

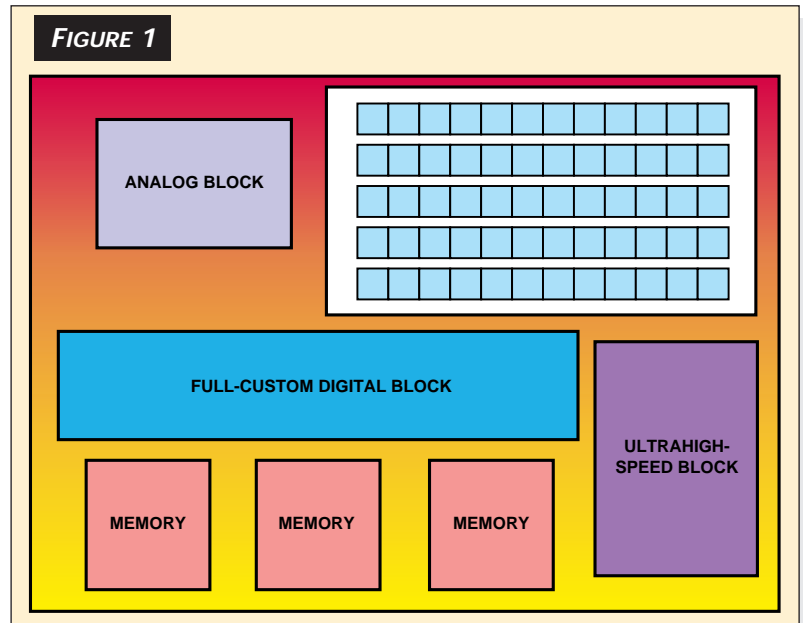
Chip place-and-route tools lay it on the line

The final step in ASIC implementation is defining your chip's physical layout. With interconnect-dominated deep-submicron designs, the tools and techniques you use determine whether your design is successful.

JIM LIPMAN, TECHNICAL EDITOR

After you check the logic functionality and simulate the timing of your digital-chip design, it's time to physically implement the design with a layout. This stage is when automated place-and-route (P&R) tools appear. The idea of connecting megagate chips with dozens of blocks, thousands of cells, hundreds of thousands of nets, and interconnect metal totaling a kilometer or more would be inconceivable without EDA P&R tools.

The number of commercial tools to place and route your chip is small, but products differ greatly in capability and price. Virtually all P&R tools do automatic routing, most with preroute block placement and timing constraints on at least some critical nets. Beyond this basic capacity, features expand to cover more complex routing for signal-integrity compliance, differential-pair matching, clock-tree synthesis, power-grid design, and very-high-gate-count chips. Understanding the features of available P&R tools and matching them with your chip's requirements saves you time and money when you're ready to go to silicon.



Place-and-route tools for complex core-based chips deal with a range of design blocks. The topology and constraints of these blocks require different routing tools for dissimilar pieces of your design (courtesy Cadence Design Systems).

@ a glance

- Routing is much more difficult than cell or block placement in digital-chip layout.
- Deep-submicron chips require constraint-driven routing to meet timing and other specifications.
- Different types of block and chip routing usually require different routing tools.
- Available routers vary widely in capabilities and prices.
- Clean place-and-route-tool interfaces to other design tools, such as floorplanners, are necessary.

The P&R tools you choose depend on the kind of chip you’re designing. Your design can be cell-based, gate-array, embedded-array, or full-custom. P&R tools for programmable-logic chips, specifically targeting coarse-grained FPGA or CPLD architectures, usually come from a programmable-logic vendor. These tools target the vendors’ own programmable-logic chips ([Reference 1](#)).

Cell-based chips contain two types of logic blocks: standard-cell-based blocks and embedded cores. EDA tools generate standard-cell blocks from rows of logic cells, with wiring channels between adjacent cell rows. Embedded cores are soft or hard. A soft core is either in an HDL or is a netlist of library cells, optimized during implementation for a specific process. A hard core has a predefined layout for a target process and is embedded in a chip as a fixed block. Cell-based chips have all mask layers defined for a given chip.

A gate array is an array of predefined logic gates, including a power grid, but without defined logic interconnectivity. The manufacturer processes the array through local interconnect within the defined gates. When targeting a design, you define the gate array’s signal routing between gates, and final-metallization processing implements these connections.

