Solving IC Interconnect Routing for an Advanced PA-RISC Processor

This paper discusses some important new block routing technologies that were required for the HP PA 8000 processor chip. These technologies are implemented in a new block routing system called PA_Route.

by James C. Fong, Hoi-Kuen Chan, and Martin D. Kruckenberg

The design complexities of today's microprocessors have grown significantly, with the number of transistors climbing to well over a million, silicon die sizes larger than 1.6 cm^2, clock speeds exceeding 150 MHz, and short design cycles caused by competition. These issues create tremendous pressure on design teams and the tools they use. The PA 8000 CPU design team used powerful design automation tools to achieve their design goals.

Layout of the interconnect metal on the chip is one of the key components of advanced designs. It is vital to address the increasingly complex layout problem to achieve smaller die sizes, higher performance, and quicker time to market. Since the early 1980s, HP has been working to solve the top-level IC interconnect problems associated with many of the larger HP-designed and HP-manufactured ICs. This paper will discuss some important new block routing technologies that were required to implement the HP PA 8000 microprocessor. These technologies are embodied in a new in-house block routing system called *PA_Route*.

Buy or Build Decision

Frequently we at the HP Integrated Circuit Business Division (ICBD) are approached by HP design teams who are about to embark on the design of a new chip to be manufactured by ICBD. We are asked to enhance our routing technology to address issues critical to the chip's successful routing.

This was the case when the PA 8000 design team approached us with some ideas for new features needed to take advantage of new technologies. Like most aggressive designs, they were pushing the limits of every technology where they thought they could get a significant return on investment. Block routing was one area they thought they could improve.

Our existing block router is called HARP (Hewlett-Packard Automatic Routing and Placement). HARP had been evolving for over a decade and had some legacy code that was becoming difficult to extend.

Using customer surveys to complement our own knowledge, we did a detailed analysis of various existing block routers. We wanted to see how they address this new class of block routing problems. Design teams are hesitant to switch layout tools unless alternatives can be found that match their design requirements well enough to justify the risks of switching tools and the cost of the new tool, not only the capital cost but also the cost of learning how to use the tool effectively. Using radar charts (see Fig. 1), we were able to determine that the less aggressive style of chips represented by the PA 7100LC processor

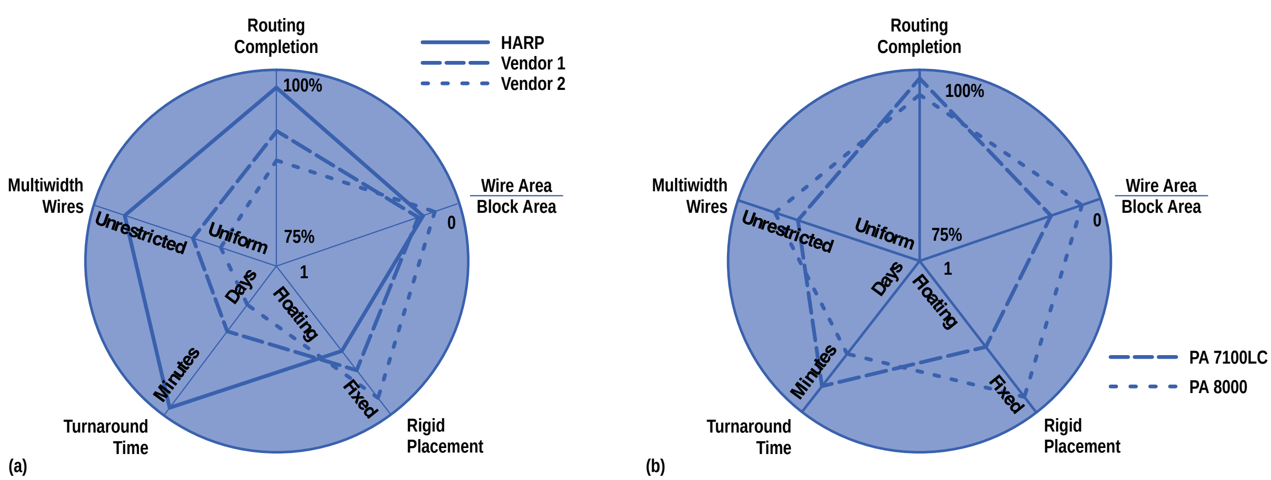

Fig. 1. *Radar charts showing (a) the capabilities of HP's existing HARP block router and third-party routers and (b) the needs of less aggressive chips like the HP PA 7100LC and more aggressive chips like the HP PA 8000.*

were well-suited for the existing HARP system. The more aggressive style of chips represented by the PA 8000, however, did not map to any existing block router offerings.

