In Part I of this two-article series we described how the combination of logic libraries and embedded memories within an EDA design flow can be used to optimize area in CPU, GPU or DSP cores. In Part II we explore methods by which logic libraries and embedded memories can be used to optimize performance and power consumption in these processor cores.

**Maximizing Performance in CPU, GPU and DSP Cores**

Clock frequency is the most highly publicized attribute of CPU, GPU and DSP cores. Companies that sell products that employ CPU cores often use clock frequency as a proxy for system-level value. Historically, for standalone processors in desktop PCs, this technique has had value. However, for embedded CPUs it’s not always easy to compare one vendor’s performance number to another’s, since the measurements are heavily influenced by many design and operating parameters. Often, those measurement parameters do not accompany the performance claims made in public materials and, even when the vendors make them available, it’s still difficult to compare the performance of two processors not implemented identically or measured under the same operating conditions.

Further complicating matters for consumers of processor IP, real-world applications have critical product goals beyond just performance. Practical tradeoffs in performance, power consumption and die area — to which we refer collectively as “PPA” — must be made in virtually every SoC implementation; rarely does the design team pursue frequency at all costs. Schedule, total cost and other configuration and integration factors are also significant criteria that should be considered when selecting processor IP for an SoC design.

Understanding the role common processor implementation parameters have on a core’s PPA and other important criteria such as cost and yield is key to putting IP vendors’ claims in perspective. Table 3 summarizes the effects that a CPU core’s common processor implementation parameters may have on its performance and other key product metrics.
Elusive Critical Paths
To achieve optimal performance designers must reduce the delay in the critical paths of the CPU, GPU and DSP designs. These critical paths can be in the register-to-register paths (logic) or the memory access paths to/from the L1/L2 caches. Critical paths can move between memory and logic during the design process and sometimes feel like playing Whac-A-Mole® but having well characterized logic and memory IP, a solid EDA flow and mastery of design techniques can help designers to achieve timing closure.

<table>
<thead>
<tr>
<th>Core options¹</th>
<th>wafer cost</th>
<th>clock frequency</th>
<th>consumption</th>
<th>area</th>
<th>yield</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Active</td>
<td></td>
<td></td>
</tr>
<tr>
<td>More Low-V, cells (LVT)</td>
<td>▽</td>
<td>▲</td>
<td>–</td>
<td>▽</td>
<td>–</td>
</tr>
<tr>
<td>Use overdrive voltage (OD)</td>
<td>–</td>
<td>▲</td>
<td>▼</td>
<td>▽</td>
<td>–</td>
</tr>
<tr>
<td>Memory BIST, repair²</td>
<td>–</td>
<td>▼</td>
<td>▽</td>
<td>▽</td>
<td>▽</td>
</tr>
<tr>
<td>Power gating</td>
<td>–</td>
<td>▼</td>
<td>▽</td>
<td>▽</td>
<td>–</td>
</tr>
<tr>
<td>Operate at high temperature</td>
<td>–</td>
<td>▲</td>
<td>▽</td>
<td>▽</td>
<td>▼</td>
</tr>
<tr>
<td>Process corner</td>
<td>Fast</td>
<td>▽</td>
<td>▽</td>
<td>▽</td>
<td>▽</td>
</tr>
<tr>
<td></td>
<td>Slow</td>
<td>▼</td>
<td>▽</td>
<td>▽</td>
<td>▽</td>
</tr>
<tr>
<td>More clock uncertainty</td>
<td>–</td>
<td>▼</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Advanced Modeling (AOCV)</td>
<td>–</td>
<td>▼</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

Legend: Effect of parameter on attribute; e.g., using more low-V, cells strongly increases clock frequency

Notes:
1. Impact of logic for memory BIST and repair for CPUs, GPUs, and CPUs, is minimized if built into memories and integrated with BIST software.
2. Each additional threshold voltage adds mask steps to the fabrication process and additional wafer costs.
3. Operating at high temp usually improves frequency and hence degrades active power.
4. Process centering can improve negative effects of process spread.
5. Fast process improves frequency, so degrades active power.

