Carnegie Mellon University
Research Showcase

Department of Electrical and Computer Engineering

Carnegie Institute of Technology

1-1-1979

Measuring designer performance to verify design automation systems

Donald E. Thomas
Carnegie Mellon University

Daniel P. Siewiorek

Follow this and additional works at: http://repository.cmu.edu/ece

Recommended Citation
Thomas, Donald E. and Siewiorek, Daniel P., "Measuring designer performance to verify design automation systems" (1979). *Department of Electrical and Computer Engineering.* Paper 87.
http://repository.cmu.edu/ece/87

This Technical Report is brought to you for free and open access by the Carnegie Institute of Technology at Research Showcase. It has been accepted for inclusion in Department of Electrical and Computer Engineering by an authorized administrator of Research Showcase. For more information, please contact research-showcase@andrew.cmu.edu.

NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:
The copyright law of the United States (title 17, U.S. Code) governs the making of photocopies or other reproductions of copyrighted material. Any copying of this document without permission of its author may be prohibited by law.

MEASURING DESIGNER PERFORMANCE TO VERIFY DESIGN AUTOMATION SYSTEMS

by

^D.E. Thomas & D.P. Siewiorek

DRC-18- 9-79

January 1979

Departments of Electrical Engineering and Computer Science

620.0042
C28d
DRC-18-9-79

Abstract

Design automation at the register transfer level of design is still in its infancy and it is not yet completely understood what the appropriate measures used in directing the automated design process should be. To establish these measures, results of these design automation systems must be compared with some near optimal designs. A set of statistically based experiments is developed to estimate near optimal designs. A method is demonstrated for gathering data on designer performance, specifically at the different levels of systems design, and in general, for calibration of design automation systems where the intuitive designer still performs more capably than the present design algorithms. An analysis of variance is used to indicate the relative importance of various decisions in a system design. It is shown that the algorithm to be implemented and the hardware design style account for 90% of the variation in the results. Thus selecting the design style (e.g., distributed, microprocessor, pipelined, etc.) is the most important parameter for a design automation system.

Keywords: Design automation, register transfer level design, design of experiments, analysis of variance, design styles

1. INTRODUCTION

Computer aided design systems have reduced the time and cost of the design process by making a computer perform many of the routine and often mundane tasks of logic design. Historically, such systems were limited to bookkeeping and consistency checking. Occasionally optimization techniques for combinatorial logic (e.g., Quine-McCluskey [1]) were incorporated into these systems. But with the advent of MSI/LSI chips and the demand for more complex systems, conceptual design moved from the logic design level to the register transfer (RT) level [2].

Recent research has been aimed at the generation of "near optimal[1]' register transfer level designs from a description of the algorithm to be implemented [3]. Design automation of this level of design is still in its infancy and it is not yet known what the appropriate measures used in directing the automated design process should be. To establish these measures, results of these design automation systems must be compared with some "near optimal" designs. In some cases, analytic bounds of the optimal solution can be shown to exist. However, when the design task becomes too complex there may be no other method to evaluate the automated design algorithms than by statistical experimentation. This paper discusses the design of a statistically based experiment to evaluate such design algorithms.

The digital system design process is a long chain of decisions starting from an abstract systems level considering various alternative implementations, down to specifying the fanout capability of each transistor at the circuit level in an LSI design (Figure 1) . At each stage of the process, the designer must

---

This work was supported by the National Science Foundation under Grant GJ-32758X.

UNIVERSITY LIBRARIES
CARNEGIE-MELLON UNIVERSITY
PITTSBURGH. PENNSYLVANIA 15213

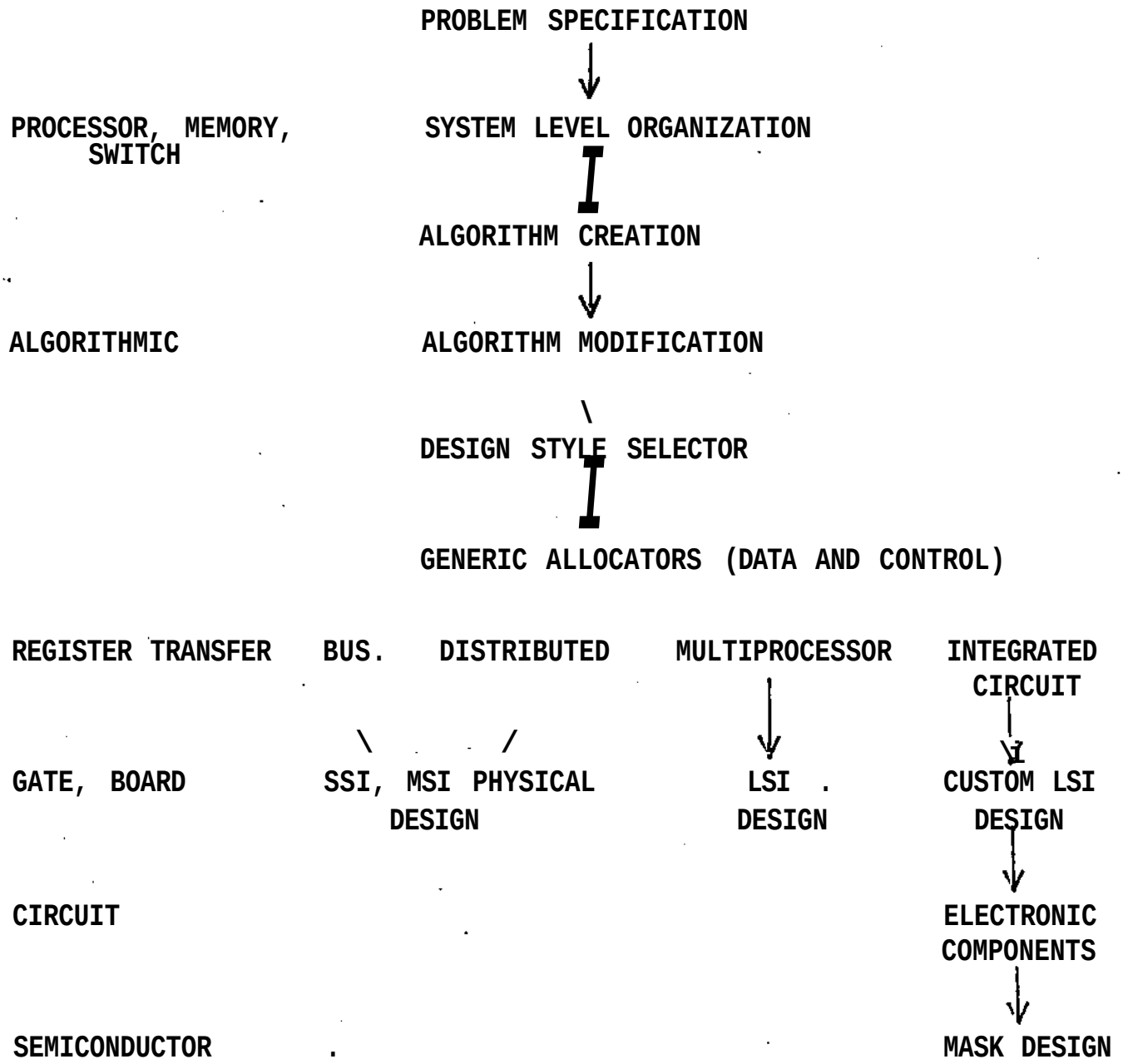

Figure 1. The Levels of Design.

consider the overall algorithm structure, the constraints to be put on the final implementation, and the modules (be they SSI chips or standard cells) that can be used to produce the implementation. Each of these considerations suggest trade offs that may be made at the different levels of the design process.

A design system has been produced which will explore serial-parallel trade offs at the RT level of design [4]. The alternative implementations can be evaluated as per the designer's constraints. Using these evaluations as dimensions, a multi-dimensional output design space for the system to be designed can be conceived. The role of a high level design system is to prune this space so that only a few alternatives need be examined.

