

SENSIBLE FLOORPLANNING AND POWER-PLANNING TECHNIQUES PAY OFF WITH SMOOTH BACK-END DESIGN FLOWS.

Design-planning guidelines prevent chip surprises

DESIGN-PLANNING STRATEGIES become common sense once you grasp the complex trade-offs they involve. One way to achieve that intuitive grasp is to understand some rules of thumb for good design planning. These practical considerations come from the day-to-day experience of dealing with floorplanning and power-planning issues. Getting good at design planning can help you prevent time-consuming iterations and chip respins. Design planning has become crucial for big chips with hierarchical design flows because such chips are more likely to have long interblock paths whose delays make timing closure impossible. For any complex chip, you need power planning to prevent problems due to IR drop and electromigration. The rules of thumb range from basic to advanced, and they can benefit both a COT (customer-owned-tooling) design flow and an ASIC flow, in which the ASIC vendor handles the actual back-end design. **Figure 1** shows a typical design-planning flow.

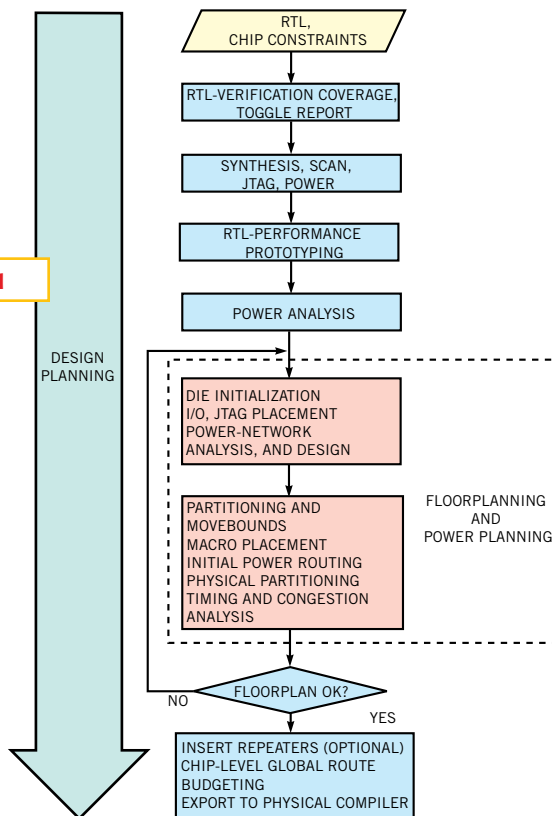
When creating a floorplan, you need to consider some basic characteristics of the process technology. For example, a typical standard-cell library defines cell rows as horizontal, and the routing direction of each layer follows the alternating pattern: metal1 horizontal, metal2 vertical, and so on. At any level, avoid routing that goes against the preferred routing direction for that level. When creating metal rings around cores and blocks, designers must remember to allow room for routing access to pins. When you push metal down into blocks, be careful to avoid congestion at the block corners; look at blockage effects inside the block.

When placing blocks, designers must avoid creating four-way intersections in top-level channels; T intersections create much less congestion. This consideration can be critical to leaving the necessary space for routing channels, depending on how much over-the-cell routing is possible. Using flylines can help determine optimized placement and orientation (**Figure 2**). Once you place the blocks, you can place block-level pins. You first determine the correct layer for the pins and spread out the pins to reduce congestion. Avoid placing pins in corners where

routing access is limited, and use multiple pin layers for less congestion.

Never place cells within the perimeter of hard macros. To keep from blocking access to signal pins, avoid placing cells under power straps unless the straps are on metal layers higher than metal2. Use density constraints or placement-blockage arrays to reduce congestion, because these strategies help spread cells over a larger area, thereby reducing the routing requirements in that area. Finally, when you place top-level buffers, remember to create the corresponding regions for power and ground connectivity. You also need to supply power and ground to

Figure 1



Design planning occurs in parallel with the stages of RTL maturity. Early floorplanning and power planning help guide RTL development and vice versa.

any areas where they might be useful for placing buffers or repeaters during the postplacement timing-convergence optimization. Avoid creating any blockage that increases congestion. It is essential to understand the requirements of your target process technology in any physical-design work. The design-rule document describes many process factors that you need to take into account. For example, most processes now require the insertion of holes in large metal areas, in a step known as “slotting,” or “cheesing.” Slotting relieves stress-related problems in the metal due to thermal effects but may change the metal’s current-carrying characteristics. Consult the design-rule document for this variable and many others.

PRACTICAL PLANNING STEPS

Along with the multitude of practical considerations you have to keep in mind when you create a floorplan, you need a philosophy that meets your business goals. Specifically, you have to decide what to optimize. Common optimization choices include repeatability, timing, project schedule, power consumption, and die size. This choice determines the margins you set for area usage and impacts other parameters. As an example, if you want to optimize the project schedule, then an obvious choice is to set usage lower than you would if you were optimizing for die size. The lower usage results in a larger chip but usually prevents routing problems.

