

*Massachusetts Institute of Technology*  
*Engineering Systems Division*

**Working Paper Series**

**ESD-WP-2003-01.03-*ESD Internal Symposium***

---

**PHYSICAL LIMITS TO MODULARITY**

---

**Daniel E. Whitney**

**MAY 29-30, 2002**

## PHYSICAL LIMITS TO MODULARITY

by

Daniel E Whitney

### ABSTRACT

Architecture, specifically the definition of modules and their interconnections, is a central concern of engineering systems theory. The freedom to choose modules is often taken for granted as an essential design decision. However, physical phenomena intervene in many cases, with the result that 1) designers do not have freedom to choose the modules, or 2) that they will prefer not to subdivide their system into as small units as is possible.

A distinction that separates systems with module freedom from those without seems to be the absolute level of power needed to operate the system. VLSI electronics exemplify the former while mechanical items like jet engines are examples of the latter. It has even been argued that the modularity of VLSI should be extended to mechanical systems. This paper argues that there are fundamental reasons, that is, reasons based on natural phenomena, that keep mechanical systems from approaching the ideal modularity of VLSI. The argument is accompanied by examples.

This paper is an updated version of one written 6 years ago,<sup>1</sup> which in turn is based on several limited circulation reports and working papers.<sup>2</sup>

### 1. Introduction and Historical Note

A number of important military and commercial systems fall into the class of “complex electro-mechanical-optical” (CEMO) items, examples of which include missile seeker heads and instant cameras. Each of these contains motors, sensors, control systems, optical trains, and, in the case of the camera, a complete chemical system for developing images on film. About 15 years ago, the author and his colleagues had research support from DARPA to advance the science of modeling and designing complex mechanical assemblies. The authors made some progress in their research [De Fazio, et al] but DARPA turned its interest increasingly toward electronics and VLSI. Our sponsors urged us to “get smart” and emulate VLSI design and fabrication methods in

---

<sup>1</sup> [Whitney, 1996]

<sup>2</sup> [Whitney, Nevins, De Fazio and Gustavson]. The author wishes to acknowledge the contributions of his co-authors of this report.

order to advance the art of CEMOs. We, in turn, argued, unsuccessfully, that CEMO systems were fundamentally different from VLSI and could not be conceptualized or designed as VLSI systems are. [Whitney, Nevins, De Fazio and Gustavson] [Whitney, 1996].

The distinction between typical mechanical systems and VLSI has gained new relevance as attention has turned to developing a theory of engineering systems. Key to that theory is the concept of architecture, the scheme by which functions are allocated to physical objects and the scheme by which those objects interact. [Ulrich and Eppinger]. Architectures are often characterized by the degree to which they are “integral” or “modular,” and many arguments are advanced in favor of modular architectures. In this paper we will argue that designers of mechanical systems do not have as great freedom to define modules or to choose the degree of modularity as do designers of low power systems like VLSI. To the extent that this is true, the theory of engineering systems will have to take account of such fundamentals while evolving metrics for evaluating architectures and defining system design techniques.

## **2. The Attractiveness of the VLSI Model of Engineering Design**

It is widely agreed that design methods and especially computer support of design is generally more mature in electronics than it is in CEMO products. This realization has given rise to speculation that VLSI digital design and manufacturing methods might be applied to CEMO products with good results. The question is whether there are fundamental blockages to such a transfer of method, or whether the transfer has not taken place simply because of inertia or lack of appreciation of the potential benefits.

Claimed benefits of the VLSI design paradigm include:

Design benefits: VLSI systems are extremely complex, small, and efficient, and can be designed by relatively few people empowered by well-integrated design tools; a microprocessor with 3 million "parts" can be designed and brought to full production in three years by about 300 people, whereas a car with about 10000 parts requires the efforts of 750 to 1500 people over about four years, and an airplane with 3 million parts may require 5000 people five years. Furthermore, the different VLSI modules can be designed relatively independently and thus in parallel, saving time. VLSI modules can be given standard interfaces, permitting plug and play design and opening up whole industries to new kinds of competition and innovation.

Manufacturing benefits: the "same" manufacturing processes or even the same manufacturing equipment can be used to make an endless variety of VLSI items; by contrast, especially at the most efficient high volumes, CEMO production facilities are dedicated to one design or at most a few variations of limited scope.

