PRINTED CIRCUIT DESIGN

is published monthly by:
UP Media Group Inc.
2018 Powers Ferry Road, Ste. 600
Atlanta, GA 30339

Tel (678) 589-8800
Fax (678) 589-8850

Autorouting Tips and Techniques

Part II of our series on how to better use your routing tools

Compiled by PCD Staff

In the September issue of *Printed Circuit Design*, we turned to the vendors for primers on how to get the most out of their autorouters, featuring Electronics Workbench, Intercept Technology, Mentor Graphics and Zuken. This month, with a little help from Altium, Cadence, DDE, McCAD and a user of CadSoft, we complete the two-part series.

Altium

By Phil Loughhead, Protel product manager

Altium's Situs autorouter uses advanced topological mapping to define the routing path and then calls on a variety of proven routing algorithms to convert this "human-like" path into a high-quality route.

1. Set up sensible routing design rules appropriate to the board technology. Poorly targeted or inappropriate design rules can adversely affect autorouting performance.

2. Check for special clearance requirements, such as fine-pitch components with lands closer than standard board clearances. Good tools cater to these using suitably scoped and prioritized design rules.

3. Use the design rule checker and eliminate all violations before starting the autorouting process.

4. Choose appropriate layer directions to suit the connection line flow. Topological mapping is not constrained to horizontal and vertical routing. Typically, it is best to have outerlayers as horizontal and vertical. For a multilayer board with a large number of connections at a 2 o'clock angle, however, set this as the preferred routing direction for an internal layer. Choosing the right directions can make a significant difference to routing performance in terms of time and quality.

5. Use routing priority rules to set a higher priority on difficult nets or those for which the cleanest routing is desirable.

6. Run fanout passes by themselves first, and assess their quality. Problem areas may need to be fanned out manually.

7. Pre-route critical nets, and if it is essential that they remain unchanged by the routing process, lock them. However, avoid unnecessary locking; a large number of locked objects can make routing much more difficult.

8. Identify dense or problematic areas. Use the autorouter interactively to route single components, or nets to assess routability.

9. Don't be afraid to experiment. If the results are not acceptable, change the router's approach. Add intermediate cleanup passes, make more room around dense areas, or change the layer directions.

10. Always follow good component placement practices. Ultimately, placement will have the largest effect on routing performance.

Cadence Design Systems

By Josh Moore, product marketing manager

Autorouting has advanced to address high-speed, complex board designs. Now that design densities and high-speed rules dominate, hand-routing is impractical. SPECCTRA is a constraint-driven autorouting environment targeted for these challenges. Here are some tips on how to realize its benefits.

1. Develop an overall route strategy. How do you normally route (placement evaluation, fanout, critical nets, buses, etc.)? A combination of applying known influences of the router and following a methodology that closely mimics interactively routing a design typically yields the most acceptable results. Perform a route 5 pass "what if" analysis to validate features of the design.

2. Route, placement, layer and rule feasibility. Use general convergence criteria to help determine if this route will hit a satisfactory result. First, take a simple approach to prove if the design is routable. A simple run file may look like: route 5, clean 2. Take a look at the highest conflict areas for potential placement changes and "unroutes" to determine if rules can be met. Look in the status file for such signs as:

- Less than 2% of the connections are unrouted after pass 1.
- There are less than five conflicts per connection after pass 1.
- Conflict resolution is 15 to 30% after passes 2-5.

3. Quality fanouts are crucial to routability. Work with the different component types/pitches separately to get the best fanout possible. Don't assume one setting works on the whole board.

Although SPECCTRA does not require a grid, use one to your advantage. Run with a larger grid at first and then reduce the grid in subsequent passes until finished. For example, set via grid to permit two wires between vias:

via grid = (2 x wire_via clearance) + (2 x wire width) + wire_wire clearance + via diameter

For dense designs, this may be too restrictive. Reduce the grid and try again, but don't proceed until fanout is done. Use SPECCTRA reports to visit and fix each pin with a missing fanout.

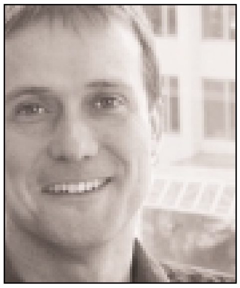
4. Simultaneous routing of a complex set of physical, spacing and electrical constraints is critical for high-speed designs. Ensure all constraint values and any constraint specific areas are entered into the system before routing.

5. Interactive routing is good but remember: too much interactive routing affects routers. Why? What's the first thing you do when a connection is done? Fix it in place. Now it is another obstacle to get around. Routers must be able to move all copper to converge on a clean DRC solution.

