Assessing Design Activity in Complex CMOS Circuit Design

Gautam Biswas, Susan R. Goldman, Doug Fisher, Bharat Bhuva, and Grant Glewwe
Vanderbilt University

Technical Report
March, 1994

Reproduction in whole or part is permitted for any purpose of the United States Government.

This research was sponsored by the Cognitive Science Program of the Office of Naval Research, under Grant Number N00014-91-J-1680, Contract Authority Identification Number NR4421571–03.

Approved for Public Release; Distribution Unlimited.
This chapter characterizes human problem solving in digital circuit design. We analyze protocols of designers with varying degrees of training, identifying problem solving strategies used by these designers, discuss activity patterns that differentiate designers, and propose these as a tentative basis for assessing expertise in digital design. Throughout, we argue that a comprehensive model of human design should integrate a variety of strategies, which heretofore have been proposed as individually sufficient models of human design problem solving. We close by describing an automated tool for design and its assessment.
Assessing Design Activity in Complex CMOS Circuit Design*

Gautam Biswas, Susan R. Goldman, Doug Fisher, Bharat Bhuva, and Grant Glewwe†

Vanderbilt University
Nashville, Tennessee 37235

Tel. (615)343-6204, (615)322-8070
e-mail: biswas@vuse.vanderbilt.edu

March 1, 1994

Abstract

This chapter characterizes human problem solving in digital circuit design. We analyze protocols of designers with varying degrees of training, identify problem solving strategies used by these designers, discuss activity patterns that differentiate designers, and propose these as a tentative basis for assessing expertise in digital design. Throughout, we argue that a comprehensive model of human design should integrate a variety of strategies, which heretofore have been proposed as individually sufficient models of human design problem solving. We close by describing an automated tool for design and its assessment.

*Supported by Office of Naval Research grant: N00014-91-J-1769.
†Herman Vandermolen, Julio Ortega, Carolyn James, Laura Novick, and Olin Campbell were part of the ONR project group and made significant contributions to this project.
1 Overview

Cognitive diagnosis of expertise relies on having a characterization of expertise in the domain. The focus of our work is on the design of digital circuits. In this domain, previous work was limited either to analog circuits or to much simpler elements of circuit design than that involved in the complex circuits we were interested in. However, the design of complex circuits allowed greater possibilities for observing multiple levels of expertise as well as individual differences in design problem solving. In earlier work (James, Goldman, Vandermolen, 1993; Vandermolen, etc.), we had focused on the design of a simpler digital circuit and the need for a richer design problem became apparent. Our investigation of complex digital circuit design had two major goals: to identify characteristics of design problem solving in this domain, and to determine how to appropriately characterize and differentiate among individuals using different design problem solving processes. When designing complex circuits, expert designers attempt to catch flaws as early in the design process as possible. Hence, we were interested in characterizing the processes involved in producing the final design, as well as the completeness and correctness of the final design itself.

We begin by describing a framework for analyzing design processes. Our empirical work uses this framework to analyze the design processes of subjects with varying levels of experience. However, from a cognitive diagnosis perspective, process assessment is resource intensive and it would be nice if classification of individuals could be solely based on the final product. We evaluate the relative information supplied by process and product characterizations for classification of expertise. We conclude that both process and product measures provide important information. We describe the design of an automated tool for collecting process and product information so as to make such assessment tractable on a wide scale.

2 Introduction to Engineering System Design

Engineering system design maps a specified function onto a realizable physical structure [Tong and Sriram, 1992]. This typically requires that the functional specification
of a system be decomposed, components be identified that satisfy various subfunctions, and that these components by integrated in a way that satisfies the overall system specification. Brown and Chandrasekaran [1986] and Sriram [1986] classify design tasks as follows.

1. **Routine** design is indicated when effective problem decompositions are known, the mapping from subfunctions to physical components is clear, and the task is to select the most appropriate set of components that optimize well-established criteria.

2. **Innovative** design assumes that top-level functional decompositions are known, but the physical realizations of subfunctions requires considerably more effort, such as constructing a solution from scratch, or making substantial functional and structural modifications to a known solution.

3. **Creative** design is called for when the functional specification is *open-ended* and/or ill-specified, effective decompositions are unknown, and the designer must evaluate a number of different options to determine an appropriate design plan.

This was proposed as a taxonomy of design tasks, but we feel that it better describes design tasks *conditional on* abilities of a designer. For example, the design of digital circuitry such as a network controller, is likely to be routine to a designer with considerable experience on such problems. In contrast, persons who are knowledgeable of circuit design and understand controller functionality, but have little experience designing such controllers are likely to take an innovative design approach. Functional decomposition will be easily achieved, but considerable effort will be expended in picking and composing suitable components to satisfy the overall functionality. Lastly, subjects who understand the domain of digital circuits and systems but have no experience in performing complex design, may try a number of decompositions and build subfunctionalities incrementally by trial and error. This is best characterized as creative design.

In sum, a routine task for an expert may well be a creative design task for a novice. We believe that characteristics of routine, innovative, and creative design process, as
well as characteristics of the final design, are important in assessing the expertise of
designers. We distinguish three broad levels of design activity.

1. Design activities at the function level study the specifications of a system and
generate functional units that collectively meet the required specifications.

2. The transition level includes design activities that look to functional units and
decide how to implement them as artifacts (e.g., gates and higher level blocks
such as shift registers and counters). Actions at this level may also guide the
coordination of component artifacts so that higher-level functionality is satisfied.

3. Implementation level activities deal with the actual generation of the artifact. In
complex design problems this often happens in stages, for example, simulation,
testing, and evaluation of partial designs.

These levels are closely tied to our earlier classification of design tasks: routine
design is revealed by ease in addressing subtasks at each of the function, transition,
and implementation levels; innovative design is suggested by behavioral determin-
ism by a designer at the function level, but nondeterminism at the transition and
implementation levels; designers with difficulty at each level are, by virtue of their
apparent abilities, performing creative design. The activities occurring at these levels
constitute indices of design problem solving processes. The final result of these pro-
cesses can be evaluated to determine whether the design meets the necessary circuit
specifications.

3 Expertise in Design

The previous section distinguished function, transition, and implementation levels;
these relate design activities to views or abstractions of the particular system being
designed. Researchers have also described orthogonal gradients that characterize the
cognitive processes of the designer, independent of the artifact being designed. For
eexample, Korpi, Greeno, and Jackson [1991] model design problem solving with two
spaces: (a) the problem construction space, and (b) the design construction space. The
problem construction space is populated by what they call senior managers who guide the design and do most of the high level planning. The design construction space has middle managers, who act as facilitators in deciding what part of the design to work on and also assess the goodness of an emerging solution, and builders who actually perform the detailed design tasks. Similarly, Goel and Pirolli [1991] look upon design activity in two parts: (a) the focus of the activity (monitoring, development, etc.), and (b) the problem solving step being executed, which includes a set of operators, such as add, modify, evaluate, propose, and request. Ullman et al. [1988] do not differentiate a small number of discrete levels of cognitive processing, but they nonetheless posit that designers manipulate behavioral episodes at varying levels of detail.

