Obstacles to Using CAD Tools for Embedded System Design: an automotive case study

Philip Koopman
Engineering Design Research Center
Carnegie Mellon University
1205 Hamburg Hall
Pittsburgh, PA 15213, USA

koopman@cmu.edu
http://www.cs.cmu.edu/~koopman/

Abstract

Historically, Computer Aided Design (CAD) research for digital electronics design has emphasized support for the largest and most technically difficult projects. However, the success of many embedded system design efforts depends more on system-level and lifecycle cost optimization than on an ability to synthesize hardware containing millions of transistors. Thus, existing CAD tools may not offer embedded system developers the capabilities they need. This paper reports the results of a case study using a digital design synthesis 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. Problems having to do with electronics design, system-level design, and engineering/business processes are reported. At least some of these obstacles will have to be overcome by CAD tools in order to address the special needs of embedded system developers.

1. Introduction

Computer Aided Design (CAD) tools continue to progress 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. Additionally, simulation and modeling capabilities have matured to the point that first-time working silicon is an attainable design goal. Historically, CAD improvements have focussed 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], [White94], [Gajski94], [Lightner95]). Additionally, a research community in hardware/software codesign (*e.g.*, [Thomas96]) has formed to explore CAD support for cross-disciplinary tradeoffs in embedded system design.

An important fact about embedded systems is that the volume of "low-end" systems having modest computing requirements is quite large. In fact, out of 2683 million embedded microcontrollers shipped in 1994, fewer than 3% were 32- or 64-bit processors [Cole95]. 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. But, the problems that CAD tools must solve to effect improvements might be different for many embedded systems than for desktop computers.

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 (HDL) 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 their benefits were demonstrated. However, the result of this case study is that the digital design synthesis tool evaluated 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 is generally applicable to many situations.

2. 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.

2.1. 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. Typi-

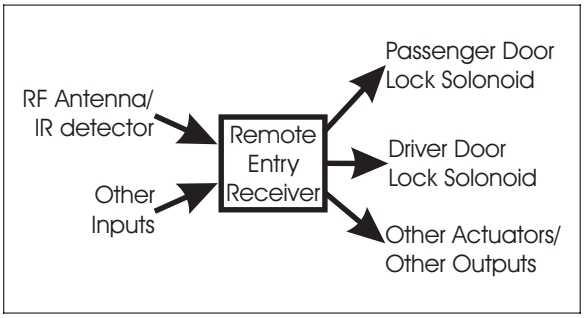

Figure 1. Generic automotive Remote Entry receiver unit.

cally 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 to a high-end 8-bit microcontroller.

Each RER board design is slightly different, depending on OEM requirements for interfaces and features. The RER unit is connected to solenoids that drive door locking mechanisms and other peripherals, and in some cases may be connected by digital serial lines or even an embedded network to other automotive subsystems. Radio frequency-capable RER units typically have a low-cost antenna implemented a trace on the circuit board. The RER unit must be able to handle different levels of output current driving capacity, as well as different input signal types.

2.2. The synthesis tool

This case study evaluated the effectiveness of a commercially available design-by-composition synthesis tool, Fidelity [Omniview95]. Fidelity is a commercialization of the Micon [Birmingham92] 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 output correspond to inputs and