A designer goes through the same process of pruning alternative implementations that his intuition/experience tells him are unfruitful. Since every designer has a different background, different designers will make different trade offs at the various levels of system design, thus producing different resultant implementations. Thus each implementation can be treated as a statistical observation. This paper describes an experiment designed to observe a group of digital systems designers. The results of the experiment are used to gain insight into the design process and to verify the results of an RT level Computer Aided Design (RTCAD) system being developed at *QSB* [5].

Briefly, each experiment consisted of confronting several designers with the task of implementing several digital systems described in a hardware descriptive language (ISPS [6]) using several design styles, (microprocessors, TTL chips, etc.). The designers made three implementations, each using a different description and style so that individual designers never used an ISP description or design style more than once. An analysis of variance

(ANOVA) using designers, descriptions and styles as major factors is used *to* analyze the results.

The organization of the RTCAD system is explained in Section 2 and two experiments are outlined in .Section 3 that gather data on designers performances in digital design situations. Section 4 describes the experimental methodology and approach to the analysis. Section 5 discusses the factors considered in the design of the experiment. Sections 6 and 7 analyze the results of the experiment. Conclusions of these sections indicate that similar experimental methodology can be used for other design situations such. as component placement in wire routing systems where designers still outperform automated systems. By the appropriate selection of factors to be modelled, the importance of various decisions within the design systems can be gauged.

2. MODEL OF THE DESIC2* PROCESS

The CMD RTCAD system is an outgrowth of an earlier design system [3] and is organized as shown in Figure 2. The goal of the system is to produce a logic design given a behavioral description of the digital system to be designed. The boxes represent components of the system. The solid lines indicate the path taken by a behavioral description as it is transformed into a physical Implementation. The dashed lines represent the flow of other design information. This section will describe the basic RTCAD system and present the idea of a design style upon which the system is partially based.

RTCAD System

There are three inputs to the design system:

1) ISPS parse tree - This is the behavioral description of the digital system to be designed translated by the ISPS compiler into a form that can be readily scanned by the design system.

2) Optimization criteria - These are parameters for the objective function to be optimized by the design system. They weight the cost, speed and other measures of the final implementation.

3) Module set library - This describes the physical modules available to the module binder for fabricating systems.

The design process starts by abstracting a minimum kernal of information which describes the behavior of the system to be designed but makes as few assumptions about its hardware realization as possible. This is a major departure from earlier design systems which derive much of their structural design information from the user specification. As the design moves through

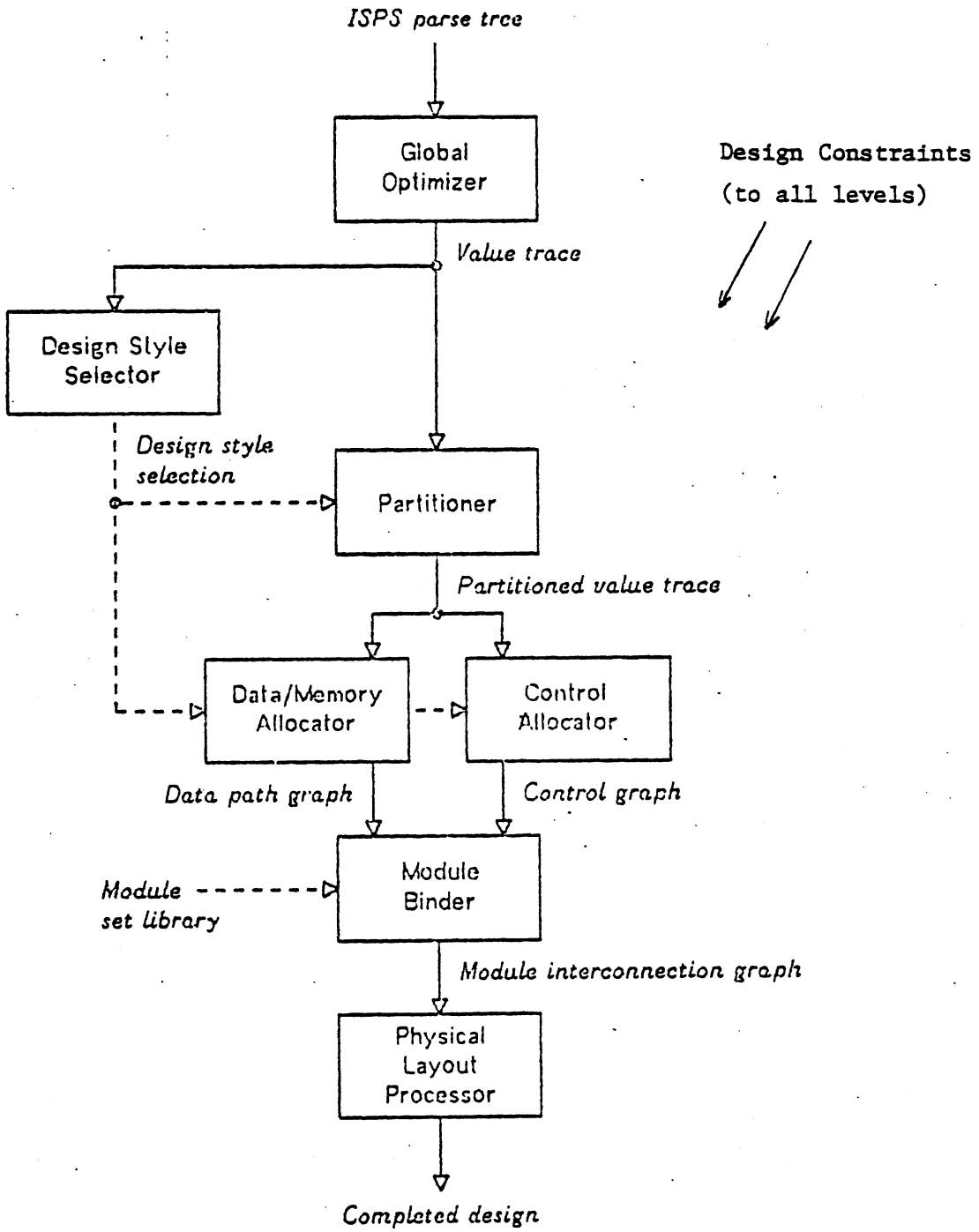

Figure 2. The CMU RTCAD System.

the system, each design system component binds certain implementation decisions appropriate to its level. At the end of the process a complete hardware description emerges. One novel feature allowed by this independence from structural decisions is the use of a variety of different hardware design philosophies and module sets instead of a single fixed allocation scheme.

As shown in Figure 2, an ISPS description is supplied to the global optimizer algorithm which can make high level transformations on the control structure of the ISP description. Example transformations are: increasing/decreasing parallelism, winding/unwinding control loops, and pipelining/unpipelining the control structure. The style selector, which encompasses abstract knowledge about the different approaches (styles) to digital systems design (e.g., microprocessors, pipelining) supplies high level trade offs between design styles to the partitioner and eventually selects the design style which will best fulfill the designer's constraints. The resulting transformed descriptions are passed along to a physical allocator [7] which will first make all of the allocations of the abstract data operators, data paths, and memories, and then map actual physical components onto them. The physical evaluator evaluates the design and supplies the results to the global optimizer algorithms.

The physical allocator program includes the algorithms and heuristics to implement a design using the design styles' module sets. The allocator first makes all of the allocations of the abstract data operators, data paths and memories. Then the actual modules which are needed to perform a register transfer have to be selected from a set of module templates, several of which may contain alternative choices. A module template is an internal data base representation of an actual hardware module. The concept of templates allows new modules to be added without changing the design programs.

In this manner the design process may be kept relative to advances in the technology. Board layout and wire routing as well as cell layout and channel routing programs can be run from the output of these allocators.

Design Styles

The decomposition of the design process shown in Figure 2 is based on the idea that at each stage (level) of the design process, there are certain trade-offs and design decisions that can be made; each lower level binding more detailed information.