In our work, we have found two descriptive levels of reasoning sufficient: the strategy and unit operator levels. This chapter focuses exclusively on activity at the strategic level, roughly corresponding to the high-level management activities of Korpi, Greeno, and Jackson (1991). This and other work suggests high-level problem-solving strategies that might populate this space.

1. Designers often rely on a decomposition strategy that decomposes a complex design task into smaller subtasks until they are directly solvable. Examples of design systems that employ decomposition are Hi-Rise [Maher, 1988] and R1 [McDermott, 1982].

2. A transformation/refinement strategy converts initial specifications into a final design solution through a sequence of primitive operator applications in some formal representation scheme (e.g., shape grammars [Mitchell and Stiny, 1978]).

3. A case-based strategy assumes that a catalog of previous design solutions is stored in memory; a new design problem prompts the designer to search through this catalog, and select one or more designs that best match the characteristics of the goal design. Examples of case-based approaches to design include the Bogart system [Mostow, Barley, and Weinrich, 1989] and Struple [Zhao and Maher, 1988].

Most automated design systems and studies of the human design process have tended to focus on one strategy. In contrast, Maher [1990] discusses the relevance of all
three strategies to design and states “the value in identifying multiple models of design lies in the richness of the representation of design knowledge and experience provided by each and in the ability to choose a model that more closely fits the knowledge readily available for the domain being considered.” For example, a designer may decompose a specification, access previously constructed components or cases that best match the subfunctions identified through decomposition, and transform these cases into new component structures that perfectly fit the requirements of the new system being designed. Thus, all three strategies may be present in the solution of a single design problem.

We hypothesize the presence of each of these strategies in designers. The possibility of multiple strategies in a single design solution provides a much richer framework for characterizing the reasoning methods that designers employ in complex design problem solving. We look upon design as an application of reasoning processes that start at the function level, go through a transition level, and produce an implementation-level description of the artifact. Several strategies can be applied in this process. The directedness with which these strategies are employed and coordinated is indicative of routine, innovative, or creative design, and we believe, can serve as a basis for assessing a designer’s expertise relative to a design problem or class of problems.

4 Complex Circuit Design

Our objective is to determine whether designers employ multiple problem-solving strategies, and if so, what is the interaction of these strategies across levels of expertise. We have selected circuit design as a testbed for these studies. In particular, consider the problem of designing a network controller, which coordinates communications between computers that are linked by cable. A more complete specification is given in Appendix A. Our experience with this and similar problems suggested numerous activities that would be observed across a range of designers. We classify these activities into the function, transition, and implementation levels as introduced earlier.
4.1 Function Level

A complex circuit is usually made up of many functional blocks. The design task is eased considerably if the designer identifies these blocks. Functional blocks may or may not be independent, but nonetheless their identification is generally critical. Based on our experience, a natural solution to the network controller problem would identify many functional components, including a serial-to-parallel buffer, busy bit logic, and a system state buffer (idle or sending). The activities that designers demonstrate in this process are generally best characterized by decomposition strategies.

A second functional activity is to coordinate functional blocks. Typically, this takes the form of a flow chart, state machine, or signal-flow diagram. An example of a flowchart for the network controller is shown in Fig. 1. In general, flowcharts and similar structures are used by designers to explicate interactions between modules. In a network controller these interactions typically relate to signals between components; communication between components must be coordinated if overall functionality is to be correct. The construction of an intermediate form (IF) of the circuit such as a flowchart is evidence of a designer's global plan for further design.

<Figure 1 about here>

4.2 Transition Level

The next step for the designer is to implement each of the functional blocks and to coordinate these implementations. A number of different strategies are possible. Very often, designers use the transformational strategy of iterative addition or refinement by picking a core functionality, implementing this functional block in some way, and then implementing functions around this core block until a complete design is obtained.

In contrast, experienced designers may be reminded of a more complete prototype and exploit a block diagram representation of that prototype system. This case-based strategy is typically augmented by a transformational strategy of iterative refinement, where local changes to the prototype are made until it satisfies the required specifications.
In still other cases, after constructing a global plan for the solution of a system, the designer may find that certain component functions require further decomposition due to their complexity. Component functions may be decomposed in this manner until functions corresponding to known physical components are produced. More generally, however, there is considerable room for variance in the level of abstraction at which decomposition stops. For example, decomposition may continue down to basic logic functions that can be implemented as gates, or it may stop at more complex functions that are implementable as physical components such as shift registers, counters, or multiplexers.

4.3 The Implementation Level and Interactions between Levels

Finally, at the implementation level physical components are being manipulated. These manipulations often introduce interactions between components that were not anticipated at the function and transition levels. In some cases, interactions can be patched immediately at the implementation level, whereby additional physical components are linked in to overcome an undesirable interaction. Patches may be deferred as well; a designer may feel that an interaction can be patched, but it is best to await implementation of other functional blocks. Patching is best viewed as the transformational problem solving strategy. In other cases, a designer may return to the function level, abandon all or part of a previous decomposition, and reanalyze the system or part of it at the function level. This is symptomatic of creative design.

In sum, the design process may not be a single, clean iteration through the function, transition, and implementation levels. Furthermore, we expect that more than one of the problem solving strategies that we discussed – decomposition, transformational, and case-based – will be used.

5 Characterizing Circuit Designers

There are several questions implied by our discussion that are relevant to the characterization of circuit designers.
5.1 Characteristics of the Function Level

Questions stemming from our discussion of the function level are:

1. How much does a designer's function level description cover the system's overall required functionality?

2. Does the designer organize functional units into an intermediate form, and if so, what is it (e.g., flow chart)?

3. Collectively, does the functional decomposition and intermediate form indicate that the designer is employing a global plan to implement the circuit.

Our experience suggests that designers producing intermediate forms tend to produce better and quicker designs than those that do not do this sort of planning. For complex designs this process is likely to happen through a series of functional decompositions. More experienced designers are likely to be reminded of prototypes that they will employ to cover one or more functional blocks. Experienced designers also tend to spend more time in functional planning and decomposition producing greater functional detail; this behavior makes the transition process quicker and more concise.

In sum, we define the following features for characterizing strategic design activity at the function level:

- Number of components the designer considered
- Type of Intermediate Form (IF) generated (e.g., flowchart).
- Number of the Components in the Intermediate Form.
- Amount of detail the designer introduced into the Intermediate Form.
- Global Plan.

The amount of detail in the intermediate form can range from low, medium to high. The greater the amount of detail the more strategic planning the user has done at the function level. Global plan refers to whether the designer explicates some form of global strategy in generating the design.
5.2 Characteristics of the Transition Level

Questions of a transitional nature include:

1. **What strategies (decomposition, transformational, case-based) does the designer use after constructing an intermediate form?**

2. **If the designer employs further decomposition, then at what level of detail does decomposition stop?**

3. **If a transformational strategy of iterative addition or refinement is used, then what functional component is selected as the core component?**

At this level, observations are made as to whether the designer specifies a prototype design that covers a number of functional units, or starts implementation on a single functional block and then includes others. Note that the designer may start with a complete prototype, partial prototypes, a single core component, the first unit in a sequence, or a random component. Designers who separate out the individual functional blocks that make up the designed circuit during the transition level and explicitly define the required communication signals between the blocks are usually more successful in generating correct and complete designs. The modularity created allows the designer to simulate and evaluate the current circuit with much less effort than a single large circuit.