An embedded array combines cell-based and gate-array features. The chip

TABLE 1—REPRESENTATIVE PLACE-AND-ROUTE TOOLS

Company	Product	Cell-based	Gate array	Placement	Routing Area	Routing Channel
Avant! Circle No. 315	Apollo	X	X	X	X	
Cadence Design Systems Circle No. 316	Silicon Ensemble	X		X	X	
	Gate Ensemble		X	X	X	
	IC Craftsman	X		X	X	
Cascade Design Automation Circle No. 317	Epoch	X		X	X	X
Everest Design Automation Circle No. 318	Wire Wizard	X			X	
Gambit Automated Design Circle No. 319	Grandmaster	X	X	X	X	
Silicon Valley Research Circle No. 320	GARDS		X	X	X	
	SC	X		X		X
	SonIC	X		X	X	X
Snaketech Circle No. 321	cellSnake	X		X	X	
	gateSnake		X	X	X	
Tanner Research Circle No. 322	L-Edit/SPR	X		X	Late 1998	X

comprises a gate array in which one or more cell-based cores substitute parts of the array of gates. The core vendor predefines the embedded cores, and you use the array to “personalize” chip logic around these cores. You can preprocess embedded-array chips until the stage at which the chip needs final metallization to interconnect the array’s uncommitted gates and embedded cores for a design. This preprocessing is similar to that of gate arrays.

A full-custom chip has all or most of its circuitry “designed from scratch” for a design. Full-custom chips often contain timing-critical digital circuitry, analog circuits, or both. These types of chips, because of the critical nature of their embedded functions, are generally not amenable to full-chip automatic P&R tools. You can use a shape-based

router for some portions of a full-custom chip layout, but most designers still do a large part of such designs manually (see [box “Shape-based routing enhances constraint management”](#), on pg 76).

Wire-dominated designs

Deep-submicron chips are interconnect-centric. This fact means that the wire parasitics between transistors, rather than transistor-delay characteristics, dominate performance. Chip performance refers not only to delay times between nodes on the chip, but also to a variety of other parameters that affect how well your chip works or even if it works at all. Wire length influences chip timing and maximum clock speed. Wire-route congestion helps determine chip size and yield. Route

Input format	Output format	Platform	Starting price	Comments
EDIF, Verilog, GDS II, CLP, .lib, SDF, PDEF	EDIF, SDF, DSPF, PDEF, GDS II, Verilog	Workstation	\$315,000	Includes floorplanning, clock-tree synthesis, wide wires
Verilog, EDIF, LEF/DEF, GDS II	GDS II, DEF	Workstation	\$196,000	
Verilog, EDIF, LEF/DEF, GDS II	GDS II, DEF	Workstation	\$184,000	
GDS II, EDIF, LEF/DEF	GDS II, DEF	Workstation	\$150,000	Shape-based router for chip assembly (gridless)
Verilog, VHDL, GDS II, Spice, PDF, schematics	Verilog, VHDL, SDF, PDEF, GDS II, Spice	Workstation	\$200,000	Includes floorplanning, timing-optimization, and timing-analysis tools
LEF/DEF	GDS II, DEF	Workstation, PC	\$200,000	Shape-based router for all levels of chip routing
Verilog, EDIF, GDS II, SDF, DEF, GARDS	DEF, GDS II, SDF, Flat EDIF, Verilog	Workstation	\$250,000	Can run place-and-route jobs on multiple processors
SDF, EDIF, GDS II, PDEF, LEF/DEF, Verilog	SDF, EDIF, GDS II, PDEF, LEF/DEF, DCS, BAF	Workstation	\$70,000	Also runs on HAL 64-bit workstations
SDF, EDIF, GDS II, LEF/DEF, Verilog	SDF, EDIF, GDS II, DCS, LEF/DEF, BAF	Workstation	\$50,000	Also runs on HAL 64-bit workstations
SDF, EDIF, GDS II, PDEF, LEF/DEF, Verilog	SDF, EDIF, GDS II, PDEF, LEF/DEF, DCS, BAF	Workstation	\$180,000	Also runs on HAL 64-bit workstations
EDIF, SDF, Spice	GDS II, CIF, SDF	Workstation, PC	\$20,000	Includes clock-tree synthesis and a floorplanner
EDIF, SDF, Spice	GDS II, CIF, SDF	Workstation, PC	\$20,000	
EDIF, .spr	GDS II, CIF, .tdb	Workstation, PC	\$7995	

spacing and quality affect signal skew and crosstalk. Wire width affects signal delay and chip reliability due to electromigration problems, particularly on clock and power nets. The ability of P&R tools to handle these types of problems varies widely from vendor to vendor.