A design style is an abstraction of a class of hardware module sets, each of the sets within the style being more or less the. same. High level design can be performed with respect to the design rules, intermodule protocol, and design trade-offs that apply to all the module sets of a design style. The existence of definitive design styles have been observed [8], [9] and are dictated by the following:

1) Hardware Modules. The type of hardware building blocks exhibit different characteristics in terms of cost, speed, power, and level of functionality. Example module sets include: microprocessor chip sets, TTL chip sets, and Register Transfer Modules (RTM's) [10].

2) Design Constraints. Structural and physical constraints (e.g., cost, speed, size, etc.) may imply or rule out design styles. For example a requirement for high speed may dictate parallel or pipeline computation whereas low cost may dictate microprocessor or serial computation.

3) Algorithm Structure. Properties of the algorithm may specify a design style. An algorithm with a high degree of parallelism may

not favor a microprocessor implementation while an algorithm with either a large number of variables or a high degree of interconnectivity will.

The next section describes two experiments designed to gather data on the design process with which to verify this approach to design automation.

3. THE EXPERIMENTS

Two experiments were conceived to use digital hardware designers to gather data at both the physical allocator level and the higher, more abstract global optimizer level of the RTCAD system. The experiments discussed here are aimed at experimentally justifying the decomposition of the RTCAD system into levels of design and gauging the impact of the design decisions at the design style selector and physical allocator levels of design. The following provides a description of the tasks the designers were to perform at each of these levels of design.

3.1 Experiment I

Physical allocator level. This measures a designer's performance in implementing a specific functional hardware description using a specific design style and module set. Some of the design decisions at this level were: memory allocation (whether to allow a variable to reside in different physical registers at different points in the algorithm); operator allocation (how many and what type of data operator to use); etc. All designers were provided with an identical goal for their design.

3.2 Experiment II

Global optimizer level. The task here is to transform the functional hardware description and make the best implementation that can be realized in a specific design style. At this level, transformations such as parallel vs. serial implementation; extraction or insertion of loop counters; and subrouting vs. inline implementation may all be considered. Of course these must all be considered in light of the design style since the transformations affect each style and its final implementation differently.

, A group of designers was asked to design various described objects using certain design styles* Due to the complexity of the problem, none of the implementations would be identical and a statistical variance could be measured. In both of the experiments, the variance observed between the designers was a measure of the difficulty of the actual implementation, of the differing amounts of creativeness possessed by the designers, and of the designers[T] performance under complicated design situations. The overall range among designers performing similar design tasks provides a range in which to expect other designer's results to fall. The output of the RTCAD system can then be compared to the designer's results.

4. EXPERIMENTAL METHOD

A main issue in designing an experiment to measure the quality of the product of an automated design system is the cost of the man-power. A number of designers must be used so as not to bias the results by one designer's expertise (or lack thereof). In addition, various factors representing different aspects of the design process may need to be considered in the experiment. These factors are expected major causes of variation in the design process and may have different qualitative levels representing the different values a factor may take on (e.g., each individual style is a level within the total style factor). This section discusses the statistical methodology involved in designing such an experiment.

If several factors or aspects of the design process are to be considered, a traditional approach is to iteratively hold all except one of the factors constant, noting the related change as the selected factor varies. However, a factorial experiment, one in which all levels of a given factor are combined with all levels of the other factors, has two main advantages:

- All the data can be used in computing all of the effects, and,
- Information on possible interaction between the factors can be observed.

In the experiments described here, two of the three factors are designers and design styles. An interaction between these factors would occur if a specific designer was above average (of all designers) in one style but below average in another. In general, there is an interaction between two factors if a change in one factor produces a different (and possibly opposite) response at the different levels of the second factor. This can be visualized in a mythical two factor experiment as shown in Figure 3. The figure shows responses of factor A as a function of factor B. If there are no interactions between the factors (solid line) the lines would be nearly parallel. The

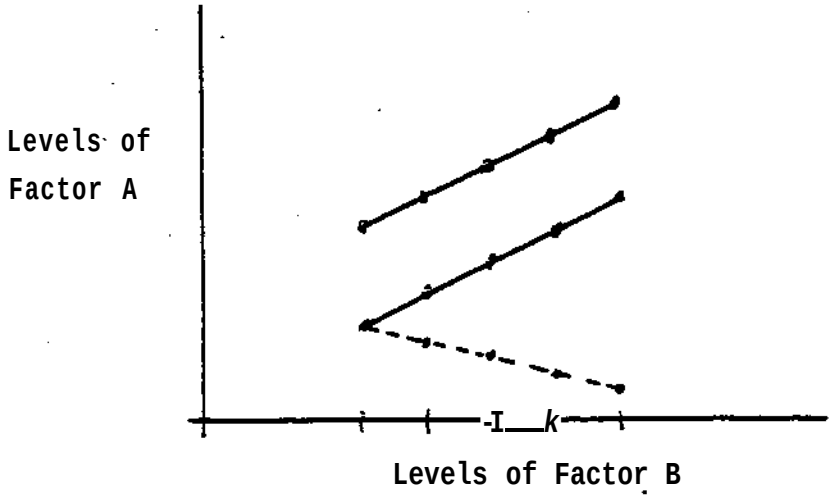

Figure 3. Interaction Between Factors of an Experiment,

dashed line indicates interaction. Thus the factorial experiment can provide interesting and useful data on related issues.

It can be seen though that as the number of factors and the number of levels within a factor grow (each design style is a different level within the style factor, above) so does the nunber of observations to produce a full design. In some cases, it is not always economical to run a full experiment. Nearly as much information can be derived from a fractional factorial experiment. However, since only a fraction of the observations are going to be made, only a fraction of the major and interaction effects can be calculated. The others will not be totally distinguishable from each other. This is because certain observations, which could have been used to make all of the effects distinguishable, have been eliminated. In this case, both effects will contribute to the same value but will be indistinguishable or aliased. Aliases must be identified so that the experiment can be run to provide the needed unaliased data. The references [11],[12] discuss aliasing.

An analysis of variance (ANOVA) [11] will be calculated for each of the experiments outlined in Section 3. The rationale behind the ANOVA is that the total sum of squares can be broken down into the sum of: squares between the means of each factor, the means of each interaction, and an error sum of squares. The experiments described here can be pictured generally in Figure 4. The columns represent the levels of the style factor, the rows represent the levels of the description factor and the letters in each cell (row, column intersection) represent the levels of the designer factor. (Each letter representing a group of three designers.) The model for such an experiment is

$$Y_{ijk} + \mu + A_i + B_j + AB_{ij} + C_k + AC_{ik} + BC_{jk} + ABC_{ijk} + \varepsilon_{ijk} \qquad (1)$$

	BUS	DIST	uP
Video	A	C	B
Elevator	C	•B	A
Change	B	A	C

Figure 4. Assignment For Designer Groups in Experiment I

	BUS	DIST	uP
Video	B	A	C
Elevator	A	C	B

Figure 5. Assignment For Designer Groups in Experiment II

The single letter terms (e.g., A) represent factors and the double and triple terms represent the interaction terms. The subscripts represent the level of the factor being represented, μ represents the grand mean of all the data and ε_{ijk} is a normally and independent distributed error term with mean μ and variance σ^2. Each experiment has three observations per cell and is a one-third fractional factorial design. Thus in the case of each experiment, certain factors or interaction factors will be aliased. These will be identified in later sections.

The variance due to the factors and the interaction factors can be calculated as follows. Define x_A to be a coding of the level of factor A (say the style factor). In this case x_A could take on three values: distributed style, bus style, or microprocessor style. These are numerically coded, e.g., $x_A = 1$ indicates the distributed style. Then the deviation about the mean due to a level m of a factor is:

$$(A)_m = \frac{1}{n} \Sigma Y_{ijk} - \mu, \quad \text{for all j where } x_A = m . \tag{2}$$

That is, the deviation about the mean μ produced by level m of factor A (say the bus level of the style factor) is the average of all n data points in Y where style is at level m. Deviations, about the mean, of interaction terms are similarly defined:

$$(AB)_m = \frac{1}{n} \Sigma Y_{ijk} - \mu, \quad \text{for all j where } x_A + x_B = m , \tag{3}$$

where the deviation is due to those levels of factors A and B which when codings are summed (mod 3 because of the three levels of each term) equal m. The only difference between these deviations are the terms of Y selected for summing. The sum of squares associated with factor (say A) is the result of equation (2) squared and summed over all levels of factor A;

$$n \underset{O}{Z} (A)^2_o$$

and the sum of squares due to an interaction term (say Afi) would be:

$$n \underset{m}{Z} (AB)^2_m$$

In the general case, regression analysis can be used to fit a multi-dimensional surface to the data points 7. The dimensions of this surface correspond to the certain major factors and interaction factors being modelled.