The following characteristics are defined to analyze behavior at the transition level:

- Did the designer pick on a core component for the design?
- Did the designer use case-based reasoning and prototypes?
- How much detail did the designer incorporate during decomposition?
- What kind of signal definitions did they generate to connect individual functional blocks?
5.3 Characteristics of the Implementation Level

Questions at the implementation level include:

1. What complexity of physical components does the designer use to implement functional blocks? These may be high-level components, such as registers, or low-level gates. In some cases, a designer assumes custom components, which cannot be taken ‘off-the-shelf’, but which nonetheless are easily implementable.

2. Does the designer check for problematic interactions concerning timing of signal interactions between components, and if so, are these checks made locally within a single functional block, or globally across functional blocks?

3. How does the designer correct undesirable interactions? Does the designer patch problems immediately, or does the designer defer patching, partially or entirely until other blocks are implemented? Does the designer simply abandon an evolving design in the face of complications?

A primary characteristic here relates to the specific strategy employed by the designer in generating the implementation. Choice of higher level prototype blocks (e.g., multiplexer) and custom components (9-bit shift register) that simplify design are indicative of higher levels of expertise. Iterative refinement and improving the design implementation by making local patches with global verification is better than employing iterative additions which implies implementation one block at a time and a complete evaluation of the emerging design with the addition of each block.

Partial checking and postponement of refinement to account for later interactions amongst functional blocks is indicative of higher levels of expertise. This produces greater efficiency because a designed block does not have to be repeatedly redesigned after new dependencies are discovered during implementation.

The following summarize the features used for analyzing behavior at the implementation level:

- The strategy employed,
- Use of high level components,
4 Use of custom components,

- Interaction checks between modules, and

- Method of problem resolution.

6 Experimental Study

We administered the network controller problem to eleven subjects. Three were considered experts. One is a faculty member at Vanderbilt University, but he has designed digital systems for industrial applications (S8). The second expert works in the communications technology industry (S7). The third expert is a graduate student with extensive CMOS digital circuit design in industry (S11). Two subjects, S4 and S10, had related industrial experience, but neither could be considered to be experts in complex circuit design. Four subjects (S2, S5, S6, and S9) are graduate students with little or no professional design experience. The other two subjects were seniors in our undergraduate program at Vanderbilt university. They both had completed a senior level course in digital circuit design. However, subject S3 was regarded as a high performing student, whereas S1 was considered to be average in academic performance.

6.1 Protocol Analysis

The subjects were given a hardcopy of the problem description (see Appendix A), and asked to develop their solution with pencil and paper. They were requested to think aloud as they developed their solution, and the entire session was videotaped. The subjects could use any material (notes, books, etc.) that they felt would assist them in generating a solution. A complete transcript of each session was generated.

Later a group of raters (Biswa, Glewwe, Bhuva) studied the tapes, transcripts, and the subjects’ written output. To encode strategic activity, we used the following two-step process to analyze subjects’ data:

1. In Step 1, we encoded the implementation-level blocks created by the subjects, and the sequence in which they added, deleted, and modified blocks as
they went through the design process. In other words, this trace records the designer's activities at the implementation level. Most implementation level blocks originated from a function-level description and retain the same name as the functional block (e.g., serial-to-parallel buffer). Subjects usually put down this name when they drew this block as part of the implementation on paper, or they verbally expressed the name of the block as they drew it on paper.

2. In Step 2, we used the Step 1 trace and information from the tapes and transcripts to answer the set of questions that define our framework for characterizing design activity (discussed in Section 4). This provides information that describes strategic level behavior of subjects at the function, transition, and implementation levels.

The trace generated in Step 1, in addition to providing information for creating Step 2, also helps us check the correctness of the design. If the subject has errors in the design, the trace helps us understand where and how a subject went wrong. In combination with the Step 1 trace, it helps determine the subject's characteristics, such as (a) did the subject attempt to pick up higher level blocks and off-the-shelf components to simplify the design, (b) did the subject implement more at the gate level, (c) if so, did the subject try and optimize the design in terms of the number of gates, and (d) did the subject think about chip design in component selection.

The Step 2 trace is the key to recognizing levels of expertise. The sequence of reasoning methods applied at the function, transition, and implementation levels is a key indicator. For example, two designers may produce the same final solution. The first designer may employ case-based reasoning methods to generate a complete prototype system, while a second designer may not have access to a ready-made prototype model. Therefore, this designer uses function-level description to understand what is required, and then completes his/her design primarily from first principles. The first designer's Step 2 trace will show function-level activity, followed by transition-level activity, and then implementation-level activity. The second designer's Step 2 trace is likely to demonstrate back-and-forth activity between the function, transition, and implementation levels. Circuit implementation is likely to occur more by incremental
additions.

6.2 Process Characteristics

The dimensions described in Section 4 were applied to analyze the Step 2 behavioral characterization of the 11 subjects. Tables 1-3 summarize the individual behaviors along the function, transition, and implementation levels, respectively.

Table 1 shows the evaluation of each subject at the function level. For this analysis, we assumed a normative solution, consisting of 7 functional blocks, and an intermediate form given by the flowchart in Figure 1. This constitutes the 'gold standard' against which we compared subject solutions. We were interested in assessing how close to this solution each subject came. Because subjects might differ in the number and organization of functional blocks from the norm, we developed an algorithm for counting the amount of 'functional' overlap. If a designer defines a functional block that covers more than one of the 7 specified blocks, all 'covered' blocks in the normative solution are counted as considered. If the designer uses finer-grain blocks than the ones specified, a set of finer-grain blocks are counted as a larger normative block if the designer effectively 'implements' this block with appropriate links between the finer grained blocks. For example, subject S3's functional description covered 6 blocks of the normative solution.

In addition, Table 1 differentiates between the number of functional blocks initially considered (obtained from recorded protocols), and the number of blocks that were represented in the final intermediate form; these are given in columns 1 and 3, respectively. Other features used for assessment are the type of Intermediate Form (IF) used, details specified in IF (i.e., detail in specifying links between blocks), and verbal evidence for the presence of a global plan (i.e., some explication by the subjects of the steps that they will follow in fleshing out the design). Some of the subjects did not use IF and they were evaluated based on the rest of the design features. The use of higher number of components, greater detail in IF, higher number of components in IF, and the presence of a global plan imply higher expertise. However, in assigning
an overall rating of observed competence at the function level (column 6), we focused
on the coverage and detail of the intermediate form: no IF (rating 1), low/medium
coverage of IF (relative to normative solution) and low/medium detail (2), at least
medium coverage and detail (3), and both high coverage and detail (4).