Significant changes in functionality and features usually entail a high amount of risk. On any design it is critical to manage the risk. Being an internal supplier, we are able to work much more closely with our customers by giving them greater visibility and control of the risks involved. This level of access is generally not available when dealing with third-party tool providers.

ICBD, as HP's internal chip supplier, is in the business of making and selling chips, not tools, so we face stricter requirements to justify any internal tool development. It is rarely cost-effective to build a router for a single chip. However, it has been our experience that the microprocessor chips have always pushed the limits of the technologies and the more general ASIC chips follow later after the bumps have been smoothed out. In looking at what was special about the PA 8000, we spotted several new technology trends that radically changed the block routing problem and might be adopted by future ASIC chips.

First, fabrication processes are starting to add many more layers of metal for interconnect. This change begins to invalidate the basic model used by traditional block routers of separate routing channels and blocks. Second, the need for higher off-chip connectivity is forcing a change in packaging technology. Solder bump packaging looks like the most viable means of addressing that need. Solder bump packaging is also being looked at for reducing packaging cost by mounting chips directly to boards. However, having solder bump pads in the middle of the chip breaks the traditional block router model of placing the pads at the periphery of the chip. Last but by no means least is a general trend of wiring delays becoming more significant than gate delays. Thus, the emphasis of routing is switched from minimizing chip area to minimizing interconnect delay.

Working with our PA 8000 customers, we prioritized the features and came up with a manageable subset needed for them to be successful. We then circulated a proposal to build the PA_Route block routing system. Given the time constraints and the ambitious goals, we had to take the drastic step of freezing the old system, HARP, with minimal support. We got agreement from all parties on the basis of strong support from the PA 8000 developers.

New Technologies Lead to New Constraints

The PA 8000 design team had decided that to be competitive, the PA 8000 chip would not only be more aggressive in its design, using superscalar, out-of-order instructions, but would also use a new process and new packaging. It is not uncommon for a microprocessor to use a new process, but this time they were moving from a three-metal-layer process to a five-metal-layer process. In addition, the increased I/O requirements of the design ruled out conventional packaging. The only mature packaging approach available was solder bump technology. With solder bump technology, the I/O pads are spread across the whole chip and are not just restricted to the periphery.

Analysis of the possibilities for extending or adapting the existing block router in the HARP system showed that its basic design intentions were not well-matched with the new requirements and could not be changed to make full use of the new technologies. HARP was based on the traditional channel routing paradigm, in which there are expandable routing channels between solid blocks. The channel router was limited to routing in three metal layers, while the new process had five. The solder bump I/O pads could be anywhere, but the block router could only attach to ports on the edges of a block.

Another more important restriction was that the solder bump port frame could not change and the blocks could not move with respect to the pads. There were two reasons for this. First, the board to which the PA 8000 was to be connected had a relatively long lead time, so its designers could not wait for the chip to be completed before getting started. Secondly, the solder bumps used to connect to the I/O pads emit alpha particles. If the placement were changed so that the pads became coincident with sensitive circuitry then unpredictable circuit behavior could result.

These requirements meant the block router could not grow the channels or move the blocks, a constraint for which our existing block router had only weak support. We jointly decided it was not feasible to attempt to automate the routing of the fifth layer of metal, since it was overly complicated by the requirements of the solder bump I/O pads.

In PA_Route we addressed as many of the new requirements as we could in the time available. We worked with the PA 8000 design team to pick the most important issues to address. This came down to two major features: being able to use the third and fourth metal layer resources over the top of some of the blocks and being better able to control the growth of the placement.

Our team is not often given the time to redesign our system, so we took advantage of the opportunity to add some long-desired capabilities. We added a more sophisticated port and net model, which we call *foliage*. With foliage we can describe the electrical characteristics of a port and a net. Pieces of artwork representing a port, for example, can be considered electrically equivalent (allowing stitching), electrically resistive (allowing connection to one of many but without stitching), or electrically open (specifying that all pieces of artwork must be connected). Foliage allows the router to be more flexible in using ports, since it uses this electrical model of the ports and it allows for more complex routing of nets in a channel. We also took the time to use more advanced software development techniques. We switched our design style from structured custom programming in the Ada language to object-oriented programming in the C++ language. This allowed us to attempt more complex algorithms and reuse existing component libraries.

