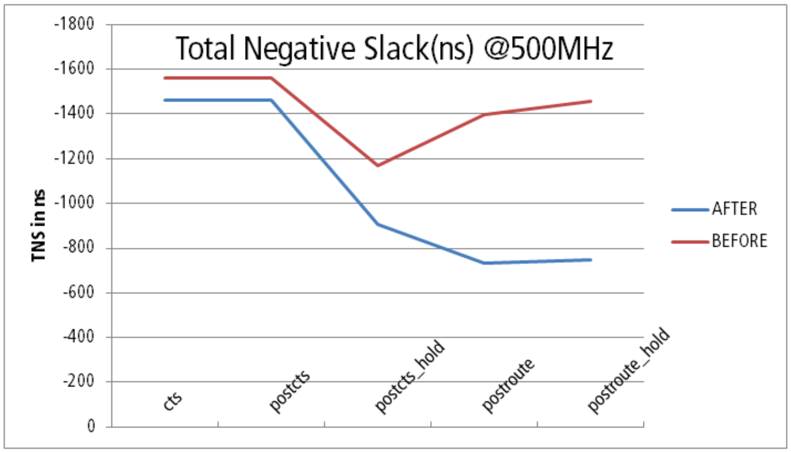

Figure 5: Total Negative Slack (ns) Chart

In-Route Optimization: SI-Aware Optimization Before Routing to Achieve Final Frequency Target

"In-route optimization" for timing optimization happens before routing begins. This is a very close representation of the real routes, which does not account for the DRC fixes and the leaf-cell pin access. This enables us to get an accurate view of timing/SI and make bigger changes without disrupting the routes. These changes are then committed to a full detail route. In-route optimization technology utilizes an internal extraction engine for more effective RC modeling. The timing QoR improvement observed after post-route optimization was significant at the expense of a slight runtime increase (currently observed at only 2%). A successful usage of an internal extraction model during in-route optimization helped reduce the timing divergence seen as we go from the pre-route to the post-route stage. This optimization technology pushed the design to achieve the targeted frequency.

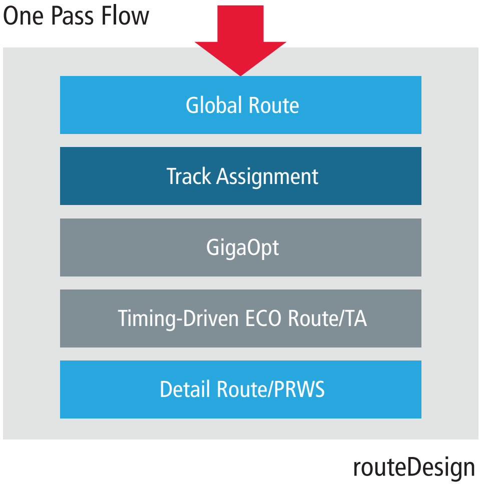

Figure 6: In-Route Optimization Flow Chart

Design Changes and Further Dynamic Power Reduction

In the majority of present-day electronic design automation (EDA) tools, timing closure is the top priority and, hence, many of these tools make the trade-off to give priority to timing. However, opportunities exist to reduce area and gate capacitance by swapping cells to lower gate cap cells and by reducing the wirelength. To address the dynamic power reduction in the design, three major sets of experiments were done to examine the above aspects.

In the first set of experiments, two main tool features were used in the process of reducing dynamic power. These were the introduction of the “dynamic power optimization engine” along with the “area reclaim” feature in the post-route stage. These options helped save 5% of dynamic power @400MHz and enabled us to nearly halve the gap that earlier existed between the actual and desired power target.

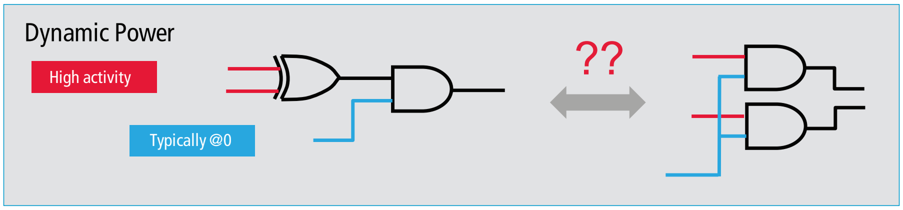

Figure 7: Example of Power Optimization

In the second set of experiments, the floorplan was soft-blocked by 100 microns to reduce the wirelength. This was discussed in detail in an earlier section. This floorplan shrink resulted in:

- Increasing the density from ~76% to 85%
- Wirelength reduction by 5.1% – post route
- Area (with combo of #1 and shrink) shrinkage by ~4% – post route

This helped saved an additional 2% @400MHz, and the impact was similar across the frequency sweep.

The third set of experiments was related to design changes where flop sizes were downsized to a minimum at pre_cts opt and the remaining flops of higher drive strengths were set to “don’t use”. This helped to further reduce the sequential power. An important point to note is that the combinational power did not increase significantly. After we introduced the above technique, we were able to reduce power significantly, as shown in the charts below.