<Table 2 about here>

Table 2 shows the evaluations of each subject at the transition level. The features
used are the details in decomposition and the use of interaction signals during this
level. Both of these measures are relative measures. In general, the more fine-grained
the decomposition and the more detailed the anticipation of interactions through
signals, then the greater the ease with which design should proceed and the greater
the presumed level of expertise. Column 4 lists overall ratings obtained as follows:
low decomposition and use of signals (1), at least medium decomposition and no more
than medium use of signals (2), high decomposition or use of signals, but not both
(3), both high decomposition and high use of signals (4).

<Table 3 about here>

Table 3 shows the evaluations of each subject at the implementation level. The
features used for assessment are the use of higher-level components, use of custom
components, types of interaction checks, and the strategy employed in problem res-
olution. This table does not give an overall rating, but as we noted earlier, the use
of high level and custom components demonstrates greater expertise, as does global
versus local interaction checks. In addition, most subjects used patching to correct
for interactions. In contrast, subject S1 abandoned a partially flawed design, which
is indicative of low expertise.

6.3 Product Characteristics

In addition to characterizing subjects in terms of process characteristics, we can
also characterize them in terms of the completeness and correctness of their final
designs. Table 4 gives a qualitative score for the completeness of each subject’s final
implementation (i.e., what proportion of the specified functionality did the final design cover) and a score for correctness (i.e., the proportion of specified subfunctionalities covered by the final implementation). Table 4 also classifies each subject on a four point scale, on the basis of the combined completeness and correctness ratings. Level 1 scores reflect minimal completeness and correctness in final implementations. Level 2 scores reflect medium completeness and correctness. Level 3 subjects scored high on one dimension and medium on the other. One subject was classified at Level 4, and scored high on both dimensions of completeness and correctness. We have left subject S7 unclassified, because he took a radically different approach to the controller design than other subjects. S7 employed a programmable logic array in solving the problem, and was stopped early by the experimenter prior to full implementation.

Table 4 about here

6.4 Integrating Process and Product Characteristics

The product rating and the previously-described process characteristics are generally consistent with one another. An exception to this was S3 who performed well at the function and transition levels. This exception would have gone unnoticed had we only looked at product scores. We were concerned that other important differences in design activity were being masked by focusing on the product measure. We developed a more integrated rating scheme consisting of four levels as follows.

1. Novices show little proficiency in complex design. Novices scored low in completeness and correctness, and had low functional and transitional scores as well.

2. Average designers scored medium for completeness and correctness, or performed well at the functional and transitional level, which is indicative of a sound understanding of the problem and a strategy for proceeding.

3. Above average designers demonstrate good understanding of complex design, and the ability to see the design process through to an adequate solution. Their average completeness and correctness was above medium.
4. *Expert designers* have a very good understanding of design. In our study we classified one subject as expert. This individual scored high on both dimensions of completeness and correctness, and demonstrated mastery at the functional and transitional levels.

These categories are exemplified in the descriptions of the design problem solving behavior of our subjects, as follows.

### 6.4.1 Novice Designers

S1 and S2 were ranked as novices. S1 understood some basic principles of digital design, but did no function-level design. S1 could implement parts of the circuit, but showed complete lack of understanding of others. He made no attempt to decompose the circuit before implementation. Instead, he tried to iteratively add new components, but could not handle flaws that arose because of interactions. S2 was more proficient in function-level design, but had no idea how to convert the functional blocks into a circuit implementation. His main problems were in transition-level design. He did not use signals to separate functional blocks of the circuit and became bogged down with trivial details. Both subjects might be regarded as performing creative design: failure to adequately decompose and isolate functionality led to non-determinism at lower levels, which were difficult for these subjects to handle.

### 6.4.2 Average Designers

S3-S5 were ranked average designers. S3 and S5 did some function-level design but none of the subjects had a complete understanding of the problem, therefore, they had difficulty in transition. All three did break down the system into some modules (incomplete) and were able to define signals to connect these modules. S5 did not include a portion of the circuit because he misunderstood the problem and was not corrected by the experimenter. S3 could not complete the implementation, as the two N/A entries in Table 3 might suggest, but nonetheless demonstrated proficiency at the function and design levels of the design process. Note that if we had only taken into account completeness and correctness aspects of S3's final design, then S3 would
have been classified as a novice. S4’s background in analog circuit design oriented him to focus on signal generation and timing analysis, but this approach did not produce a working implementation. He had more difficulty than the others assimilating the problem.

6.4.3 Above-Average Designers

Five subjects, S6-S10, were placed in this category. Three of the subjects produced good function-level design and all of them generated good solutions. S9 and S10 ignored function-level design and selected a shift register as a ring data buffer, respectively to start off the design process. This was notable, but isolated evidence of case-based reasoning. They did not have complete prototypes, however, and used iterative refinement to add to these components and complete their design. S10 used signal analysis to keep track of his overall design. Overall, barring the differences discussed, S6, S8, S9, and S10’s approach were similar, and their solutions were comparable. In contrast, S7 produced a non-standard solution and was stopped early by the experimenter prior to full implementation. Although his solution was not completely specified, he seemed to know exactly what he was doing and probably would have generated the complete solution had he been forced too. We can safely say that S7 was at least an above average designer, and might well have qualified as an expert had he finished the problem.

6.4.4 Expert Designers

S11 undoubtedly had a superior approach and produced the best solution to the problem. He performed detailed function-level design, and worked on refining the solution at this level before transitioning to the implementation level. Most of his analysis and checking occurred at the function level, therefore, error correction and recovery proved to be much easier for him than the other subjects who did a lot of evaluation and checking at the implementation level. The relative determinism at each stage suggests that S11 came the closest of our subjects to performing routine design.
6.5 Implications for Assessment of Expertise

We noted at the outset that one issue for purposes of assessment is whether both process and product characterizations are necessary indicators of expertise or are important in making expertise classifications. It seems clear to us that there is important information to be gained by looking at how people went about producing a final design. For example, having only looked at product we would not have known that subjects 9 and 10 used a primitive form of case-based reasoning and did not use functional decomposition. Likewise, subject 3’s performance was due to difficulties at the implementation level, while her performance at the function and transition levels was quite good. We would also not have known that subject 4 attempted to transfer his expertise in analog circuit design to the digital case. Furthermore, subject 11’s behavior suggests the importance of early error checking and evaluation at the functional level in obtaining a good design. The information gained from the process level descriptions provides a basis for comparisons with other domains in which expertise has been described. In addition, a comparison of the processes used by the most expert subject in our group with the processes of the less expert subjects provides a basis for instructional interventions. On the other hand, it is also clear that there are circumstances under which the product characteristics are sufficient for classification.

7 Ongoing Work on an Automated Design Tool

Assuming that one is interested in the richer characterization that results from considering both process and product in digital circuit design, it is important to make the collection of process data more tractable than through the analysis of think-aloud verbal protocols. Towards this end, we are building an automated design tool that can be used by subjects for purposes of assessment. The design tool has the following components:

1. problem-description browsing tool
2. high-level formalization and simulation tool
3. circuit-diagram drawing tool, and
4. circuit-simulator tool.