Table 3: Effects of processor core design and operating parameters on key product metrics
High Performance Critical Path Optimization Techniques

Performance of CPU critical paths can be maximized by selecting the high speed logic libraries and memories and using various design techniques including starting with a proper floorplan, library usage, incremental synthesis, script settings, path group optimization and using over-constraints.

One of the best ways to minimize these critical paths is to start with a good initial floorplan to minimize the physical distance between the memory I/O pins and the critical registers within the processor logic. The ability to change this floorplan is critical as the design progresses and engineering tradeoffs are made to achieve the goals. A good floorplan based on the number of cores and the rest of the high performance core interconnectivity requirements can minimize the physical distance in the top level of the design and reduce timing bottlenecks.

Library usage refers to selecting the best library architecture (in this case High Speed), and selecting the optimal VT and channel length libraries to introduce in the synthesis and place and route flows. This also refers to the practice of *don’t use* lists to encourage the tools to select the highest performance cells by “hiding” some of the more area-efficient cells to trade performance for area.

Incremental compile techniques include running synthesis multiple times, sometimes introducing different synthesis options and additional libraries or cells to improve performance with each run.
Minimizing Power
Minimizing Power in CPU, GPU and DSP Cores

CPU, GPU and DSP cores must achieve high degrees of performance while consuming a minimal amount of energy to achieve longer battery life and fit into lower cost packaging. Power optimization is often the most important constraint, making the design challenge getting the best performance possible within the available power budget. Each new silicon process generation brings a new set of challenges for providers of logic library and memory compiler IP and a new set of opportunities to create more power-efficient IP.

Taking Advantage of Multiple VTs and Channel Lengths for Power Optimization

Most silicon processes support multiple VTs and at 28 nanometer (nm) and smaller nodes, multiple transistor gate lengths with the same gate pitch. This process feature enables multi-channel libraries without the area penalty of designing to the worst-case channel length to achieve footprint compatibility. These interchangeable libraries facilitate late-stage leakage recovery performed by automatic place-and-route tools and very fine granularity in power optimization. Additional VT cells (ultra-high VT, ultra-low VT) provide even more granularity, but with increased costs due to wafer add-ons.

![28-nm graph](image)

*Figure 10: This 28-nm graph plots the relative leakage (at the leakage corner) of a library on the vertical axis and the relative performance of a library (at the signoff corner) on the horizontal axis. The*

With all of the possible library options, the amount of data presented to the synthesis and place-and-route tools can seem overwhelming. The aggressive use of *don’t_use* lists (initially hiding both very low and very high drive strength cells) and the proper sequencing of libraries provides an efficient methodology for identifying the optimal set of high-speed and high-density logic libraries and memory compilers that will achieve optimum performance and power tradeoffs at minimum cost. These methodologies are effective on many different circuit types—CPUs, GPUs, DSPs, high-speed interfaces—and are dependent on the specific circuit configuration and process options being used. With a good understanding of the synthesis and place-and-route flow, designers can determine the
optimal library combination and sequence for a given configuration of a design.

<table>
<thead>
<tr>
<th>Design requirements</th>
<th>Optimization</th>
<th>Methodology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ultra high performance</td>
<td>Selective LVT usage</td>
<td>Initial synthesis with LVT_mid and SVT_min Library. Resolve remaining timing violations with LVT_min. Recover leakage for non-critical path using SVT_mid.</td>
</tr>
<tr>
<td>High performance</td>
<td>Selective LVT usage Only</td>
<td>Initial synthesis with LVT_mid and SVT_min Library. Resolve remaining timing violations with LVT_min. Recover leakage for non-critical path using SVT_mid, HVT_min, and HVT_mid.</td>
</tr>
<tr>
<td>Medium performance</td>
<td>No LVT or very selective LVT usage</td>
<td>Initial synthesis with SVT_mid and SVT_min library. You should also use LVT_mid if chip plans to use LVT. Recover leakage for non-critical path using HVT_min and HVT_mid.</td>
</tr>
<tr>
<td>Low performance</td>
<td>No LVT with leakage optimization</td>
<td>Initial synthesis with SVT_mid and HVT_min. Resolve remaining timing violations with SVT_min. Recover leakage for non-critical path using HVT_mid.</td>
</tr>
</tbody>
</table>

Table 4: Library selection and sequence recommendations for synthesizing blocks to achieve different levels of target circuit performance

Acquiring specific libraries for each different type and configuration of CPU, GPU and DSP core implemented on an SoC can be inefficient and costly. A properly designed portfolio of logic cells and memory instances can deliver optimal PPA results in processor core hardenings if it includes a full selection of efficient logic circuit functions, the right set of variants and the right set of drive strength granularities.