“Business” benefits: Product architectures can be tailored to the way a product will be sold or distributed. A more modular architecture permits modules to be identified as the differentiators that will be customized for different purchasers. Differentiation can occur at attractive points in the product delivery process, such as at the very end of the assembly line, at the distributor’s place of business, or even by the customer. [Lee] Modular architectures lend themselves to outsourcing, permitting companies to share risk or gain access to knowledge and capabilities not available in-house. [Fine and Whitney] It has even been argued that modularity is a fundamental source of value in systems because it affords opportunities for innovation, provided that certain “design rules” are followed. [Baldwin and Clark]

Are these benefits transferable from VLSI to CEMO items? To begin the discussion it is necessary to classify CEMO items and choose one class for further discussion.

CEMO products can be classified roughly as follows:

- those that are primarily signal processors
- those that process and transmit significant power

Examples of the two classes can be found in Table 1.

SIGNAL PROCESSORS	PROCESS AND TRANSMIT SIGNIFICANT POWER
four digit mechanical gear gas meter dial (1 mW?)	• Polaroid camera (30W peak?)
ball-head typewriter (30 mW peak at the ballhead?)	• missile seeker head( 50W peak?)
sewing machine (1 W?)	laser printer (1 KW, much of which is heat)
Marchand calculator (10W?)	automobile automatic transmission (50 KW+)
	automobile (100 KW+)(half or more dissipated as heat from engine)

	airplane (10 MW±)
	ship (40 MW+)

**Table 1. Examples of CEMO Signal Processors and Power Processors**

The distinction is not merely academic, for two reasons. A major trend in recent decades has been the replacement of mechanical signal processors first by analog electronics and more recently by digital electronics. Signal processing behavior is generally carried out more economically, accurately, and reliably by electronics. The replacement is physically possible because signal processing is, or can be, accomplished at very low power levels because the power is merely the manifestation of a fundamentally logical behavior. The power itself is not actually required to perform any physical function, such as motion.

However, the replacement has not occurred where significant power is the basis for the system's behavior and the main expression of its basic functions. The discussion that follows focuses on such power-level CEMOs. The presence of significant power in CEMOs and its absence in VLSI is the root of the reasoning in this paper.

### 3. Sketch of VLSI Design

There are basically three classes of VLSI, distinguished by the aggressiveness of their design in terms of circuit density, size of individual circuit elements, and width of connecting lines: dynamic random access memories (DRAMs), microprocessors, and application-specific integrated circuits (ASICs). DRAMs represent the cutting edge, requiring the smallest features and redesign of individual device elements at every new generation. ASICs are at the other end, having relatively large devices and line widths and relatively fewer devices on a chip. Microprocessors are in between.<sup>3</sup>

---

<sup>3</sup>Information for this section was obtained from the following sources: interview 9/19/94 with Fred Harder of Hewlett-Packard; various discussions with Gene Meieran of Intel during 1994 and 95; presentation to the MIT VLSI Seminar series by Ted Equi of DEC, March 15, 1994; presentation "Trends in Integrated Circuit Design" by Mark Bohr of Intel, 11/22/94; proceedings of NSF Workshop on New Paradigms for Manufacturing chaired by Dr Bernard Chern, May 2-4, 1994 and discussions with symposium participants, especially Carver Mead and Carlo Sequin; members of the ad hoc National Research Council study team on Information Technology in Manufacturing, especially Louise Trevillyan of IBM and Gene Meieran of Intel; presentations "Component Design Process" by John Dhuse and "Mask Design" by Barbara Christie, Mark Chavez, and Tara Brown, all of Intel, Feb 1, 1994.

A generic approximate list of the steps comprising design of a microprocessor is as follows:

In Stage 1, elementary devices are created, validated, and entered into a library along with design rules and associated analysis tools that reasonably guarantee successful fabrication. In Stage 2, complex systems are created as designers draw standard validated components from the library and hook them into systems. In stage 3 the item is manufactured. A more complete description is in Table 2.