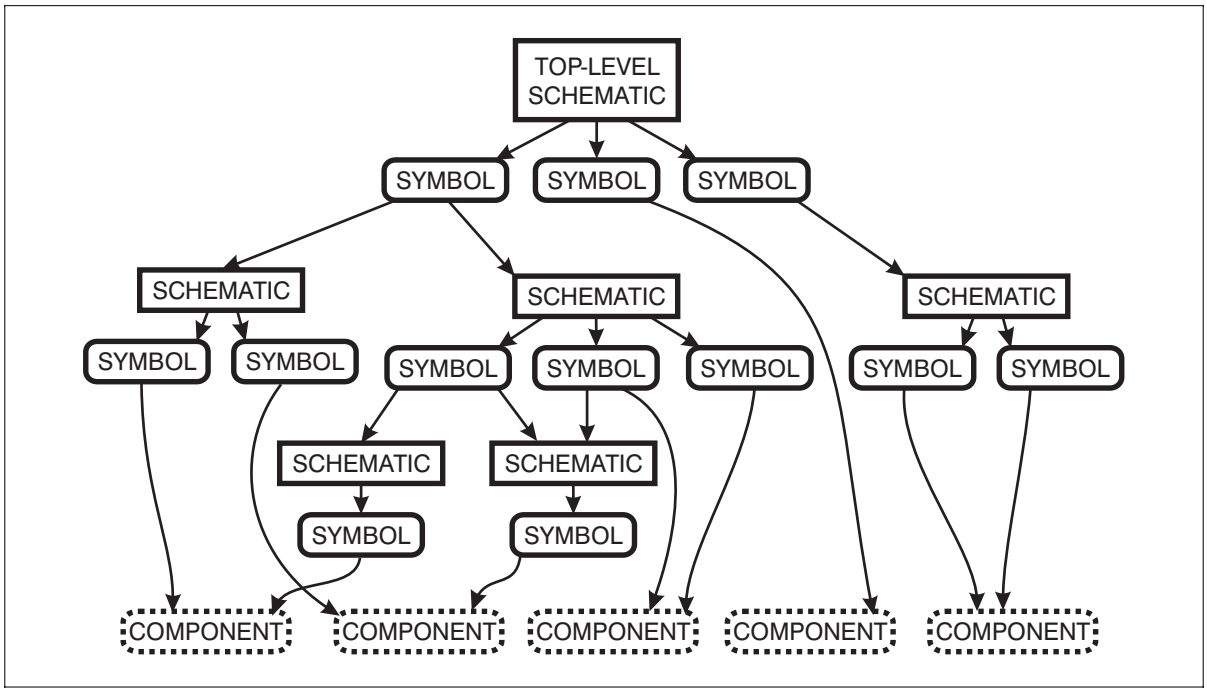

Figure 2. Fidelity uses a design hierarchy of alternating components and schematics.Multiple schematics associated with a component indicates the potential for alternative design decisions.

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).

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 a 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.,* the current carrying capability of 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 (Figure 3). Fidelity then generates a succession of all feasible design instances.

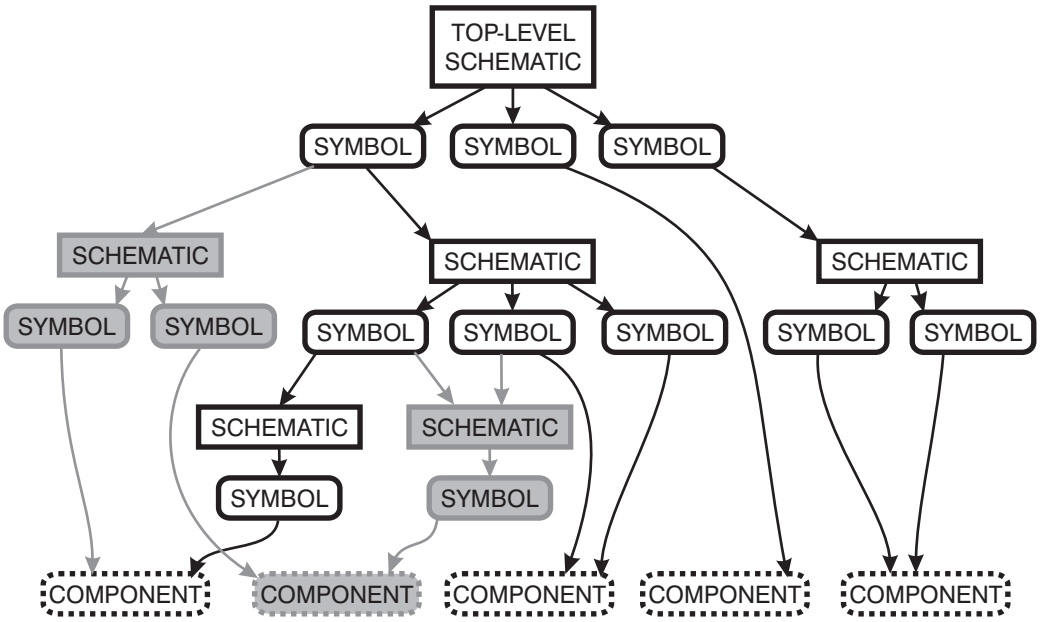

Figure 3. A single synthesized Fidelity design is created by selecting one, and only one, alternative for each symbol.The grey areas indicate schematics and symbols not used by this particular implementation alternative.