**Multi-bit Flip-Flops**

Using multi-bit flip-flops is an effective method to reduce clock power consumption and is now fully supported in mainstream EDA flops. Multi-bit flip-flops can significantly reduce the number of individual loads on the clock tree, reducing overall dynamic power used in the clock tree. Area and leakage power savings can also be achieved simply by sharing the clock inverters in the flip-flops with a single structure.

![Figure 11: Combining two single-bit flops into a dual flop with shared clocking](image)

Multi-bit flip-flops provide a set of additional flops that have been optimized for power and area with a minor tradeoff in performance and placement flexibility. The flops share a common clock pin, which decreases the overall clock loading of the N flops in the multi-bit flop cell, reduces area with a
corresponding reduction in leakage, and reduces dynamic power on the clock tree significantly (up to 50 percent for a dual flop, more for quad or octal).

Multi-bit flip-flops are typically used in blocks that are not in the critical path of the highest chip operating frequency. They range from small, bus-oriented registers of SoC configuration data that are only clocked at power up, to major datapaths that are clocked every cycle and with a number of variants in between. SoC designers use the replacement ratio, measured by how many of the standard flops in the design can be replaced by their multi-bit equivalents and the resulting PPA improvements, to determine their overall chip power and area savings. The single-bit flip-flops to be replaced with multi-bit flip-flops must have the same function (clock edge, set/reset, and scan configuration).

**FinFETs Extend the Power Curve at 16nm/14nm**

FinFETs are replacing planar FETs (also called “planar CMOS”) in today’s 16/14-nm process nodes. The clear advantages of FinFETs over planar transistors include excellent short channel control that leads to lower leakage due to lower drain-induced barrier leakage (DIBL), short channel effects and lower Vt variability due to lower channel doping. There is also less variability caused by random dopant fluctuations and the lower operating voltage can lead to that 50 percent dynamic power savings.

![Figure 12: This 22-nm tri-gate (FinFET) graph plots normalized transistor gate delay on the vertical axis and the operating voltage on the horizontal axis. It shows how these advanced transistors can operate at lower voltages with good performance, reducing active power by more than 50 percent.](image)

**Figure 12:** This 22-nm tri-gate (FinFET) graph plots normalized transistor gate delay on the vertical axis and the operating voltage on the horizontal axis. It shows how these advanced transistors can operate at lower voltages with good performance, reducing active power by more than 50 percent.

FinFET logic libraries and memories bring their own set of challenges due to quantized widths (and channel lengths) of the fins, and body biasing (often used to achieve lower leakage or perform process compensation) is totally ineffective. Also, the higher parasitics of the three-sided gate that enable lower leakage can increase the dynamic power. There are potential self-heating issues and the thermal aspects of electrostatic discharge (ESD) must also be managed. Finally, degradation and aging can be a factor as PBTI and NBTI are looking worse than they were with planar transistors. Designing logic libraries and memories for FinFETs that shield SoC designers from these challenges requires expertise in TCAD, device and parasitic extraction, transistor modeling, FinFET-specific layout and place-and-route tools. SoC design at the 16/14-nm FinFET node will be hard enough.

**Conclusion**
Each new SoC process generation brings a new set of challenges and a new set of opportunities for logic library and memory compiler IP to enable optimal SoC PPA. SoC designers need to be aware of and know how to take advantage of advances in library IP using the latest EDA tools. This is especially true when hardening CPU, GPU and DSP cores for high performance, low power and low area. A single source of logic and memory IP that enables this optimization can significantly reduce design time of hardening cores to SoC-specific requirements. Synopsys delivers effectively architected, efficiently designed, accurately modeled logic libraries and memory compilers, thoroughly integrated into EDA flows, silicon-proven and rapidly delivered through an experienced worldwide support infrastructure.