Short Contribution

Using CAD Tools for Embedded System Design: Obstacles Encountered in an Automotive Case Study

PHILIP KOOPMAN

Institute for Complex Engineered Systems, Carnegie Mellon University, 1201 Hamburg Hall, Pittsburgh, PA 15213, USA
Tel.: +1 412 268-5225; Fax: +1 412 268-6353; E-mail: koopman@cmu.edu

ABSTRACT: Historically, digital system CAD research has emphasized increasing the size and complexity of the largest feasible design. However, the success of embedded system design efforts may depend more on design flexibility and lifecycle cost optimization than on an ability to synthesize hardware containing millions of transistors. This paper reports the results of a case study using a commercial CAD tool to redesign an automotive electronics product. Although the tool was in fact able to perform the required design synthesis, the case study uncovered obstacles to the adoption of CAD tools by some classes of embedded system designers.

INTRODUCTION

Computer Aided Design (CAD) tools continue to evolve in their ability to handle ever-larger designs as well as incorporate more comprehensive models and simulations. One way of characterizing electronic CAD tool development is that it has been largely driven by the need to keep up with exponentially increasing transistor counts. Historically, CAD improvements have focused on helping designers work at the upper limits of complexity possible with any given technology, and have thus been driven by the needs of those designing chips and circuit boards for mainframes, desktop computers, and signal processing applications.

Recently there has been increased research interest in the embedded systems area, in which processors are embedded into non-computer products (e.g., Kluwer, 1996; White et al., 1994; Gajski et al., 1994; Lightner, 1995). Additionally, a research community in hardware/software codesign (e.g., Thomas and Ernst, 1996) has formed to explore CAD support for cross-disciplinary tradeoffs in embedded system design.

An important fact about embedded systems is that "low-end" systems having modest computing requirements dominate the demand for embedded CPUs. In fact, out of 2683 million embedded microcontrollers shipped in 1994, fewer than 3% were 32- or 64-bit processors (Cole, 1995). Given the huge volumes of products that use these smaller embedded controllers, there could be a significant opportunity for CAD tools and advanced design methodologies to reduce cost and improve product quality.

Many embedded systems are designed using modest CAD tools such as schematic-based netlist creation and semi-automated circuit board layout. Additionally, High-Level Description Language synthesis tools are used for programming configurable logic devices, and occasionally for designing Application Specific Integrated Circuits (ASICs). It would seem plausible that increased levels of CAD tool support might improve design efficiency and decrease development time significantly, and would be readily adopted once benefits were demonstrated. However, the result of this case study is that the digital design synthesis tool eval-

Integrated Computer-Aided Engineering, 5(1) 85–94 (1998)

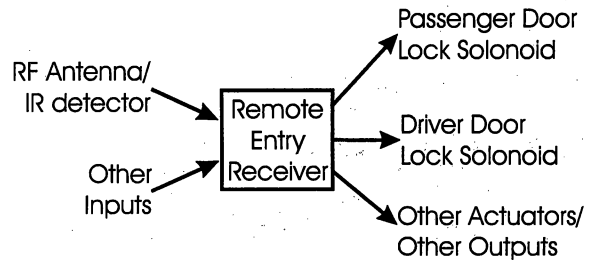

Figure 1 Generic automotive Remote Entry Receiver unit.

uated did not encompass a broad enough part of the design space to be worth adopting. Experience in a variety of other embedded system domains suggests that this result may be widely applicable.

THE CASE STUDY

The results presented in this paper are grounded in a case study done on an automotive subsystem. The case study used a schematic design synthesis tool to attempt to recreate and possibly improve upon four hand-optimized designs within a product family. In addition to evaluating whether the CAD tool was mature enough to be deployed into a mainstream engineering process, the study set out to determine what barriers exist to deploying advanced CAD tools of any kind into a real-world embedded design environment.

The Application

The application example chosen was a Remote Entry Receiver (RER) unit (Figure 1). This unit is installed in high-end vehicles as a convenience feature to receive commands from a small radio-frequency or infrared transmitter on the driver's key ring. Typically RER units have the ability to unlock car doors from a distance of several meters, unlatch the trunk (boot), and other product-specific functions. Because this is an automotive application, the design is optimized for extremely low cost, constrained by meeting minimum specified performance, lifetime, and quality requirements.