The Building of PA_Route

The PA_Route system is composed of many components, including a netlist reader, an artwork reader (which models obstacles), a global router, a channel scheduler, and a detailed router. A viewer is used to examine intermediate and final results. Eventually the artwork is produced and then verified. Even though the design time of the PA 8000 is long compared to most ASIC chips, we did not have time to rewrite the whole system, so only three main parts were designed and implemented from scratch: the main database, the global router's over-the-block grid model, and the new over-the-block detailed router. We leveraged the rest of the system from the old HARP system with modifications to interface with the new database.

To minimize development time, we partitioned our development team into two parallel groups. One group started on the new database and started porting the old programs, while the other started implementing some of the new global router features in the old system. When most of the old HARP system was ported we ported the modified global router. This meant that the global router stayed in structured custom Ada code, which was the language used in HARP.

Database Changes

The capability needed by the PA 8000 design to route over the blocks required us to improve the expressiveness of the underlying database models used in routing regions. The new database allows us to model the obstacles and internal ports that we see in these over-the-block regions. The advanced port and net models (foliage) we implemented also required significant changes to the database. This not only allows us greater control and flexibility in routing, but also allows us to separate the act of global routing from the channel scheduler that calls the detailed router.

We developed automatic code generation technology to transform a graphical model of the database into code. The code generation technology was extended to support the C++ language and we began to work on the new input and output programs. With the database changes completed, we could begin porting the old HARP programs to the new database.

Global Routing

The general global routing problem is described at right. PA_Route incorporates a global router that understands rectangular blocks. The global router needed to be extended to support L-shaped blocks for the PA 8000. An L-shaped block is cut either horizontally or vertically into two rectangular components and special control is imposed on the channel between the cut components to keep the components linked together. The routing plane is divided into rectangular routing regions that meet only at T-intersections such that only two sides of a routing region have constrained ports. This somewhat restricted routing model comes from a conscious decision to avoid situations in which a routing region becomes a “switchbox” with ports on all four sides constrained to fixed locations. The more constrained switchbox routing problem generally requires more run time, creates more constraint cycles, demands clever rip-up and reroute strategies, and tends to leave more shorts for manual repair. We opted instead to concentrate our effort on providing more flexibility in the PA_Route global router for meeting user requirements and for achieving the smallest possible overall chip area.

The old HARP global router, like any traditional global router, assumes that blocks are black boxes, that the points for connections are on the edges of the black boxes, and that routing is confined to the channel areas between the blocks. That is, all routing resources inside the blocks are dedicated to the blocks' internal implementation only and therefore routing at the global level is not allowed to traverse through the blocks. This simplistic assumption was largely accurate in the days of two-layer and three-layer IC processes. With the advent of the HP CMOS14 process, which can have up to five routing layers, the assumption that routing resources inside a block are dedicated only to the block is no longer realistic. For the PA 8000, a good amount of metal 3 and metal 4 resources inside some child blocks are available for routing global nets.

Being able to use such over-the-block routing resources can lead to reduced signal timing, decreased channel congestion, and ultimately smaller overall routed chip size. Having judged that over-the-block routing was a critical factor to the success of PA 8000, the PA_Route team undertook a revolutionary change in the global router to support the routing of global nets over any block, provided that there are routing resources available over the block. The traditional global routing graph was augmented with a virtual grid model over each child block, a sophisticated net flow optimizer, and an efficient routing resource estimator. The grid model allows the lowest-cost path of a global net to traverse through any region over a block as long as there are free routing resources. The global router builds a detailed model of routing resources in each region (channel or block) and tracks free spaces in the regions based on a sophisticated density estimator that understands obstacles. The net flow optimizer minimizes jogging and distributes unavoidable jogs of different nets to different regions to reduce congestion. For connecting to the new solder bump I/O pads, which are inside some child blocks, the new global router was extended to support ports inside any block, with the restrictions that the ports be on selected port layers and that there be available routing resources in the block. The global router takes care of avoiding obstacles and ports on the edges of a block when inside ports are brought out of a block to form a lowest-cost path. The net flow optimizer also plays an important role in choosing an optimal exit point for the inside port so as to reduce unnecessary jogging.