You need different types of P&R tools for different types of chip implementations. For example, gate-array routing requires different algorithms than does cell-based routing (**Table 1**). In addition, the type of routing tool needed for detailed intracell routing is different than the tool you use for chip assembly (see **box** “[Channel vs area routing](#),” pg 75). Assembling the blocks on a chip means connecting a relatively small number of cores, memories, and user-defined logic blocks (**Figure 1**). A typi-

cal chip-assembly router has to handle as many as 50 blocks, each block with as many as several hundred connecting points, or pins, on several layers of metal interconnect. The maximum nets that a chip-assembly router addresses are on the order of 10,000 to 20,000.

Interblock routing tools address a different problem. These tools may have to connect thousands of individual cells, each with just a few connecting points, on one or two metal layers. Problems associated with special nets, such as clock lines and power grids, are smaller with intracell routers than with chip-assembly tools.

P&R constraints

Deep-submicron (0.5 μm and below) chip P&R is more than just placing cells and blocks in a minimum area

and connecting the appropriate pins. With stringent timing specifications for many chips, P&R tools are generally timing-driven. This factor means that you can control routing with signal-delay limits on critical nets. P&R tools vary widely in how many nets can have timing constraints. The more constraints placed on the router, the longer the router takes to finish a job and the greater the possibility that the router will not complete all routes (some constraints may be impossible to meet).

Routing today's high-speed chips requires more than just routing timing constraints. You also have to meet many other electrical specifications, including signal integrity, electromigration limits, and power dissipation (**Reference 2**). Although all popular

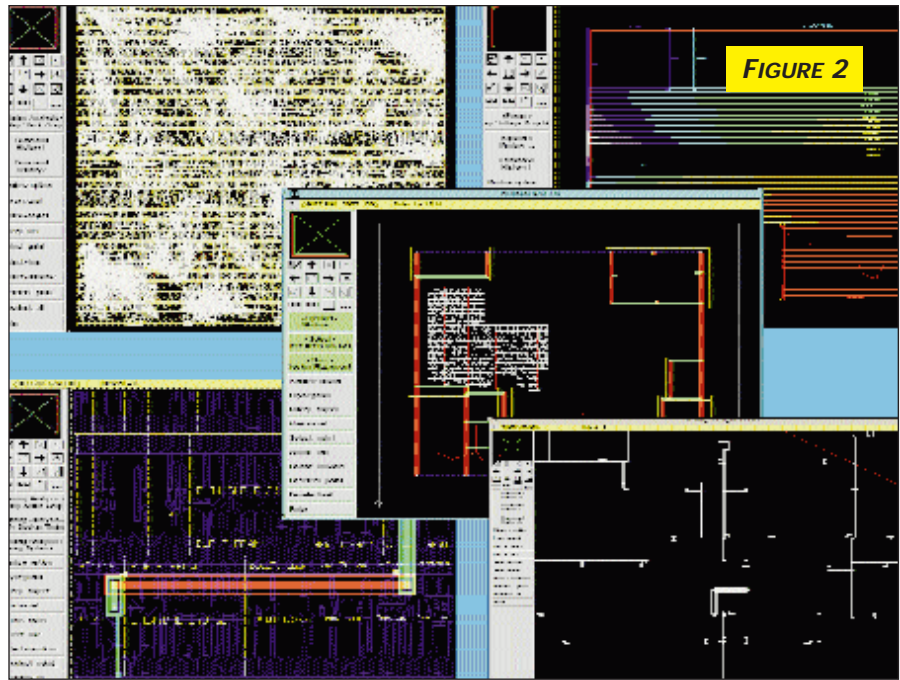
CHIP P&R TOOLS

chip routers are timing-driven, few are signal-integrity-driven as well. Silicon Valley Research links SonIC, its cell-based P&R tool, to its Analog Tool Box, bringing analog-circuit simulation into the routing operation. Analog Tool Box provides a means of interactively checking routed lines for crosstalk and other signal-integrity problems. IC Craftsman from Cadence, on the other hand, is a shape-based router that is both timing-driven and signal-integrity-driven (see **box** “[Shape-based routing enhances constraint management](#)”). You use IC Craftsman for full-chip assembly and a Cadence routing tool, Silicon Ensemble, for block routing.