RER units are, in general, different for each Original Equipment Manufacturer (OEM — a company that produces entire vehicles) as well as each class of vehicles from a single OEM. In order to gain a sense of the range of applications, we obtained design information for four existing RER units sold to three OEMs. One of the RER units uses computation-intensive encryption of messages as a theft deterrent, whereas the other three older designs used fixed serial numbers to match transmitters with receivers. CPUs used in these designs ranged from a low-end 8-bit microcontroller for the fixed serial number subsystems to a high-end 8-bit microcontroller for the encrypted message design.

Each RER board design is somewhat different, depending on OEM requirements for interfaces and features. The RER unit is connected to actuators that drive door locking mechanisms and other peripherals, and in some cases may be connected by serial interfaces or even an embedded network to other automotive subsystems. The RER unit must be able to handle different levels of output current driving capacity on different actuator circuits, as well as different input signal types. The RER units studied all use RF communications, and so have a low-cost antenna implemented as a trace on the circuit board.

The Synthesis Tool

This case study evaluated the effectiveness of a commercially available design-by-composition synthesis tool, Fidelity (Omniview, 1995). Fidelity is a commercialization of the Micon (Birmingham et al., 1992) work done at Carnegie Mellon University.

Fidelity is a schematic-based synthesis tool that composes designs from a database containing a parametrized hierarchy of function blocks (symbols) and design templates (schematics) as shown in Figure 2. At the top level is a single schematic that acts as the root of a graph specifying the entire potential design space within a larger multi-product design database. Each schematic is used to define connections among one or more symbols. Schematic sheet inputs and outputs correspond to inputs and outputs of any symbol that might refer to that schematic from above.

Each symbol used in a particular schematic in turn refers to multiple schematics at the next lower level. At the lowest level, each symbol refers to an individual electronic component. The multiple schematics referred to by a single symbol constitute multiple alternative implementations for the function represented by the symbol, and are the basis for enumerating design alternatives. It is possible for any number of symbols to point to any particular schematic, making the design hierarchy a directed acyclic search graph with each path through the graph consisting of alternating AND nodes (schematics) and OR nodes (symbols).

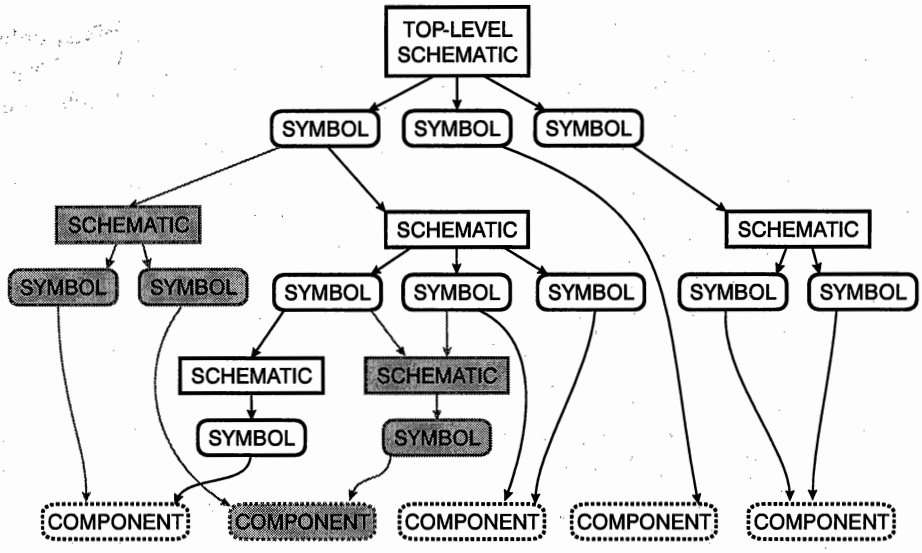

Figure 2 Fidelity uses a design hierarchy of alternating components and schematics. Multiple schematics associated with a component indicates the potential for alternative design decisions. A single synthesized Fidelity design is created by selecting one, and only one, alternative for each symbol. The grey areas indicate schematics and symbols not used by this particular implementation alternative.