Regression analysis is used to find the parameters b in equation (1), which is rewritten here in matrix form, by the method of least squares.

$$\underline{Y} = \underline{Xb} + \underline{\varepsilon} \quad . \tag{4}$$

]T is the vector of observations (the costs of the designs produced by the designers) and $\underline{X}$ is the matrix of independent variables indicating which observations correspond to which style, description and designer, £ is a vector of errors.

It is possible that a curvilinear regression using a second degree (or higher) model of the form

$$Y - b_Q + b_x X_j + b \wedge^2 + \ldots \tag{5}$$

may be needed* Of interest to the experiment designer is finding the smallest degree of polynomial in X which can best fit the data. Equation (5) can be rewritten as:

$$Y \text{ " } C t_0 + a_i Z_l + a_K Z_K + \ldots \tag{6}$$

where the Z^fs are functions of X and the a^fs are the corresponding coefficients. The Z^fs are chosen to be orthogonal polynomials. An advantage of

writing the model in this form- is that higher order polynomials which are orthogonal to-(and independent of) the ones already considered may later be included [13].

The least squares method is used and the sum of squares of the deviation of each Y_i from its predicted Y_i is minimized by differentiating and setting equal to 0. The normal equations of regression analysis [14] can then be written in matrix form as:

$$\underline{X}'\underline{Xb} = \underline{X}'\underline{Y} \tag{7}$$

(where X^1 is the transpose of matrix X) and thus b is found to be:

$$\underline{b}. - \underline{(X'}xr^1\underline{X}^f\underline{Y} \ . \tag{8}$$

Using these calculated b's, the original data, can be observed in terms of the fit of each data point by calculating the residual (or the error e) of each data point.

$$£ - \underline{res} \ « X \bullet \underline{*b}. \tag{9}$$

The residuals can be used to note data points which are not close to the fitted surface. Large residuals may indicate the need for higher order approximation. Jj can also be used to calculate the correlation coefficient as follows. The total sum of squares can be found by

$$Y^fY \ - n * Z_i \ (YJ^2_x \ . \tag{10}$$

This can be apportioned to two sources: 1) the sum of squares about the mean which will later be apportioned to the factors being modelled in the ANOVA, and 2) the residual sum of squares Y^fY - b'X'Y which is the variance

due to effects not considered in this regression model. The value R^2 is the correlation coefficient defined by:

$$R^2 = 1 - \frac{Y^T Y - b^T X^T Y}{Y^T Y - \frac{(\sum_{i=1}^{n} Y_i)^2}{n}} \qquad (11)$$

and reflects that portion of the variance which the model determines. It is also a measure of the fit of the data to the multidimensional surface.

The elements of the row vector $\underline{b}^T \underline{X}/\underline{Y}$ are, in the general case, the variances due to each of the modelled effects. However, it must be remembered that in a partial factorial design all of the elements may not be independent of each other as aliasing may occur.

The described statistical methods will be used in the analysis of two experiments performed on the RTCAD system. Finally the results will be analyzed.

5. DESIGN OF THE EXPERIMENT

An issue in designing this type of experiment is the learning experience that a designer undergoes in the process of creating a design. If, somehow, a designer's memory of a previous design could be completely erased, then he could be expected to complete a set of designs where each design's result is independent of the others. However, a human designer will learn from the early designs and offset, for the better, the results of the latter. In this section, an experiment will be designed using the previously discussed methods to observe a group of digital systems designers.

5.1 Factors

The experiment was designed around three major factors: designers, design styles, and descriptions. The act of a designer implementing a description using a design style provided an observation for statistical analysis. It was imperative that each observation be statistically independent of the other observations. As described previously, between any two of these factors there can be interactions which can affect the independence of any observation made. There is an interaction say, between styles and designers if in one style designer 1 does better than designer 2 but just the reverse for another style. For instance, if a designer implements more than one description in any single design style, the second implementation would not be an independent observation because he would be more familiar with the style. Therefore the designers in the experiment will be restricted to using each design style only once.

A similar situation arises between a designer and a description if a description is used more than once by the same designer. Interactions between the design style and the description are possible if the styles and descriptions do not both represent a broad range of objects. Otherwise a set of unbalanced descriptions might be trivial in one design style rather

than challenging to the designer. To increase the independence of the observation, these interactions were minimized by providing a range of styles and descriptions for implementation. The next several subsections expand upon the problems of interactions and what has been done to minimize them.

5.2 Algorithmic Features of the Descriptions

Several ISP descriptions of varying sizes were considered for the two experiments. To guarantee a diversity of descriptions, a range of algorithmic features were defined. Three descriptions were finally selected which represented a range of these algorithmic features. The features considered were:

I/O vs. Internal calculations. Some of the descriptions should represent that class of object which mainly does arithmetic calculations, only rarely stopping to present its results to the outside world. The others should have a greater amount of interaction, probably through the manipulation of flags, with the outside world.

Low level serial vs. Los level parallel structures. Some of the descriptions should exhibit a degree of parallelism between register transfers.

High level parallel structures. Some of the descriptions should have asynchronous, process level, parallel structures which intercommunicate through global variables and flags.

Memory types. Arrays, registers, and shared memories should all be represented among the descriptions.

Performance requirements. Various realistic timing requirements should be represented.

Complexity. Various sizes of descriptions as represented by the number of defined register transfers and hence the degree of complexity, should be represented.

5.3 Algorithms Used in Exper-*™*nts

The descriptions chosen are listed in order of ascending complexity in Table 1. A discussion of these descriptions is provided below. These are the levels of the description factor.

Coin Change.isp. This is a description of the coin receiving and change mechanism for a vending machine offering multi-priced items. It contains 48 register transfers, 60% of which include variables defined as system input and outputs. The 15 variables Included single bit flags as well as several multibit registers. Several low level parallel constructs existed. Performance requirements were not demanding.

Elevator.isp. This is a description of a simple elevator controller and scheduler. It consisted of two high level processes operating asynchronously in parallel: one to scan the call buttons and update a call memory, the other to control the door, motor, and scheduling, 35% of the first processes' 19 register transfers are input or output as opposed to 34% of the second processes' 59 register transfers. Single bit flags, multi-bit registers and an array were included in the 23 variables. There were no low level parallel structures. All performance requirements were easily met in all design styles.

Video.isp. This is a description of a video terminal which consists of 126 register transfers divided among five high level, parallel processes. These structures control the raster scan, horizontal sync, vertical sync, keyboard, and the communications interface. The percentage of register transfers associated with system inputs and outputs varied from 1.5% to 50%. There are several instances of low level procedures operating in parallel. The 50 variables included single bit flags, multi-bit registers and arrays. The performance requirements ranged

FEATURE	Change	Elevator	Video
Register Transfer Count	48	78	126
Process Level Parallelism	0	2	5
Percent I/O Instructions	60%	• 857. - 34%	15% - 50%
Procedural Parallelism	yes	no	yes
Memory Types			
Flags	yes	yes	yes
Registers	yes	yes	yes
Arrays	no	yes	yes
Strict Performance Requirements	no	no	yes

Table 1. Algorithmic Features of the Descriptions.

from stringent in the rastor scan section to loose in the output (keyboard) communications section.

As can be seen from Table 1 these descriptions represent a range of systems to be designed and should help minimize statistical interactions. Later results will show that these interactions were not significant.

5.4 Design Styles

Several different design styles (pipelining, bit slice micro architecture, etc.) representing different design rules and trade-offs were considered but only three were selected for use. The styles are the levels of the style factor.

