UCID- 17433

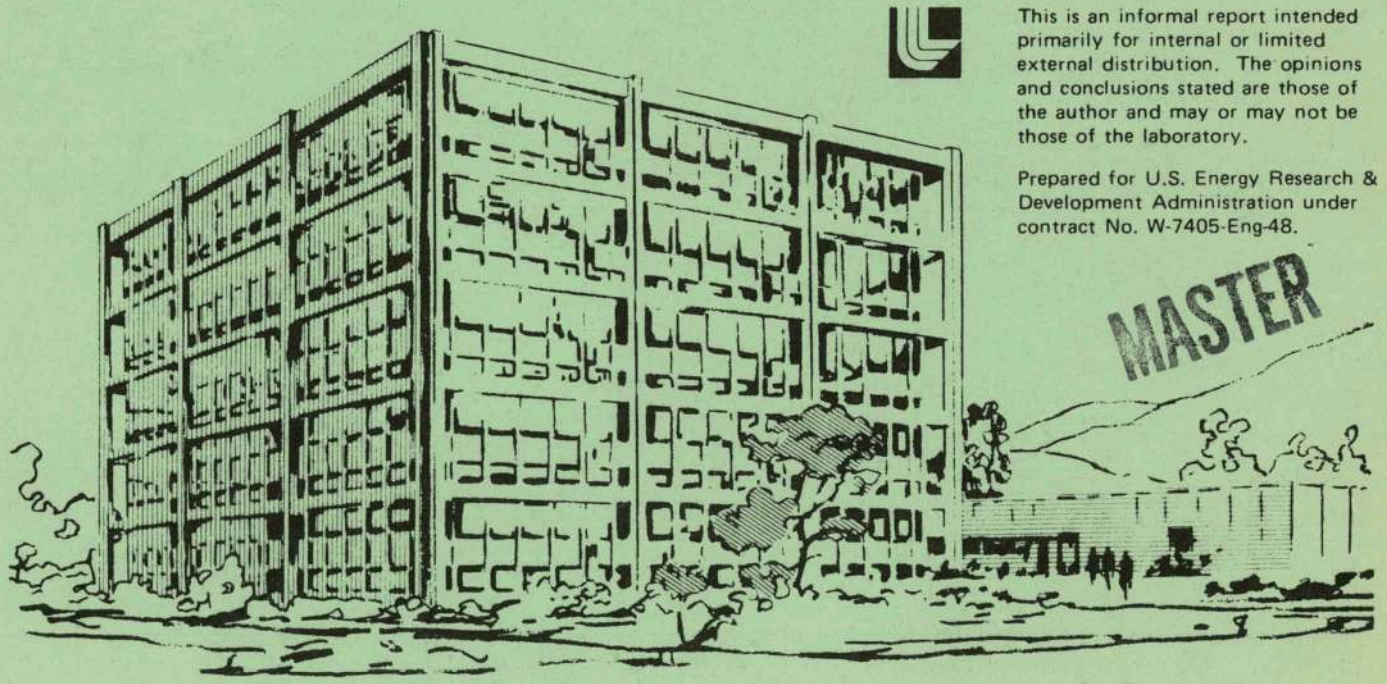
Lawrence Livermore Laboratory

STRUCTURED COMPUTER-AIDED DIGITAL LOGIC DESIGN: THE S-1 DESIGN SYSTEM

Lawrence C. Widdoes, Jr., and Thomas M. McWilliams

Special Studies Group
Physics Department
University of California Lawrence Livermore Laboratory
Livermore, California

March 31, 1977



Prepared for U.S. Energy Research & Development Administration under contract No. W-7405-Eng-48.