Within the schematic hierarchy, constraints and capabilities are communicated among levels by the use of interval arithmetic. An equation can be attached to the input and output ports of each symbol (although they are called equations, they can in fact contain almost any valid conditional expression clause). When a candidate schematic is being evaluated for feasibility with respect to implementing a particular symbol, equations in the symbol pass requirements downward (e.g., a particular output must be able to provide at least 600 mA of current), while equations in the schematic pass capabilities upward (e.g., a particular schematic for switching is limited to 20 mA of current).

Fidelity performs design exploration by brute-force enumeration and elaboration of all possible alternatives subject to equation-based design constraints. Starting at the top-level schematic, Fidelity examines all combinations of linking symbols to each of their underlying schematics. The design equations are used to immediately discard infeasible designs, such as a switch that cannot handle a required amount of current. All feasible design alternatives for each symbol are explored until a complete set of all possible fully elaborated implementations is generated. This process results in automatic top-down requirements refinement combined with bottom-up capability determination. Fidelity prunes schematics with infeasible implementations from the search space, and only traverses those portions of the schematic database that are part of the acyclic graph rooted at the top-level design.

A particular design instance is created by selecting one, and only one, design alternative at each node in the design hierarchy (the shaded blocks in Figure 2 indicate regions of the design space omitted from a particular design instance). Fidelity then generates a succession of all feasible design instances. At the measured synthesis rate of several thousand designs per minute on a workstation, Fidelity took less than an hour to enumerate all possibilities for an RER example.

The design instances that Fidelity generates are ranked according to a weighted sum criteria with cutoff thresholds. Any design exceeding a cutoff threshold is discarded out of hand. The ranking criteria available are total component cost, power, area and weight. These criteria are combined with a user-specified weighted average to form a single figure of merit. The case study followed common automotive practice, and ranked designs solely by cost, applying threshold limits to power and area measurements.

Micon, the predecessor of Fidelity, was originally developed to synthesize personal computer motherboards. So, in all fairness, it is unreasonable to expect Fidelity to be perfectly suited to this embedded system example. However, the equation-solver in Fidelity facilitates some analog and power component selection, and at the time of the case study the tool seemed better suited than any other to the RER designers' needs. Given that Fidelity was not specifically designed for embedded sys-

tems, the conclusions and observations in later sections are not meant to be critical of using that tool for digital design, but rather to illustrate the gap that exists between digital design tool support and the needs of embedded system designers.

The Experiment

When the case study started, a key concern of the RER designers was whether an automated tool could duplicate the high degree of optimization possible with manual methods. Therefore the focal point of the case study was an experiment to see whether Fidelity could exactly reproduce (or improve upon) a hand-created optimal design, and do so in a matter of hours once an initial design database was seeded.

The need for very rapid creation of optimized designs is driven by the way the U.S. automotive subsystem industry does business. OEMs issue Request for Quote documents on subsystems, including RER units. Engineers typically have a one- or two-week window in which to perform a preliminary design and come up with a cost estimate. Because much of the automotive subsystem business is driven by price, this estimate must be low enough to win the competitive bidding process, but not so low that the company loses money on the product. As an additional challenge, it is common for purchase agreements to specify that the cost of subsystems supplied to an OEM must be reduced yearly, even in the face of economic inflation.

Therefore, the primary attraction of using Fidelity was the possibility that it could be used to create optimized designs within a one- or two-week time span. This would permit more aggressive bidding based on knowing the details (and therefore accurate cost) of a design, rather than an approximation based on experience and similar designs. Furthermore, it was envisioned that the tool could continually redesign subsystems so as to create cost-reduced versions as component prices, market conditions, and part availability changed.

Fidelity further was chosen because it uses schematics (making it compatible with existing practice), and because it is capable of handling analog and power portions of the design as well as digital design. Additionally, Fidelity is optimized for creating designs by composing off-the-shelf components rather than performing ASIC design, and that is exactly how current RER products and many other embedded systems are created.

