International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

IJIEMR Transactions, online available on 16th Aug 2017. Link :http://www.ijiemr.org/downloads.php?vol=Volume-6&issue=ISSUE-7

Title: **ARTIFICIAL INTELLIGENCE ON VLSID**

Volume 06, Issue 07, Pages: 69 – 76.
Paper Authors

*** ROBERT S. KIRK GOULD AMI ,VAIN HARTE.**

* CALIFORNIA

.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

ARTIFICIAL INTELLIGENCE ON VLSID

ROBERT S. KIRK GOULD AMI ,VAIN HARTE.

CALIFORNIA

ABSTRACT : Twenty years ago, artificial intelligence technology promised to revolutionize the world. As time would tell, advances in artificial intelligence have taken significantly longer than expected. Slow progress created skepticism and disinterest in the technology. Today there is a great deal of renewed interest in the field, tempered by the slow progress of the past twenty years. This new interest is focused on domain specific artificial intelligence applications, rather than the broad problem solving capabilities originally proposed. In addition this interest is focused on domains offering exceptional return on investment, either through direct profits or through leveraging of scarce resources. This paper surveys the potential impact of artificial intelligence technology on the VLSI design domain. This domain is characterized by a fifteen year evolution of computer aided design tools, a chronic shortage of skilled integrated circuit designers and ever growing demands for shorter design spans, reduced costs and design error rates.

INTRODUCTION

Advances in integrated circuit (IC) fabrication technology are rapidly outpacing IC design capabilities. The first ICs contained small scale integrated (SSI) functions, which were straightforward to design. As device counts increased, early design methods quickly became obsolete. Today computerized tools are widely used to design very large scale integrated (VLSI) circuits. These software tools are themselves very complex. Yet with advances in IC fabrication technology, the pressure to solve more difficult design automation problems continues to increase. Present software technology has been pressed to the practical limit in the current generation ofVLSI design tools. Entirely new approaches to the VLSI design problem are required.1,2 Artificial intelligence (AI) technology may hold the key to solving this problem. This paper explores the potential contributions of AI technology to VLSI design and attempts to answer the following questions. Will AI offer truely useful solutions, or will it go the way of the past twenty years? What significant changes in VLSI design can be expected in the next three, five or ten years due to AI? The degree to which AI impacts VLSI design will significantly affect the entire computing community. Advances in VLSI supports advances in computing hardware and together they feed advances in AI research. Before describing how AI technology will be used to advance VLSI design capabilities, VLSI design requirements are reviewed, followed by an overview of existing AI based VLSI design tools. Then the salient features of AI technology are examined to draw some conclusions on their impact on VLSI design tools and methodology.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

TRENDS IN VLSI DESIGN

The VLSI design tool domain is commonly referred to as Computer-Aided Design (CAD) or Design Automation (DA) as it applies to integrated circuit design. The complexity of ICs is increasing so rapidly that ICs are no longer limited to simple logic devices. Rather complete digital systems are being designed on a single silicon chip.3 Thus the term VLSI design means both digital systems design and IC design.

Digitizing

In the early days, CAD tools were developed to automate the tedious and error-prone task of creating IC photomask artwork. By digitizing the photomask drawings or layouts, a computerized editing system could be used to make changes. New artwork was then generated automatically on a photo The Impact of AI Technology on VLSI Design 127 plotter. A digitizing system is the graphical analogy to a textual word processing system. The main drawback of digitizing systems is that they did little to help a person perform the design task.

Checking and Analysis

Next came a generation of checking and analysis tools. These tools are similar to a spelling checker in a word processing system. Their purpose was to automate the incredibly difficult and tedious task of checking a layout for design rule violations or analyzing the performance of an electronic circuit. These tools helped the designer immensely by eliminating a great deal of mechanical work. A person, however, still had to perform all of the creative design work. Because these CAD tools were primarily mechanical in nature, they lent themselves to implementation by algorithms that were not unreasonably complicated. This is not to say that these algorithms were trivial, only that AI techniques were not required.