DISCLAIMER

STRUCTURED COMPUTER-AIDED DIGITAL LOGIC DESIGN: THE S-1 DESIGN SYSTEM

Lawrence C. Widdoes, Jr., and Thomas M. McWilliams

Special Studies Group
Physics Department
University of California Lawrence Livermore Laboratory
Livermore, California

ABSTRACT

Structured computer-aided digital logic design is defined and an example of it given, the S-1 Design System. It is noted that such a system has resulted in a very large reduction in the amount of designer effort required to complete the design of a large-scale digital processing system in the first exercise of its capabilities. This basic and generally applicable advance in digital design technology may be expected to quite favorably impact all complex digital systems developments.

INTRODUCTION and SUMMARY

This report briefly describes a novel computer-aided digital logic design system which is currently being used in the design of a high-performance multiprocessor. The S-1 Design System is a set of computer programs which allow digital logic to be represented in "structured" or hierarchal drawings. The system provides extensive interactive support during the creation of drawings by the designer, i.e., the specification of the system design. It subsequently expands these drawings automatically to produce input for a gate-level logical system simulator program, and also input for computer-driven machines in the industrial sector which perform the automatic construction of the specified digital logic system, e.g., a multiprocessor module. Its use allows a very large savings in design manpower, relative to that required by currently employed digital design techniques, and can reduce the total time required for the creation of a superprocessor from the current 5-8 year periods to under a year.

STRUCTURED DESIGN: NEED and NATURE

In general, ideal structured design involves the design of arbitrary modules, each in terms of a few other modules, relatively independently, which communicate through well-defined interfaces. Advanced programming languages and programming disciplines exemplify these goals; arbitrary subroutines are written, each in terms of a few other subroutines, relatively independently (i.e., they can often be written by different programmers), and they communicate through interfaces defined by their input and output parameters.

Historically, logic design has lagged far behind program design in terms of these ideals; logic is still typically hand-drawn by draftsmen, and is specified by showing on drawings the actual physical connections

between the primitive logical elements which are available in sections of integrated logical circuit packages ("chips") from semiconductor manufacturers. Traditional logic design thus supports a single level of abstraction; the design is not specified in terms of the circuits which implement logical gates, but in terms of the gates themselves. A good structured design system would facilitate the creation of multiple levels of such abstraction; the S-1 Design System is the most powerful such system of which we are aware.

THE S-1 DESIGN SYSTEM

The heart of the S-1 Design System contains a Graphics Editor and a Macro Expander. The Graphics Editor allows the designer to draw prints (i.e., do logical design) at a computer-serviced graphics terminal, and to create hard copy of these prints on standard graphics output devices, such as a Xerox Graphics Printer or Calcomp Plotter. Instead of drawing prints only in terms of chips, macros (precisely defined abbreviations) can beused which are defined at will by the designer himself. Macros represent logic modules of arbitrary size and arbitrary physical configuration. The important characteristics of such a macro are:

1. It has a small number of predefined shapes which may appear in a drawing, and all of which are defined in a library.
2. It has a set of parameters which allow logical signal vectors to be passed to the macro each time it is called.
3. It has a name, which links the macro call to its definition.
4. It has a definition, which is used to expand the design out to individual chips, thereby allowing the production

of a wire list which can be used to specify the automatic construction of the system being designed.

5. It has a logical location name, which serves to uniquely identify the use of a macro within the enclosing definition.

Macros are connected by buses, which represent vectors of signals, and are drawn as single lines for the convenience of the designer. Each bus may have a name or set of names (synonyms).

The designer thus manipulates shapes (representing macros) on a graphics terminal, connects them by buses, and enters labels for buses, macro names, and macro locations. He is able to define new (understandable) shapes to represent complex functions. The designer's freedom of expression in this context is constrained by the discipline implicit in the five characteristics above.

A sample of what the designer manipulates on his screen is shown in Figure 1. The lowest level of abstraction shown in this example is a single ECL gate, a section of a 10105 chip. The gate is shown in one of its two traditional forms, the other being an "or" shape. The drawing also includes a RAM and a register, which are drawn as boxes. The middle body is a mnemonic shape representing a 16-bit ALU (arithmetic-logic-unit).

