

Research Opinion

The Potential for Mechanical Design Compilation^{*†}

Erik K. Antonsson, Ph.D., P.E.
Engineering Design Research Laboratory
Division of Engineering and Applied Science
California Institute of Technology
Pasadena, California 91125

March 17, 1997

Abstract

Significant potential exists for engineering design research to produce viable approaches to automatic compilation of mechanical systems.

In a recent article in a prior issue of this journal [9], Dr. Whitney articulated his view of why the design of mechanical systems is not now, and will not be, similar to the design of VLSI systems. An alternative view suggests that his conclusions are overly negative, and that significant potential exists for the development of approaches to compilation of mechanical systems (in analogy to compilation of digital VLSI).

Dr. Whitney nicely summarizes some of the basic aspects of VLSI and mechanical design in the first few pages of his article. In the latter sections he raises several good points about the intrinsic aspects of mechanical design that introduce difficulties, including:

- the back-loading of upstream subsystems from downstream functions:

“... VLSI elements do not back-load each other. That is, they do not draw significant power from each other but instead pass information or control in one direction only. These facts combine to

*Manuscript prepared for submission to *Research in Engineering Design*.

†Word Count: 1825 words.

permit VLSI circuit elements to be connected together in building-block fashion [in a way that mechanical components cannot.]” [9, Page 136];

- the importance of “scale-dependent” effects:

“... in the all-important arena of time- and scale-dependent side effects like fatigue, crack growth and corrosion, ... geometric details too small to model or even detect are conclusive in determining if the [failure] will occur.” [9, Page 136];

- the complex and desirable sharing of function in order to achieve size, weight and energy efficiencies:

“[M]echanical designers ... depend on the multifunction nature of their parts to obtain efficient designs. Building block designs are inevitably either breadboards or kludges.”

“For example, the outer case of an automatic transmission for a car carries drive load, contains fluids, maintains geometric positioning for multitudes of internal gears, shafts and clutches, and provides the base for the output drive shafts and suspension system.” [9, Page 135],

- and the profound problems introduced by the need of mechanical systems to control the flow of significant levels of power:

“Multiple side effects at high power levels are a *fundamental* characteristic of mechanical systems.” [9, Page 134].

These differences between mechanical system design and VLSI design are real and significant, and the difficulties that they introduce must be surmounted by any viable approach to automatic generation of mechanical designs.

However, much of Dr. Whitney’s paper mis-compares mechanical and VLSI design. The many virtues that he (and others) ascribe to the design of VLSI systems are limited to digital VLSI. Vast areas of electrical engineering beyond this narrow area are not included and do not benefit from highly automated design techniques. For example, analog design (of all types), microwave design, electric motors, power systems, and many others do not enjoy the benefits of highly automated design systems. Therefore his comparison of the design methodologies for the narrowly defined area of digital VLSI should be with design systems for similarly limited areas of mechanical design, rather than with the immensely broad field of “complex electro-mechanical” design.

Design of digital VLSI systems may yield useful lessons for design in other domains, however, initial inquiries into the applicability of these lessons should be in similarly narrow sub-domains. Al Ward's mechanical compiler of hydraulic systems [5, 6, 7] is an instructive example that borrows many of the essential aspects of digital VLSI design. The Nippondenso automobile panel meter component interchangeability example [8] that Dr. Whitney cites on Page 137 of his article [9] to illustrate an exception to his thesis (*e.g.*, when the fabrication process is assembly) is indeed a telling counterpoint. One could argue that a viable and likely successful approach to mechanical design automation will be (initially) in areas where design is dominated by "assembly" of "components", as in VLSI cell libraries, automobile panel meters, hydraulic systems, etc. This approach to design (assembly or integration of modular components) is nicely described by Dr. Whitney as follows:

"... a library of *generic* product components. Design of a *specific* product occurs at the system level by combining modules made of verified devices found in the library." [9, Page 130]

"[a] modular approach works sometimes, but not in systems subjected to severe weight, space or energy constraints; in constrained systems, parts must be designed to share functions, or do multiple jobs; ..." [9, Page 131]

Most mechanical consumer products are not subject to severe weight, space or energy constraints. Kitchen appliances, office appliances, children's toys, pumps, machine tools, even internal combustion automobiles fall outside these severe constraints. Clearly aircraft and other high-performance devices are subject to these (and other) constraints, but there remains considerable room in the commercial world where the design and performance constraints are not so severe as to preclude some design automation.

As a further counterexample, consider a bicycle. Clearly these devices are subject to severe weight and energy constraints. However, even world-class racing bikes are nearly completely composed of modular components,¹ with the exception of the frame. Even the frame can be (and increasingly is) welded from tubes selected from a library/catalog of standard lengths and semi-custom built to fit the customer.² While it is only one additional counterexample, it indicates that Dr. Whitney's pessimism concerning the promise of mechanical design compilation is not justified.

Throughout his article, Dr. Whitney puts forward the view that digital VLSI design is undergoing a fundamental change away from automated instantiation of

¹Hujsak Bicycles, Wimberly, Texas, <http://internetnow.com/hujsak/>

²Performance Elite Catalog, Spring, 1997, Performance Bicycle Shop, P.O. Box 2741, Chapel Hill, NC 27514, <http://www.performancebike.com/>

components from standard cell libraries. Unfortunately, he does not cite any references for this position (other than personal communications). Interestingly, an article published in *IEEE Spectrum* [3] simultaneously with Dr. Whitney's article strongly refutes this view. This article indicates that digital VLSI is indeed changing, but changing further in the direction of design by instantiating ever more complex modular sub-systems from standard libraries (thus the required increases in complexity, speed, and density are achieved by making libraries of digital logic sub-functions (shift registers, multipliers, etc.) with greater functionality, complexity, speed, and density, instead of libraries of lower-level functional units such as logic gates).