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 summed 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. Fidelity also has the ability to use external design advisers to take into account system-level effects and give more refined measurements of emergent system properties such as board area in the face of trace layout congestion, but these advisers were not available at the time of the study.

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 is able to perform 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 systems, the conclusions and observations in later sections are not meant to be critical of 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.

2.3. The experiment

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 centerpiece 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 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 periodically redesign subsystems so as to create cost-reduced versions as component prices, market conditions, and part availability changed.

Fidelity in particular was chosen because it uses schematics (making it compatible with existing practice), and because it seemed readily capable of handling analog and power design as well as digital design. Additionally, Fidelity is optimized for created 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 re-synthesize all four example RER designs. Because of time and resource limitations only one design was actually created. However, it was obvious at the conclusion of the work that once one design was created there were few if any additional obstacles to creating the other designs. 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 users from five other embedded system manufactures were briefed on the case study and in return 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, wasn't comprehensive enough to be immediately useful.

In a strict sense, Fidelity was successful. It was able to synthesize an optimized design within an hour. As with any beta-version software tool there were setbacks associated with installing, learning, and using Fidelity. In time, the tool could have been deployed and been made to be productive for its advertised function. However, it turned out that rapid synthesis of schematic designs under the limitations that Fidelity imposed did not solve enough of the problem at hand to make the cost of adopting the tool worthwhile.

The case study raised a number of issues with respect to electronics design, system design, engineering process issues, and business issues. While these problems are presented in the context of this specific case study, experience with a broad range of other embedded systems suggests that most of them are of general importance [Koopman96]. Many if not all of these areas would have to be addressed for Fidelity, or any CAD tool, to be useful for the RER case study area.

The observations in this paper come from the author's experience with not only the RER system, but with several commercial as well as military applications, development methodologies, and life-cycle support activities. All characterizations of embedded systems beyond the RER example are implicitly qualified to indicate a typical, representative, or perhaps simply an anecdotal case rather than a definitive statement about all embedded systems. While it is understood that each embedded system has its own set of unique requirements, it is hoped that the generalizations and examples presented here will provide a broad-brush basis for discussion and evolution of CAD tools and design methodologies.

3. 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. Rather than simply offering a semi-automated hierarchical schematic database methodology, Fidelity has the added ability to include some design rationale information in the schematic database in the form of design equations. This allows fully automated synthesis to take place once a top level schematic and any additional supporting schematics have been added to the database. 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.

However, it turned out that excellence at digital design level 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 incremental design updates.

3.1. Digital, analog, and power components

A typical RER system has only one or two digital components — a single-chip microcontroller and potentially an accompanying program 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 is much more complicated than the digital circuit design. Many embedded systems are able to use a small, highly integrated microcontroller with many I/O pins, and thus it is not unusual to have non-digital components comprise the bulk of a circuit board design. (The design complexity of selecting that microcontroller is discussed in the next section.)

As with most digital design tools, Fidelity has the ability to represent analog components. But, the real issue is 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. However, Fidelity selects from pre-designed circuits rather than synthesizing combinations of components from scratch. Therefore the

non-digital sections of the RER design were decomposed into building blocks of analog design circuits. For example, output switching circuits were arranged so that different circuits could be used depending on drive current requirements.

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 modelling 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.*, [Sullivan96]).