- Distributed: A design style representing the TTL chip set (74200 and below) and interconnected with simple link connections and multiplexors. No busing structures.were allowed. This is also very similar to LSI standard cell design.
- Bus: A design style represented by the PDF-16 module set with several extensions. This style requires the designer to allocate variables to registers and central data elements all attached to a bus structure. Such problems as register allocation for high performance sections and selection of different types and sizes of data operators are typical in this design style.
- Microprocessor: A design style represented by the Zilog Z-80 microprocessor. This style was selected because of its different problems in implementation as compared to the above two styles. In this case there is only one data operator (the microprocessor chip) and the problem is to produce a program to implement the description.

Again to minimize interactions, it was considered important that the previously defined features had different effects on the different styles. The following is a discussion of how each of the algorithm features affect the above design styles.

I/O vs. Internal calculations. In terms of I/O, the microprocessor style is more constrained than the others in that special I/O ports must be pro* vided. The distributed style is more applicable for small special purpose calculations than either of the other styles.

Low level serial vs. Low level parallel structures. Parallelism is easily handled in the distributed style but the other two styles generally require large overheads in their control and data structure. The bus style allows certain operations in parallel without large overhead.

High level parallel structures. The .ease with which each style can implement high level parallelism depends on the number of global variables which are shared. The distributed style is easily implemented no matter how many variables are shared. The bus style allows for a fair amount of intercommunication through separate flag and I/O modules. The microprocessor style allows intercommunication by way of more expensive multiport memories and I/O ports.

Memory types. The distributed style is the only non-constrained style of the three. The other styles limit the bit width of all of their memories. The distributed also allows more functionality in its single registers (clear, increment, etc.).

Performance. The microprocessor style has the overhead of instruction fetching and thus is ultimately slower than the other styles. The bus style having a hardwired control is faster than the microprocessor style but slower than the distributed style because of its common data bus.

Complexity. As a design becomes more complex because of a higher number of data operators, data paths and memories, systems tend to be better implemented with the more centralized design styles.

It can be seen from the above that the styles are not the same in terms of the effect that these algorithmic features will have on a final implementation. The next several subsections note further considerations in the design of the experiment.

5.5 Assignment of Designers

The total manpower for the project was twelve people. These people, six of whom were to do twice as much work, were to be assigned to the two experiments. Different people could not have a different number of assignments within one experiment since those doing more would cause statistical interactions by learning and becoming familiar with either a specific description or design style. Rather, the double load people split their time between the two experiments, first completing Experiment I and then Experiment II. Along with the six double load people on the Experiment I, three single load people were chosen to make the total of nine designers. Three other single load people were chosen to work with the double load people on the second experiment to make a total of nine designers.

Consider the design assignments for Experiment I in Figure 4. For a full design plan, each of the nine designers would have had to produce an implementation in each box for a total of 9x9=81 implementations. However, since this was considered to be an unrealistic work load, only three implementations were produced by each of the nine designers making this experimental design a one third (3x9=27) plan. This is illustrated in Figure 4 where A, B, and C each represent groups of three designers. Figure 5 shows

the plan for Experiment II (global optimizing level) where with nine designers there would be a possibility 6x9«54 implementations. In this more difficult experiment, only two Implementations were produced by each of the nine designers which also makes it a one third (2x9-18) plan.

The above design plans are fractional factorial experimental designs, used to create a balance between the major factors of designer, description and style* The experimental design in Figures 4 and 5 show that in Experiment I each designer (say one of the three designers in Group A) will implement a different description in each of the different styles, and that in Experiment II each designer will implement a different description in two of the different design styles.

In Experiment II, it was observed that the double load designers would be more familiar with the style and descriptions than the other three single load designers grouped with them. This was a major concern since they might do significantly better than the three single load designers. To alleviate the problem, none of the double load designers repeated a style/description combination they had done previously, and the three most experienced single load designers were used to balance the situation by working only on Experiment II. Results of Experiment II will, show that there was no major unbalance caused by these designers working on both experiments.

5.6 Instructions to Designers

Each designer was provided with a booklet describing the aim of the experiment. The booklet contained a description of each style and what was involved in designing with it. Included in these sections was a statement of the design goal, a definition of the measures to be applied, the modules to be used, the results to be reported and an example using the design

styles' modules. In addition, several hours were devoted to teaching the ISP language, and describing the design styles and descriptions. At no time were the designers taught how to design using the styles.

The basic design goal given for the distributed and bus design styles was to produce "the least expensive design which meets the designer's timing requirements." The requirements were stated by the author of the ISP description and could be found in the text of the descriptions. The design goal for the microprocessor design style was to find "the minimum program memory space which meets the designer's timing requirements." The byte count was selected rather than the cost because the cost of a microprocessor system would show very little variability. This was upheld by the microprocessor cost figures provided in addition to the byte count by the designers.

5.7 Calculations of Costs

The data for the bus and distributed styles are the actual costs of building the systems using the formula

$$\text{COST} = \text{total cost of chips} + \$3.00 \text{ overhead/chip [15]} \qquad (12)$$

The microprocessor style, where the data indicated the number of bytes to implement the program, required a conversion from bytes to dollars. This was derived by comparing the byte count of the microprocessor and the control part of the bus style since the control parts of the two styles were similar.

The conversion factor was calculated to be $1.59 per ROM byte. This figure does not suggest that the cost of a ROM chip is $1.59 per byte but rather that by comparing the control parts of two design styles a conversion

factor can be calculated by which to compare bytes and dollars.*

These costs are to be used in an analysis of variance. One of the assumptions in ANOVA problems is that of homogeneous variances. That is, the variance of a sample is not proportional to its mean. Whenever large departures from homogeniety occur it is felt that the data should be transformed using a variance stabilizing transformation in order to produce meaningful ANOVA results. There is no specific test for "too much" heterogeneity.

By understanding the variation to be expected in the data, a transformation can be determined. Intuition led us to estimate that as the cost (complexity) of a design increased, so did the variance. That is, as a designer moved on to different parts of a design, say a new process, he would be little affected by the previous portions of the design. This linear changing of the variance with the cost of the design indicated that a square root transformation of the raw data would be appropriate [11]. There are not enough data points to estimate the true relation of the cost and the variance.

*Considering all the overhead associated with a board level microprocessor implementation, this cost is realistic. This value and the perturbations around it produced "costs" that were in the range of the dollar costs (also provided by the designers) in the microprocessor style.

6. RESULTS OF EXPERIMENT I

6.1 Designer Performance

The raw data from Experiment I is shown in Table 2. The largest variation (48%** and 51%) between designers occurred in both the bus and distributed implementations of the change mechanism respectively. Also high in variation was the distributed implementation of the elevator (46%). This indicated that at least one designer produced a design that cost half again as much as the other designers in the block.

The above variations were all calculated from the raw data. The smaller designs showed more variation because of the lower absolute value of their costs. To account for this, the square root transformation as described in the previous secion was used as a variance stabilizing transformation. Table 3 shows the square root transformed data, the residuals pertaining to each point and the average result of the three designers in each cell. Examination of the square root transformed data indicates no clear cut trends. That is, the data showed little bias, providing increased confidence in the validity of the results.