The predetermined solder bump I/O pad locations for the PA 8000 force the placement to be unperturbed during routing. This is a hard problem for the global router. Not having the luxury of actually routing the nets during global routing, utilization of routing resources over the block as well as in the channel regions is controlled using a close estimate of detailed routing. A reasonably accurate and fast density estimator was incorporated into the PA_Route global router. Since routing over the block is allowed, the density estimator must understand prerouting and obstacles. A density check phase was introduced after the evaluation of the lowest-cost path of a net. If the path would exceed the routing capacity of one or more regions, the grid model in the global routing graph is modified to forbid further routing through the congested portion of the regions and an alternate lowest-cost path is sought. This checking process is repeated until either a clear path is found or no clear path is found, in which case the net is left unrouted to preserve the fixed placement. In addition to performing accurate density calculations, the global router also attempts to achieve minimal placement perturbation by automatically assigning nets to the less-dense layer wherever there is more than one routing layer available, and by optimizing net flow at region interfaces to reduce routing congestion.

For multipin nets, for which connectivity can have significant performance and density implications, port foliage was added to the PA_Route database to give the global router a model for determining port equivalency, while net foliage was introduced to allow the global router to generate more sophisticated physical connectivity for the shortest path. This combination of port and net foliage results in a high degree of control over the physical connectivity of a net. A designer can specify foliage explicitly or allow the PA_Route global router the freedom to optimize the global route by creating foliage as necessary to minimize the total wire length and to avoid congestion.

Over-the-Block Routing

To handle two important aspects of the PA 8000, a new over-the-block detailed router was required. The router had to handle obstacles in any routing layer and it had to be able to connect to ports located anywhere within the routing region.

Restrictions on the topology of the obstacles and ports were negotiated with the PA 8000 design team to relax the constraints of the over-the-block router to meet an aggressive schedule.

Child blocks were constructed by lower-level composition teams. Their design used the lower layers of metal to perform local interconnect and the upper levels of metal for intermediate levels of interconnect. The result was that partially used layers were made available to the over-the-block router to complete the intermediate and global level interconnect. The over-the-block router was given the responsibility of avoiding artwork created by the lower-level composition teams.

A review of existing detailed routing algorithms is presented ***Subarticle 5a***. The PA_Route over-the-block router is built on a new routing algorithm. It is based on a channel-like paradigm although it handles obstacles with arbitrary configurations. Layers are generally assumed to run in either the x direction or the y direction. Wires are routed assuming one direction is preferred, that is, in the preferred direction the wire runs for a longer distance and carries most of the current between a source and its sinks. By handling obstacles in arbitrary configurations, the over-the-block router extends channel routing concepts into an area-based routing regime. It retains many of the benefits of channel routing while being flexible enough to handle more complex routing topologies. This type of routing methodology will become more common as more layers are made available for routing. The over-the-block router can connect to ports not only on the sides of routing regions, but also in the middle of a routing region. The over-the-block router supports variable wire width and spacing, which gives the designers greater control over the timing delays of a signal. The over-the-block router reads the layout rules directly and does not abstract them into arbitrary routing constraints. Unlike other algorithms, our proprietary over-the-block algorithm does not require wires to be “binned” according to their width and spacing, and it does not rely on a compaction process to achieve optimal density.

The over-the-block router contains the features needed for high-speed, performance-driven critical designs such as the PA 8000. It handles complex via structures necessary for high-performance designs by allowing the intersection area to be expanded beyond the size dictated by individual metal connections. While it supports a high degree of manual control, the over-the-block router is also reasonably fast, making multiple design turnarounds feasible.

The over-the-block router models the routing problem as two-dimensional line segments that represent the largest wiring component of a net. This *trunk* is assumed to cause most of the parasitic delay and the overall goal of the algorithm is to find an optimal ordering of these trunks to generate a dense packing and avoid obstacles. Each trunk and each obstacle becomes a node in a graph. The edges in the graph model the vertical pin constraints of each wire and the horizontal constraints of that trunk’s placement relative to other trunks (see Fig. 2). The total graph contains the weighted constraints of all trunks in the routing region. Thus, each trunk is considered for placement during each phase of edge direction assignment, and the net ordering difficulties of other routing schemes are avoided. The general nature of the edge selection allows other constraints such as cross talk and delay to be modeled in future versions.

The algorithm can handle any number of layers and is not rigidly required to follow layer-per-direction constraints for vertical components (i.e., connections to ports) or trunk components. When constraints occur, the over-the-block router tries several schemes to alter the topology of the wire, such as removing the constraints. The scheme includes *jog insertion* and *wrong-side segmenting* in various forms.

If the over-the-block router cannot complete a route, it produces a spacing violation or short circuit along with associated diagnostics and completes the route. When this occurs, the user has the option of fixing the short manually or altering the routing problem for the region by such methods as growing the placement, constraining the global routing with capacity controls, or other means.