<u>Stage 1</u>
1. a set of design rules (line widths, device sizes) is established
2. elementary devices, such as gates, are designed, and simulations of their behavior are generated
3. processes are designed and validated for these devices (this can be a very lengthy and difficult step, involving severe materials, chemical, optical, thermal, stress, and other problems; dozens or hundreds of people and millions of dollars may be required) <sup>4</sup>
4. validated devices are entered into a cell library, along with their simulations; the design rules for these devices, which reasonably guarantee successful fabrication, are imbedded in the cell layouts

<u>Stage 2</u>
1. systems are designed on a top-down basis, proceeding from flowcharts to logic diagrams to circuit diagrams or device interconnections, each step of which can be simulated to check function as well as side effects like heat dissipation
2. designers draw devices from the library and combine them into large systems, following the interconnection plan or logic layout, and obeying a grid structure to allocate real estate; this method restricts designers' freedom to use space in the most efficient manner; the alternative, used until about 1980, is to permit designers to create devices to fit the space; in effect this combines device design and validation with system design, which would be fatal to rapid design of complex systems
3. systems are again simulated by combining device simulations
4. logic and timing errors are found and eliminated

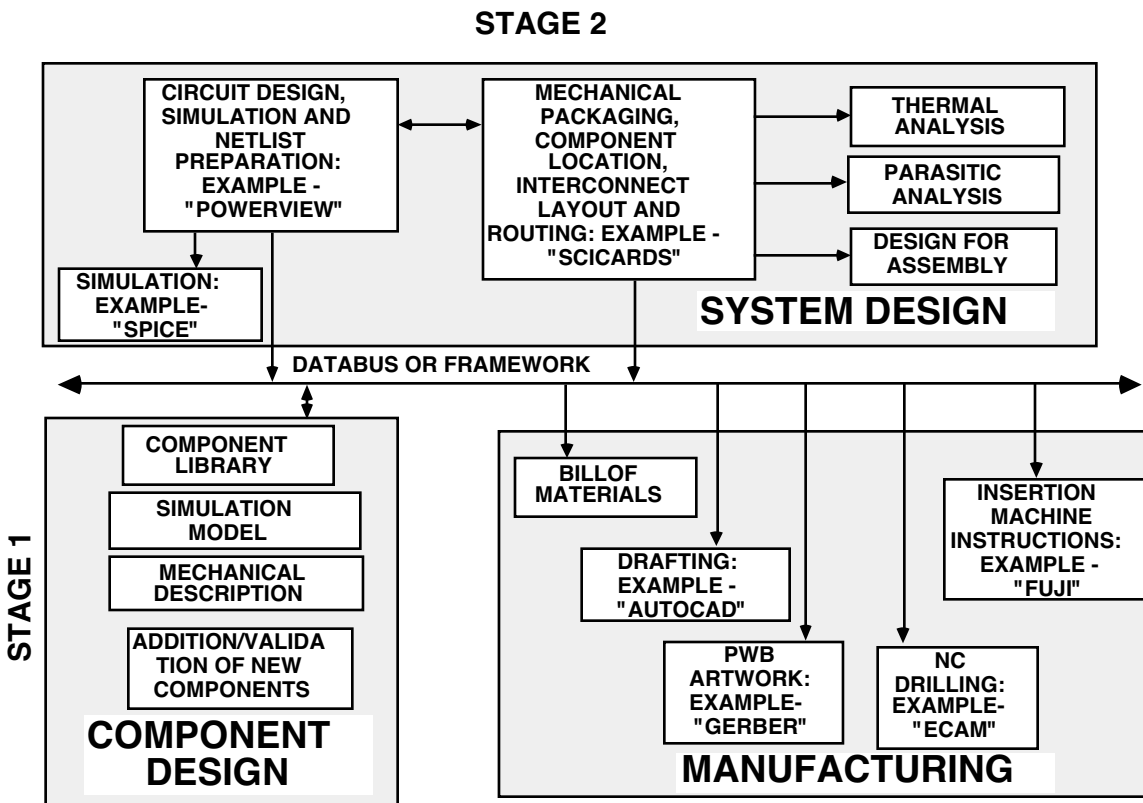
<sup>4</sup> When people in this industry are asked about difficulties designing VLSI systems, they usually respond by indicating how difficult the systems are to fabricate rather than how difficult they are to design.

5. device geometries are drawn to scale, photomasks are made, and entire systems are fabricated
6. The next generation of smaller devices is usually created by geometrically shrinking the library devices. The next generation after that requires new devices, new processes, and new verifications. New material systems may also be created across generation boundaries.

**Table 2. Stages of Design of VLSI Systems (not including manufacturing)**

Few items in all of technology can be designed so automatically by proceeding from step to step, algorithmically converting requirements and symbolic representations of behavior into specific geometry without intervention by a person.

Figure 1 captures the essence of the above process, dividing the effort into three distinct stages: component design, system design, and manufacturing process design.