You might also want to set a power-planning margin that puts more metal in the power mesh than you need to avoid failure under dc power loads. This strategy increases die area but may improve the schedule by avoiding downstream power problems. Additionally, you can fill any open spaces with power-mesh metal. The semiconductor process requires a certain percentage of metal anyway, so you might as well use that metal to help avoid power problems. But make sure the extra metal does not push signal wires closer together, thus increasing capacitance, power-consumption, and signal-integrity problems. As these considerations imply, floorplanning and power planning constitute an integrated process.

If you have multiple instances of any logical hierarchical element, consider grouping these items to form one hierarchical physical element. Look for logical modules in the RTL (register-trans-

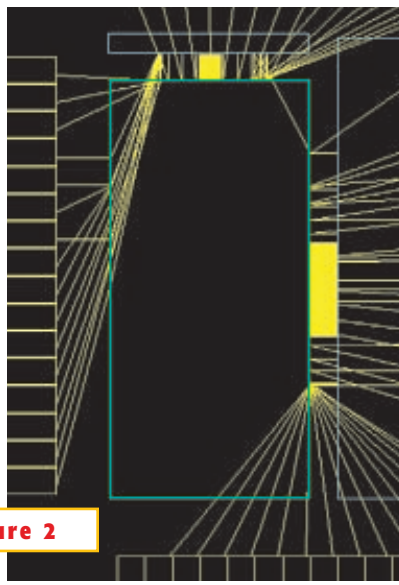


Figure 2

Flylines display the connectivity of nets and the effect of block placement.

fer-level) design representation that you can group in hierarchical blocks. You also want to group small blocks into one larger block, because it is easier to floorplan with same-sized blocks. Try to work with midsized blocks. A design partitioned in six to 12 roughly equivalent-sized blocks constitutes a reasonable candidate for floorplanning. Depending on the package design, you usually want to start the floorplan with I/Os at the periphery.

Designers should consider all the parts of the design that are not typical standard cells: memories, analog circuitry, PLLs, logic that works with a double-speed clock, blocks that require a different voltage, exceptionally large blocks, and unusual design-specific instances. You usually want to place these elements first to ensure that you can accommodate their special needs. Make sure you understand the special needs at the beginning. Flash memory, for example, has a high-voltage programming input that must be within a certain distance of an I/O pin, so you should place the flash memory first.

If you have two or more large blocks or other features that make a reasonable floorplan impossible, you may have to increase the die size or rearrange I/Os. Finding this problem early in the flow makes it easier to make a business decision about whether the chip will be financially viable with a larger, more expensive die. If any of the large blocks are soft (synthesizable) IP (intellectual property) or otherwise available as RTL, you

might be able to avoid going to a larger die by repartitioning that block into smaller pieces. Complete the floorplan by arranging the rest of the blocks in the remaining space based on their I/Os and power consumption, and try to avoid placing blocks that consume a lot of power near the chip’s center. Area usage varies, depending on the library, technology, and characteristics of the design, but, for average libraries, the sweet spot is usually around 70% usage. An unusually high percentage of registers or hard IP increases the percentage; large numbers of multiplexers or other small, pin-dense cells decrease it. Run initial synthesis to find out how big your blocks are.

There is no common method of setting the power budget for a design. Engineers determine the power dissipation at the block level using design tools, spreadsheet, back-of-napkin calculations—whatever is appropriate for the information they have. You must begin by calculating the power consumption of the core (non-I/O blocks). You need to know this parameter within about 30% of the final value, so you can calculate the chip’s supply currents.

Analyze the chip supplies for the effects of simultaneously switching I/Os. Mostly concern yourself about outputs, because they draw the most current. Most I/O-library suppliers perform extensive characterization of the I/Os and recommend the ratio of power and ground pads you need to have, given the number of simultaneously switching pads in your design. When creating a power mesh for your floorplan, consider that, even if you spread power consumption evenly over the entire chip and look at it from the block level, IR drop still is worse in the chip’s center, due to the length of the wires.

Some wires in the mesh therefore carry more current than others, so you need to calculate the current on every wire, junction, and via. Vias that are too small act like fuses and blow when the current is too high. In fact, you need to analyze via arrays for IR, current density, and electromigration. Also, bear in mind that IR drop in the center of the chip causes the logic placed at that location to run slightly slower. You can avoid electromigration problems in power meshes by meeting the maximum current-density limits for the process. Foundries publish these limits, and tools such as Astro-Rail from Synopsys can estimate current density throughout the power mesh and

highlight areas of concern.