Figure 2 shows part of the expansion process accomplished by the Macro Expander. Within the set of drawings created by the designer exists one with the title "16-Bit ALU," and that drawing defines the logical structure of the 16-Bit ALU macro in terms of several other macros. A list of bus names below the keyword "parameter" identifies the formal parameters of the 16-Bit ALU; those vectors of signals are implicitly connected across the boundary between the macro call and its definition.

The parameter-passing mechanism is an extremely important aspect of the design system; its function is to establish well-definied interfaces through which modules defined relatively independently can communicate. Note that the names of the formal parameters of a macro are defined by the designer, just as the names of formal parameters of a subroutine are defined in the subroutine, thus providing effective isolation between different sections of the design, in the same way that the use of subroutines isolates various portions of a computer program.

Just as global variables can be used in programming languages, global buses can be used in the S-1 Design System. Such buses must be declared (in order to protect the designer from certain modes of error) and can be used by any macro definition lower in the expansion tree than the declaration. The availability of global buses allows the designer to limit parameters to those buses which increase the understandability of the design.

The similarities between structured computer-aided logic design and computer-aided program design are pervasive. Figure 3 illustrates those similarities. In the S-1 Design System the designer uses a graphics console and interacts with a graphics editor, just as a programmer uses a keyboard and text display when interacting with a text editor. The graphics editor produces a description of the design which serves as input to the macro expander, just as the text editor produces input to a compiler. Syntactic errors are detected by the macro expander just as is done by a compiler. The macro expander finally produces a wiring list which builds the machine, just as the compiler produces machine code which implements the desired program.

EXPERIENCE WITH STRUCTURED DESIGN

We have had extensive experience with the S-1 Design System during the design of a 4000-chip superprocessor-scale module of a multiprocessor system. The system has been of enormous utility throughout the design process. Most importantly, the imposition of a structured design discipline has greatly increased the understandability of the design, reducing the design time by a large factor, enhancing design correctness, and facilitating the generation of final documentation. The design itself serves as a major portion of the final documentation because it is so readily understandable; thus, the need for expensive and relatively inaccurate post facto documentation is greatly reduced. Another major advantage of the structured design approach has been that it has increased the changeability of the design; since macros are inherently isolated, changes in one macro definition usually require minimal changes in other parts of the design.

Finally, we feel that the imposition of structure on the design will facilitate machine verification of large designs; that is, although simulation of a large digital logic system design at the chip level is currently difficult because of computer time and memory requirements, doing the design in a structured manner allows part or all of the design to be simulated at a high level comparatively readily.

CONCLUSION

Structured design consists of extending to digital logic design the essential power of the concepts and the tools which have been developed for simplifying digital computer programming over the past quarter century; the S-1 structured design program package is used to

cause computer-controlled equipment to create digital logic in hardware from a set of near-minimal specifications by the designer.

Savings in human labor expended in digital systems design realizable by this advance are potentially as large as those which the application of compilers caused in the specification of complex arithmetic and logical computations; very large effort reductions have already been realized in the first utilization of this structured design technology. Very substantial subsequent advances doubtless will be made as this technology matures, with correspondingly large reductions in the cost (and therefore greatly increase cost-effectiveness) of large digital logic systems.