3.2. 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 cheaper 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 designs in more than one industry.
- Using standard parts avoids the initial component engineering/mask costs, inventory requirements, and risk for custom-made components. One issue is what happens if the RER order from the OEM is canceled or reduced in size. Standard parts can find a use elsewhere if they have not already been programmed or had memory mask layers fabricated, whereas custom parts may have to be thrown away.
- 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, *not* synthesizing digital circuitry. This selection is generally performed based on cost, I/O pins, and memory configuration. Because software is only a few hundred bytes in size, instruction set of the microprocessor is not as important as cost and reputation of the supplier.

Fidelity is oriented to composing a solution out from defined building blocks rather than searching a multi-parameter space for a single matching component, and so was not able to readily accommodate microcontroller selection. It is possible that other CAD tools designed to synthesize high performance computer systems from components will have similar problems.

Conclusion: While very large embedded systems may have 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.

3.3. 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 features, 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 micro- or milli-Amps 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. Newer versions of Fidelity may have increased flexibility with respect to 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 laptop and mobile computing applications will spur digital CAD tools to provide more comprehensive power management capabilities, which will be to the benefit of many embedded system designers.

3.4. Layout vs. logic design

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

While the RER designs often have the luxury of a multi-layer circuit board, some 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 layout optimization to reduce cost. Barring special-purpose layout optimization tools, support for manual intervention of layout decisions as well as back-annotation to the design source is essential, even for automatically synthesized designs.

3.5. Incremental design updates

It is common for an embedded system such as an RER to be partially redesigned in order to provide additional features, interface to additional I/O devices, or inject new technology in the form of updated components. Ideally, the investment in a machine-readable design database can be exploited to simply resynthesize a design with minor changes. In general, synthesis tools are not concerned with constraining results to maintain a large degree of similarity with previous designs, but rather in producing globally optimal designs.

However, substantial investments may already be made in component inventories, tooling, training, and purchasing agreements when it is time to update a design. Additionally, it is often desirable to avoid re-engineering circuit board layout details that resolve noise, thermal, or mechanical clearance issues. Therefore, in order to minimize total lifecycle cost (given the history of the design and its production), it may be desirable to accept a suboptimal circuit schematic that has a high degree of similarity in components and layout to the old design.

Conclusion: CAD tools that are meant to be used to create updates as well as initial designs must take into account the possible desire to limit the scope of changes, even at the expense of forgoing an optimal design.

4. System design constraints

At a level above electronic design, there are constraints and problems that must be addressed at the system-wide level. Two important problems encountered in this case study are the issues of design rule variation and clock speed limitation.

4.1. 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. Although not currently applicable to RER units, in many industries embedded products must meet safety code requirements that vary considerably between countries.

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 component integrated into their vehicles.

These factors result in a situation where 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. An obvious way to accomplish this is to include a parameter that selects either more expensive or less expensive parts that presumably result in a reliability vs. cost tradeoff (whether this in fact is a useful approach is unclear). Another potential strategy is to incorporate different design schematics in the database for use in different design situations, such as selecting input protection circuits by setting a variable with the name of the OEM the design is being created for. But, there was not time to attempt to implement either strategy, and the issue remains open.

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. Digital design tools concentrate on correct vs. incorrect designs, and may optimize for fast/large or slow/small design tradeoffs. However, in embedded designs there may need to be many different design parameters and a desire to use a single master design database rather than separate designs or component libraries for each application situation.

4.2. Clock speed limitation

One of the RER designs employed cryptographic encoding of messages in order to deter theft. A special encoding was invented [Koopman94] not only to reduce memory costs, but to reduce costs by limiting computation requirements. In these days of multi-hundred megaHertz processors, an RER CPU is limited to less than approximately five megaHertz by Radio Frequency Interference (RFI) concerns (a too-fast CPU can interfere with the receiver antenna for RF units, as well as with the AM/FM radio in the vehicle, and shielding is an added cost). As a result, special tradeoffs had to be made in the cryptographic algorithms to increase speed given a limited computation capability.

Conclusion: In addition to power limitations, RFI concerns occur in many embedded systems. The amount of computation per clock cycle may be a more important design metric than maximum theoretical system throughput. This is a design tradeoff that is not of primary importance in desktop computing applications.

