The future of design automation

Successfully automating an electronic product development process can speed up design, but only if all constraints defining the process fall within the limits of the design environment. As design disciplines converge, the scope of design automation is changing. This article examines what makes for good design automation and how this influences the way your tools will evolve in the future.

Someone – probably famous – once said that in product design, having the idea is the hardest part. That person almost certainly never had to lay out a 32-bit processor board by hand using tape and tracing paper. If they had, then they might have said that the hardest part of product design is balancing the complexity of the design with the time needed to implement it.

It’s no accident that PCB design was one of the first electronics design tasks to which computer technology was applied in an effort to automate the process. The Electronic Design Automation (EDA) industry really had its roots in the need to handle the explosion in circuit complexity brought about by the advent of digital electronics and, in particular, the rise of the microprocessor in electronics design. Traditional manual design techniques became too cumbersome to manage when faced with the onslaught of higher pin counts, multi-line signal buses and growing numbers of board layers.

Today it’s unthinkable that any modern electronic design would be done without the use of computer-based tools to automate many of the processes across all the design disciplines. However, design automation is not a simple matter of feeding data to a software algorithm. It must tread a fine line between assisting the designer to get a task done more quickly and efficiently without hindering the desired outcome by assuming too much control.

Automating the routing process

Most PCB designers today take the inclusion of an autorouter in a PCB design system as a given. Autorouters are rarely if ever used to route an entire board, but the majority of designers these days will use an autorouter on significant portions of a board. It’s fair to say that autorouters are now recognized as a productive and useful tool in the layout process. But this hasn’t always been the case.

Early autorouters were notoriously difficult to set up, requiring an in-depth knowledge of the application in order to achieve acceptable results. From the PCB designer’s perspective, the time saved by the automation process was often lost in the time taken to set up the autorouter; the complexity was simply moved from one area to another.

In order to make the automation useful, by necessity the tool vendor must hide complexity from the user and make decisions on the appropriate way to handle certain situations. Striking a balance between lowering complexity and providing enough control over the process is a perennial problem facing all EDA tool developers. In the case of autorouters, this balance has largely been achieved by a combination of better configuration tools and a greater understanding of the technology by both the vendors and the users.
Contrasting with autorouting is the area of automatic component placement. Like routing, placement is an iterative process involving many conflicting constraints, but to date, automatic placement tools have not achieved widespread acceptance in the industry. But this is not necessarily, as is commonly believed, a product of the quality of tool or the sophistication of the algorithms used.

Where autoplacement differs from autorouting is in the constraints that drive the process. With routing, the constraints are largely dictated by the characteristics and parameters of the board itself. These properties are easily modeled within the PCB design environment because they are properties of the board design – number of layers, spacing between tracks, connection distance, layer direction, etc.

The constraints dictating component placement, however, can be driven by mechanical considerations such as the shape of the product casing, by ergonomic issues such as the positioning of buttons and connectors, or by other concerns such as heat dissipation and optimization for pick-and-place machines during fabrication.

The fact that these constraints are not properties of the board design, but belong to the wider product design, means that trying to accurately model these properties in the PCB space to drive placement introduces significant overheads for the PCB designer. Until electrical and mechanical CAD systems are fully integrated, and the design constraints imposed by both realms can be exposed in an integrated system, the application of automation to board-level component placement is an inefficient prospect.

With the recent surge of development in FPGA technology, some industry analysts believe we are at the start of a paradigm shift in electronics design. While FPGAs have been around for a while, a watershed has been reached in the price/capacity curve for these devices. They offer many of the benefits of ASICs without the high NRE costs and long development cycles. The inherent reprogrammability of these devices offer huge cuts in design time, and the tantalizing prospect of field hardware upgradeability. In short, they have the potential to change the way we can approach electronic product development, similar to the way that affordable and powerful microprocessors opened up new design possibilities decades ago.

To take full advantage of the potential offered by programmable hardware however, engineers need tools that allow them to exploit the benefits of the technology within the wider design process. For example, the large number of configurable I/O pins on high-capacity FPGA devices makes the task of managing the synchronization of pin mapping between the FPGA design and the board design a complex one. To do it manually can slow down the design cycle considerably. Indeed the benefits of working within a programmable platform can be negated by the lack of automated solutions for dealing with these devices at the PCB level.

But as we have seen with autoplacement, successfully automating a process that spans different design disciplines is difficult because of the lack of integration between the different design environments involved. The constraints that drive the process – in this case the limitations on pin interchangeability and the connectivity paths of the FPGA pins on the PCB – exist with different design tools.

To intelligently automate the process of optimizing the pin-out of a large FPGA for routing at the board level requires that the PCB design editor has an intimate knowledge of the physical characteristics and capabilities of the target FPGA device and any overriding constraints placed on pins by the FPGA designer. It also requires tight interaction between the FPGA and PCB design environments in order to be able to quickly iterate between FPGA and PCB-level design constraints in order to arrive at an optimal solution that balances the timing requirements within the FPGA with routability issues on the board.
A unified design environment

The true potential of programmable technology such as FPGAs lies in the ability to blur the lines between hardware and software. As we increasingly make use of this potential, the design of hardware, software and programmable hardware is converging. From a design tool perspective, this convergence will drive the need to automate processes across traditional design tool boundaries. To do this successfully will require more than just superficial integration between different sets of design tools. It will require a complete unification of the different design environments involved.

Altium has recognized this imperative in its Altium Designer system. Altium Designer provides a single, unified environment that brings together PCB-level hardware design, programmable logic design and embedded software development. It does so in a way that allows all engineers, designers and developers to take full advantage of the benefits of programmable hardware to create more intelligent electronic products, regardless of their level of FPGA expertise.

By unifying design, Altium Designer supports practical automation across design disciplines – such as between the FPGA and the PCB. In doing so it opens up new design possibilities and unlocks the full potential of the latest generation of programmable devices.

Raising the level of abstraction

Electronics design is an incredibly complex process. When you consider that we’ve come from circuits containing just several transistors to several hundred million in the space of less than half a century, it almost defies belief that this complexity can be managed and harnessed by an engineer working in an office on a desktop computer. Indeed, the increased power of PCs themselves have greatly facilitated the advanced automation that most designers now take for granted.

Increased automation of the design process does not necessarily drive innovation in the electronics industry. What it does is raise the level of abstraction that a designer works at, empowering them to do what they do best. Design time can then be spent adding value at a creative level, rather than dealing with the complexities of implementation.

For automation to be useful, however, it must be done within an environment that encompasses all of the constraints necessary to fully specify the boundaries of the process being automated. High-capacity, low-cost programmable devices provide a vast design space that facilitates the convergence of hardware and software. In order to provide effective automation within this design space, the design system we use must unify what are currently considered separate design disciplines.

Ultimately the goal of all design automation is to maximize the effectiveness of the most valuable design asset – the creativity of the designer operating the controls. Operating within a unified design environment provides the freedom necessary to harness the full range of electronics design technology and devices available today, and supports the automation necessary to work at higher levels of abstraction, add increasing levels of intelligence to products and get the job done faster.