We are currently studying issues related to the high-level formalization and circuit-diagram drawing tool.

1. How natural is the high-level formalization tool for creating an intermediate representation of the design problem. More specifically, we are studying how easy it is for subjects to express their high-level functional conceptualizations in a flow-chart like form. A side issue we are trying to capture is how many levels of decomposition they produce at the function level.

2. Use of the circuit diagram drawing tool. We noticed that subjects went through a number of iterations when creating their implementation level designs. They added blocks to existing blocks, changed and refined blocks and often got rid of blocks and redesigned them. Sometimes they had to move them around. On paper most of these tasks required them to redraw figures, which is a nuisance. Also, but for the full protocols it would be hard to recognize what blocks were being deleted and replaced, or refined. However, the computer tool lets them save intermediate implementations for future reference and also to import past representations onto the current drawing area and modify them. This should make it much easier to capture and interpret the activities discussed above.

A preliminary version of this tool was evaluated within the group and with a few subjects during development. The general consensus was that this tool was too limiting in its ability as a CAD tool. Features in the high level functional tools and the circuit implementation tool were not extensive enough. It was determined that a framework for combining multiple design assistant tools was needed. Designers must be able to spend almost all of their energy on design, not on the tool’s workings.

We believe some important requirements for a usable assessment tool are:

- Designers must be able to guide the design process at all times. Currently, most tools center around complete synthesis of a circuit from a very formal user-specified design language such as a VHDL, or the tools force the designer
to specify all of the low-level design (just a drawing tool). Something between these two is necessary; a tool that can handle tedious details of synthesis but still allows the designer to intervene and force specific choices.

- Tools must be developed for working completely at the function design level. These tools need to handle flowchart creation or state machine creation by the user in a graphical format. A graphical format is necessary for the designer to more easily grasp what is being entered. Completely textual systems describe circuits such that they are difficult to understand.

- Simulators for functional tools are necessary that can directly simulate the graphical representations of flowcharts and state machines. This implies that the functional descriptions will have to be somewhat formal.

While these components are certainly the most important to us, the framework built for integrating these tools should be expandable so new tools can be added as needed.

8 Conclusions

This study has suggested a more complete and rich process model for capturing behavior in complex design. We have proposed characterizing CMOS circuit design behavior at different levels of abstraction — function, transition, and implementation levels. We believe that this provides a richer language for classifying designers into different levels of expertise. Empirical studies demonstrate considerable variance along these levels over subjects with differing design abilities.

We are currently working toward developing a computer-based tool for data gathering and assessment of design skills. Once this tool is complete, it will be important to study the reliability of computer-generated data in classifying design behavior as think-aloud protocols and paper-and-pencil problem solving. Lastly, we need to extend our study and perform a more systematic analysis of strategic and lower (i.e., unit operator) reasoning levels in order to derive a more formal model for assessment.
References


Appendix A

A small company selling networking supplies is experiencing problems with one of their IC suppliers and decides to design their own network controller. The type of network the company sells is a proprietary token ring architecture. The token ring controller will be manufactured as a CMOS chip by an external firm, using 2 micron technology. The network normally connects 5 to 100 computers, with each computer containing one network controller. You are asked to design the sender part of the controller. The final design should be a gate level design specification. You are free to use higher level modules like shift registers, just specify the implementation of one bit of the register at the gate level.

<Figure A1 about here>

The computers are connected in a ring topology. Each computer has a unique 7 bit address. A frame of data is of variable length and consists of the following elements:

1. an 8-bit token, indicating the head of a frame (1000 0000)
2. a 1-bit busy signal, indicating whether frame is full or empty (1 = full, 0 = empty)
3. a 7-bit address identifying the intended receiver
4. a 1-bit signal acknowledging reception by the receiver (1 = accepter, 0 = other)
5. a 7-bit address identifying the sender
6. an N-byte data packet

<Figure A2 about here>

The controller has the following lines:
When the controller receives the token, it checks the next bit to see if the following frame contains data. If it does: the controller does nothing (receiver part checks to see if data is intended for this system, and initiates reception). If the frame is empty the local system is allowed to send data if it has any available (DF = 0). In that case the next 7 bits are set to the address of the intended receiver. The following bit is set to 0 (acknowledge signal), and the next 7 bits are set to the address of the sender. Finally data transmission begins. When no more data is available (DF = 1) transmission stops. While the system is sending it keeps watching for the return of the token. When it receives a busy bit following return of the token (busy bit generated by itself) it HAS TO stop sending and reverse the busy bit to 0. Generation of sender and receiver addresses all happens externally on the Data In (DI) line. NOTE: You are responsible for two outputs (Ring Out and Fetch Data) derived from four inputs (Ring In, Data In, Data Finished, and Clock); see the figure on the previous page for explanations of these signals.

Additional Constraints:

- Minimum buffer size per controller: 8 bits
- Maximum buffer size per controller: 16 bits
- Speed of operation: 10 Mbps

Please do not worry about the following aspects:

- generation and synchronization of clock signals
- power down situations in any of the computers on the ring
• conflicts between data and marker bit patterns
• reception of packets
• loss of token
• no acknowledgement from receiver
• limited buffer capacity of the net
Figure A1: Ring Topology
Frame Format

<table>
<thead>
<tr>
<th>Data</th>
<th>Sender</th>
<th>Receiver</th>
<th>Token</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Acknowledged

Busy

Figure A2: Frame Format
Figure A3: Controller Lines

Controller

Clock → Ring In → Ring Out

Data In → Fetch Data → Data Finished

Controller Lines

File
Table 1: Function level evaluation.

<table>
<thead>
<tr>
<th>Number of Components</th>
<th>IF</th>
<th>Number in IF</th>
<th>Detail in IF</th>
<th>Global Plan</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>3</td>
<td>none</td>
<td>n/a</td>
<td>n/a</td>
<td>No</td>
</tr>
<tr>
<td>S2</td>
<td>3</td>
<td>Rough State Machine</td>
<td>7</td>
<td>Low</td>
<td>No</td>
</tr>
<tr>
<td>S3</td>
<td>6</td>
<td>Flowchart</td>
<td>6</td>
<td>High</td>
<td>No</td>
</tr>
<tr>
<td>S4</td>
<td>5</td>
<td>none</td>
<td>n/a</td>
<td>n/a</td>
<td>No</td>
</tr>
<tr>
<td>S5</td>
<td>3</td>
<td>Flowchart</td>
<td>3</td>
<td>Low</td>
<td>No</td>
</tr>
<tr>
<td>S6</td>
<td>6</td>
<td>Flowchart</td>
<td>4</td>
<td>Medium</td>
<td>No</td>
</tr>
<tr>
<td>S7</td>
<td>Data Flow Diagram</td>
<td>6</td>
<td>Medium</td>
<td>No</td>
<td>4</td>
</tr>
<tr>
<td>S8</td>
<td>Informal State Machine</td>
<td>5</td>
<td>Medium</td>
<td>Yes</td>
<td>3</td>
</tr>
<tr>
<td>S9</td>
<td>5</td>
<td>none</td>
<td>n/a</td>
<td>n/a</td>
<td>No</td>
</tr>
<tr>
<td>S10</td>
<td>7</td>
<td>none</td>
<td>n/a</td>
<td>n/a</td>
<td>Yes</td>
</tr>
<tr>
<td>S11</td>
<td>7</td>
<td>State Machine</td>
<td>6</td>
<td>High</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 2: Transition level evaluation.