5. 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 costs models, system certification costs, CAD tool proficiency, database maintenance, and electronic- vs. paper-based design representations.

5.1. 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 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 [Demes93]. 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. Thus, substituting a new component in one place may increase the cost of all other instances of using the old component by lowering total order volume.
- Cost of purchasing components, which includes time spent researching availability and obtaining competitive price quotes. The time spent making purchases is tied more to the number of different components being purchased and the number of vendors involved than to the quantity of any given component.
- Cost of qualifying a new component for operation, including environmental, stress, and lifetime testing (the vendor may be able to supply such data, but accepting it requires qualifying the vendor to make the tests). In some cases the OEM must be asked for permission to use a component (which may involve OEM qualification procedures), which consumes additional time and effort.
- Cost of qualifying a new vendor if the component is unavailable from existing vendors. It is vital that vendors reliably deliver components as part of a "just-in-time" manufacturing strategy. Also, in many cases vendors must be qualified to perform inspections in order to relieve the embedded system manufacturer of the burden of doing so.
- 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. Selecting more component types than fit onto a single placement machine can result in a need to purchase an additional placement machine, or reduce manufacturing throughput by using multiple component placement passes. 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.*, [Harris96])
- Likely cost changes for the components over the multi-year production life of the system, driven by factors such as emerging competition, demand from other manufacturers, and availability of fully-depreciated integrated circuit fabrication plants.

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. Ways that CAD tools can support this include:

- Identifying opportunities to use a single component type within a design (for example, substituting a tight-tolerance component for a low-tolerance one if the tight-tolerance version is required elsewhere in that or other designs).
- Consolidating component usage across multiple designs.
- Using suboptimal components from existing vendors rather than forcing new-vendor qualification.
- Balancing component placement machine loads when selecting top- or bottom-side placement

5.2. 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 exploit changing component costs to continually re-optimize production costs. Given productions runs of up to a million units a year, costs for retooling would likely be minimal 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 and customer acceptance of changes.

5.3. 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 transfer to 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, this can be perceived as reducing the number of engineers available to do "real" work (even if the net effect is increased total productivity). More importantly, the one or two CAD experts become an organizational vulnerability, and in the high-turnover automotive industry, their loss could make it impossible for the organization to do any design at all. 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 practical approach in the case study was to only use relatively simple CAD tools, and only those with intuitive user interfaces.

5.4. Model & library database maintenance

When switching from a schematic design approach to an automated synthesis approach, considerable additional information must be entered into the design as machine-readable information. A significant part of the work 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 particular needs of 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. In addition to initially placing a component into a database there are several additional costs, including:

- Updating pricing at various quantity levels and lead-time information for component acquisition.
- Translating to multiple formats which may have incompatible database fields if several different CAD tools are being used.
- Translating from vendor part numbers to internal part numbers. Companies may use a single internal part number that maps to multiple vendor part numbers in order to maintain flexibility in component suppliers choices (and thus garner negotiating power).

- 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.

5.5. 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 electronic as well as mechanical design information, qualification test results, and other data in a central repository.
- The increased risk of theft or compromise of centrally archived on-line data (especially on networked computers) 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 vendor-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. The risks and costs associated with maintaining obsolete CAD tools and their execution environments make it difficult to justify keeping only electronic copies of designs on any but the largest design efforts. However, 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. These issues seem to constitute a quiet but prevalent unsolved problem across many embedded system designs.

5.6. 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 problems discussed 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 environments, a compelling advantage to adopting advanced CAD tools 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.

6. Observations from other domains

This paper is focussed specifically on applying digital CAD tools to an example high volume/low cost embedded system design. A previous conference paper [Koopman96] 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 therefore summarized here to present a more complete picture of embedded system design issues.

6.1. Real time performance

Many embedded systems have real-time performance requirements in which they must meet certain deadlines for computational results. However, some architectural features of modern high-end processors, such as cache memory, result in varying and difficult-to-predict timing behavior [Koopman93]. Many embedded systems must be over-designed to ensure they meet worst-case performance deadlines. Improving the accuracy of guaranteed performance bounds (*e.g.,* [Yen95]) can permit using a somewhat less expensive processor, or permit increased capability by adding features while retaining a given processor.