6. Diagonal routing may leave little room to converge on a "shorter" solution; work on an orthogonal solution, then let the Miter routine do its job. This permits easier length tightening during routing. Use the various tuning patterns to your advantage. If the error spread is low, then a meander route with no vias may be a good choice. If the nets are in a tighter area and spacing permits, use a trombone or accordion pattern to obtain maximum trace density. Fine tuning to within 0.001″ or 0.002″ can be achieved through iterative tightening of the short/long range rule followed by execution of the "elong" command.

7. Use a similar strategy as noted by paying attention to the short/long range. Buses often use relative delays based off a specific clock net. Route the clock and obtain its delay. Convert the bus relative delays into hard min/max numbers and use the min/max routing technique to reduce creep associated with relative delays. Via limited buses should be routed auto-interactively without length rules, providing space along bus members to add elongation when the bus is completed.

8. Virtual pins are gaining in usage as they are the only way to achieve accurate timing from a common source to multiple loads. Ideally, topologies with virtual pins should be built into the core PCB constraint system; however, they can also be added directly within SPECCTRA. SPECCTRA will attempt to dynamically optimize virtual pin placement while routing, if permitted. Preferred locations may be controlled by positioning virtual pins interactively or in a Do file, and restricted or prevented from moving by adding a radius tolerance value.



Altium's Loughead

Cadence's Moore

DDE's Viklund

9. In the event tha1 100% convergence is not achieved, the status file can help predict where the problems occurred. Things to check for include:

- High crossing counts, which may indicate via starvation or an insufficient number of signal layers.
- High clearance and crosstalk counts, which may indicate design rule errors or incorrect route directions.

Look for:

- Protected wires blocking the router.
- Unrouted fanouts.
- BGA fanout.
- Placement density.

10. All routers make unnecessary jogs in traces that must be cleaned up to ensure manufacturability. The routines named Critic, Miter and Clean can help with this task. Since these routines maintain all constraints that are entered, they will not create new or additional conflicts. Specifically, use Miter to automatically optimize those nice diagonals that all designers strive for manually.

DDE USA Inc.

By Per Viklund, technical manager

Supermax ECAD is tightly integrated with Cadence's SPECCTRA autorouter. Although these hints are seen specifically in the light of this integration, all SPECCTRA users could benefit from this information.

1. No router can turn a bad component placement into a good routing solution. Supermax ECAD comes with a set of functions to cluster components and to place and optimize placement by cluster, block, and room.

2. SPECCTRA Do files? No thanks! The design rule model is compatible with SPECCTRA's. Set up rules in Supermax ECAD; all rules and settings are transferred by the integration.

3. Perform fanout in Supermax ECAD. It often gives a cleaner result, with less routing blocks.

4. Ensure 100% fanout before proceeding with general routing. If fanout cannot find a solution for every surface mount lead, it is not likely that the router will be able to resolve it later. Typical reasons for fanout failures are improper component placement, or (sometimes) incorrect clearance rules. Adjust and rerun fanout.

5. Regardless of experience, you can't see on the design which layer directions will give the best routing result. Instead, try various solutions, such as routing two passes to see where it's heading. Then restart with the best setup. This can well be the difference between 100% completion and many unroutes.

6. Here is a little-known but powerful feature. If logical information is present in the design data, SPECCTRA can do pin/gate-swap on the fly while routing. (Supermax ECAD permits enabling/disabling this feature.)

7. So-called optimal metal fills that connect to one net, permitting other nets to pass through isolated, are common. The router, however, does not understand this concept, and treats these as keep-outs for all but one net. Don't pass fills to the router as they will become obstacles that consume routing time. When performing redesigns, metal fills can become serious obstructions for the router.

8. Go totally gridless! It increases the completion rate. Supermax ECAD is still able to snap in interactive editing afterward. By the way, a very small grid provides faster routing than gridless.

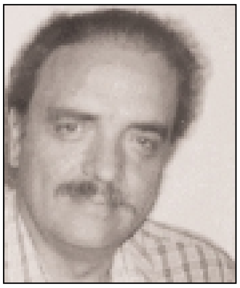
9. Monitor progress. It's easy to spot early on if routing is not going to converge. If the status report indicates that the design is not converging, abort! Change "something" and re-run.

10. Users who take the time to learn how to control the router will get better results faster. For tough designs, there is no such thing as a push-button solution.

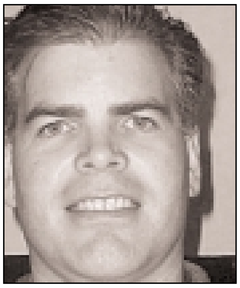
McCAD

By George Soluk, director of software development

Autorouter users want a tool that does everything perfectly. The router has to work



MCAD's Soluk



Dallas Logic's Tauch

quickly, without much to clean up. McCAD's Trailblazer autorouter features unattended and fully interactive routing in a variety of modes. Designers can improve results by observing the following points:

1. Placement can make or break the design. Position and orient components within the mandatory constraints of the physical requirements, and in such a way as to minimize crossovers as well as localized congestion. When finished, ask yourself: Could I route this manually using the constraints that will be imposed on the autorouter? If the answer is no, then in all likelihood the autorouter won't either.