Consider the best and worst design style for each of the nine designers as shown in Table 4. From Table 3 a designer can be compared with the average of all of the designers in the cell. In the video/bus cell, designer 1 was below average in cost by 1.7%, below average by 1.5% in the change/distributed cell, and below average by 7.5% in the elevator/microprocessor while the other two performances were within 15% of each other ((1.7 - 1.5)/1.5 = 13%). Whereas the worst design style for each designer was clear cut

---

**(1598-1078)/1078.

	BUS		DISTRIBUTED		MICROPROCESSOR	
	Designer	Cost	Designer	Cost	Designer	Cost
Video	1	5281	3	1322	5	1962
	4	6014	6	1290	8	1503
	7	5029	11	1457	10	1663
	Designer	Cost	Designer	Cost	Designer	Cost
Elevator	3	2326	5	406	1	588
	6	1835	8	377	4	743
	11	1899	10	278	7	716
	Designer	Cost	Designer	Cost	Designer	Cost
Change	5	1598	1	187	3	633
	8	1078	4	282	6	636
	10	1091	7	273	11	483

Table 2. Raw Data - Experiment I

	BUS			DISTRIBUTED			MICROPROCESSOR		
	Des	Cost	Res	Des	Cost	Res	Des	Cost	Res
Video	1	7267	-0.3	3	36.4	-3.7	5	44.3	i 4.1
	4	78.13	4.9	6	35.9	-3.3	8	38.76	-1.4
	7	70.92.	0.02	11	38.17	0.5	10	40.78	-0.8
	Mean •	73.91		Mean -	36.82		Mean •	41.25	
	Des	Cost	Res	Des	Cost	Res	Des	Cost	Res
Elevator	3	48.23	-0.5	5	20.15	0.8	1	24.25	-0.7
	6	4283	-3.3	8	19.41	1.9	4	27.25	1.3
	11	43.6	0.8	10	16.69	-21	7	26.75	1.6
	Mean -	44.84		Mean »	18.75		Mean =	26.08	
	Des	Cost	Res	Des	Cost	Res	Des	Cost	Res
Change	5	40.0	21	1	13.66	-0.3	3	25.16	-1.7
	8	328	-1.4	4	16.79	1.9	6	25.22	-0.6
	10	33.0	-24	7	16.53	4.3	11	21.99	-2.0
	Mean -	35.27		Mean »	15.66		Mean *	24.12	

Table 3, Transformed Data - Experiment I.

(4 designers had the bus style as their worst), the best design style frequently had two contenders. Dividing the credit between the designer's performance on his best two design styles if their difference from the average for the style and design was within 15% provides the second row of Table 4.'

	BUS	DIST	UP
Worst	4	3	2
Best	2.5	3.5	3

Table 4 - Number of Designers Scoring Best and Worst in Each Style.

Even the number of designers producing below average cost designs was evenly distributed: 2 designers had all three designs below average (Table 3), 3 designers'had two, 2 designers had one, and 2 designers had none. Variations from the average ranged from 13% over for a bus/change design to 11Z under for a distributed/elevator design. The largest spread was 20% for both the bus/change design and distributed/elevator design. This spread provides the designers of physical allocator algorithms with a powerful figure. If they can assume that they have a reasonable designer to design at the physical allocator level, then they only need one observation to estimate the range (using transformed data) in which the design algorithms should fall.

6.2 Regression Analysis and Analysis of Variance

The major factors used in the first experiment were: grand xoean, styles (label A), descriptions (B), designer psuedo factor 1 (C), designer

pseudo factor 2 (D).* Also modelled was the Interaction between the major factors Including Interactions between styles and descriptions, styles and designers and descriptions and designers. In the model selected, though, these Interaction terms were aliased. They were none-the-less calculated and shown to be Insignificant.

Table 5 shows the analysis of variance results. The first column indicates the sources of variance, and their abbreviation (e.g., A - style). Each source had 2 degrees of freedom. The first four rows are the major factors. The next three are the aliased interaction terms, and the last six are interaction terms calculated using orthogonal polynomials of order 2. It can be seen that the major portion of the variance is attributable to the difference with styles (3730.2/7591.3 - 49%) and within descriptions (i.e., the object to be designed, 44%). Only .48% is attributable to variation in the capabilities of the designers. This can be understood from the original data in Table 3 since the range of costs between descriptions or styles is greater than between designers within any particular style and description. 6.4% of the variance is attributable to all of the modelled interaction terms.

The results of this section indicate the relative differences with the styles, descriptions and designers. They show that the selection of the proper style has a profound effect on the final design produced by a CAD system. The next section will explore perturbations of certain of the parameters.

---

*Instead of modelling each designer separately, the designers were modelled as two factors each at three levels. The results are expressed in terms of two pseudo factors [11] whose sum is the total variance due to designers. The defining of pseudo factors in this case made the calculations simpler.

**This last number is found by pooling, adding, the variances due to the two designer factors.

SOURCE	SSQ*
Style (A)	3730.2
Description (B)	3338.2
Designer (C)	14.8
Designer (D)	21.9
AB-CD	207.0
AC-BD	7.5
AD-BC	0.5
AB2	194.2
AC2	10.9
AD2	19.2
BC2	1.0
BD2	9.7
CD2	36.0
TOTAL	7591.3

Analysis of Variance
And
Deviation About the Mean
of Different Levels of
Factors A and B

*Each source has 2 Degrees of Freedom

Table 5. ANOVA - Experiment I.

6.3 Perturbation Analysis

Two parameters were identified as potential sources of errors: the cost per byte of ROM conversion factor in the microprocessor style, and the width of the bus in the bus design style. The cost of the designs from the experiment were re-evaluated for the cost per byte of ROM of 0.47 [16] , $1.59 (derived from RTCAD data and used in the main tables above) and $3.62 (derived from a different bus style control module.

Since some of the designs had variables composed of a small number of bits, 4, 8, and 16 bit bus widths were also explored in the bus design style. This was done by scaling the data part of the design by the appropriate factor.

Figure 6 depicts the expected cost in respect to the grand average of all designs of each design style as a function of the cost per byte of ROM. The bands represent the spread due to different bus widths. In general, the distributed design style is best (relative cost of 0.6 to 0.35) and the bus design style is worst (relative cost 1.2 to 1.6). However, for low cost per byte of ROM (below about 1.0) the microprocessor design style is best. For high cost per byte of ROM (above about 3.5) the microprocessor design style is worst. R^2 for all these cases ranged from 0.958 to 0.988.

Figure 7 depicts the average relative cost in respect to the grand average of all designs as a function of cost per byte of ROM. The video terminal design is about 1.4 versus .8 for the elevator and .7 for the changer. This merely demonstrates that there was a difference in cost and complexity of the objects being designed.

Figure 8 depicts the percentage of the variance due to the design style and object designed as a function of cost per byte of ROM. Again the band represents the range assuming different bus widths. The design style has more impact for cheaper cost per byte of ROM. The sum of the design style