6.2. Size and weight limitations

Although the RER example had some limitations on size and weight, they were not as severe as those encountered in aircraft, spacecraft or portable electronic device applications. Design for non-planar geometries may help with fitting electronics into unusual form factors.

6.3. Safety and low-cost reliability

Many safety-critical applications require highly available systems. Traditionally, military and aerospace systems have employed multiply-redundant computers or distributed consensus protocols in order to ensure continued operation after an equipment failure (*e.g.*, [Levi94], [Siewiorek92]). However, in commercial embedded applications these approaches are too costly. Designing low-cost, highly available systems using components with only ordinary reliability properties remains an open problem. Additional safety issues include system-level tradeoffs between digital and electromechanical components as well as software safety concerns [Leveson94].

6.4. Harsh environment

Many embedded systems have to endure extremes of temperature, vibration, shock, lightning, power supply fluctuations, water, corrosion, fire, and general physical abuse. More comprehensive treatment of this issues is required, as well as integration with synthesis tools [Lightner95].

6.5. End-product utility

In many embedded systems the efficient employment of the fastest available processor is not an overriding concern. Rather, it is often the selection of a processor that is adequate and has the lowest system/lifecycle cost that is of interest. Unlike the desktop computing or digital signal processing domains, it is not the availability of compute cycles that drives the design, but rather the I/O requirements and required software functionality that is paramount. This suggests that strategies that start with a software and I/O specification and move toward hardware may at times be more attractive than approaches that start with a more hardware-oriented specification.

6.6. Controlling physical systems

The RER example design is actually a subsystem within distributed control system. Several main computers control the engine, transmission, and "body" (peripheral) functions of an automobile. The RER unit is used to control some of the peripheral functions, thereby reducing the number of wires running to the body controller. Such distributed embedded system designs may become more common with time, and bring with them a need to perform design tradeoffs involving balancing embedded communication network costs with partitioning of functionality (*e.g.*, [Beck95], [Hou96]).

6.7. Long-term component availability

Automotive products have a lifetime that is shorter than some other embedded systems. Automotive subsystems may be manufactured for only a few years, and may be supplied as spare parts for approximately ten years after they have left production, depending on manufacturer and country of sale. Given the fact that few automotive components are routinely replaced during the life of a vehicle and that they are relatively low in cost, it is reasonable to simply manufacture spares while the units are in production and stockpile them. However, other embedded systems can have significantly longer design cycles, longer lifetimes, and poorer statistical predictability for future demand of relatively expensive spare components. These systems can be threatened by components going out of production while the units are still being manufactured, or when spare units are needed. In these cases an additional argument in favor of design synthesis is present, namely that synthesis forces a complete and unambiguous specification of the design so that it may be recreated from different technology at a later date. This potential benefit might outweigh some of the disadvantages to synthesis observed in the RER case study.

6.8. Computer culture *vs*. other cultures

In many embedded design organizations business policies and management decisions are made by people who do not understand the current quantitative, simulation-based culture (*e.g.*, [Patterson90]) of computer designers. This is not because they are poorly trained, but rather because they may not be computer engineers. Computers form a relatively small part of a much larger embedded design in many cases, and company staffing often reflects this.

This lack of familiarity with computer design methodologies raises a problem in both managing projects and acquiring tools. Managers who have not been trained with an emphasis on simulation may not permit sufficient time for modeling and simulating the electronics within an embedded system, and may not have confidence in the results of anything other than actual hardware prototypes. A challenge is educate embedded system developers to the tools and methods available to them (*e.g.*, through [Miller-Freeman]), and to make the tools dramatically easier to use and thus accessible to non-computer experts.

7. 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.