Startup company Everest Design Automation is developing a shape-based tool that you can use for both block routing and chip assembly. Besides being a unified tool for all levels of cell-based routing, Everest's Wire Wizard will be available for both Windows and Unix platforms. Two other companies, Snaketech and Tanner Research, offer this dual-platform flexibility in chip-routing tools.

Router limitations

The major problem with today's chip P&R tools is the hierarchy inherent in



Despite a high price tag, a full-featured P&R tool is necessary for optimizing area, timing, signal integrity, and power on a complex deep-submicron chip. You use different P&R-tool capabilities for various design domains. Color-coded data helps you interpret interconnect layers and the degree of severity when analyzing problems (courtesy Avant!).

core-based chips. Chips with embedded blocks have two major hierarchical levels, one within the blocks them-

selves and another for the entire chip containing the blocks. As previously noted, the different parameters and

REPRESENTATIVE ASIC PLACE-AND-ROUTE EDA-TOOL COMPANIES

For more information on companies selling chip place-and-route tools, circle the appropriate numbers on the Information Retrieval Service card or use *EDN's* Express Request service. When you contact any of the following manufacturers directly, please let them know you read about their products in *EDN*.

Avant!
Fremont, CA
1-510-413-8000
fax 1-510-413-8080
www.avanticorp.com
Circle No. 315

Cadence Design Systems
San Jose, CA
1-408-943-1234
fax 1-408-943-0513
www.cadence.com
Circle No. 316

Cascade Design
Automation*
Bellevue, WA
1-800-258-8574
fax 1-206-649-7600
www.cdac.com
Circle No. 317

Everest Design Automation
Fremont, CA
1-510-668-0640
fax 1-510-668-0642
www.everest-da.com
Circle No. 318

Gambit Automated Design
San Jose, CA
1-408-345-3555
fax 1-408-241-6741
www.gambit.com
Circle No. 319

Silicon Valley Research
San Jose, CA
1-800-624-9978
fax 1-408-361-0330
www.svri.com
Circle No. 320

Snaketech
Grenoble, France
011-33-4-7607-1408
fax 011-33-4-7607-1067
www.snaketech.com
Circle No. 321

Tanner Research
Pasadena, CA
1-818-792-3000
fax 1-818-792-0300
www.tanner.com/eda
Circle No. 322

VOTE . . .
Please also use the Information Retrieval Service card to rate this article (circle one):
High Interest 590
Medium Interest 591
Low Interest 592

Super Circle Number

▶ For more information on the products available from all of the vendors listed in this box, circle one number on the reader service card.

Circle No. 323

*At press time, Cascade had been purchased by Duet Technologies: www.duettech.com.

constraints needed for core or block routing vs those associated with full-chip assembly often require different routing.

Jon Levi, development-support manager of product development, and Charley Kahle, technical director, both of Cadence Spectrum Design (the design and design-support group of

Cadence), summarize the deficiencies of available chip routers. Being responsible for many high-density, high-speed chip designs, Levi and Kahle agree that although many current chip routers do an efficient job on “flat” (no physical hierarchy) designs, core-based chips present problems. Blockages in embedded cores prevent globally rout-

ed wires from traversing these cores, thus wasting silicon area. Furthermore, clock-tree synthesis is more difficult for core-based designs than for flat designs. You really need two levels of synthesis: one for cores and another for the top-level chip. Meeting critical clock constraints, such as latency and skew, becomes much more difficult

CHANNEL VS AREA ROUTING

You can choose between channel and area routers for chip-interconnect routing. Some routers use only one type of router; others combine the two to handle the routing job more effectively. Both channel and area routers are based on a grid that the tool defines across the region being routed. Another type of router, a shape-based router, is a gridless form of an area router (see box “[Shape-based routing enhances constraint management](#)”).

