EDP Symposium Brainstorming

Brent Nelson, Brad Riching, and Richard Black
Fulton College of Engineering and Technology
Department of Electrical and Computer Engineering
Brigham Young University, Provo, Utah 84602

***Abstract*—This paper philosophically explores the trade-offs of using a text-based language versus a graphical tool to perform schematic capture of printed circuit board designs. An enumeration of features that would be needed to make the text-based alternative favorable is given.**

***Index Terms*—PCB, HDL, schematic, design, EDA, CAD, layout, hierarchy**

I. Introduction

This document serves as the breeding ground for and core dump of ideas regarding the philosphical implications of using a text-based HDL for circut board schematic capture as opposed to a graphical tool.

The topics that will be explored are as follows:

1. Historical Perspective
2. Asthetics Versus Content
3. Design Reviewability
4. Reusability
5. Collaboration
6. Development Speed
7. Features Needed for Normalcy
8. Features Needed for Superiority

II. Historical Perspective

An exploration of the first logic-synthesis HDLs advent reveals several similarities to the introduction of a hypothetical circuit board language. When ISP, KARL, and Verilog, et. al, were first invented, they were categorized as inefficient compared to and costlier than schematic capture. In modern industry, however, a digital engineer is considered gravely unqualified without proficiency in an HDL.

This begs the following questions, (1) what caused HDLs to be unpopular at their inception, and (2) what caused HDLs to become streamlined among engineers.

. . . flesh out ideas here . . .

It follows, therefore, that the introduction of a circuit board HDL would provide similar benefits to those of logic-synthesis languages.

III. Asthetics Versus Content

At the end of the day, the most important result of a design is its functionality. In schematics, designs can become so complex that the layout of devices and wires becomes more important than the fundamental connectivity. This can shift the priority from functionality to asthetics. The vast majority of electrical engineers are hired for their inventive ideas and not their artistic ability.

Through the use of a text-based schematic, the boundary between layout and design becomes well-defined; the primary goal becomes achieving the desired functionality. This will be demonstrated with a simple example: a ratsnest.

Fig. 1. Wire Ratsnests Schematic

In Figure 1 above, a collection of devices is interconnected in various ways. Assuming that the desired functionality has been achieved, the only problem with this schematic is the illegibility of its connectivity. If such a design is fully functional, there is little benefit in moving devices and modifying paths to make the drawing pleasing to the eye. Potentially, in the course of rearranging the various components, errors might inadvertently crop up due to the clean-up.

Alternatively through an HDL, once proper connectivity is confirmed, there exists little benefit to perform revisions. The ratsnest is completely circumvented, and the overall design flow is expedited.

All-in-all, a circuit board HDL eliminates the need to create a carefully organized drawing and instead focuses the engineer's attention on the connectivity and functionality of the design. This does not limit those that prefer to see physical drawings of schematics as they are still free to pull out their pencils and sketch.

IV. Design Reviewability

On the major arguments in favor of graphical tools is the ability to visually inspect and review designs for proper functionality. Design reviews are a necessary standard to prevent expensive mistakes by fabricating en masse a faulty circuit board. This is a potentially fatal flaw with an HDL approach to schematic capture.

In order to rectify this flaw, it is wise to learn from the success of other languages, such as VHDL and Verilog. The question to be answered in examining these HDLs is how is a design review accomplished without visual schematics.

One approach that VHDL and Verilog take to simplify design reviews is reductionism: the use of smaller modules that make up the greater. This allows a reviewer to verify proper behavior in manageable chunks. Such an approach would largely benefit a circuit board HDL.

Another approach used by mainstream HDLs is the utilization of simulations. The reviewing typically consists of the plotting of several waveforms and checking for proper behavior at various points of interest. In order to be maximally effective, a text-based schematic would need mixed-signal simulation capabilities.

A unique and unconventional approach is to create and output rudimentary schematics to facilitate design reviews. These images need not be flawless; they must, however, convey enough information such that the verification is straightforward to somebody not familiar with the HDL.

It is important to note that graphical tools are not exempt from scrutinization in this topic. One particular case where graphical tools make reviews difficult is the use of very high-pin-count devices.

Fig. 2. High-Pin-Count Device Schematic

In Figure 2 above, there is a single device in the center of the schematic that has so many pins that it fills the majority of the page. As a result, the reviewer is required to turn several pages in order to perform verification. It is hard to debate that such a schematic is easier to review than a text-based one.

In summary, the need for design reviews are a necessity and must be facilitated by the tools used to create the schematic. If a circuit board HDL is to be successful, it must take from the features that make other languages easy to verify. Such features include module compartmentalization and simulation.

V. Reusability

A significant pitfall in the use of most graphical schematic capture tools is the inability to easily reuse

working designs in others. Often this process involves marqueeing the desired elements and copying them to the clipboard for use in another project. Unfortunately the marquee tool does not exhibit the desired behavior; sometimes an entire device must be enclosed in order to be selected, other times only a portion must be in the selection area. Furthermore, a straight copy and paste will also copy reference designators that might conflict with those in the newer design. In either case, the engineer emerges the process with a splitting headache.

An HDL would not have this problem since all attributes and connections can easily be grouped together and selected with ease. If the language included some method of autogenerating reference designators, the trial of modifying them upon design reuse becomes insignificant.

VI. Collaboration

VII. Development Speed

VIII. Features Needed for Normalcy

IX. Features Needed for Superiority

X. Conclusion