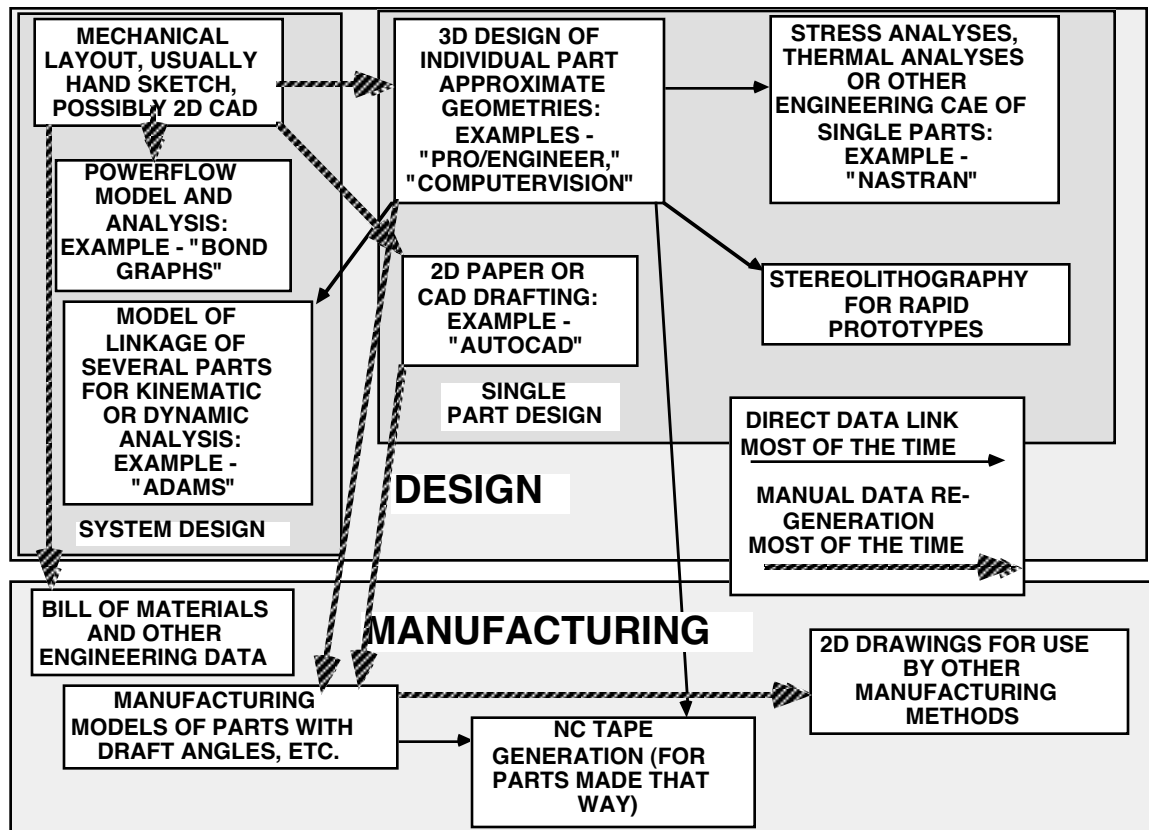


combining modules, once verified devices are in the library. The result is that extremely complex digital systems can be designed with dramatic reduction in cost and product development time. This basic point will be expanded in Section V of the paper.

#### 4. Sketch of CEMO Design

The situation in CEMO design is quite different from VLSI. The Boeing 777 has, by various estimates, between 2.5 million and 7.5 million parts. Design took about 5 years and involved about 5000 engineers at Boeing plus some thousands of others at avionics, engine, and other subcontractors.

In CEMO design, there is nothing comparable to Stage 1 and there is no cell library from which parts can be drawn, with a few exceptions. These exceptions are mainly such items as fasteners, motors, valves, pipe fittings, and finishes like paint. They are typically catalog items supplied by subcontractors and are not often designed to suit the CEMO product. See Figure 2.





**Figure 2. Simplified Sketch of Typical CEMO Product Design.** Design steps shown occur after generation of requirements is complete. Compared to Figure 1, there is no equivalent in the form of a free-standing Stage 1 Component Library Preparation and Stage 2 System Design. Main function carriers must be designed from scratch or adapted prior designs and modified to be consistent with evolving system concepts. System analysis and verification tools are almost totally absent. Analysis tools are not integrated and cover one physical medium or phenomenon at a time. Only in special cases can manufacturing tooling or processes be created directly from CAD data.

