Report 82-05
Stanford -- KSL

Scientific DataLink

PALLADIO: An Expert Assistant for
Integrated Circuit Design.
Harold Brown, Mark J. Stefik,
Apr 1982

card 1 of 1

Heuristic Programming Project
Report No. HPP-82-5

April 1982

Palladio: An Expert Assistant for Integrated Circuit Design

HAROLD BROWN
HEURISTIC PROGRAMMING PROJECT
DEPARTMENT OF COMPUTER SCIENCE
STANFORD UNIVERSITY

MARK STEFIK
VLSI SYSTEM DESIGN AREA
XEROX PALO ALTO RESEARCH CENTER

The Stanford University component of this research is funded by the
Defense Advanced Research Projects Agency under Contract MDA-903-80-c-007.

Palladio: An Expert Assistant for Integrated Circuit Design[1]

Harold Brown
Department of Computer Science, Stanford University

Mark Stefik
VLSI System Design Area, Xerox PARC

Abstract. *We are currently developing a system, Palladio, which serves as a vehicle for experimentation with various integrated circuit design methodologies and with knowledge-based expert system design aids. This paper describes the basic design concepts underlying Palladio, the overall architecture of Palladio and the current development status.*

1. Introduction

The Palladio[2] system is a framework for experimentation with integrated circuit (IC) design methodologies, expert system techniques, and symbolic circuit simulation concepts. Palladio serves as the focus for the Knowledge-based VLSI Project (KB-VLSI project), a collaborative activity between the Heuristic Programming Project, Stanford University and the VLSI System Design Area, Xerox Palo Alto Research Center.

The KB-VLSI project is concerned with understanding the processes by which artifacts, in particular, integrated circuits, are designed. The long-term goals of the project are:

> Identify and articulate the expert knowledge used in integrated circuit design. An objective here is to gain an understanding of the design process and to develop cognitive models of the process.
>
> Develop methods for representing and reasoning with design knowledge. Such reasoning involves design constraints, goals, and tradeoffs.
>
> Develop knowlege-based expert systems for assisting in the IC design, test and debug cycle. The systems include aids for entering and recording IC design specifications and aids for transforming abstract design specifications into more detailed specifications.

---

[1] The Stanford University component of this research is funded by the Defense Advanced Research Projects Agency under Contract MDA-903-80-c-007.

[2] *Andrea Palladio* (1518-1580) was an Italian Renaissance architect of great reknown. He is perhaps best known because he developed a methodology of proportion and formal architectural style that has become known as *classical* architecture. In a sense, he was the first *knowledge engineer* of design principles and his influential published works are still in print four hundred years after his death.

Palladio is the primary research vehicle for the KB-VLSI project.

2. Palladio's Model of the Design Process

An IC design process can be viewed as the creation of behavioral and structural specifications of a circuit. This usually involves a sequence of transformations from abstract specifications of the behavior and structure of the circuit to more detailed implementation specifications. For example, the design of a combinational logic circuit may involve first transforming a specification of the circuit in terms of boolean equations which relate the inputs and outputs into a specification in terms of logic gates and interconnection networks, and then transforming this latter specification into a layout specification expressed in terms of "colored" rectangles.

A useful metaphor for this transformation process is that design is search [7]. The designer searches in a solution space of implementation specifications. Moves in this space are design decisions. Each design decision involves considering alternative implementations, testing the alternatives against the constraints and goals imposed by the abstract specifications, and using tradeoffs to differentiate between "satisficing" alternatives and to resolve conflicts between incompatible constraints and goals. The design decision process is difficult because: (a) the solution space is large, (b) the generation of alternative solutions is expensive, (c) only partial information is available, (d) it is not possible to predict all of the consequences of a decision.

2.1. Design Hierarchies

IC designers have, in part, coped with the difficulty of making design decisions by exploiting hierarchies in the design process. One powerful hierarchical technique is to decompose a device into semi-independent subdevices and to focus attention on each subdevice individually. For example, a 4-bit register can be considered as four 1-bit registers and their interconnections. The focus on a subdevice reduces the size of the solution space under consideration.

The device-subdevice hierarchy is only one way of partitioning the design process. Design using *description levels* (abstract models of circuits) is a complementary way to do it. Each description level provides languages for describing the behavior and structure of a device which suppress particular details of physical implementations of the device. The use of description levels reduces the complexity of the elements in a solution space and makes the generation and comparision of alternatives less expensive.

Description levels also permit a designer to partition concerns by concentrating on subclasses of design decisions. For example, at an architectural level a designer can work out certain storage and communication decisions before worrying about power considerations. The derivation of useful design description levels requires significant domain-specific knowledge – a sort of "engineering of knowledge" [8].

We are currently experimenting with four description levels in the Palladio system: *Layout*, *Clocked Primitive Switches* (CPS), *Clocked Registers and Logic* (CRL) and *Linked Module Abstraction* (LMA). Collectively, these levels factor the concerns of a

digital designer [6].

