



There is no clear line dividing the early floor planning phase from the malleable phase. (We use the term “malleable” because it implies a degree of solidity, but also a degree of adjustability.) The malleable floor plan is beginning to firm up, but it is by no means frozen—the malleable phase’s primary feature is incremental refinement of the floor plan. We move and adjust blocks, pushing them together or apart as needed, keeping their relative positions fairly constant. Similarly, block size remains reasonably consistent in the malleable phase, generally changing by no more than 10 percent.

Malleable floor planning involves incremental improvement of the general floor plan created in the early phase. With the block size estimates firming up, we begin getting meaningful congestion data from trial routes of the top-level netlist. We run trial routes periodically throughout the floor planning process (early and malleable) for two reasons. First, they ensure that the evolving floor plan can be routed and they give early warning to possible routing hot spots. Second, trial route data provides more accurate timing estimates because they include detailed route information.

The malleable phase tends to have significant content of place-and-routed blocks, rather than simple rectangular estimates. These blocks function as hard macros that enable the trial routes to accurately reflect over-the-block routing. Similarly, the block timing models for placed and routed blocks reflect actual extracted timing rather than estimates based on synthesis or physical synthesis runs or both.

As the dummy block rectangles are replaced with the real Library Exchange Format, the malleable floor plan models timing and congestion with increased accuracy. Fewer, subtler refinements are made until timing is met and the ability to route is assured. Then this final floor plan is frozen and becomes the blueprint for assembling the chip.

#### The designer’s perspective

Physical designers begin with the design in the early floor planning state. It is important at this stage to estimate the size of the block from synthesis runs and to guess how much logic will be added to the block. Normally, customers give only block diagrams of what they believe the chip will look like; from these diagrams, guesses of major buses and critical timing signals are identified. Clocking and power needs are also identified at this stage. These estimates are then put into a specification that is used by the designer assigned to floor planning.

Early estimates are important because the best way to have a successful chip is to catch problems early. If the RTL has not been written or is still incomplete, there is a greater chance that the specifications and protocols

**FIG. 2** Early floor planning enables rapid exploration of the physical design space. Early analysis of the design highlights potential RTL, timing and congestion problems even before the logical design is completed.