Results

By using these latest tool technologies and design techniques, we were able to achieve 10% better frequency and reduced the dynamic power by 10%. Results are shown here based on the 400MHz and 200MHz for the dynamic power reduction.

Run Details	Relative Dynamic Power Reduction
400MHz - RC/EDI 13.2	100
400MHz – RC/EDI 14.1	96
400MHz - 14.1 + PowerOpt	91
200MHz - 14.1	75
200MHz - 14.1 + PowerOpt	71
200MHz - 14.1 + PowerOpt + GigaPlace	67
200MHz - 14.1 + PowerOpt + GigaPlace + Relax Clock Skew	62

Table 1: Dynamic Power Reduction Results

The joint ARM/Cadence work started with addressing challenges at two points/scenarios on the PPA curve:

1. Frequency focus with optimal power (400MHz)

2. Lowest power at reduced frequency (200MHz)

For scenario #1, out of box 14.1 allowed us to reach 400MHz. With the use of PowerOpt technology, available in Encounter Digital Implementation System 14.1, we were able to reduce power to an optimal number. For scenario #2, additional use of GigaPlace technology and inherently better SI management allowing relaxed clock slew, and much higher power reduction at 200MHz was possible. With the combination of ARM design techniques and Cadence tool features, we were able to show 38% dynamic power reduction (for standard cells) going from 400MHz – 13.2-based run to 200MHz – 14.2 best power recipe run.

Summary

Reducing the wirelength and slack profile-based placement, and predicting the detailed routing impact in the early phase of the design, are important aspects to improve the performance and reduce the dynamic power consumption in designs. Tools perform better when given the right floorplan along with the proper directives at appropriate places. With a combination of design changes, advanced tools, and engineering expertise, today's physical design engineers have the means to thoroughly address the challenges associated with timing closure while keeping the dynamic power consumption of the designs low.

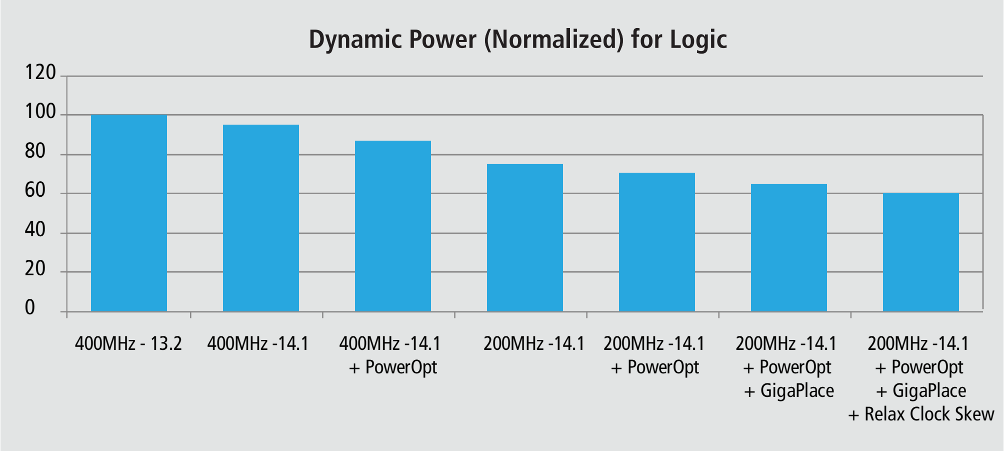

Figure 8: Dynamic Power (Normalized) for Logic

Several months of collaborative work between ARM and Cadence, driven by many trials, have led to optimized PPA results. Cadence tools – Encounter RTL Compiler/Encounter Digital Implementation System 14.1 – have produced better results out of box compared to Encounter RTL Compiler/Encounter Digital Implementation System 13.x. The continuous refinement of the flow along with design techniques such as floorplan reduction and clock slew relaxation allowed a 38% dynamic power reduction. The ARM/Cadence implementation Reference Methodology (iRM) flow uses a similar recipe for both scenarios: lowest power (MinP) and highest frequency (AFAP).

References

[1] D. Liu and C. Svensson, "Power consumption estimation in CMOS VLSI chips," IEEE Journal of Solid-State Circuits, vol. 29, pp. 663-670, June 1994.

[2] A.P. Chandrakasan and R.W. Broderson, "Minimizing power consumption in digital CMOS circuits," Proc. of the IEEE, vol. 83, pp. 498-523, April 1995.

[3] G. Gerosa, et al., "250 MHz 5-W PowerPC microprocessor," IEEE Journal of Solid-State Circuits, vol. 32, pp. 1635-1649, Nov. 1997.

Cadence Design Systems enables global electronic design innovation and plays an essential role in the creation of today's electronics. Customers use Cadence software, hardware, IP, and expertise to design and verify today's mobile, cloud and connectivity applications. www.cadence.com www.cadence.com