The most widely used description level in integrated circuit design is the artwork or layout level. This level describes integrated circuits in terms of "colored rectangles" (representing material on a chip) that can be composed to build up large designs. Associated with the colored rectangle terms of the layout level is a set of *composition rules*, called layout design rules. The layout composition rules provide a simple *shallow model* of composition that is based on a *deep model* of electrical properties and fabrication tolerances. If designers follow these rules, their designs are guaranteed to have adequate physical spacing on a chip [3, 4].

The layout description level has several important properties which make it useful for the synthesis of designs. First, primitive terms can be combined to form larger terms and subsystems. Second, there are rules of composition that define the allowed compositions of these terms. These rules apply both to composite objects and primitive terms. Third, there is a well characterized set of *bugs* that are avoided when the composition rules are obeyed. At the layout level, these bugs correspond to the function and performance problems caused by incorrect physical spacing.

All of our proposed more abstract description levels have properties analogous to those of the layout level. The CPS level distinguishes between different uses for logic and is concerned with the digital behavior of a system. Different uses of logic include steering logic, clocking logic, and restoring l gic. The composition rules at this level prevent bugs of non-digital behavior caused by charge sharing and invalid switching levels. The CRL level is concerned with the composition of combinational and register logic. The composition rules at the CRL level preclude various bugs related to clocking in a two-phase system. The LMA level is concerned with the sequencing of computational events in a digital system. It describes the paths along which data can flow, the sequential and parallel activation of computations, and the distribution of registers. The composition rules at the LMA level preclude bugs such as starting computations before the data are ready, and deadlock bugs that arise from the improper use of shared modules.

2.2. Design Knowledge Bases

Much of the design of ICs is done by using parts of existing designs, possibly with modification. This technique exploits the fact that there are common constructs used in many circuits; for example, registers, NAND gates and input-output pads. The use of previously designed (and debugged) components in a current design is analogous to the use of subroutine packages in software development.

In Palladio, knowledge about previously defined circuits is kept in community knowledge bases. These knowledge bases can contain not only exisiting designs but also collections of knowledge about the composition and the optimization of circuit components. For example, at the CPS level we are developing a knowledge base which includes a collection of prototype logic gates, a set of rules that define the allowed composition of these gates and a set of optimization rules for reducing various costs of circuits composed of networks of gates.

The use of community knowledge bases in Palladio is supported by the LOOPS system [1]. LOOPS is an object and data oriented programming system implemented in the Interlisp [9] programming environment. LOOPS was created, in particular, to support a design environment in which there are community knowledge bases that people share, and to which they can add incremental updates.

2.3. Design Evolution

The design of a complex artifact such as an integrated circuit is an evolutionary process that follows an iterative cycle: create a candidate design — test the candidate design against current requirements — modify the design and/or requirements to create a new candidate design. An IC design system should have facilities for interactive simulation to provide a rapid feedback between proposed changes and their exercise on test cases.

Within the Palladio framework, we have begun experiments with interactive, rule-based symbolic circuit simulators. These simulators use symbolic reasoning on a hierarchy of behavioral and structural specifications for a circuit in order to predict the outputs of the circuit given a set of inputs. The simulators include a dynamic display capability, *i.e.*, "animated simulation cartoons." Our objective is to develop a design environment based on simulators, interactive editors, and debugging tools comparable in power and flexibility (and in concept) to, for example, the Interlisp Programmer's Assistant [10].

3. Palladio's Architecture

A major purpose of the Palladio system is to provide a common starting point and a framework for the research activities of the KB-VLSI project. As such, Palladio must be sufficiently general so that it can be easily extended as our research continues. At the same time, Palladio must admit sufficient specialization so that we can rapidly experiment with particular design concepts. To achieve these goals we have used a knowledge based architecture for Palladio.

The overall architecture of the Palladio system is shown in Figure 1.

