AI Magazine Volume 10 Number 4 (1989) (© AAAI)

some insightful comments. If you are not familiar with them, you can learn by tracing the development of some foundational ideas. This book should have a deserving place on any bookshelf.

Design Automation: Automated Full-Custom VLSI Layout Using the Ulysses Design Environment

Marwan A. Jabri

*Design Automation: Automated Full-Custom VLSI Layout Using the Ulysses Design Environment (*Academic Press, Boston, Massachusetts, 1988, 463 pages) by Michael L. Bushnell deals with an interesting and challenging idea in very large-scale integration (VLSI) computer-aided design (CAD): the implementation of a top-down design process based on a script. A system called Ulysses that implements such a script-based approach within a blackboard architecture is described. The primitive operations in the script are the actions of specialized design tools such as floor planners, automatic layout generators, and signal routers. Ulysses is destined to be used in a top-down design methodology, which achieves compact and efficient VLSI circuits, now primarily a manual, time-consuming process.

On the positive side, the book is generally well written and well presented. The chapters are well structured, and the flow of ideas within them is smooth. Paragraphs chain well, and arguments are generally well constructed.

On the negative side, I found it disruptive to have Chapter 3, "The VLSI Design Process," coming between Chapter 2, "Prior Design Environments," and Chapter 4, "Using AI in Environments." To appreciate prior design environments, readers need to have a feeling for the VLSI design process. Reading Chapter 3 before Chapter 2 is a solution. Also, few graphic supports are used in the presentation, which is surprising for a text on CAD and VLSI. The quality of the figures is mediocre when you consider what can easily be achieved with today's desk top publishing systems. Additional work on the (few) figures used would have greatly helped the reader.

Although the book deals with many important technical issues, I have a number of concerns. The biggest concern is that although the major theme is interesting, the actual implementation is quite poor and disappointing. The idea of the top-down design script is to have an environment where CAD tools can easily interface and be used in the design process, without the burden of file and syntax transformations. Therefore, the contribution of the work described in the book would be based on the demonstration of such a script system, with a number of design tools to prove the usefulness of the idea. However, only two tools were integrated (the third was manually simulated, which itself is awkward) in the script environment, which considerably reduces the power and authority of the demonstration. This disappointing demonstration might be the result of the project's ambitious nature and the immaturity of the CAD tools as low-level operators to a high-level design system.

The book is misleading in its treatment of some key points. The text reads:

> Also Weaver is not set up for interactive control of routing. The designer's input can be manually added to the blackboard before routing begins, but he cannot change this information once Weaver has begun to route. It would be highly desirable if the designer could interrupt the routing system and add his own manually-constructed routes of subnets when he observed that the router was floundering. Of course such interruption capability would require an additional user communication expert, so that the user would not have to edit raw blackboard structures directly. (p. 51)

The author is criticizing the capability of the Weaver system (a knowledge-based circuit interconnections router) to restart, continue (that is, to be interrupted), or accept that a user might specify some routing channels. This criticism implies a misunderstanding about what current state-of-the-art routing tools are able to achieve. The problem here is not the communication expert in the blackboard architecture. Any routing expert will have difficulty trying to deal with incomplete (partial) solutions while being interrupted and having its current solution modified. Expecting the router to continue under these circumstances is too much.

A big surprise in the book's technical content was that important work in AI-based VLSI CAD was not referred to, yet seemingly unrelated work was repeatedly cited. Here are some examples:

The work of the AT&T group of Brian Ackland was not even mentioned. This group has completed a number of AI-based VLSI CAD systems, including Cadre, which is the most relevant to the book. However, the book mentions the Redesign system (pp. 55–56), which deals with MSI TTL circuits, which are only distantly related to VLSI circuits. Bushnell also discusses the Vexed and Critter systems:

> The work on Vexed and Critter looks extremely promising; however, the work is not really extendible to the VLSI design domain because these systems do not consider chip floor planning, cell placement and routing problems.

Bushnell's mention of SADD, also an MSI TTL target system, is another reference to an irrelevant system:

> SADD appears to be successful, but like Vexed, it is difficult to extend to the VLSI IC design domain because SADD does not consider floor planning, cell placement and routing. (p.59)

Some details might be confusing to the unprepared reader; for example, Bushnell says:

> Certain modules must also be designated as primitive logic gates and transistors. Various attributes must be defined for the VLSI layout task. For modules, these include: possible module aspect ratios, area....(p.72)

This confusion is typical where module aspect ratios are being used as a parameter with no mention of the target technology being used, whether it is semicustom or full custom. The target technology is important: This book targets full-custom design, and aspect ratios have no meaning in a full-custom design methodology when no dimensions are specified.

The author also draws some conclusions that his supporting evidence does not seem to uphold:

> Ulysses, like HASP, is driven by models of its design activities, which are a codified design methodology. VLSI design methodology can be viewed as an extreme generalization of the HASP system operating in model-driven mode. High level design knowledge sources hypothesize designs, and low-level KS's verify or reject these hypotheses. This approach significantly reduces computation costs of HASP and is the only viable one for a design synthesis system. (p. 110)

Saying that this approach is the only viable one for a design synthesis system is quite a strong statement to leave unsupported. Such a statement needs to conclude a discussion on different viable systems for design synthesis.

The author is apparently unaware of the work at Sydney University or Adelaide University, where quality factors have been used to measure the quality of a solution and to backtrack if it is not above a certain threshold. On page 155, the text reads:

> Several additional interactive control facilities have been pioneered by Ulysses. Both the designer and the rating policy KS can decide that the current design point is inferior to another one and switch design activities to the other design point.

On page 168, automatic rule generation is mentioned. It is surprising that a claim is made about the capability of Ulysses to automatically generate rules (machine learning) without any reference to the work of Ross Quinlan on machine learning using induction decision trees. Only the work of Meta-Dendral is mentioned.