8. Acknowledgements

Phil Lemay did the detailed work with the Fidelity synthesis tool, and discovered techniques needed to successfully synthesize an RER design. The engineers of the Input Controls group within United Technologies Automotive provided significant expertise and help in order to make this case study possible. In particular, Keith Christenson provided the access RER to design information. In addition to United Technologies support, portions of this work were sponsored by DARPA contract DABT63-95-C-0026, and ONR contract N00014-96-1-0202.

9. References

[Beck95] James Beck, *Automated processor specification and task allocation methods for embedded multicomputer systems,* Ph.D. Thesis, Dept. of Electrical and Computer Engineering, Carnegie Mellon University, 1995.

[Birmingham92] William Birmingham, Anurag Gupta & Daniel Siewiorek, *Automating the Design of Computer Systems: the Micon project*, Jones & Bartlett, Boston, 1992.

[Cole95] Bernard Cole, "Architectures overlap applications", *Electronic Engineering Times*, March 20, 1995, pp. 40,64-65.

[Demes93] Georgette Demes *et al.*, "The Engineering Design Research Center of Carnegie Mellon University," *Proceedings of the IEEE*, **81**(1) 10-24, January 1993.

[Gajski94] Daniel D. Gajski, Frank Vahid, Sanjiv Narayan & Jie Gong, *Specification and Design of Embedded Systems*, PTR Prentice Hall, Englewood Cliffs NJ, 1994.

[Harris96] "Harris EDA introduces EDA*assimilator* software for manufacturing synthesis," Harris Electronic Design Automation Inc., 1995

[Hou96] Junwei Hou & Wayne Wolf, "Process partitioning for distributed embedded systems," *Proceedings: Fourth International Workshop on Hardware/Software Co-Design*, IEEE Computer Society, Los Alamitos CA, 1996, pp. 70-76.

[Kluwer] *Design Automation for Embedded Systems: an international journal*, Kluwer Academic, ISSN 0929-5585.

[Koopman93] Philip Koopman, "Perils of the PC Cache", *Embedded Systems Programming*, May 1993, **6**(5) 26-34.

[Koopman94] Philip Koopman & Alan Finn, *Pseudorandom Number Generation and Cryptographic Authentication*, US Patent #5,363,448, Nov. 8, 1994.

[Koopman96] Philip Koopman, "Embedded System Design Issues — The Rest of the Story" *Proceedings of the 1996 International Conference on Computer Design*, October 7-9, Austin, Texas, 1996.

[Leveson94] Nancy Leveson, *Safeware: system safety and computers*, Addison-Wesley, Reading MA, 1994.

[Levi94] Shem-Tov Levi & Ashok Agrawala, *Fault Tolerant System Design*, McGraw-Hill, New York, 1994.

[Lightner95] M. Lightner (ed.), *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).

[Miller-Freeman] *Embedded Systems Programming*, Miller Freeman, San Francisco, ISSN 1040-3272.

[Omniview95] *Fidelity User Guide*, Omniview Corporation, Pittsburgh, PA, 1995.

[Patterson90] David Patterson & John Hennessy, *Computer Architecture: a Quantitative Approach*, Morgan Kaufmann, San Mateo CA, 1990.

[Siewiorek92] Daniel Siewiorek & Robert Swarz, *Reliable Computer Systems: design and evaluation (2nd edition)*, Digital Press, Burlington MA, 1992.

[Sullivan96] Michael Sullivan, "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).

[Thomas96] Don Thomas & Rolf Ernst (eds.), *Proceedings: Fourth International Workshop on Hardware/Software Co-Design*, IEEE Computer Society, Los Alamitos CA, 1996.

[White94] Stephanie White, Mack Alford & Julian Hotlzman, "Systems Engineering of Computer-Based Systems." In: Lawson (ed.), *Proceedings 1994 Tutorial and Workshop on Systems Engineering of Computer-Based Systems*, IEEE Computer Society, Los Alamitos CA, 1994, pp. 18-29.

[Yen95] Ti-Yen Yen & Wayne Wolf, "Performance Estimation for Real-Time Distributed Embedded Systems," In: *Proceedings of the 1995 International Conference on Computer Design (ICCD '95)*, 1995.