EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH
CERN - LEP DIVISION

CERN/LEP-BI/84-7
(01501)

LOGIC DESIGN WITH A DAISY SYSTEM

L. Pregernig

ABSTRACT

Electronic design has been carried out with the aid of a Computer Aided Engineering workstation. The paper describes this new approach to designing, the facilities which were used and tries to indicate some implications and gains of using such a system.

Presented at the ESONE Seminar on Computer Aided Design
of Electronics in Scientific Applications
PADUA, Italy, September 26, 1984

Prévessin, 14th September 1984.

1. INTRODUCTION

During the last few years, a new type of computerized tool, the Computer Aided Engineering (CAE) workstation, has been introduced on the market. This tool should be of great help for the electronics engineer at any level of his design work.

Designing consists of various tasks, such as specifying the particular design, establishing the circuit diagram, producing and debugging a prototype, modifying the design and preparing the documentation etc.. A design evolves gradually towards the "final version" requiring several iterations.

Designing with the aid of a CAE workstation still remains an iterative process but with the important difference that "software prototyping" is being employed rather than "hardware prototyping".

In the following, this way of designing will be described. The paper is based on the experience gained from carrying out two jobs on a workstation produced by Daisy Systems*. The workstation which has been rented, was operated by an application engineer of the firm. The first circuit contains 300 integrated circuits in ECL technology, the second one contains only 12 integrated circuits in ECL and TTL technology, but includes programmable array logic. The two circuits may be considered as "typical" jobs. At present, the first one exists as a wire wrap module and testing of it has started. For the second one which will be produced as a hybrid circuit, the printed circuit board is currently layed out. The confidence level is such that no wire wrap module of this circuit was considered necessary.

2. CIRCUIT DIAGRAM - DESIGN DATA BASE

In both cases hand-drawn circuit diagrams had been established before transferring the design to the workstation, but none of the designs had been tested or debugged previously. The large circuit has a two level hierarchical structure, the small circuit is a "flat" design.

On the workstation, the schematics may be entered by means of the keyboard or a tablet and the respective software package. At first the "components" are extracted from the component library and placed on the "drawing sheet". Then the connections in form of individual signal lines, busses or sub-busses, are made. "Connectors" establish the connections to other drawing pages or the outside world (inter-page or hierarchical connectors).

Next a program is run to check the drawing at the connectivity level (unused pins, multiple drives, duplicate names, netlist errors etc.) and to translate the source data into object form. If there are no errors found, this preprocessed information is stored in an intermediate file.

* LOGICIAN V workstation.

The intermediate files generated for each drawing of the design are then linked together to resolve inter-page and hierarchical references and finally formatted to produce input files for simulation or design automation programs.

At this level the complete design data base has been created, which is the software equivalent of a hardware prototype.

Preprocessing and linking take roughly from some 10 seconds, up to 1-2 minutes, per page, the complete linking of the large design (32 sheets, size A3) taking about 40 minutes. In that context, it was useful that on this workstation, a design change not touching the hierarchical structure required only preprocessing and linking of the affected page.

Particular attention should be payed to the component library which holds for each element information on its graphic representation, the electrical behaviour and design automation parameters. For the graphic representation of a component, symbols can be chosen according to various standards (e.g. IEEE Standard 91). The electrical model uses primitives (e.g. D-Flip-Flop, Latch, etc.) and/or Boolean equation-like expressions to describe the logic behaviour of the component and, in addition, properly attached timing values to describe the timing behaviour of the component. This concept is illustrated by a Quad 2-Input Multiplexer, the 10158 type, as an example :

```
Inputs : SEL,A1,A2,A3,A4,B1,B2,B3,B4.
Internal State : IS[7,4,15];
Outputs : O1[25,11,38],
          O2[25,11,38],
          O3[25,11,38],
          O4[25,11,38],

SEL = IS ;
O1=(A1 AND NOT IS) OR (B1 AND IS) ;
O2=(A2 AND NOT IS) OR (B2 AND IS) ;
O3=(A3 AND NOT IS) OR (B3 AND IS) ;
O4=(A4 AND NOT IS) OR (B4 AND IS) ;
```

The numbers in the square brackets are the values for the typical, minimum and maximum propagation delays in 100 ps units. The internal state, IS, has been introduced to model correctly the different propagation delays from either the Select or the Data (A,B) inputs to the outputs.

The design automation parameters refer to physical implementation data of the component, such as the number of pins per package, the number of equivalent functions within one package, etc., etc..

Implementing a new component into the library does not require any special programming experience but the internal structure of the component has to be known.

3. DESIGN VERIFICATION

Tools which allow the designer to verify his design exist on the workstation for circuit analysis, logic simulation, timing verification, testability analysis and fault simulation.

The circuit analysis tool is the widely used program SPICE.

Logic simulation, decoupled from timing verification, allows to evaluate interactively the logic behaviour of the design at the gate and/or functional level. In order to simulate different technologies accurately, 12 logic states have been provided. The state of each node of the design is defined by one of three levels (0,1,Unknown) and by one of four strengths (Forcing, Resistive, High Impedance, Unknown). The strongest source will dominate and drive the node. Which strength is assigned to a level depends on the technology, e.g. in ECL technology the "1" level is "Forcing" and the "0" level is "Resistive", thus WIRED-OR conditions will be handled correctly.

In order to perform a logic simulation, the user specifies an input stimulus, the number of time units for which the simulation should run (100 ps time units for ECL) and the sampling rate for displaying the simulation results.

The computing time for a simulation run depends on the complexity of the design and the number of simulation steps. It takes about one minute to simulate one page from 0 to 1800 ns in 100 ps time steps. In this way, in both designs, errors have been detected. A design error is corrected by changing the circuit diagram and by preprocessing and linking the parts of the design which were affected by the change.

Timing verification, intended to analyse the timing behaviour of a digital circuit, checks the setup and hold time of any signal in respect to the corresponding reference signal, pulse width requirements, the relationship of signal edges, contention on busses, etc., etc.. The timing verifier of the used workstation recognizes 6 levels (0, 1, Rising, Falling, Changing, Stable) and 4 strengths (Forcing, Resistive, High Impedance, Unknown). The computing time requirements of the timing verification program are similar to those for the logic simulation. Some functions of the small design which is an interface circuit, depend on signals from the user of that interface. The final timing specifications for these signals were obtained from the results of the timing verification program.

The testability analysis program provides a measure of the difficulty of testing a design after manufacturing. It calculates the controllability and observability values for each node in the circuit.

The fault simulation program checks, whether faults which have been introduced into the circuit by this program, can be detected with a given test vector.

4. CONCLUSION

Carrying out the first job on the workstation gave a very good evaluation of its performance. The results of that evaluation were so impressive that it was considered indispensable to carry out the design of the interface circuit which will be produced in larger quantities, on the workstation.

At the present stage it is felt that this way of designing is a great improvement compared to the traditional way and should lead to a product of much higher quality.

A final report, describing also the design automation part of the experiment, will be available after the completion of the two projects towards the end of the year 1984.

5. ACKNOWLEDGEMENT

I wish to thank M. Rabany for support and interest in this experiment.