“The complexity of large designs calls for a shift in the design paradigm to one based on reusable, high-level building blocks.” [3, Page 35]

It is indeed true that the three-dimensional field effects of signal and power flow in a digital VLSI chip become more significant as line widths decrease and clock speeds increase. However, the overwhelming need for automated design methodologies to permit the design of systems with many millions of components (transistors) also increases. Dr. Whitney seems to suggest that the design of digital VLSI systems will become more and more manual as the complexity of these devices increases. This design task will certainly become more complex and difficult, however, he offers no independent support for his position, and it defies credulity. If humans are unable to economically design a microprocessor with 10^6 components using manual methods, on what basis would one suggest that it will be necessary to (more) manually design one with orders of magnitude more components?

Importantly, Dr. Whitney does not mention a significant element of design automation, namely the cost of the time and other resources to create a design. In looking back at software design, the advent of assembly languages and assemblers created concerns among machine-language programmers that assemblers would not exploit the nuances of the machine instructions, and that sub-optimal code would be created. It certainly was, but at such a savings in design (coding) time that the trade-off was clearly in favor of assembly languages. Similarly, the advent of compilers created a similar controversy among assembly language programmers. A skilled assembly language programmer could develop highly optimized code (at the cost of many hours of programming time). A compiler could create sub-optimal (but satisfactory) code in a tiny fraction of the time. Naturally, the sub-optimal compiled code has won (historically) because of the enormous savings in design resources [1, 2, 4].

Digital VLSI has experienced the same transition, away from specialists laboring long over completing (and optimizing) one particular design, to a wider

community of less-specialized designers rapidly creating many sub-optimal (but satisfactory) designs with significantly less design time and cost.

Dr. Whitney mentions part of this issue in connection with the early developments of structured design methods for digital VLSI:

“The practical result is that [modular, standard-cell digital VLSI] designs take up more space than space-optimized designs produced by EEs in the 1970s.” [9, Page 132]

However, he does not mention the concomitant advantage of these (somewhat) sub-optimal designs, namely the enormous savings in design resources needed to create them. Indeed, economic forces, not legislation or coercion, resulted in the adoption of structured (and therefore sub-optimal) digital VLSI design methods by the commercial microelectronics industry. Structured methods of digital VLSI design are pervasively used because they produce (and appear to be able to continue to produce [3]) systems at an economically favorable trade-off between performance, efficiency, fabrication cost, and design cost.

Many authors, including Dr. Whitney, have raised the important point that the nature of many mechanical designs requires them to be nearly optimal in order to operate at all (he uses the example of balancing a turbine assembly), however, one shouldn't conclude from this that the idea of mechanical compilers or mechanical design automation is impossible, only that developing them will be challenging, and that initial work in this area should focus on narrowly defined sub-areas of mechanical design.

I would turn Dr. Whitney's "Final Remarks" section around and pose his negative conclusions as positive research challenges:

- What (narrow) sub-domains (of mechanical systems) can be determined to permit high-quality designs through the re-use of library elements?
- What languages can be developed to specify mechanical function?³
- In what sub-domains can a “clean separation” between design and manufacturing in mechanical systems be found?
- In what sub-domains can systems be designed independently of components?

³Indeed, despite his negativism, Dr. Whitney himself repeatedly points out a powerful example: bond graphs, but also indicates that they are limited to representing power flows. Should one conclude from this one example that no others can be developed? Or should one conclude that this one example suggests that others might be developed (particularly for carefully specified narrow sub-domains of mechanical system design, equivalent in narrowness of definition to digital VLSI)?

- In what sub-domains can logical homologues or homomorphisms (or other mappings) between system requirements and embodiments be found or established?

Dr. Whitney articulately illuminates the many challenges in pursuing this work, and clearly indicates the enormous (and increasing) complexity in the design of mechanical systems. Stated positively, instead of negatively, his paper outlines a productive and valuable agenda for research in engineering design.

Acknowledgement

Tom Peters' many helpful thoughts greatly improved this article.

References

- [1] ASPRAY, W., AND BURKS, A., Eds. *Papers of John Von Neumann on Computing and Computer Theory*. MIT Press, Cambridge, MA, 1987.
- [2] ECKHOUSE, JR., R. H., AND MORRIS, L. R. *Minicomputer Systems: Organization, Programming and Applications (PDP-11)*, second ed. Prentice-Hall, Englewood Cliffs, NJ, 1979.
- [3] HUNT, M., AND ROWSON, J. A. Blocking in a system on a chip. *IEEE Spectrum* 33, 11 (Nov. 1996), 35–41.
- [4] SEBESTA, R. W. *PDP-11 Structured Assembly Language Programming*. Benjamin/Cummings Publishing Company, Menlo Park, CA, 1985.
- [5] WARD, A. C. *A Theory of Quantitative Inference for Artifact Sets, Applied to a Mechanical Design Compiler*. PhD thesis, MIT, 1989.
- [6] WARD, A. C., AND SEERING, W. P. The performance of a mechanical design compiler. memo 1084, MIT Artificial Intelligence Laboratory, 1989.
- [7] WARD, A. C., AND SEERING, W. P. Quantitative inference in a mechanical design compiler. memo 1062, MIT Artificial Intelligence Laboratory, 1989.
- [8] WHITNEY, D. E. Nippondenso Co. Ltd: A case study of strategic product design. *Research in Engineering Design* 5 (1993), 1–20.
- [9] WHITNEY, D. E. Why mechanical design cannot be like VLSI design. *Research in Engineering Design* 8, 3 (1996), 125–138.