The original plan was to use Fidelity to resynthesize all four example RER designs. Because of time and resource limitations, only one design was actually created. However, there is no reason to believe that other designs could not have been created in a similar manner. Fidelity was able to exactly recreate the hand-optimized schematics, as well as propose alternate optimal designs depending on the relative costs of various components in the design database. These designs included all analog and power components as well as the digital circuitry.

During the course of the case study, RER design engineers critiqued the results and utility of the tool. Additionally, CAD tool experts from five other embedded system manufacturers provided constructive criticism. What was found was that the requirements for embedded system design were different enough from digital computer design that the tool, although it worked, was not comprehensive enough to be immediately useful.

In a strict sense, Fidelity was successful. It was able to synthesize an optimized design identical to the hand-created design, and do so within an hour. However, it turned out that rapid synthesis of schematic designs by itself did not solve enough of the problem at hand to make the cost of adopting the tool worthwhile.

ELECTRONIC DESIGN

The major impetus for adopting a CAD tool for this case study was to quickly perform automated design of new variants of existing products. In order to perform fully automatic synthesis from an hierarchical schematic database, Fidelity uses design rationale information in the form of design equations. In order to create an optimal design, the designer simply provides high level information such as the number and type of input/output connections to the RER unit, and then lets Fidelity search the design space.

Unfortunately, excellence at digital design synthesis was not sufficient for the RER problem. Additional capabilities are necessary for analog and power component design, selection of highly integrated off-the-shelf components, power optimization, layout-influenced design changes, and design rule variation.

Digital, Analog, and Power Components

A typical RER system has only one or two digital components — an off-the-shelf single-chip microcontroller and potentially an accompanying pro-

gram ROM chip. In addition to these digital components, the board may have two orders of magnitude more analog and power switching components. So, in this system the analog circuit design could be considered more complicated than the digital circuit design. Many embedded systems are able to use a small, highly integrated microcontroller, and thus it is not unusual to have non-digital components comprise the bulk of a circuit board design.

As with most digital design tools, Fidelity has the ability to represent analog components. But, there is a problem representing the semantics of those components. Fidelity can attach equations to the function block interfaces in order to do parametrized component selection, and so can do simple calculations such as selecting resistor values to form a voltage divider automatically. But, complex calculations involving engineering analysis software were beyond the capabilities of the version of Fidelity used for this case study.

Conclusion. Embedded CAD tools must deal with the possibility that most of the design is non-digital. It is not sufficient to simply accommodate analog and power components as "second-class citizens" that are largely devoid of simulation and modeling semantics (e.g., decoupling capacitors and pull-up resistors on typical digital design tools with switch- or clock-based simulations). Fortunately, the issue of analog components in design synthesis tools is receiving attention (e.g., Sullivan, 1996).

Digital Design vs. Digital Selection

While ASICs are popular for many applications, the RER design specifically requires off-the-shelf components, and in particular the use of standard-product microprocessors. This is the case even though an ASIC may be less expensive on a per-unit basis. The reasons for this include:

- Standard parts may have reduced cost and more stable supplies when they are used by multiple companies.
- Using standard parts avoids the initial component engineering/mask costs, inventory requirements, and risk for custom-made components. Additionally, standard parts might be usable in other applications if an order for a particular RER variant is canceled or reduced in size.
- The generality of standard microprocessors increases the likelihood that design changes can be accommodated quickly and inexpensively with a software change, causing minimal disruptions in manufacturing and procurement.

The result of these design constraints is that digital "design" for an RER unit consists of selecting a microprocessor and writing software, not synthesizing digital circuitry. This selection is generally performed based on cost, I/O capability, and memory configuration. Fidelity did not handle this task well.

Conclusion. While very large embedded systems may require a significant digital design task, in smaller systems it may well be that choosing a single microcontroller is all that is needed. In such cases, selecting from among a wide variety of pre-integrated off-the-shelf components is the design task of importance, and may not be well suited to synthesis-oriented digital design tools.