To achieve a given steady state and worst-case power drop and to maintain current densities within acceptable limits, you need a certain total width of metal for power busing. Designers trade off the number of metal straps versus the spacing between straps to achieve the total metal width. Designers also must consider other issues in creating a power-distribution network. The most common issue is that, at some point in the design process, straps may end up with slots wider than the process-slotting size, thus reducing the conductivity that you had previously calculated for them.

Metal atoms migrating along metal grain boundaries that run parallel to the direction of current flow cause many of the failures due to electromigration. As metal wire width decreases,

fewer grain boundaries are parallel to the direction of current. When wire widths are in the “bamboo region,” where almost no grain boundaries are parallel to the di-

rection of current flow, the perpendicular grain boundaries give the wire an appearance similar to a stalk of bamboo. These wires can handle as much as an order of magnitude more current density than wider wires before experiencing electromigration failures.

Both power and ground wires create return paths for inductive coupling effects; the more power and ground wires you use, the more signal wires are protected from inductive noise effects.

Floorplanning and power planning of hard IP have some interesting special characteristics. As an example, consider the hardening of an ARM946E core using TSMC 0.13-micron technology (**Figure 3**). Core hardening involves steps similar to those of any planning flow, but the plan must be

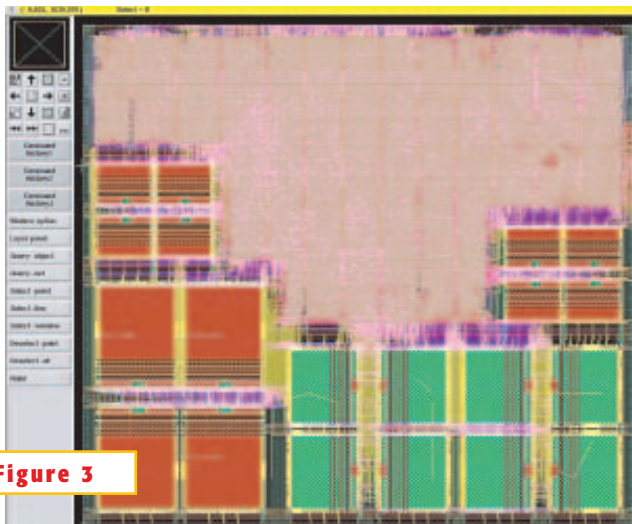


Figure 3

Hardening a core, such as the ARM946E-S, demands utmost versatility from the design-planning process. You need to implement features such as power and signal pins so that system-on-chip designers can use the core in a variety of contexts.

more flexible to allow for variations in the way designs use the core. In the ARM-core hardening described here, for example, the Synopsys Professional Services team configured the layout to allow you to rotate the core through 90°.

The team implemented a power grid over the entire core on the top three metal layers using 1.2-, 1-, and 1.6-micron straps placed every 24, 22, and 32 microns, respectively. One of the floorplanning challenges in this project was to arrange the horizontal straps such that they did not compromise the standard cell prerouting. Perl scripts helped automate this process. The provisions for rotating the core provided another floorplanning challenge. The team duplicated all the power and signal pins on a second layer and generated suitable via arrays to connect the two layers inside the core. The customer could therefore use the core in any orientation and still connect to it using preferred-direction routing for both signal and power-supply pins.

The design team created the required two sets of pins by starting with a single

set of pins on one layer. A Perl script collected the information about this first set of pins and duplicated the pins on another layer. Another Perl script created the via arrays to connect the two sets of pins.

All the signal pins were wide enough for access at the chip level without the need for off-grid routing, no matter how the core was aligned to the grid. The team also generated the RAMs such that the supply rings inside them were on layers chosen to suit the orientation of the RAMs in the floorplan, thus making it easier to connect the RAMs to the power grid. They routed the clock nets double-width, double-spaced, with double vias to minimize signal-integrity and electromigration issues. They built clock trees using integrated clock-gating cells inserted by Power Compiler. All logic inputs and outputs to the macro included protection from charge-collecting-antenna problems by the addition of a reverse-biased diode cell near the I/O pin. □

AUTHORS' BIOGRAPHIES

Steve Lloyd is a staff IC-design engineer

working for Synopsys Professional Services, where he serves as a project leader and performs layout design for RTL-to-GDSII design services. He holds a BSc with honors from the University of Leeds, UK.

Rick Mitchell is a staff engineer at Synopsys Professional Services. His job responsibilities include flow development and physical implementation. He holds a BSEE from Rose-Hulman Institute of Technology (Terre Haute, In).

Ron Spillman is a staff IC-design engineer at Synopsys Professional Services. A veteran of the semiconductor industry, Spillman is a project leader for RTL-to-GDSII design services.

Jonathan Young is senior manager, Physical Design Centre of Excellence, with Synopsys Professional Services, where he is responsible for physical-design projects. He holds a bachelor's degree with honors in electrical and electronic engineering from the University of Reading, UK.