Test and Diagnosis

At roughly the same time that checking and analysis tools were under development, interest in the test and diagnosis area increased. Procedures for testing and diagnosing problems in SSI complexity ICs were totally inadequate for testing complete VLSI systems on a chip. Numerous algorithmic approaches were tried, but there has been relatively little success to date. Researchers are now turning to AI techniques to see if this difficult problem can be solved.

Synthesis

With reasonable performance from checking and analysis tools, research attention shifted to the general area of design synthesis and is now the most active area in the VLSI design tool domain. The idea behind synthesis is to solve the problem of automating the design process. Most synthesis systems operate in a series of steps known as decomposition and refinement as illustrated in Figure 1. Within the area of synthesis, there are two major'topics: silicon compilation, and automatic test generation.

Silicon Compilation

The term silicon compilation was first coined by Johansen4,5 to describe a

process, similar to a software compiler, whereby a textual chip description would be automatically compiled From the collection of the Computer History Museum (www.computerhistory.org)128 National Computer Conference, 1985 into layout artwork. Today this concept has been widely expanded and now divides into two broad areas: functional specifications to structural netlist compilation, and structural netlist to layout artwork compilation as illustrated in the Ydiagram in Figure 2. A structural netlist contains the information found in an engineer's logic diagram. Logic symbols are converted into a netlist format usable by the computer. The information is said to be structural because it describes which logic gates will be used and how they are to be interconnected. Information on

ASYNCHRONOUS SYSTEM DESIGN

BEHAVIORAL DESCRIPTION

REGISTER TRANSFER

MACRO CELL

GATE! SWITCH

ABSTRACT GEOMETRY

PHYSICAL GEOMETRY

DESION REFINEMENT LEVEIS

Figure 1-Decomposition and refinement steps

STRUCTURAL REPRESENTATION

PROCESSOR MEMORY SWITeK

CELLS

LAYOUT PLANNING

GEOMETRICAL REPRESENTATION.

Figure 2-The silicon compilation process

FUNCTIONAL REPRESENTATION

how it works (function) and details of the chip layout are absent. An example of a logic diagram and netlist are shown in Figure 3. Structural netlist to layout artwork compilers include common tools such as placement and routing for gate array and standard cell chips. These tools automatically perform the design steps involved in deciding where to place cells and how to route the interconnections so as to minimize chip area. Some such tools can actually perform a better job than human designers, at a fraction of the time and with no mistakes.6 Another form of layout artwork compiler generates layouts directly from a two dimensional layout language.7 Layout generators create detailed layout cells for use by a structural netlist to layout artwork compiler. In a sense the cell generator is used to build up the target machine language instructions (cells) of the compiler in terms of a micro code sequence (transistors and connections). Figure 4 shows a simple generator input and output. In some cases, generators are used to create more complex layouts such as a complete datapath for a CPU.8,9 Functional specifications to structural netlist compilers perform the task of converting abstract English language descriptions of the system's desired performance into a logic diagram. The first problem is that deciphering English language descriptions is non-trivial. This problem is avoided by inventing a constraining hardware description language (HDL) for writing functional specifications. The first

functional specification compilers performed tasks such as generating programmable logic arrays (PLA). Boolean equations were converted to structural information and then to layout artwork. Current efforts in this area are much more ambitious. The long term goals are to be able to compile a very high level HDL into logic for any type of digital system. Most research From the collection oftheComputer,History,Museum(www.co mputerhistory.org)in this area is employing some sort of AI techniques to achieve this goal. 10 Automatic Test Generation and Diagnosis There seems to be a consensus that the goals of automatic test generation and diagnosis require higher level solutions than have been employed thus far. Researchers are realizing that structural netlist descriptions contain far too little information about the "function" of the system. 11,12 With progress in the functional specifications compilation area, it is hoped that breakthroughs will be found in automatic test generation and diagnosis.