<table>
<thead>
<tr>
<th>Core Component</th>
<th>Detail in Decomposition</th>
<th>Use of Signals</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>Switch</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>S2</td>
<td>Shift Register</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>S3</td>
<td>Abstract Controller</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td>S4</td>
<td>Shift Register</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>S5</td>
<td>System State Flop</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td>S6</td>
<td>Shift Register</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>S7</td>
<td>ROM Sequencer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S8</td>
<td>Shift Register</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>S9</td>
<td>Shift Register</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>S10</td>
<td>Shift Register</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>S11</td>
<td>High Level Blocks</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>
Table 3: Implementational Level

<table>
<thead>
<tr>
<th>Strategy</th>
<th>High Level Components</th>
<th>Custom Components</th>
<th>Interaction Checks</th>
<th>Problem Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>Iterative Addition</td>
<td>Yes</td>
<td>No</td>
<td>Local, Some Global</td>
</tr>
<tr>
<td>S2</td>
<td>Iterative Refine. &amp; Addition</td>
<td>Yes</td>
<td>Yes</td>
<td>Local (very little)</td>
</tr>
<tr>
<td>S3</td>
<td>n/a</td>
<td>Yes</td>
<td>Yes</td>
<td>none</td>
</tr>
<tr>
<td>S4</td>
<td>Iterative Refine. &amp; Addition</td>
<td>Yes</td>
<td>No</td>
<td>Local</td>
</tr>
<tr>
<td>S5</td>
<td>Iterative Refine. &amp; Addition</td>
<td>Yes</td>
<td>No</td>
<td>Global</td>
</tr>
<tr>
<td>S6</td>
<td>Iterative Addition</td>
<td>Yes</td>
<td>No</td>
<td>Local</td>
</tr>
<tr>
<td>S7</td>
<td>Complete Block Layout</td>
<td>Yes</td>
<td>No</td>
<td>none</td>
</tr>
<tr>
<td>S8</td>
<td>Global Layout; Iterative Refine.</td>
<td>Yes</td>
<td>Yes</td>
<td>Local w/ a Final Global</td>
</tr>
<tr>
<td>S9</td>
<td>Iterative Refine. &amp; Addition</td>
<td>Yes</td>
<td>Yes</td>
<td>Local</td>
</tr>
<tr>
<td>S10</td>
<td>Iterative Refine. &amp; Addition</td>
<td>Yes</td>
<td>Yes</td>
<td>Local</td>
</tr>
<tr>
<td>S11</td>
<td>Iterative Refine. on Signals/Blocks</td>
<td>Yes</td>
<td>Yes</td>
<td>At Block Level</td>
</tr>
</tbody>
</table>
Table 4: Evaluation

<table>
<thead>
<tr>
<th>Completeness</th>
<th>Flaw Detection</th>
<th>Overall Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1 Low</td>
<td>Low</td>
<td>1</td>
</tr>
<tr>
<td>S2 Low</td>
<td>Low</td>
<td>1</td>
</tr>
<tr>
<td>S3 Low</td>
<td>none</td>
<td>1</td>
</tr>
<tr>
<td>S4 Medium</td>
<td>Medium</td>
<td>2</td>
</tr>
<tr>
<td>S5 Medium</td>
<td>Medium</td>
<td>2</td>
</tr>
<tr>
<td>S6 High</td>
<td>Medium</td>
<td>3</td>
</tr>
<tr>
<td>S7 Medium</td>
<td>n/a</td>
<td>-</td>
</tr>
<tr>
<td>S8 Medium</td>
<td>High</td>
<td>3</td>
</tr>
<tr>
<td>S9 High</td>
<td>Medium</td>
<td>3</td>
</tr>
<tr>
<td>S10 High</td>
<td>Medium</td>
<td>3</td>
</tr>
<tr>
<td>S11 High</td>
<td>High</td>
<td>4</td>
</tr>
</tbody>
</table>
Figure 1: Flowchart form: Token Ring Controller
Distribution List

Dr. David L. Alderton, Code 131
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. Terry Allard
Code 342CS
Office of Naval Research
800 N. Quincy St.
Arlington, VA 22217-5660

Dr. John R. Anderson
Department of Psychology
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

Dr. Nancy S. Anderson
Department of Psychology
University of Maryland
College Park, MD 20742

Dr. Thomas H. Anderson
Center for the Study of Reading
174 Children's Research Center
51 Getty Drive
Champaign, IL 61820

Dr. Stephen J. Andriole, Chairman
College of Information Studies
Drexel University
Philadelphia, PA 19104

Prof. John Annett
University of Warwick
Department of Psychology
Coventry CV4 7AL
ENGLAND

Dr. Gregory Anrig
Educational Testing Service
Princeton, NJ 08541

Technical Director, ARI
5001 Eisenhower Avenue
Alexandria, VA 22333

Edward Atkins
13705 Lakewood Ct.
Rockville, MD 20850

Dr. Patricia Baggett
School of Education
610 E. University, Rm 1302D
University of Michigan
Ann Arbor, MI 48109-1259

Dr. William M. Bart
University of Minnesota
Dept. of Educ. Psychology
330 Burton Hall
178 Pillsbury Dr., S.E.
Minneapolis, MN 55455

Leo Beltracchi
United States Nuclear
Regulatory Commission
Washington DC 20555

Dr. William O. Berry
Director of Life and
Environmental Sciences
AFOSR/NL, N1, Bldg. 410
Bolling AFB, DC 20332-6448

Dr. Thomas G. Bever
Department of Psychology
University of Rochester
River Station
Rochester, NY 14627

Dr. Menucha Birenbaum
Educational Testing Service
Princeton, NJ 08541

Dr. Werner P. Birke
Personalamt der Bundeswehr
Kolnstrasse 262
D-5000 Koln 90
FEDERAL REPUBLIC OF GERMANY

Distribution List

Dr. John Black
Teachers College, Box 8
Columbia University
525 West 120th Street
New York, NY 10027

Dr. Deborah A. Boehm-Davis
Department of Psychology
George Mason University
4400 University Drive
Fairfax, VA 22030-4444

Dr. Kenneth R. Boff
AL/CFH
Wright-Patterson AFB
OH 45433-6573

Dr. J. C. Boudreaux
Manufacturing Engineering Lab
National Institute of
Standards and Technology
Gaithersburg, MD 20899

Dr. Richard L. Branch
HQ, USMEPCOM/MPECT
2500 Green Bay Road
North Chicago, IL 60064