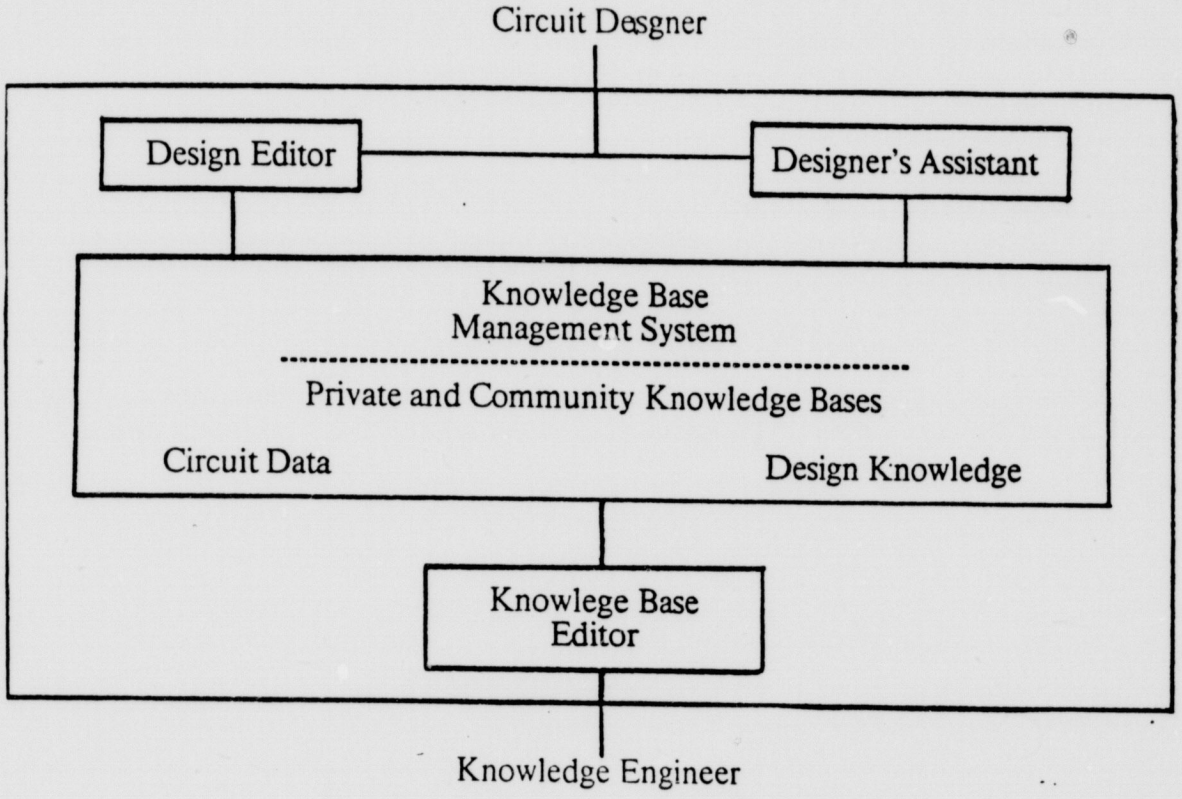

Figure 1. Palladio System Block Diagram

There are two classes of users of Palladio: knowledge engineers and circuit designers. Knowledge engineers use the *knowledge base editor* to enter concepts and rules of design that define Palladio's design methodologies. This knowledge is kept in community knowledge bases. Circuit designers interact with the *design editor* and *designer's assistant* to create circuit designs. The design editor enables a designer to enter and modify circuit descriptions at various levels of description. The editor uses the composition rules of each design methodology to assure that the design is "legal" with respect to that methodology.

The designer's assistant is an active element that can propose design decisions. The two programs are integrated with a single graphics interface from which the user can control the activity and participation of the designer's assistant. The design editor, designer's assistant, and knowledge base editor all communicate with the knowledge base via the *knowledge base management* component of LOOPS.

4. Current Status

Most of the supporting framework for the Palladio system is currently in place. The LOOPS programming system [1], has been fully implemented. A high-level, object-

oriented graphics package has been developed for the Xerox Dolphin personal computer, the development machine for the KB-VLSI project. This package, HILGA [2], is interfaced with the LOOPS system. The GLISP language [5], has been interfaced with LOOPS. GLISP provides LOOPS with optimized data and procedure access.

Prototype community knowledge bases for the CPS and LMA description levels are substantially completed. The initial knowledge bases for the layout and CRL levels are under development. A rule-based design editor for the CPS level is partially implemented.

A prototype "animated" simulator for the LMA level has been implemented. The implementation of an interactive simulator for the CPS level has been started.

Research has been initiated on expert system design assistants to aid in transforming abstract design specifications into more detailed specifications [11]. This work includes research on the use of tradeoffs in the design process.

By the end of this year we plan to have enough of the Palladio system in place so that it can be used to create designs of simple, "student – level" integrated circuits.

5. Concluding Remarks

An important purpose for Palladio is as a vehicle for community building. Opportunities for developing systematic bodies of design knowledge will appear in many parts of the VLSI design community. Although knowledge engineering provides effective techniques for capturing and debugging this knowledge, these techniques are not widely understood or practiced in the VLSI community. In particular, there is a shortage of trained knowledge engineers and suitable computers for this work. Because of the intellectual and computational hurdles, it is unlikely that expert systems will be widely available in the community for several years. It is our intention to keep this part of our research open and to invite experimentation with our facilities and participation by other members of the VLSI design community as opportunities arise.

6. Acknowledgements

The development of the Palladio system is a team effort involving all of the members of the KB-VLSI project. The current members of the project are: Gordon Foyster, Phillip Gerring, Gordan Novak, Narinder Singh, Christopher Tong and the first author at Stanford University; and Alan Bell, Daniel Bobrow, Lynn Conway and the second author at Xerox Palo Alto Research Center.