The designer puts most of the effort into

- converting an elaborate set of requirements on function, size, space, power, longevity, cost, field repair, recurring maintenance, and user interface into a geometric layout,
- identifying subsystems that will carry out the functions
- allocating functions and space to the subsystems within the allowed space
- breaking the subsystems into individual parts
- designing those parts and fitting them into the allocated space
- determining allowable variations in part and system parameters (tolerances on geometry, voltage, pressure, temperature, hardness, surface finish, etc.)
- predicting off-nominal behaviors and failure modes and designing mitigators into the parts and systems
- identifying fabrication and assembly methods, their costs, and yields
- identifying design verification plans (simulations and prototypes of both parts and systems at various levels of fidelity)
- revisiting many of the initial decisions up to the system level if their consequences, as discovered in later steps, result in technical or financial infeasibilities

While this list sounds superficially like the tasks of VLSI design, the process is profoundly different because each part and subsystem is an individual on which all the above steps must be applied separately. Each part will typically participate in or

contribute to several functions and will perform in several media (gas, solid, electricity, heat...)

Put another way, CEMO and VLSI items differ in how one designs the "main function carriers," the parts that actually carry out the product's desired functions:

- in VLSI these parts are made up by combining library devices; a few device types are leveraged into systems with millions of parts; a modular approach to system design works, in which parts can be designed and operated independently

- in CEMO these parts are designed specifically for the product, although they may be variants of past parts designed for similar products; thousands of distinct parts must be designed to create a product with a similar total number of parts, and many must be verified first individually and again in assemblies by simulation and/or prototype testing; 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; design and performance of these parts are therefore highly coupled.

## 5. Fundamental Differences Between VLSI and CEMO Design

The previous sections were primarily preparation, review and restatement of things known to many readers, and establishment of vocabulary and assumptions. This section comprises the heart of the author's contribution, an attempt to restate the foregoing in a more logical way, appealing to fundamental factors and avoiding to the extent possible any historical factors or artifacts.

I think there are fundamental reasons why VLSI systems are different from, and substantially easier to design than, mechanical systems, and I think the differences will persist. My conclusions are summarized in Table 3 and the reasoning is sketched below. An essential feature of the argument is to distinguish carefully between parts or components on the one hand and products or systems on the other.

ISSUE	VLSI	Mechanical Systems
-------	------	--------------------

<b><i>Component Design and Verification</i></b>	Model-driven single function design based on single function components; design based on rules once huge effort to verify single elements is done; few component types needed	Multi-function design with weak or single-function models; components verified individually, repeatedly, exhaustively; many component types needed
<b><i>Component Behavior</i></b>	Is the same in systems as in isolation; dominated by logic, described by mathematics; design errors do not destroy the system	Is different in systems and in isolation; dominated by power, approximated by mathematics, subject to system- and life-threatening side effects
<b><i>System Design and Verification</i></b>	Follows rules of logic in subsystems, follows those rules up to a point in systems; logical implementation of main functions can be proven correct; system design is separable from component design; simulations cover all significant behaviors; main system functions are accomplished by standard elements; building block approach can be exploited and probably is unavoidable; complete verification of all functions is impossible	Logic captures a tiny fraction of behavior; system design is inseparable from component design; main function design cannot be proven correct; large design effort is devoted to side effects; component behavior changes when hooked into systems; building block design approach is unavailable, wasteful; complete verification of avoidance of side effects is impossible
<b><i>System Behavior</i></b>	Described by logical union of component behaviors; main function dominates	No top level description exists; union of component behaviors irrelevant; off-nominal behaviors may dominate

**Table 3. Summary of Differences Between VLSI and Mechanical Systems**

The primary fundamental factors distinguishing CEMO and VLSI systems are stated in the five points below:

Point 1: CEMO systems carry significant power, from kilowatts to gigawatts. A characteristic of all engineering systems is that the main functions are accompanied by side effects or off-nominal behaviors. In VLSI, the main function consists of switching between 0 and 5 (or 3 or 2.4) volts, and side effects include capacitance, heat, wave reflections, and crosstalk. In mechanical systems typical side effects include imbalance of rotating elements, crack growth, fatigue, vibration, friction, wear, heat, and corrosion. The most dangerous of mechanical systems' side effects occur at power levels comparable to the power in the main function. In general there is no way to "design out" these side effects. A VLSI system will interpret anything between 0 and 0.5 volts as 0, or between 4.5 and 5 volts as 5. There is no mechanical system of interest that operates with 10% tolerances. A jet engine rotor must be balanced to within  $10^{-2}\%$  or better or else it will simply explode. Multiple side effects at high power levels are a fundamental characteristic of mechanical systems.