Power Consumption

Many embedded applications are sensitive to power consumption, either because of power supply restrictions or because of heat dissipation limits. In a power-limited embedded system design, optimization has to take into account operating power, various power-saving modes, standby power, and the computation duty cycle. In low duty cycle applications such as a RER unit, standby power can be the dominant factor, not operating power.

In the case of the RER unit, the subsystem has to co-exist with all other standby electric loads on a car battery for at least a month and still leave enough battery energy left for a wintertime engine start. This results in a power budget of about a milli-Amp of standby current. Given that the receiver must continually search for incoming messages, this is an aggressive power target.

The Fidelity tool only supports a single power optimization field, which is intended to represent full-speed operating power. While a workaround is available by simply redefining this power field to be standby power, this requires that the component database be modified to include standby power numbers, and does not permit optimization on both power attributes. Future versions of Fidelity might permit such power management optimization.

Conclusion. Standby power as well as duty cycle and intermediate "sleep" modes are critical when optimizing for power on embedded systems. It is possible that mobile computing applications will spur digital CAD tools to provide more comprehensive power management capabilities, which will benefit many embedded system designers.

Layout vs. Logic Design

Schematic and printed circuit board layout are often performed using two different tools, and are loosely coupled through netlist and back-annotation files. Component changes must sometimes be made in order to aid in circuit board routing (for example, changing I/O pin assignments on the microcontroller to reduce routing congestion), resulting in back-annotation changes to the schematic. This was the case with RER designs.

While the RER designs often have the luxury of a multi-layer circuit board, other low-cost embedded designs use a single layer of routing, so each and every wire cross-over may have to be reflected back into the schematic as a zero-ohm resistor component. Thus, cost-sensitive embedded systems may require a coupling between component-level design and layout. Fidelity had essentially no capability for back-annotation, although workarounds were available.

Conclusion. Cost-sensitive embedded systems may require extensive layout optimization. With current layout tools support for manual intervention of layout decisions as well as back-annotation to the design source is essential, even for automatically synthesized designs.

Design Margin & Customer Variation

Environmental conditions and cost/performance tradeoff points vary widely for embedded systems. Even within the RER designs examined, there was a range of system robustness and lifetime requirements. While engineers in general may wish to make a high-quality product that is unlikely to break, the fact remains that different designs require different degrees of robustness traded off against cost. This is especially true in highly cost-conscious industries such as automotive products and consumer electronics.

In addition to simple cost issues, regional requirements may differ as well. For example, RER units must meet different levels of ElectroMagnetic Interference (EMI) restrictions depending on the country they are sold in. Because meeting severe EMI requirements can require more expensive circuit designs, only units designed for certain markets are built to meet the most stringent requirements.

Another issue is that similar subsystems may have different interface requirements for input protection or voltage levels to other equipment. For example, different OEMs have different required input protection circuits on any unit integrated into their vehicles.

These factors result in a situation in which different designs or variations of designs must be created with different "robustness", degrees of safety, or other varying requirements in order to provide a competitive price while meeting varying regulatory and competitive strategy requirements. A challenge found in the RER case study was to find a way to vary the "robustness" or other attributes of a design without having to create entirely different design databases.

Conclusion. If a designer is creating subsystems to be used in a variety of situations (whether by customer, market segment, or regulatory jurisdiction), there may be design rule variation for otherwise identically specified designs.

BUSINESS AND ENGINEERING PROCESS ISSUES

Because CAD tools are part of a larger engineering process, they must operate within the context of not only a technical design, but also business strategies. This is especially important in embedded systems, where computers are intimately tied to what may be largely a non-digital product. Specific areas of concern include the use of accurate lifecycle cost models, system certification costs, CAD tool proficiency, database maintenance, and electronic- vs. paper-based design representations.

Lifecycle Component Costs

As with many high-volume applications, the RER design is extremely sensitive to cost, with small changes in production cost becoming a big issue (for example, one million units times a ten cent cost increase is U.S. $100,000–a substantial amount of money compared to a few weeks of cost-reduction engineering time). Fidelity supports a single, fixed price per component. But, in the RER

application a much richer representation of cost is desired in order to optimize for true lifecycle cost.