Dr. Robert Breaux
Code 252
Naval Training Systems Center
Orlando, FL 32826-3224

Dr. Ann Brown
Graduate School of Education
University of California
EMST-4533 Tolman Hall
Berkeley, CA 94720

Dr. Pat Carpenter
Carnegie-Mellon University
Department of Psychology
Pittsburgh, PA 15213

CDR Robert Carter
Naval Medical Research
and Development Command
National Naval Medical Ctr
Bldg. 1-T-11
Baltimore, MD 20889-5606

Dr. Eduardo Casacallar
Educational Testing Service
Rosedale Road
Princeton, NJ 08541

Dr. Davida Charnay
English Department
Penn State University
University Park, PA 16802

Dr. Micheleene Chi
Learning R & D Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15260

Dr. Susan Chipman
Cognitive Science Program
Office of Naval Research
800 N. Quincy St., Code 342CS
Arlington, VA 22217-5660

Dr. Raymond E. Christal
U.S. LAMP Science Advisor
AL/HRMI
Brooks AFB, TX 78235

Dr. Deborah Claman
National Institute for Aging
Bldg. 31, Room 5C-35
9000 Rockville Pike
Bethesda, MD 20892

Dr. William J. Clancy
Institute for Research on Learning
2550 Hanover Street
Palo Alto, CA 94304
Distribution List

Chief of Naval Education and Training (N-5)
NAS Pensacola, FL 32508

Dr. Paul Cobb
104C Mayborn Building
Box 330 Peabody College
Vanderbilt University
Nashville, TN 37203

Dr. Rodney Cocking
NIMH, Basic Behavior and Cognitive Science Research
5600 Fishers Lane, Rm 11C-10
Parklawn Building
Rockville, MD 20857

Director, Life Sciences
Office of Naval Research
Code 114
Arlington, VA 22217-5000

Director, Cognitive and Neural Sciences, Code 1142
Office of Naval Research
Arlington, VA 22217-5000

Director
Training Systems Department
Code 15A
Navy Personnel R&D Center
San Diego, CA 92152-6800

Library, Code 231
Navy Personnel R&D Center
San Diego, CA 92152-5800

Dr. Magda Colberg
Office of Personnel Management
1900 E Street, N.W.
Washington, DC 20415-0001

Commanding Officer
Naval Research Laboratory
Code 4827
Washington, DC 20375-5000

Dr. Albert T. Corbett
Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

Dr. Michael Cowen
Code 142
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. Kenneth B. Cross
Anacapa Sciences, Inc.
P.O. Box 519
Santa Barbara, CA 93102

CTB/McGraw-Hill Library
2500 Garden Road
Monterey, CA 93940-5380

Dr. Charles E. Davis
Educational Testing Service
Mail Stop 22-T
Princeton, NJ 08541

Dr. Georgi Delacorte
Exploratorium
3601 Lyon Street
San Francisco, CA 94123

Dr. Sharon Derry
Florida State University
Department of Psychology
Tallahassee, FL 32306

Dr. Stephanie Doane
University of Illinois
Department of Psychology
605 East Daniel Street
Champaign, IL 61820

Dr. Michael Drinnings
Basic Research Office
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

Defense Technical Information Center
DTIC/DDA-2
Cameron Station, Bldg 5
Alexandria, VA 22314
(4 Copies)

Mr. David Dubois
Personnel Decisions Research
Institutes
43 Main Street, SE
Riverplace, Suite 405
Minneapolis, MN 55414

Dr. Richard Duran
Graduate School of Education
University of California
Santa Barbara, CA 93106

Dr. Nancy Eldridge
College of Education
Division of Special Education
The University of Arizona
Tucson, AZ 85721

Dr. John Ellis
Navy Personnel R&D Center
Code 15
San Diego, CA 92152-6800

Dr. Susan Embretson
University of Kansas
Psychology Department
426 Fraser
Lawrence, KS 66045

Dr. Susan Epstein
144 S. Mountain Avenue
Monclair, NJ 07042

ERIC Facility-Acquisitions
1301 Piccard Drive, Suite 300
Rockville, MD 20850-4305

Dr. Martha Evens
Dept. of Computer Science
Illinois Institute of Technology
10 West 31st Street
Chicago, IL 60616

Dr. Lorraine E. Eyde
US Office of Personnel Management
Office of Personnel Research and
Development
1900 E St., NW
Washington, DC 20415

Dr. Franco Faina
Direttore Generale LEVADIFE
Piazzele K. Adenauer, 3
00144 ROMA EUR
ITALY

Dr. Jean-Claude Falmagne
Irvin Research Unit in Mathematical & Behavioral Sciences
University of California
Irvine, CA 92717

Dr. Beatrice J. Farr
Army Research Institute
PERI-IC
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Marshall J. Farr
Farr-Sight Co.
2520 North Vernon Street
Arlington, VA 22207

Dr. Pat-Anthony Frederico
Code 13, NPRDC
53335 Bryne Road
San Diego, CA 92152-7250