One result of this fact is that mechanical system designers often spend more time anticipating and mitigating a wide array of side effects than they do assembling and satisfying the system's main functions. This dilution of design focus is one reason why mechanical systems require so much design effort for apparently so little complexity of output compared to VLSI. But this judgment is mistaken. A correct accounting of "complexity of output" must include the side effects, which are also "outputs" that cannot be ignored during design and are usually quite complex.<sup>5</sup>

Systems that operate by processing power are subject to a variety of scaling laws that drive the number and size of components. For example, [Thompson] shows that as steamships got larger, it was necessary to increase the number of boilers rather than simply build one larger boiler.

Point 2. VLSI systems are signal processors. Their operating power level is very low and only the logical implications of this power matter (a result of the equivalence of digital logic and Boolean algebra). Side effects can be overpowered by correct formulation of design rules: the power level in cross-talk can be eliminated by making the lines farther apart; bungled bits can be fixed by error-correcting codes. Thus, in effect, erroneous

---

<sup>5</sup> [Ulrich and Eppinger] call these "incidental interactions."

information can be halted in its tracks<sup>6</sup> because its power is so low, something that cannot be done with typical side effects in power-dominated CEMO systems.<sup>7</sup>

Furthermore, 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. VLSI elements don't back load each other because they maintain a huge ratio of output impedance to input impedance, perhaps 6 or 7 orders of magnitude. If one tried to obtain such a ratio between say a turbine and a propeller, the turbine would be the size of a house and the propeller the size of a muffin fan. No one will build such a system. Instead, mechanical system designers must always match impedances and accept back-loading. This need to match is essentially a statement that the elements cannot be designed independently of each other.

An enormously important and fundamental consequence is that a VLSI element's behavior is essentially unchanged almost no matter how it is hooked to other elements or how many it is hooked to. That is, once the behavior of an element is understood, its behavior can be depended on to remain unchanged when it is placed into a system regardless of that system's complexity. This is why VLSI design can proceed in two essentially independent stages, module design and system design, as described above.

Furthermore, due to the mathematical nature of VLSI digital logic and its long-understood relation to Boolean algebra, the performance of VLSI systems can often be proven correct, not simply simulated to test correctness. But even the ability to simulate to correctness is unavailable to mechanical system designers. Why is this so?

Point 3: Single vs multiple functions per device. An important reason why is that mechanical components themselves are fundamentally different from VLSI components. Mechanical components perform multiple functions, and logic is usually not one of them. This multi-function character is partly due to basic physics (rotating elements transmit shear loads *and* store rotational energy; both are useful as well as unavoidable) and partly due to design economy. VLSI elements perform exactly one function, namely logic. They do not have to support loads, damp vibrations, contain liquids, rotate, slide, or act as fasteners or locators for other elements.

---

<sup>6</sup> This point (that information can be blocked when desired but significant power cannot) was made by Dr. Mark Matthews of University of Bath, UK in an interview with the author.

<sup>7</sup> If fanout limits are reached, amplifiers can be inserted at some cost in space, power, and signal propagation time. But this is not fundamental.

Furthermore, each kind of element performs exactly one logical function. Designers can build up systems bit by bit, adding elements as functions are required. A kind of cumulative design and design re-use can be practiced, allowing whole functional blocks, such as arithmetic logic units, to be reused *en bloc*. The absence of back-loading aids this process. However, a kind of resource conservation dominates mechanical design: if one element were selected for each identified function, such systems would inevitably be too big, too heavy, or too wasteful of energy. For example, the outer case of an automatic transmission for a car carries drive load, contains fluids, reduces noise, maintains geometric positioning for multitudes of internal gears, shafts, and clutches, and provides the base for the output drive shafts and suspension system.