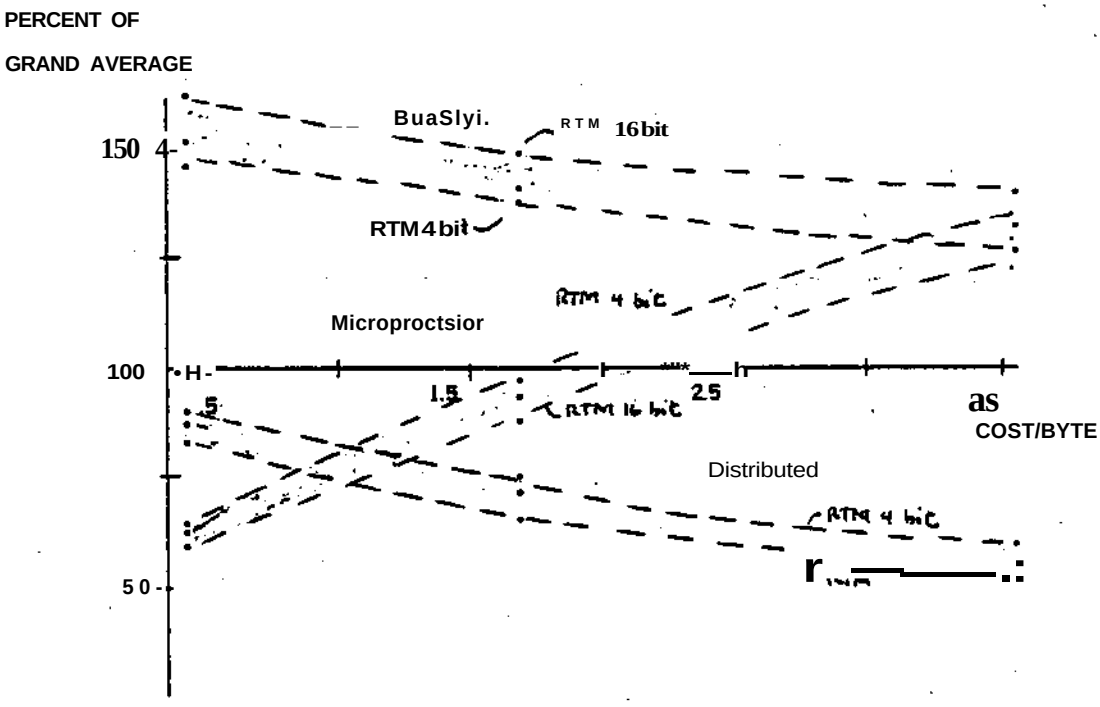

Figure 6. Cost of Style vs. Cost *per* Byte

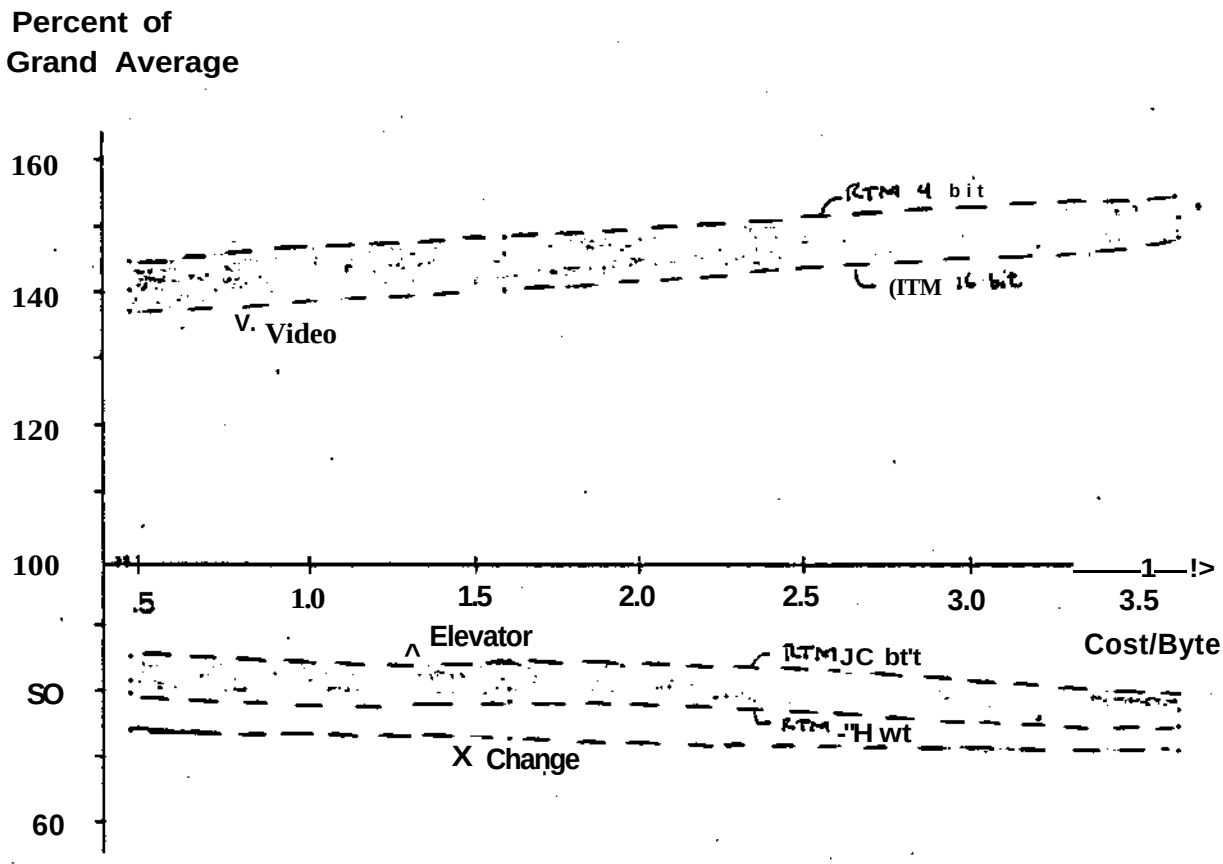

Figure 7. Cost of Description vs. Cost per Byte

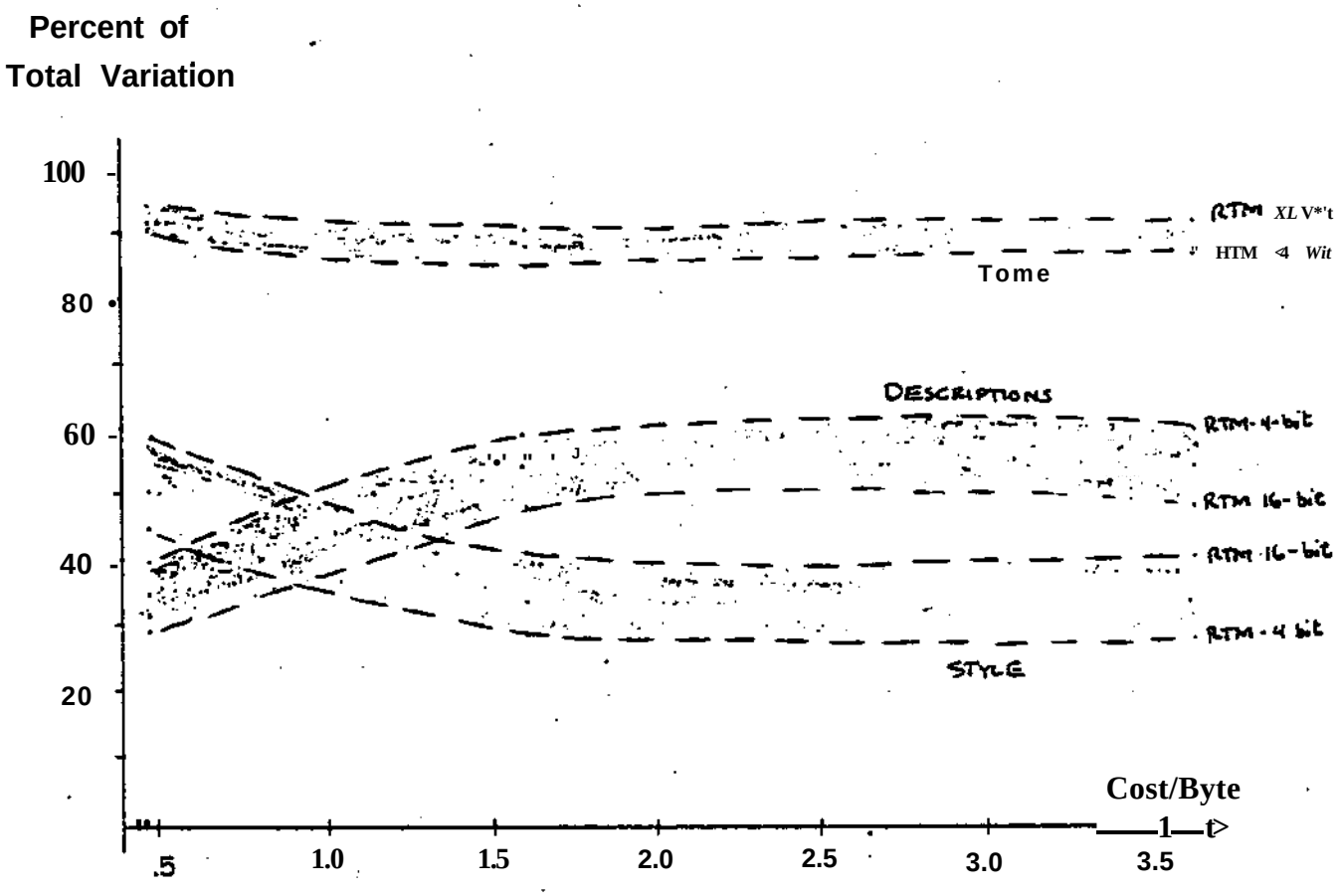

Figure 8. Percentage Variance-Due To Style.and Design

and object designed is almost a constant 90% implying that over a wide range of assumptions these two factors are the dominate contribution to design variation.