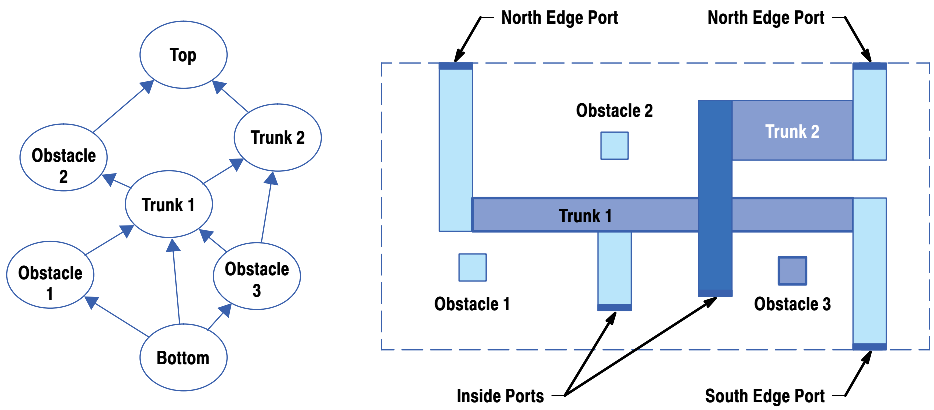

Fig. 2. *To the over-the-block detailed router, each wiring trunk is a node in a graph (a). The edges in the graph model the vertical pin constraints of each trunk and the horizontal constraints of that trunk's placement relative to other trunks. (b) Routing plan. Shades of gray represent different metal layers.*

Although specialized to handle the routing problems of the PA 8000, the over-the-block router was built to handle the general channel routing problem. No shortcuts were taken that would compromise robustness for the general case in the expectation that the router could be leveraged for other designs.

Conclusion

A new block routing system called PA_Route was built specifically to address the needs of high-performance, leading-edge IC designs. PA_Route contains significant features built on new technology while leveraging existing code to minimize risk. It was designed to be extendable to address future issues as they arise. It was used successfully to route the PA 8000 chip and did not impact its schedule despite the high levels of risk involved. Fig. 3 shows the areas of the PA 8000 chip where PA_Route performed block-level routing. The features and limitations of the system were carefully designed with close cooperation between the PA 8000 design team and the CAD development team. Many alternatives were analyzed using design-critical issues as the measurement criteria. Balancing immediate and future chip design needs was given high importance in the design of PA_Route so that the system can continue to be used for future designs.

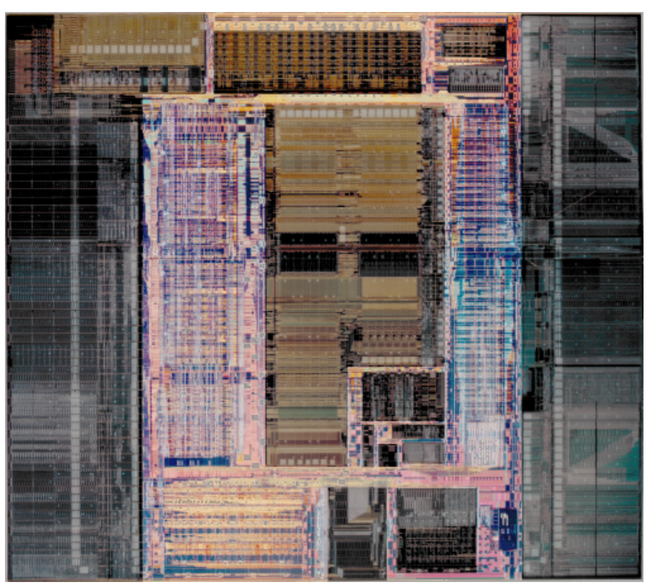

Fig. 3. *PA 8000 CPU chip with highlighted areas showing where PA_Route performed block-level routing.*

Acknowledgments

The PA_Route team would like to thank the members of the Engineering Systems Lab in Fort Collins with whom we worked closely, especially Julie van Gelder and Tom Asprey. Julie was invaluable in helping define features and requirements. Tom showed much patience and persistence with the early releases of PA_Route. We would like to acknowledge Dell Lee of the ICBD Design Technology Center, as well as Michaela Guiney and Eric Slutz, for supporting our efforts. Thanks to Sandy Fogle and Gaylord Tefft for providing a photomicrograph of the PA 8000.

PA_Route is available to designers within HP. The URL for PA_Route is http://emerald.dtc.hp.com/pa_route/general.html.

- Go to Subarticle 5a
- Go to Subarticle 5b
- Go to Next Article
- Go to Journal Home Page