Not only is there no other way to design such a case, but mechanical designers would not have it any other way. *They depend on the **multi-function** nature of their parts to obtain efficient designs.* Building block designs are inevitably either breadboards or kludges. But the multi-function nature of mechanical parts forces designers to redesign them each time to tailor them to the current need, again sapping the effort that should or could be devoted to system design. *VLSI designers, by contrast, depend on the **single function** nature of their components to overcome the logical complexity challenges of their designs.* One can observe the consequences of this fundamental difference by observing that in VLSI the "main function carriers" are standard proven library elements while in mechanical systems only support elements like fasteners are proven library elements; everything else is designed to suit.

The existence of multiple behaviors in CEMO systems means that no analysis based on a single physical phenomenon will suffice to describe the element's behavior; engineering knowledge is simply not that far advanced, and multi-behavior simulations similarly are lacking. Even single-behavior simulations are poor approximations, especially in the all-important arena of time- and scale-dependent side effects like fatigue, crack growth, and corrosion, where the designers really worry. In these areas, geometric details too small to model or even detect are conclusive in determining if (or when, since many are inevitable) the effect will occur. And when component models are lacking, there is a worse lack of system models and verification methods.

Point 4: Ability or Inability to Separate Component Design from System Design. The fundamental consequence of back-loading is that mechanical elements hooked into systems no longer behave they way they did in isolation. (Automotive transmissions are always tested with a dynamometer applying a load; so are engines.) Furthermore, these

elements are more complex than VLSI elements due to their multi-function behavior. This makes them harder to understand even in isolation, much less in their new role as part of systems. VLSI elements are in some sense the creations of their designers and can be tailored to perform their function, which is easy in principle to understand. Mechanical elements are not completely free creations of their designers unless, like car fenders, they carry no loads or transmit no power.

The fact that mechanical components change behavior when connected into systems means that systems must be designed together with components, and designs of components must be rechecked at the system level. No such second check is required in VLSI, as long as the design rules are obeyed.<sup>8</sup> For this reason, CEMO items cannot be designed by the strict top-down Stage 1 - Stage 2 process described above for VLSI systems.

Point 5. Ability or Inability to Define Interfaces. VLSI systems transmit so little power that their interfaces can be designed based on other criteria. The interfaces are much bigger, for example, than they need to be to carry such small amounts of power. The conducting pins on electrical connectors that link disk drives to motherboards are subjected to more loads during plugging and unplugging than during normal operation. Their size, shape, and strength are much larger than needed to carry out their main function of transferring information. This excess shape can be standardized for interchangeability without compromising the main function. No such excess design scope is available in high power systems. Interfaces take up space and weight and must be designed specifically to their application.

## 6. Final Remarks

[Baldwin and Clark] suggest that modularity manifests itself in three domains:

Modularity in design

Modularity in manufacturing

---

<sup>8</sup>This statement requires that "design rule" be interpreted to mean that component functions are preserved, not simply that the manufacturing process will not generate defects. Verification of design rules thus needs to include functional testing of entire devices. If or when it does not, then the above statement is invalid, and system level checking for component misbehavior will be needed. The more aggressive a design is in terms of packing components together, the more likely such checks will be necessary. Thermal effects caused by combining too many high dissipation components near each other can also cause system level problems. Thus practical VLSI will not be like ideal VLSI.

## Modularity in use

In each of these areas, CEMO systems will not be as modular as VLSI and similar systems are. Furthermore, the extreme of modularity may not be the best choice for some CEMO systems in at least some of these domains.

In design, we have seen that CEMO systems cannot be designed in a feed forward way with modules designed first followed by system design using the modules. In fact, integrated CEMO designs are often called "refined," indicating that great effort was invested in combining elements, capitalizing on multiple behaviors to achieve design objectives efficiently, and so on.

The ideal of modularity permits one to simulate the system and test or prototype only the modules. Under these conditions, the cost of a system grows essentially linearly with the number of modules. In more integral systems, testing requires building a system, and substitution of one module for another requires another whole system to be built and tested in order to uncover any emergent interactions between the new module and the reused ones.

Design is easier in VLSI than in CEMO systems because in VLSI systems, the information at the system level is entirely logical and connective. This information is transformed and augmented from stage to stage in the design process but its essential logical/connective identity is preserved all the way to the masks. This is not possible in mechanical systems, where the abstractions are not logical homologues (much less homomorphs) of the embodiments and likely never will be. Instead, tremendous conversion is needed, with enormous additional information required at each stage. A stick figure diagram of an automatic transmission captures only the logic of the gear arrangements and shifting strategy. It fails totally to capture torques, deflections, heat, wear, noise, shifting smoothness, and so on, all of which are essential behaviors. Function-sharing is not a matter of choice in CEMO systems, and side effects cannot be eliminated.