Channel routers were the first type of layout tools used for cell-based chip routing. Designers used early versions of these routers to route interconnect between rows of standard cells, each cell in the same row having the same height (Figure A). The router worked with two or 2½ layers of interconnect, and the router contacted the cells on their top or bottom edges. The “½” layer referred to polysilicon, used for transistor-gate definition and short routes in which polysilicon’s higher resistivity than aluminum would pose no problem.

Routers assumed that the cells were mainly “blocked,”

meaning that routing wires could not go horizontally over a cell but could occasionally travel over cells vertically. The router increased the channel height between cell rows as necessary to fully route the design. Because you could not determine channel heights before routing, you couldn’t use a channel router for a chip with a predetermined area.

Channel routers are easy to learn how to use, but they do not result in chips with high routing density. These types of routers do not handle core-based designs well, because the cores may vary greatly in width and height. In addition, if you have to make even a small change to a channel-routed design, rerouting the chip is as big a job as the initial routing.

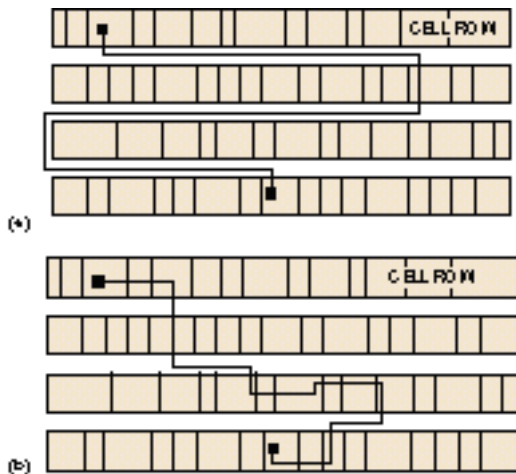
Area routers work on chips for which you have estimated a die size. This class of routers has “knowledge” of predetermined wires within placed cells or cores and, treating these wires as blockages, can route wires through porous regions of these cells or blocks (Figure A). Originally used for routing gate arrays, area routers now dominate cell-based chips as well (Table 1).

You can route denser chips with an area router than you can with a channel router. Area routers are best used interactively by a designer who can see postrouted areas that are lightly or heavily routed and can adjust block placement to optimize routing density. This feature is important because, if you start with too small a chip, the router may be unable to complete all routes. If the starting-chip area is too big, you waste silicon area and may add unnecessary interconnect delay on the bigger-than-needed chip.

Area routers are better choices for floorplanned designs—designs in which you place cores or standard-cell blocks on a chip before routing to estimate chip timing. You can floorplan manually or with a floorplanning tool. Place-and-route (P&R) tools such as Avant!’s Apollo and Cascade’s Epoch use built-in floorplanners. Other P&R tools have links to separate floorplanning tools. You also need area routers to effectively route chips with many interconnect layers. Typical deep-submicron processes offer five to six metal layers.

Not surprisingly, virtually all commercially available chip routers use some form of area routing. Epoch and Silicon Valley Research’s SoniC combine channel and area routing into one tool. SoniC goes a step further by combining two types of area routers—line probes (for initial, relatively uncongested routing) and maze probes (for routing completion of congested regions)—to optimize full-chip routing.

FIGURE A



A channel router does most routing between rows of cells (channels), under the assumption that cells generally don’t allow global routes through their interiors (a). Area routers have knowledge of “open” areas inside cells and can easily route over cells as well as between them (b).

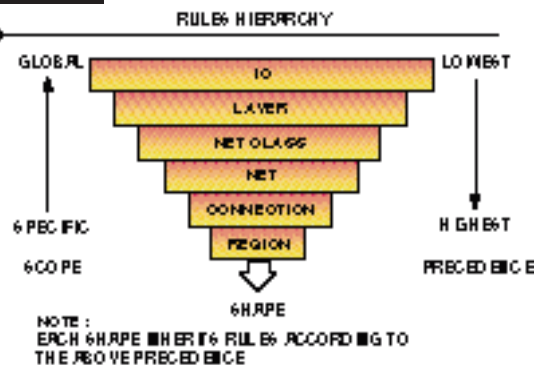
SHAPE-BASED ROUTING ENHANCES CONSTRAINT MANAGEMENT

Shape-based routing, or gridless routing, is not a new idea. Shape-based routers have been available for pc-board routers for a while. Specctra from Cooper & Chyan Technology (CCT) is the most widely used tool for board routing. CCT also developed and released its IC Craftsman tool family for shape-based chip routing. (Since IC Craftsman's release, Cadence purchased CCT.)