2. Select an appropriate strategy. Although the standard default strategy will handle 95% of designs, there are more than 15 other strategies for rare or unique situations. If none is ideally suited, select the one closest to your needs and fine-tune the particular control aspects. Selecting or fine-tuning the strategy will determine how much cleanup may be required.

3. Maximize board utilization. Set a suitable default grid/spacing/track width combination that maximizes available interconnect pathways.

4. Locate impossible spacing constraints. Perform a DRC check for pin-to-pin spacing constraints, which may prevent proper entry or exit of signal traces to these pins. Local routing using a different routing setup for only those specific areas may need to be performed.

5. Set the direction of routing to move from a dense area toward a less dense one, as a designer would do if routing manually. Connectors, for example, are usually dense areas. Routing away from the connector can be much easier than vice versa. This practice will reduce or eliminate route blockage.

6. If the design has a number of highly congested areas, clear them first through the use of restricted route area windows. Push congested areas into less congested ones. Doing so will avoid potential route blockages when routing from a less congested area into a very congested area.

7. A bogged down router could be a sign of a spacing constraint that is preventing interconnect completion. Stop the router and examine the area in question, and adjust strategies or spacing constraints as necessary to resume effective routing.

8. Use prerouted devices when appropriate. In cases of extreme densities or difficult escape paths, such as with PGAs or BGAs, prerouting a specific fanout pattern will dramatically improve completion results.

9. Spend extra time in net setup. Although most designs do not require this additional time before routing, very complex ones – 50,000 connections or more – will have substantially improved results when nets are assigned a specific order of routing and to specific routing layers. In one instance, a user reduced route time to 28 hours from 270 hours and completed all but 30 routes of a design with 180,000 unrouted connections. Routing was then completed using interactive mode.

10. Any good autorouter provides an interactive mode, in which the designer can finish any uncompleted routes. In this mode, push traces around while guiding the autorouter through placing the final routes. While doing this, the autorouter will move and carefully space the traces as the final paths are laid.

CadSoft

By Eric Tauch, senior hardware engineer, Dallas Logic Corp.

CadSoft's Eagle is a grid-based, rip-up and retry system. It uses a costing structure for varying the routing parameters of the autorouter algorithm (costs are assigned for each via, wrong way route, number of trace bends, and so on).

1. Take advantage of the CadSoft Web site. Get answers to autorouter or other questions by posting to and searching the active support forum. Download autorouter control files, ULP programs, parts libraries, user projects, and more. Hundreds of free files are available.

2. Obtain the published users manual. The built-in Help files do not cover the autorouter very well. The printed version offers details on such aspects as the autorouter costing structure and rip-up/retry control that must be thoroughly understood for optimal operation.

3. Modify the autorouter cost variables to control "big picture" routing characteristics. Lacking any common sense, the autorouter will generally route a few "silly" traces each job. Clean these manually; don't waste time trying to cost out the bad routes.

4. Always iteratively save a project, especially before autoroute. If you don't like the routing results, change a few parameters and rerun the autorouter from the saved version rather than manually removing the traces. Autoroute groups of nets in sequence to give routing priority to critical signals or power nets. Note: Iteratively routing too much of the board bounds the autorouter rip-up/retry capability.

5. Use the "net class" option in the schematic to assign trace width, drill size, and clearance rules for various net types. The autorouter will then use these rules to route these nets for you (instead of manually routing them prior to autoroute). This also provides the rip-up/retry algorithm with maximum flexibility.

5. Use the autorouter to break out power pins on surface mount devices by assigning a high cost to trace length and a low cost to vias. Then autoroute only the power nets. Save the autorouter parameters in an "auto_power.ctl" file for use on a later project.

6. Keep the grid setting as high as possible to facilitate completion time, but for a job that finishes with a few unrouted traces, drop the grid setting to a smaller value. Then rerun the job (perhaps overnight) and the autorouter may completely finish the job.

8. The autorouter has absolutely no common sense – remember that. Anything that helps it work logically vastly improves results. Minimize the task by presenting a logical component placement and assign layer routing directions that work well with the job at hand. I overcame a problem with routing dense surface mount BGA components by adding breakouts to the component footprint. Breakouts logically define routing channels to and under the components and provide access to each pin on all layers.

9. The autorouter does not always perform as well if you use odd numbered clearances such as 0.009″ or 0.011″. Use even numbers (e.g., 0.008″ or 0.012″) when possible.

10. Keep pad-to-via clearances as high as possible. This makes prototype hand assembly, debug and rework much easier, as it helps prevent solder shorts from pads to vias. Also, define general clearances to be as high as possible, but still make efficient use of routing area (maximize number of traces between pads, etc).