STRUCTURED LOGIC DESIGN EXAMPLE

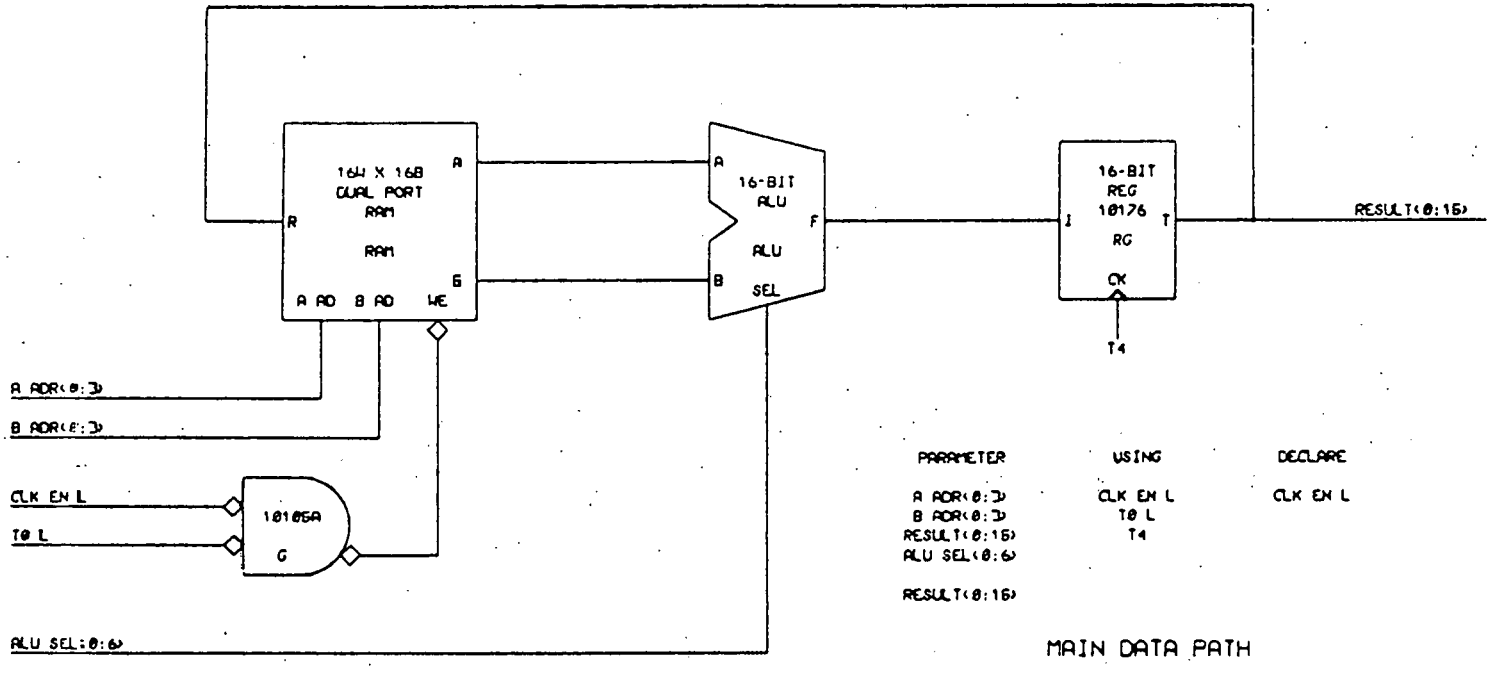


FIGURE 1

STRUCTURED LOGIC DESIGN EXAMPLE

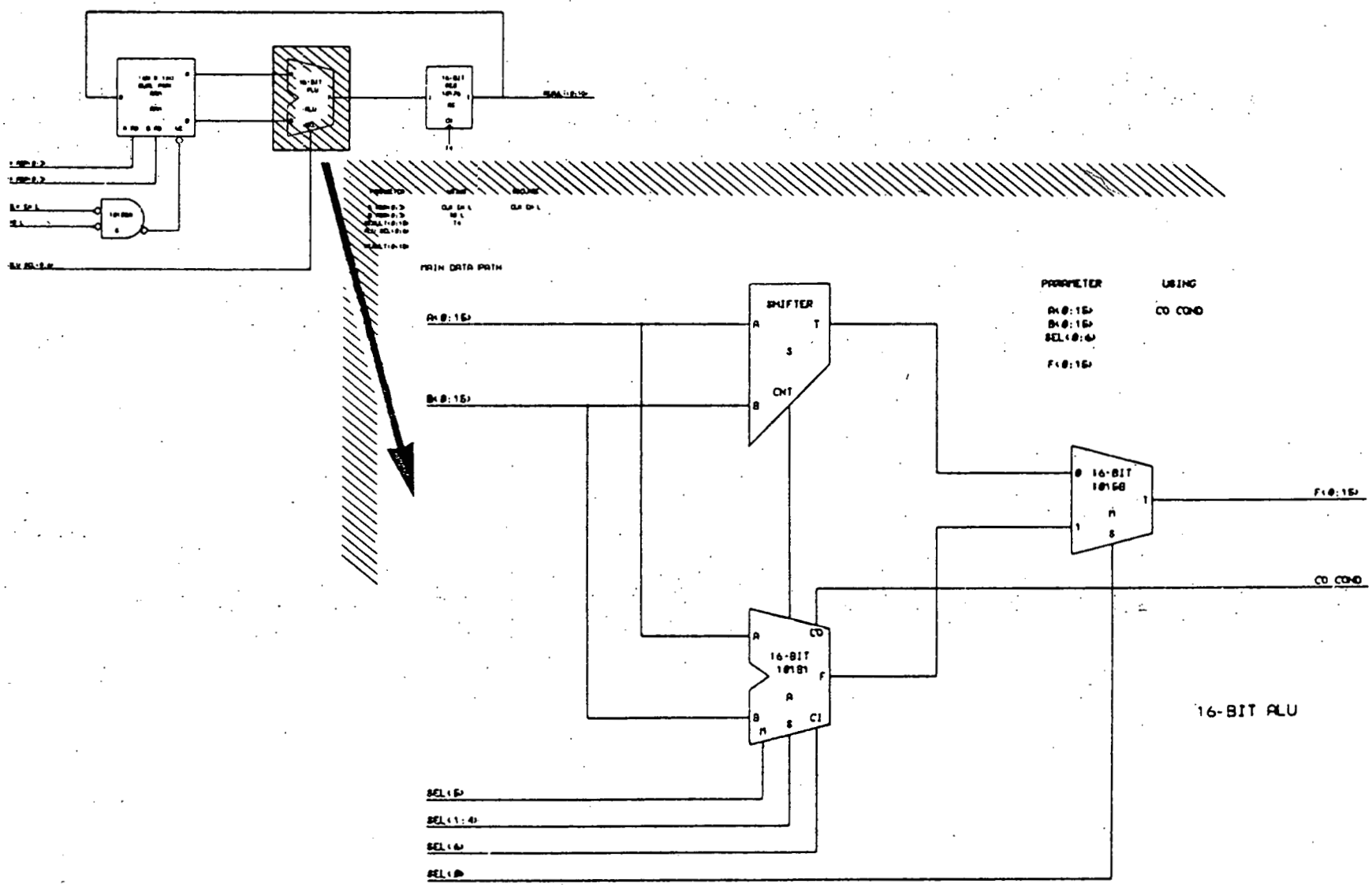


FIGURE 2

COMPUTER AIDED LOGIC DESIGN VERSUS COMPUTER AIDED PROGRAM DESIGN

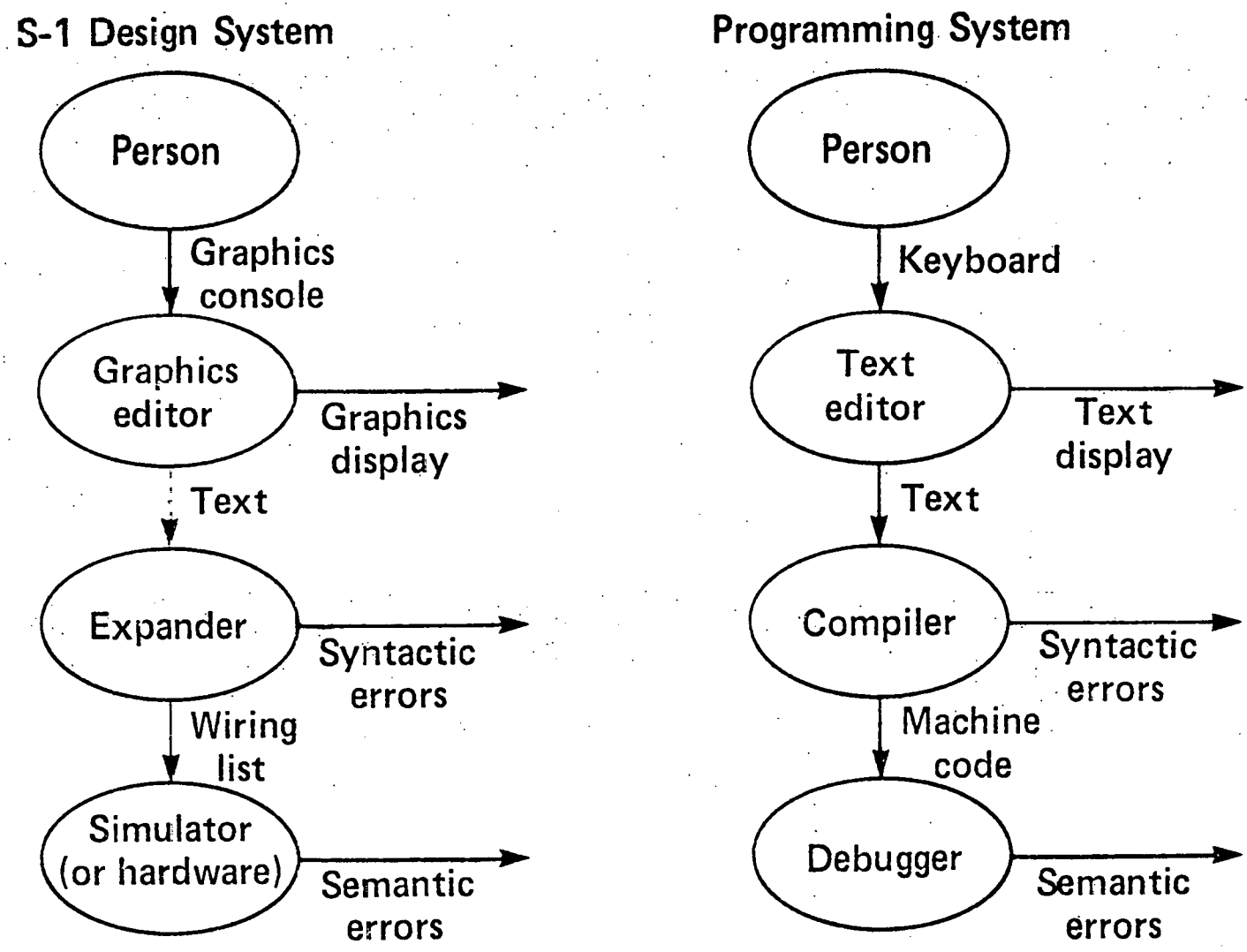


FIGURE 3

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161
Price: Printed Copy $; Microfiche $3.00

Page Range	Domestic Price	Page Range	Domestic Price
001–025	$ 3.50	326–350	10.00
026–050	4.00	351–375	10.50
051–075	4.50	376–400	10.75
076–100	5.00	401–425	11.00
101–125	5.50	426–450	11.75
126–150	6.00	451–475	12.00
151–175	6.75	476–500	12.50
176–200	7.50	501–525	12.75
201–225	7.75	526–550	13.00
226–250	8.00	551–575	13.50
251–275	9.00	576–600	13.75
276–300	9.25	601–up	*
301–325	9.75		

*Add $2.50 for each additional 100 page increment from 601 to 1,000 pages; add $4.50 for each additional 100 page increment over 1,000 pages.

Technical Information Department
LAWRENCE LIVERMORE LABORATORY
University of California | Livermore, California | 94550