OVERVIEW OF EXISTING AI.VLSI TOOLS

It is useful to examine the VLSI tools in existence today that use AI technology. Published tools include the XCON expert hardware configuration system, the CMU-DA system, and the MacPitts, Arsenic and Palladio silicon compilers. XC ON was one of the first CAD/CAM tools to use AI technology. 13 XCON performs the difficult task of configuring computer systems for the Digital Equipment Corporation. The Impact of AI Technology on VLSI Design 129 While XCON is not involved in VLSI design, it demonstrated the use of AI technology in the engineering domain. The CMU-DA system represents a significant effort to automate the design of digital systems, including CPU and VLSI design. The project has covered many different aspects of the domain. Some software components were written along the lines of conventional CAD tools, while others struck out to experiment with AI technology. These tools, such as TALIB14 and EMUCSIDAA,15 use the OPS516 production rule system. The CMU projects probably represent the most encompassing efforts to date to explore the use of AI in the systems and VLSI design domain. TALIB is an expert system for performing the mask layout step starting from a structural netlist. It is effective on small cells with approximately 20 transistors. The production sys-. tem employs over 1200 rules to construct layouts which are about 10 to 35 percent less area efficient than layouts created by human designers. Cells at this efficiency level are not too useful and the very high number of production rules must have been difficult to collect. On the other hand the EMUCS/DAA system appears to be more useful. EMUCS and DAA are expert CPU design systems that work at the architectural or functional level. The input to these systems consists of a set of desired machine instructions, and the output is a block diagram and finite state transition table for a CPU. These systems employ only about 70 production rules to obtain acceptable results. These two systems from Carnegie-Mellon University point out that

there are some problems that experts easily solve, and others which the machine can easily solve. In the case of TALIB, the problem is characterized by a relatively small amount of data (20 transistors) and a large number of design rules (at least 1200). For EMUCS/DAA, there is more data (hundreds of machine instructions) and few rules (about 70). The silicon compiler systems: MacPitts,17 Arsenic18 and Palladio,19 are a bit more conventional because they use an algorithmic approach. The unique quality of these tools however, is that they all employ a search scheme through some abstract design space. Silicon compilers attempt to evaluate a large number of tradeoffs and thereby tryout a large number of alternative designs. Figure 5 illustrates how a silicon compiler might evaluate several approaches. This approach differs from the human design approach where only one or two alternative designs are considered. While none of these systems are yet producing competitive layouts as compared to human designs, they have the potential to do so.

USEFUL AI CONCEPTS

Approaching AI from the VLSI design perspective, one would like to extract concepts from AI technology which can be put to practical use. Some concepts are actually not new but rephrased and with the rephrasing often comes new ideas about how to use or implement the concepts.

. Computer Languages

Probably the most visible contributions of AI technology are the LISP and PROLOG languages. These languages are From the

collection of the Computer History,Museum(www.computerhistory.org)130 National Computer Conference, 1985

STRUCTURAL REPRESENTATION

PROCESSOR MEMORY SWITCH

GEOMETRICAL REPRESENTATION

Figure 5-Alternative designs in the design space

FUNCTIONAL REPRESENTATION

significant in the new programming paradigms they each introduce. In addition they allow large algorithms to be implemented in significantly less code. For LISP, a 5 to 1 improvement over PASCAL is common. Since programmers write lines of code at the same rate, independent of the language, this is a significant productivity gain.

Algorithmic vs Production Rule Driven

Many people are taking advantage of the greater inherent capability of LISP to develop more powerful algorithms. These algorithms are generally more symbolic than numerical in nature, much in the same sense that algebra is more powerful than arithmetic. The algorithmic approach is behind some of the advanced VLSI design tools such as layout generators, MacPitts and Arsenic. Production rule driven systems such as TALIB and Palladio represent a significant departure from the more conventional algorithmic approach. Proponents of each approa~h strongly believe their approach is correct. From a more objective point of view, it seems

reasonable that both approaches are useful, one better than the other in particular cases.

Knowledge Database

Many design tools were originally written in such a way that they embodied the IC fabrication technology in hard coded expressions. For example, a Design Rule Checker (DRC) might check a layout designed in 4-micron NMOS technology. When the technology was changed, say to 4-micron CMOS or to 3-micron NMOS, the entire program would have to be overhauled. It was not long before the IC fabrication technology information was put into a technology file which was read by the program at start up time. These technology files are a form of knowledge database. The formal concept of knowledge databases, however, introduces new ideas. Many "tricks" embodied in present CAD tool algorithms could be pulled out and kept in a design knowledge database. This would facilitate changes to the tool for handling different design styles.