Dr. J. D. Fletcher
Institute for Defense Analyses
1801 N. Beauregard St.
Alexandria, VA 22311
<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
<th>City, State</th>
<th>Zip Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr. Charles M. Reigeluth</td>
<td>Chairman, Instructional Systems Technology</td>
<td>Bloomington, IN</td>
<td>47405</td>
</tr>
<tr>
<td>Dr. Daniel Reisberg</td>
<td>Reed College, Department of Psychology</td>
<td>Portland, OR</td>
<td>97202</td>
</tr>
<tr>
<td>Dr. Brian Reiser</td>
<td>Institute for the Learning Sciences</td>
<td>Evanston, IL</td>
<td>60201-3142</td>
</tr>
<tr>
<td>Dr. Lauren Resnick</td>
<td>Learning R &amp; D Center, University of Pittsburgh</td>
<td>Pittsburgh, PA</td>
<td>15213</td>
</tr>
<tr>
<td>Dr. Gilbert Ricard</td>
<td>Mail Stop K01-14, Grumman Aircraft Systems</td>
<td>Bethpage, NY</td>
<td>11714</td>
</tr>
<tr>
<td>Dr. W. A. Rizzo</td>
<td>Head, Human Factors Division</td>
<td>San Diego, CA</td>
<td>92152-6800</td>
</tr>
<tr>
<td>Mr. Louis Roussos</td>
<td>University of Illinois, Department of Statistics</td>
<td>Champaign, IL</td>
<td>61820</td>
</tr>
<tr>
<td>Dr. Eduardo Salas</td>
<td>Human Factors Division (Code 262)</td>
<td>Orlando, FL</td>
<td>32826-3224</td>
</tr>
<tr>
<td>Dr. Fumiko Samejima</td>
<td>Department of Psychology, University of Tennessee</td>
<td>Knoxville, TN</td>
<td>37966-0900</td>
</tr>
<tr>
<td>Mr. Drew Sanda</td>
<td>N/P/R/D Code 62</td>
<td>San Diego, CA</td>
<td>92152-6800</td>
</tr>
<tr>
<td>Dr. Capitan Jesus Bernal Santos</td>
<td>Ministerio de Defensa, Unidad de Psicologia (SEGENTE)</td>
<td>28071, Madrid</td>
<td>ES/PANA</td>
</tr>
<tr>
<td>Dr. Daniel L. Schacter</td>
<td>Department of Psychology, Harvard University</td>
<td>Cambridge, MA</td>
<td>02138</td>
</tr>
<tr>
<td>Dr. Walter Schneider</td>
<td>Learning R&amp;D Center, University of Pittsburgh</td>
<td>Pittsburgh, PA</td>
<td>15260</td>
</tr>
<tr>
<td>Dr. Mary Schratz</td>
<td>4100 Parkside, Carlsbad, CA 92008</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dr. Myrna F. Schwartz</td>
<td>Director, Neuropsychology Research Lab</td>
<td>Philadelphia, PA</td>
<td>19141</td>
</tr>
<tr>
<td>Dr. Robert J. Seidel</td>
<td>US Army Research Institute</td>
<td>Alexandria, VA</td>
<td>22333</td>
</tr>
<tr>
<td>Dr. Terrence J. Sejnowski</td>
<td>Professor, The Salik Institute</td>
<td>San Diego, CA</td>
<td>92138-9216</td>
</tr>
<tr>
<td>Dr. Michael G. Shafro</td>
<td>NASA Ames Research Ctr., Mall Stop 262-1</td>
<td>Moffett Field, CA</td>
<td>94035-1000</td>
</tr>
<tr>
<td>Dr. Valerie L. Shalin</td>
<td>Department of Ind. Engineering State University of New York</td>
<td>Buffalo, NY</td>
<td>14260</td>
</tr>
<tr>
<td>Dr. Richard J. Shavelson</td>
<td>Graduate School of Education, University of California</td>
<td>Santa Barbara, CA</td>
<td>93106</td>
</tr>
<tr>
<td>Dr. Tracy Shors</td>
<td>Dept. of Psychology, Princeton Univ.</td>
<td>Princeton, NJ</td>
<td>08544</td>
</tr>
<tr>
<td>Dr. Randall Shumaker</td>
<td>Naval Research Laboratory</td>
<td>Washington, DC</td>
<td>20375-5000</td>
</tr>
<tr>
<td>Dr. Edward Silver</td>
<td>LRDC</td>
<td>Pittsburgh, PA</td>
<td>15260</td>
</tr>
<tr>
<td>Dr. Zita M. Smutis</td>
<td>Director, Manpower &amp; Personnel Research Laboratory</td>
<td>Alexandria, VA</td>
<td>22333-5600</td>
</tr>
<tr>
<td>Dr. Jerome E. Singer</td>
<td>Department of Medical Psychology</td>
<td>Bethesda, MD</td>
<td>20814-4799</td>
</tr>
<tr>
<td>Dr. Jan Sinnott</td>
<td>Department of Computer Science</td>
<td>Towson, MD</td>
<td>21204</td>
</tr>
<tr>
<td>Dr. Derek Sleeman</td>
<td>Computing Science Department</td>
<td>Aberdeen, ABY 2FX</td>
<td>Scotland</td>
</tr>
<tr>
<td>Dr. Robert Smillie</td>
<td>Naval Ocean Systems Center</td>
<td>San Diego, CA</td>
<td>92152-5000</td>
</tr>
</tbody>
</table>
Distribution List

Dr. Richard E. Snow
School of Education
Stanford University
Stanford, CA 94305

Dr. James J. Staszewski
Dept. of Psychology
University of South Carolina
Columbia, SC 29210

Dr. Bruce D. Steinberg
Curry College
Milton, MA 02186

Dr. Frederick Steinheiser
CIA-ORD
Aames Building
Washington, DC 20505

Dr. Ted Steinke
Dept. of Geography
University of South Carolina
Columbia, SC 29208

Dr. Kurt Steuck
AL/HRTI
Brooks AFB
San Antonio, TX 78235-5601

Dr. Michael J. Tarr
Dept. of Psychology
Yale University
PO Box 11A, Yale Station
New Haven, CT 06520

Dr. Kikumi Tatsuoka
Educational Testing Service
Mail Stop 03-T
Princeton, NJ 08541

Dr. Gary Thomasson
Defense Manpower Data Center
99 Pacific Street
Suite 155A
Monterey, CA 93940

Dr. Sharon Tkacz
CAE-Link Corporation
209 Madison Street
Alexandria, VA 22314

Dr. Douglas Towne
Behavioral Technology Labs
University of Southern California
1228 Spring Street
St. Helena, CA 94574

Dean, College of Behavioral and Social Sciences
University of Maryland,
Baltimore County
Baltimore, MD 21228

Chair, Department of Psychology
University of Maryland,
Baltimore County
Baltimore, MD 21228

Dr. Kurt VanLehn
Learning Research & Development Ctr.
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15260

Dr. Frank L. Vicino
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. Jerry Vogt
Department of Psychology
St. Norbert College
De Pere, WI 54115-2099

Dr. Jacques Voneche
University of Geneva
Department of Psychology
Geneva
SWITZERLAND 1204

Dr. Douglas Wetzel
Code 15
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. Barbara White
School of Education
Tolman Hall, EMST
University of California
Berkeley, CA 94720

Alexandra K. Wigdor
NRC MH-176
2101 Constitution Ave.
Washington, DC 20418

Dr. David Wiley
School of Education and Social Policy
Northwestern University
Evanston, IL 60208

Dr. David C. Wilkins
University of Illinois
Department of Computer Science
405 North Mathews Avenue
Urbana, IL 61801

Dr. Bruce Williams
Department of Educational Psychology
University of Illinois
Urbana, IL 61801

Dr. Kent E. Williams
Virginia Polytechnic Institute and State University
Department of Industrial and Systems Engineering/MS.
1900 Kraft Drive
Blacksburg, VA 24060

Dr. Mark Wilson
Graduate School of Education
University of California, Berkeley
Berkeley, CA 94720

Dr. Robert A. Wisher
U.S. Army Institute for the Behavioral and Social Sciences
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

Dr. Merlin C. Wittrock
Graduate School of Education
Univ. of Calif., Los Angeles
Los Angeles, CA 90024

Dr. Kentaro Yamamoto
03-0T
Educational Testing Service
Rosedale Road
Princeton, NJ 08541

Dr. Masoud Yazdani
Dept. of Computer Science
University of Exeter
Prince of Wales Road
Exeter EX4 4PT
ENGLAND

Frank R. Yekovich
Dept. of Education
Catholic University
Washington, DC 20064

Dr. Joseph L. Young
National Science Foundation
1800 G Street, N.W., Room 320
Washington, DC 20550

Mr. Anthony R. Zara
National Council of State Boards of Nursing, Inc.
676 N. St. Clair, Suite 550
Chicago, IL 60611

Prof. Gerard de Zeeuw
Center for Innovation and Cooperative Technology
Grote Bickersstraat 72
1013 KS Amsterdam
The NETHERLANDS