In a cost-conscious environment, designers must optimize across the entire lifecycle (Demes, 1993). If a newly available component has a low per-unit price, it may seem obvious for a synthesis tool to select it for a new or revised design. However, there are many additional points to be considered in a highly accurate cost model, such as:

- Cost of the component itself, which decreases in quantized steps with higher order volumes. Substituting a new component in one place may lower total order size of the old component below a cost threshold, thus increasing the cost of all other instances of using the old component.
- Cost of purchasing components, which includes time spent researching availability and obtaining competitive quotes.
- Cost of qualifying a new component for operation, including environmental, stress, and lifetime testing.
- Cost of qualifying a new vendor if the component is unavailable from existing vendors.
- Cost of entering and maintaining a new component in databases for purchasing, design, simulation, and spare parts.
- Cost of carrying a component in inventories (the money spent on component stock has an opportunity cost compared to investing the money elsewhere). Depending on repair strategies, components may have to be stocked, periodically counted, and replenished at multiple locations worldwide.
- Cost of running production line component placement equipment, which in general can accept only a limited number of component types at one time. In some systems, the side of the circuit board on which a component is placed is influenced by the need to balance the component feed capacities of the two different placement machines used (one machine per board side). New CAD tools are becoming available to optimize the manufacturing process (e.g., Harris Corporation, 1995).

Conclusion. An important strategy to reduce lifecycle component costs is to use fewer types of components, even if in some cases a less expensive component must be replaced with a more expensive component.

System Certification and Lifecycle Costs for Changes

One of the initial promises of automated design synthesis as provided by Fidelity was the possibility of re-synthesizing designs on a regular basis. Resynthesis could continually re-optimize production costs. Given productions runs of up to a million units a year, costs for retooling could be small compared to potential savings. However, it turned out that certification was a major obstacle to design resynthesis.

In automotive systems, as in many other embedded systems (especially safety critical systems), designs must be recertified every time they are changed. This is to ensure that the design will function properly, have an acceptably low failure rate, and meet any governmental requirements. The cost and time consumed by such certification can be significant, and can preclude fine-tuning designs for small cost improvements.

Conclusion. Many embedded systems must undergo certification or testing after any change. Synthesis tools that attempt redesign for cost savings must weigh potential benefits against the cost of certification.

CAD Tool Specialists & Maintaining Proficiency

As in many embedded design activities, the designers of the RER units are assigned to products rather than to assembly-line-like engineering functions. Thus, one or two hardware engineers are responsible for a single product from initial proposal through support of the manufacturing facility. Within this entire lengthy effort, the amount of time actually performing circuit design can be small.

As a result of this product-oriented focus and the fact that designs are not extremely complex, RER designers do not spend the majority of their time actually using their CAD tools. But, many modern CAD tools have steep learning curves and, in practice, users can lose proficiency quickly. Thus, the use of a fairly complex workstation-based schematic capture tool combined with the specialized database and equation-solving languages of Fidelity raised the question of whether engineers, once trained, would be able to maintain proficiency with the tool. This question also arises in other embedded application areas that have

relatively modest digital design requirements.

A possible solution to the tool proficiency problem is to have a team of CAD specialists who help the other engineers use the tools. However, having one or two CAD experts was seen as an organizational vulnerability. In the high-turnover automotive industry, the loss of even one CAD expert could seriously compromise the organization's ability to design products. Furthermore, CAD experts might be difficult to hire or train because they would require not only CAD tool expertise, but application-specific knowledge as well.

Conclusion. Complex, point-design CAD tools may be unusable in organizations where engineers spend only a small fraction of their time doing actual design work. Having experts to do design may not be compatible with business and organizational strategies. The current, pragmatic, approach is to use only relatively simple CAD tools.

Model & Library Database Maintenance

When switching from a schematic design approach to an automated synthesis approach, considerable additional knowledge must be entered into the design. A significant part of the effort in using Fidelity is entering and maintaining component information such as pricing and suppliers. This is exacerbated by the fact that some of the information in the pre-supplied libraries (e.g., full-speed power consumption) is not the information that is actually needed for the RER design (e.g., standby power).