Experiment I has demonstrated the viability of the design style approach for a register transfer level CAD system. It has illustrated that the proper selection of the overall approach to the final design can have profound effects. Perturbation analysis has shown that as the cost/byte factor decreases (meaning less expensive microprocessors) the microprocessor style will become prevalent in more situations.

7. RESULTS OF EXPERIMENT II

7.1 Designer Performance

The original data from Experiment II is shown in Table 6* An important consideration in comparing the following results to those of Experiment I is that there are now only 18 data points by which to build a statistical model. Although fewer factors will be modelled, significant results are still attainable.

Analysis of the variations in the raw data show larger variations between designers (Table 6). The bus implementation of the elevator had a 97% variation and the distributed implementation of the video terminal had a 71% variation. The microprocessor style had low variation (5% and 19% for the video and elevator respectively)• This is attributed to the fact that the translation from ISP to assembly language is no more difficult here than in the first experiment.

Table 7 shows the square root transformed data. As shown in Table 8, there are no clear cut best or worst styles for the designers. Only one designer (6) did equally well on both designs so that he could not be categorized. Two of the designers had both designs above average (in cost); three had both designs below; three had one design above and one design below; and one designer had one design below average and one design right on average.

Of the designers performing only in Experiment II (12, 13, and 14) two provided the high points for the cell, one provided the low point, and 3 provided the midpoints for their cell. This upholds the previous decision to allow some designers to perform in both experiments. The uniformity of these results increases our confidence in the group of designers as being representative of a range of designers.

	BUS		DISTRIBUTED		MICROPROCESSOR	
	Designer	Cost	Designer	Cost	Designer	Cost
Video	5	8728.3	4	1289.8	6	1140.
	8	6698.3	7	1434.7	9	1171.83
	13	5825.4	12	839.1	14	1202.
	Designer	Cost	Designer	Cost	Designer	Cost
Elevator	4	1675.1	6	220.5	5	734.6
	7	1113.	9	221.9	8	615.3
	12	2187.8	14	313.5	13	704.4

Table 6. Raw Data - Experiment II.

	BUS			DISTRIBUTED			MICROPROCESSOR		
	Oes	Cost	Res	Oes	Cost	Res	Des	Cost	Res
Video	5	93.4	9.5	4	35.91	1.4	6	33.76	-1.9
	8	81.84	-20	7	37.88	3.3	9	38.64	2.9
	13	76.32	-7.5	12	29.88	-4.7	14	34.67	-1.0
	Mean »	83.85		Mean -	34.56		Mean =	35.69	
	Des	Cost	Res	Des	Cost	Res	Des	Cost	Res
Elevator	4 '	40.93	0.6	6	14.85	-0.96	5	27.1	0.95
	7	33.36	-7.0	9	17.7	1.9	8	24.81	-1.3
	12	45.58	5.4	14	14.9	-0.9	13	26.54	0.39
	Mean	-39.96		Mean -	15.82		Mean -	26.15	

Table 7. Transformed Data - Experiment II.

	BUS	DIST	uP
BEST	3	2	3
WORST	3	3	2

Table 8. Best and Worst Styles For Designers.

Analysis of the transformed data shows the max-firmm range between designers to be about 40% (in the Elevator/Bus design). This maximum range is twice that shown by Experiment I bearing out the original contention of the varying difficulties of the design tasks of each experiment. Only in the microprocessor style was there little change in the range between the designers from Experiment I to Experiment II. This is because the design task Involved is a translation from ISP to assembly language and there were not enough radical transformations made to the ISP to drastically effect the byte count*

7.2 Analysis of Variance

It was expected that the results of Experiment II would be similar to those of Experiment I. The model used to analyze the second experiment data used the following major and interaction factors: grand mean, styles (A), description (D), Styles 2 (A2) a second orthogonal linear combination of values to calculate the variance due to styles, and style-description interaction terms* No aliased terms were calculated.

Table 9 shows the results of the analysis of variances. Again, the major impact is from the style and description. Effects due to designers are in the 4% not represented by this model. The variance due to style is found by pooling factors A and A2. Style accounted for 55% of the variance and descriptions accounted for 30%. The interaction effects are down in the 10% range.

SOURCE	SSQ*
Style (A)	2914.5
Description (D)	2578.8
A2	1817.6
AD	866.6
A2D	60.8
TOTAL	8535.7

Analysis of Variance

Revised Model

$R^2 = .96$

* Each source has 1 degree of freedom

Table 9. ANOVA - Experiment II.

9. CONCLUSION

The two experiments described In this paper have Illustrated several points:

1) Despite the complexity of automated design algorithms, and the difficulty of obtaining statistical observations, a method has been demonstrated for gathering data on, specifically, designer performance at the different levels of the RTCAD system.

2) A methodology has been demonstrated, in general, for computer aided design systems where the intuitive designer still performs more capably than the present algorithms and heuristics for automated design.

3) That the selection of the design style by the RTCAD system is a major step toward finding the optimal implementation.

4) The physical allocators may be verified with hand designs from a "hand design data base" or (to within 20%) using transformed data by having a designer or two produce hand designs of newly described objects.

ACKNOWLEGEMENTS

We would like to thank Dr. Paul Shamen of the Carnegie-Mellon University Statistics Department for his consultation during the development and analysis of the design experiment. We also wish to thank the designers who spent many hours preparing the designs for the experiment.

REFERENCES

[1] Mbrreale, E., "Partioned List Algorithms for Prime Implicate Determination from Canonical Forms," IEEE Trans, on Electronic Computers, EC-16, 611-620, October 1967.

[2] Bell, C.G. and A. Newell, Computer Structures, Readings and Examples, McGraw Hill, New York, 1971.

[3] Barbacci, M.R., Automated Exploration of the Design Space for Register Transfer (RT) Systems, Ph.D. Thesis, Computer Science Department, CMU, Pittsburgh, 1973.

[4] Siewiorek, D.P. and M.R. Barbacci, "The CMU RT-CAD System, An Innovative Approach to Computer Aided Design," National Computer Conference, New York, 1976.

[5] Snow, E.A., D.P. Siewiorek and D.E. Thomas, "A Technology-Relative Computer-Aided Design System: Abstract Representations, Transformations, and Design Tradeoffs," L5th Design Automation Conference, June 1978, Las Vegas, Nevada.

[6] Barbacci, M.R., G. Barnes, R. Cattell and D.P. Siewiorek, "The Symbolic Manipulation of Computer Descriptions: ISPS Primer and Reference Manual," Research Report, Depart, of Computer Science, CMU, Pittsburgh, 1977.

[7] Hafer, L., and A.C. Parker, "Register-Transfer Level Digital Design Automation: The Allocation Process," Proceedings 15th Design Automation Conference, Las Vegas, Nevada, June 1978.

[8] Thomas, D.E., The Design and Analysis of an Automated Design Style Selector, Ph.D. Thesis, Electrical Engineering Department, CMU, Pittsburgh, 1977.

[9] Thomas, D.E. and D.P. Siewiorek, "Measuring Design Performance to Verify Design Automation Systems," 14th Design Automation Conference, New Orleans, La., 411-418, June 1977.

[10] Bell, C.G., J. Grason and A. Newell, Designing Computers and Digital Systems Using PDP16 Register Transfer Modules, Digital Press, Digital Equipment Corporation, 1972. .

[11] Anderson, V.C. and R.A. McLean, Design of Experiments, A Realistic Approach, Dekker, New York, 1974.

[12] Hicks, C.R., Fundamental Concepts in the Design of Experiments (Second Edition), Holt, Rinehard and Winston, New York, 1973.

[13] Fisher, R.A. and F. Yates, Statistical Tables for Biological, Agricultural and Medical Research (Fourth Edition), Edinburg and London, Oliver and Boyd, Ltd., 1953.

[14] Draper, N.R. and H. Smith, Applied Regression Analysis, Wiley, New York, 1966.

[15] Blakeslee, T.R., Digital Design With Standard MSI/LSI, Wiley, New York, 1975.

[16] Davidow, W.H., "General Microcontrollers. Part 1: Economic Considerations," Computer Design, 11, 7, 75-79, July 1972.