Shape-based chip routers have advantages similar to those of their pc-board counterparts. Because they are gridless, these types of routers work with complex geometries and get a higher routing density than do gridded routers. You also have more flexibility to add constraints to the automated routing operation. For example, instead of using universal rules to regulate interconnect width and spacing on a per-layer basis, you can apply design rules to classes of nets (such as all clock lines), individual nets, and specific areas of a chip within a precedence hierarchy. In IC Craftsman's rules hierarchy, for example, you have a rule precedence; connection rules supersede net rules, and net-class rules supersede layer rules (Figure A). The trade-off for this rule-based constraint flexibility is increased runtime—rule-set precedence and an increased number of rules to check result in slower routing than you can achieve with a gridded router.

Area- and net-specific routing rules let you add additional performance-driven constraints to your router. Examples of the types of constraints you can include are signal integrity (crosstalk, noise, and shielding), matched lengths (for differen-

FIGURE A



Cadence's IC Craftsman uses a design-rule hierarchy in its constraint-driven routing. This hierarchy lets you assign rules to a specific subset of your design, such as a physical region on the chip or a class of nets, that supersede the chip's general rule set.

tial pairs and clock-skew control), minimum/maximum lengths (for buses), and maximum IR drop (for power and ground lines). For analog- or mixed-signal blocks, a shape-based router is your only choice for automated routing. Gridded routers can't handle the complexity of analog-circuit layout.

with two levels of clock-tree synthesis.

A number of designers who perform chip P&R bring up what they consider to be a serious problem regarding timing-driven router accuracy. When you synthesize your design from an RTL to a gate-level representation, the synthesis tool uses a statistical wire-load model to meet timing constraints. A timing-driven router computes timing information based on wire length and parasitic information, with parasitic parameters based on a process-technology file. The inconsistency between synthesis- and router-timing models can cause difficulties in physically implementing the logic that the synthesis tool defines, resulting in additional work for the designer responsible for chip layout.

Another problem with current routers is that their interconnect-parasitic calculations are too inaccurate for chips processed at or below 0.25 μm . Automatic routers use 2-D estimations for wire parasitics; 3-D models would

make the routing operation too lengthy for full chips. Some full-chip-analysis tools, such as those offered by Simplex Solutions (San Jose, CA, www.simplex.com), have more accurate parasitic-estimation algorithms, but these algorithms are too time-consuming for router use. The accuracy problems of today's routers make the use of full-chip verification tools an important part of your postlayout chip design (Reference 3).

There are several features that Kahle, Levi, and other cell-based-chip designers feel a router must have to be useful on high-speed designs. These features include wide-wire generation, clock synthesis, power-grid capability, and both timing and signal-integrity constraint-driven routing (Figure 2). Clean interfaces to other design tools, such as floorplanners, are also important. Floorplanning lets you place blocks and estimate chip timing before full routing, saving time that you might waste by routing a chip with less-than-

optimally placed blocks. Keep all these points in mind when shopping for your chip P&R software. EDN

References

1. Dipert, Brian, "Shattering the programmable-logic speed barrier," *EDN*, May 22, 1997, pg 36.
2. Lee, Tsu-Chang and J Cong, "The new line in IC design," *IEEE Spectrum*, March 1997, pg 52.
3. Lipman, Jim, "Postlayout EDA tools lock onto full-chip verification," *EDN*, Oct 10, 1996, pg 93.

You can reach Technical Editor Jim Lipman at 1-510-606-1370, fax 1-510-606-1563, ednlipman@mcimail.com.

VOTE

Please use the Information Retrieval Service card to rate this article (circle one):

High Interest 590	Medium Interest 591	Low Interest 592
-------------------------	---------------------------	------------------------