Maintaining component databases and models is a formidable task, and must be taken into account when calculating the true cost of engineering efforts. Costs include:

- Updating pricing at various quantity levels and lead-time information for component acquisition.
- Translating to multiple formats in support of different CAD tool requirements.
- Translating from vendor part numbers to internal part numbers.
- Updating the database to meet the requirements of new CAD software releases.
- Modifying any vendor-supplied component information to conform to internal component database formats and fields.
- Policing designers to ensure that only accurate and current copies of component databases are in use (in cases where CAD tools maintain copies of component databases within a design file).
- Archiving component databases so that designs can be resynthesized at a later date even if components are altered or removed from the database (in cases where CAD tools access a central copy of a component database).
- Coordinating, enforcing standards for, and validating component database entries.

Conclusion. Using automatic synthesis requires significant on-going infrastructure maintenance. And, because the needs of different embedded system design tradeoffs vary with application area, it is not clear that it is possible for component vendors to provide and maintain databases that will satisfy the needs of most designers.

Legacy Designs, Synthesizability, and Understandability

The environment in which the RER design team works is paper-oriented. This is in spite of the fact that most of the paper is produced as output from CAD tools and word processors. Individuals or groups of engineers keep electronic copies of recent designs, but there are good reasons to have permanent archival copies of designs on paper, including:

- The difficulty and expense of coordinating and maintaining many different formats of design information, qualification test results, and other data in a central repository.
- The increased risk of mass theft or wholesale compromise of centrally archived on-line data compared to paper.
- The risk of data becoming unusable due to changes in CAD tools. Or alternately, the high cost of maintaining old versions of software, hardware, and operating systems to ensure continued access to older designs.

In order to promote synthesis, Fidelity encourages the use of deep design hierarchies. Each level in the hierarchy permits a choice among alternatives. For example, on an input conditioning circuit there might be one hierarchy level to permit replication of multiple inputs (so that the number and types of input circuits can be selected by the user without changing schematics), another level of hierarchy to select which type of input is expected (e.g., digital pulse or analog voltage), and yet another level of hierarchy to select the OEM-

specified input protection circuitry. When viewed on a computer with navigation aids, the hierarchies for a RER design were at times intrusive, but manageable.

However, once a deeply hierarchical design was printed on paper, the navigation information was essentially lost, and the RER designs became quite difficult to understand. This awkwardness was made apparent in design review meetings for this design and other embedded system designs where a CAD workstation was not available in meeting rooms. This effect seems common when dealing with deeply nested hierarchies on paper-based designs, and is one of the reasons that the existing RER designs were each on a single large schematic sheet.

Conclusion. In an environment with many designs, something must be done in order to make legacy designs readable years later. Optimizing a design for the limits of paper output seems to defeat some of the benefits of using a CAD system in the first place, such as deep hierarchies to decouple design decisions.

Compelling Advantage Required

When the idea of using hardware synthesis was initially proposed, it was met with some justifiable skepticism. The designers had previously explored advanced CAD tools, and found them lacking in a number of respects (in fact, they correctly anticipated many of the obstacles reported in this paper). But, more importantly, they were successfully doing their job without advanced CAD. Given that there was no true crisis in their ability to perform design, there was little incentive to take on the added expense, training, and risk of adopting new tools and methodologies in order to do the same work with (potentially) improved efficiency.

The RER designers, like many embedded engineers, need to see a clear and compelling advantage to adopting CAD synthesis tools beyond just incremental efficiency improvements. Unlike desktop and supercomputer designers, their designs are still tractable even with rudimentary CAD tools, and the actual digital design portion of product development is a relatively small percentage of the total engineering effort. So, to be worth adopting, a design synthesis tool has to offer them some completely new capability. In the case of Fidelity, this new capability promised to be creating a design variant within a two-week window, permitting a response to a bid request with a detailed design analysis rather than an engineering estimate. For other embedded system applications a compelling advantage to adopting advanced CAD tools is likely to be needed as well, but may take other forms.

Conclusion. Many embedded systems can be designed with little more than an inexpensive personal computer-based schematic drafting package and a board layout service bureau. Given that adopting new tools introduces risk and change, there must be some compelling competitive advantage beyond simply increasing efficiency in what is only a small part of the entire product development process.