Expert Systems

Expert systems are loosely defined as a computer program capable of performing tasks at a level equal to or better than experts in the domain of interest. Within this loose definition a number of conventional VLSI design tools could be consid~ ered expert. A more proper definition of expert systems requires the software to be based on some sort of production rule system. Yet the differences are not as great as they appear. What makes a system perform at an expert level? It usually is the number of IF-THEN conditions in the conventional programming language paradigm and the number of rules in a production system. While the numbers are on different scales, they are a metric of expertise. The benefits of the expert systems approach are that fewer rules are required as compared to IF-THEN conditions, and hence the knowledge is more clearly specified. Also, the only code in the system is the rules themselves. This introduces the notion of granularity of knowledge. The more granular the knowledge, the easier the system is to modify and extend.

Natural Language

Natural language processing holds the promise of being able to supply the ultimate user-friendly system. A major barrier to the use of VLSI design tools is the user interface. Often the designer must learn a fair amount of "computereze" to deal effectively with the host computer operating system and the individual tools. With natural language processing, a VLSI design tool would be able to deal at a more English like level. This would shorten user training time and avoid mistakes because the tool should be better at "do what I mean" as opposed to "do what I say." The current state of natural language capability supports effective program directed dialogue. The user is asked questions in English and is expected to respond with one word. User directed dialogue capabilities are beginning to emerge with limited capabilities. A natural language front end would eliminate the need for a hardware description language (HDL)

front end to the functional specifications silicon compiler. In removing the rigid constraints of an HDL, the system should do a better j{)b of capturing the sorts of vague and implied tradeoffs and constraints which engineers express in English language functional specifications.

Learning

Machine learning is an extremely attractive idea. A learning system would be able to follow the work of human experts and From the collection of the Computer,History,Museum,(www.comput erhistory.org)extract the knowledge for use on similar problems. The main attraction is the dramatic reduction in knowledge acquisition time and cost which is theoretically possible. Unfortunately, research in the machine learning area has not advanced very far at this time.

THE IMPACT OF AI

VLSI design tool developers will assimilate techniques from AI as rapidly as technology and development time permits. AI promises to bring new features and capabilities to VLSI design tools as well as improved design methodologies. It will also have limitations which will keep it from being a panacea.

Promises

Most certainly, AI based tools will make great strides forward in improving user friendliness. Starting with simple things such as program directed dialogue in the near term (three years) and moving toward user directed dialogue (full natural language processing) in the long term (ten years). The flexibility of production rule systems will enable sophisticated users to modify knowledge databases and alter tools. Because the baggage of implementation details in algorithmic systems is left behind, production rules are fairly easy to understand. The small granularity of production rules makes it more practical for a tool user to understand the IF-THEN rules and judge the impact of modifications. This will enable the user, for the first time, to modify design tools without the assistance of programmers. With more powerful programming paradigms comes the ability to create more powerful tools. Synthesis tools such as a complete general purpose silicon compiler will emerge in the long term. This tool will leverage scarce engineering resources tremendously and greatly shorten VLSI design times.

Expanded Capabilities

With long term advances in VLSI, computer hardware and AI, it is reasonable to expect performances from VLSI design tools that exceeds human capabilities. This seems quite probable, given the large amounts of data and knowledge required to design ICs. It is reasonable to assume that a machine can eventually do a better job of evaluating complex tradeoffs and selecting the best design from among many design attempts. In addition, an expert design system should be able to complete its task many times faster than a person. Eventually it may be possible to generate working chips from an English description in the time it takes to fabricate the silicon chips.

Limitations: The foregoing probably sounds a bit optimistic and may well be. Natural language understanding is still the subject of much research. Expert systems for problems with small The Impact of AI Technology on VLSI Design 131 amounts of data and a large number of rules will be developed slowly. Progress towards unperstanding learning is so slow as to virtually eliminate any chance of using learning to make expert systems acquire rules more quickly, any time in the next ten years.

CONCLUSION

From this brief survey of the VLSI design and AI fields, it is evident that AI technology will significantly alter the way VLSI design is done today. Many human design tasks will be automated, leaving designers to deal with the most difficult and obscure design problems. These advances will pave the way for major revolutions in computing hardware and AI research.