In manufacturing, the same issues can arise. If the system is to some degree integral, then several advantages of modular systems will be unavailable. These include omission of final system tests at the end of the production line<sup>9</sup> as well as easy substitution of

---

<sup>9</sup> But DELL tests each finished computer. Why? Mainly to detect module failures in the hardware. If there are software errors (other than corrupted installations) then the problems are distinctly integral.



suppliers that build “the same” module. Upgrades and engineering change orders will similarly have to be verified at the system level and cannot be counted on to follow plug and play expectations. Interestingly, much progress has been made in CEMO systems in creating even more integrated parts by means of advanced injection molding, die casting, and rapid prototyping techniques.

The reason why "an enormous variety of VLSI products can be built" from the same process is that the variety is embodied at the system level. At the component level, only one item can be made by each process. VLSI escapes the consequences of the process-dependence of components because VLSI systems can be designed independently of component design. On the mechanical side, this separation does not exist.<sup>10</sup>

In use the same limitations will exist, preventing users from quickly substituting upgrades or third party items.

In summary:

- System design methods based on extensions of the VLSI model will greatly underestimate design and debugging time.
- Methods of evaluating the excellence of a design that derive from the VLSI model will value the wrong things and fail to value the right things about good CEMO designs
- Theories based on the VLSI model aimed at evaluating architectures will not properly value CEMO integrality.
- CEMO systems will not become more modular in the future
- Design of CEMO systems will not evolve toward the two stage separation method applicable to VLSI
- Yet, technical and “business” pressures will pull opposite ways in the CEMO domain, creating ongoing tension.

---

<sup>10</sup>An exception to this may be found when the "process" is assembly. This is the case where a family of products can be created at the time of assembly by using a variety of similar parts. An example is described by [Whitney, 1993] in which Denso makes a wide variety of instrument panel meters, alternators, and radiators by this strategy.

## 7. References

[BW] "They've Got the Whole World on a Chip," Business Week, April 23, 1994, pp 132-35.

[Baldwin and Clark] Baldwin, C., and Clark, K., *Design Rules*, Boston: Harvard University Press, 1999.

[De Fazio, et al] De Fazio, T. L., and 10 co-authors, "A Prototype for Feature-Based Design for Assembly," ASME J. Mech. Des., v 115 no 4, Dec, 1993, pp 723-34.

[Fine and Whitney] Fine, Charles, and Whitney, Daniel, "Is the Make-Buy Decision a Core Competence?" Moreno Muffatto and Kulwant Pawar (eds.), *Logistics in the Information Age*, Servizi Grafici Editoriali, Padova, Italy, 1999, pp. 31-63.

[Geppert] Geppert, Linda, "Not Your Father's CPU," IEEE Spectrum, Dec., 1993, pp 20 - 23.

[Lee] Lee, H., "Effective Inventory and Service Management Through Product and Process Redesign," Operations Research, vol 44, no 1, Jan-Feb 1996, pp 151-159.

[Paynter] Paynter, H. M., *Analysis and Design of Engineering Systems*, MIT Press, 1960

[Stix] Stix, Gary, "Toward 'Point One,'" Scientific American, Feb., 1995, pp 90 – 96

[Thompson] Thompson, D. W., *On Growth and Form*, quoted in Newman, J. R., *The World of Mathematics*, New York: Simon and Schuster, 1956, p 1009.

[Ulrich] Ulrich, Karl, "The role of Product Architecture in the Manufacturing Firm," Research Policy, v 24, 1995.

[Ulrich and Eppinger] Ulrich, K. T., and Eppinger, S. D., *Product Design and Development*, New York: Irwin, 2001.

[Whitney, Nevins, De Fazio and Gustavson] Whitney, D E, Nevins, J L, De Fazio, T L, and Gustavson, R E, "Problems and Issues in Design and Manufacture of Complex Electro-mechanical Systems," C S Draper Lab Report R- 2577, Dec., 1993.

[Whitney, 1993] Whitney, D. E., "Nippondenso Co. Ltd: A Case Study of Strategic Product Design," Research in Engineering Design, #5, Dec, 1993, pp 1-20.

[Whitney, 1996] Whitney, D. E. "Why Mechanical Design Will Never be Like VLSI Design," Research in Engineering Design, (1996) 8: 125-138.