OBSERVATIONS FROM OTHER DOMAINS

This paper is focused specifically on applying digital CAD tools to an example high volume/low cost embedded system design. A previous conference paper, (Koopman, 1996), took a somewhat broader, although more anecdotally-based, view of embedded system design requirements. Although those observations were not born of a methodical case study, they were observed in enough different embedded design contexts to bear consideration by CAD tool designers, methodologists, and vendors (in fact, many of theses issues were present in the RER design, but were not directly affected by the use of the Fidelity synthesis tool). These additional issues and opportunities are listed here in order to present a more complete picture of embedded system design issues:

- Guarantee real time performance, including avoiding unduly pessimistic performance bounds.
- Size and weight limitations.
- Safety and low-cost reliability, including use of low-cost components in non- redundant configurations.
- Operation in harsh environments.
- Embedded distributed designs, including communication versus computation tradeoffs.
- Long-term component availability, including logistics; and redesign when components become obsolete.
- Cultural aspects, including distrust of simulation-based design methods.

CONCLUSIONS

The Remote Entry Receiver case study serves to point out many significant issues that might be addressed by CAD tools aimed at the embedded design market. Some issues, such as a more system-level and lifecycle based approach to optimization, are likely to be required for CAD tools to penetrate "low end" applications that are prevalent. Other problems can perhaps be endured as long as there is a sufficiently compelling advantage to justify the risk and expense of adopting new tools and techniques.

ACKNOWLEDGEMENTS

Phil Lemay performed the detailed RER design synthesis work using Fidelity. Keith Christenson and the engineers of the Input Controls group within United Technologies Automotive provided significant expertise in order to make this case study possible. In addition to mainly United Technologies support, portions of this work were sponsored by DARPA contract DABT63-95-C-0026, and ONR contract N00014-96-1-0202.

REFERENCES

Birmingham, B., Gupta, A. and Siewiorek, D.P. (1992) *Automating the Design of Computer Systems: the Micon project*, Jones & Bartlett, Boston.

Cole, B. (1995) "Architectures overlap applications," *Electronic Engineering Times*, March 20, 1995, pp. 40,64–65.

Demes, G. (1993) "The Engineering Design Research Center of Carnegie Mellon University," *Proc. of the IEEE*, 81(1): 10–24.

Gajski, D., Vahid, F., Narayan, S. and Gong, J. (1994) *Specification and Design of Embedded Systems*, PTR Prentice Hall, Englewood Cliffs NJ.

Harris Corporation (1995) "Harris EDA introduces EDAassimilator software for manufacturing synthesis," Harris Electronic Design Automation Inc.

Kluwer (1996) *Design Automation for Embedded Systems: an international journal*, Kluwer Academic, ISSN 0929-5585.

Koopman, P. (1996) "Embedded System Design Issues — The Rest of the Story," *Proc. 1996 Int. Conf. on Computer Design*, October 7–9, Austin, Texas, pp. 310–317.

Lightner, M. (ed.) (1995) *National Science Foundation Workshop on CAD Needs for System Design, Final Report*, http://dufay.colorado.edu/~lightner/system-workshop/system-workshop-final.html, April 3–4 1995, (accessed December 7, 1996).

Omniview Corporation (1995) *Fidelity User Guide*, Omniview Corporation, Pittsburgh, PA, 1995.

Sullivan, M. (1996) "IEEE DASC VHDL Analog Extensions Working Group (IEEE 1076.1)," http://vhdl.org/vi/analog/wwwpages/Welcome.html, (accessed December 7, 1996; work in progress).

Thomas, D. and Ernst, R. (eds.) (1996) *Proc. Fourth Int. Workshop on Hardware/Software Co-Design*, IEEE Computer Society, Los Alamitos CA.

White, S., Alford, M. and Hotlzman, J. (1994) "Systems Engineering of Computer-Based Systems." In: Lawson (ed.), *Proc. 1994 Tutorial and Workshop on Systems Engineering of Computer-Based Systems*, IEEE Computer Society, Los Alamitos CA, pp. 18–29.