Spring 1983

Design Methodology of Very Large Scale Integration

Ankush D. Oberai
University of Central Florida

DESIGN METHODOLOGY OF VERY LARGE SCALE INTEGRATION

BY

ANKUSH D. OBERAI
B.E.E., Shivaji University, 1981

RESEARCH REPORT

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Engineering
in the Graduate Studies Program of the College of Engineering
University of Central Florida
Orlando, Florida

Spring Term
1983

ACKNOWLEDGEMENTS

The author wishes to thank Martin Marietta's VHSIC Group for the demonstration of CALMA CAD systems. Grateful thanks to Dr. Brian E. Petrasko for his inspiration, guidance and direction in conducting this research. Thanks to Dr. Donald Malocha and Dr. Richard N. Miller for taking an interest in my research. Also, special thanks to my parents for their support and encouragement.

ABSTRACT

Very Large Scale Integration (VLSI) deals with system complexity rather than transistor size or circuit performance. VLSI design methodology is supported by Computer Aided Design (CAD) and Design Automation (DA) tools, which help VLSI designers to implement more complex and guaranteed designs. The increasing growth in VLSI complexity dictates a hierarchical design approach and the need for hardware DA tools.

This paper discusses the generalized Design Procedure for CAD circuit design; the commercial CADs offered by CALMA and the Caesar System, supported by the Berkeley design tools. A complete design of a Content Addressable Memory (CAM) cell, using the Caesar system, supported by Berkeley CAD tools, is illustrated.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

CHAPTER I

INTRODUCTION

This research report discusses VLSI (Very Large Scale Integration) design methodology and Computer Aided Design (CAD) tools that support this methodology. VLSI is a statement about system complexity, not about transistor size or circuit performance. It defines a technology capable of creating systems so complicated that coping with the raw complexity overwhelms all other difficulties. As the system complexity grows larger, the VLSI design methodology dictates a hierarchical design approach, in which complex systems can be constructed by subdividing them into less complex systems, which are again subdivided as many times as necessary until the resulting systems are simple enough to construct easily.

VLSI design techniques range from CAD to design automation (DA). CAD design tools like Design Rule Checks (DRC), Circuit Extraction (CE), Static Checks and Logic Simulation help VLSI designers make the best use of the custom manufacturing opportunities. Sophisticated CAD put the entire VLSI design process on line from schematic development all the way through Pattern Generator or e-beam tape output. Most of the above mentioned design tools are unable to cope with the continuing increase in complexity in VLSI technology. This drawback makes more intelligent systems with less user interface desirable. This is the thrust of Design Automation. Hardware DA

tools have been developed and proposed which feature larger capacity and speed than their respective software design tools.

Commercial CADs like the ones offered by Calma and Applicon differ from most of the educational CAD setups, like the Caesar system, at different universities. This is because commercial CADs feature higher speed, greater design capability, sophisticated editing capability, greater system complexity, higher resolution, more friendly prompting, larger storage capabilities, user definable libraries and menus, and a higher degree of automation and system reconfiguration.

The Silicon Broker, a third identity, provides the essential services and facilities to the designer group and various foundaries, thus accelerating the growth, design complexity and sophistication of the VLSI design methodology and production.

The following chapters present trends in custom design of VLSI. This is followed by the discussion of design methodology and features of currently available design tools. The design methodology is illustrated with the design example of a Content Addressable Memory cell, along with its evaluation. This report is concluded with the discussion of the role of an electrical engineer in the improvement of the design methodology, process design and DA tools needed to cope with the continuing increase in VLSI technology.

CHAPTER II

DESIGN METHODOLOGY USING COMPUTER AIDED DESIGN (CAD) AND DESIGN AUTOMATION (DA)

This section of the report discusses the CAD/CAM concept and its basic organization. This is followed by the description of Total Design Automation. The features of Gate Arrays are discussed. The role of CAD in IC design and production is then discussed with detailed discussion starting from its evolution straight through to its role in mask design and the final IC. This section of the report is concluded by the discussion of the role of the Silicon Broker in implementing VLSI systems.

CAD/CAM Concept

CAD/CAM is an acronym for Computer-Aided Design/Computer-Aided Manufacturing. The term CAD/CAM refers to the integration of computers into the production process to improve productivity from the standpoints of yield and quality.

The heart of any CAD/CAM system features a design terminal or work station. This design terminal helps the engineer to interact with the main computer to develop product design in detail, at the same time enabling him to monitor his work constantly on a TV-like graphics display. The engineer can, during the process of design creation, manipulate, modify and refine his creation with the help

of system commands and proper response to system prompts. Having completed the design, the engineer can command the system to make a "Hard-Copy" or generate a computer tape to guide computer-controlled machine tools in manufacturing and testing the part.

During the process of design development, the computer graphics system accumulates and stores physically related data, identifying the proper location, descriptive text, dimensions and all other properties of every design element that helps to define the new part or product. This design-related data enables the computer graphics system to help the engineer to do detailed and complex engineering analysis, generate special tests and reports, and detect and flag flaws in the design before the design is sent out for manufacturing. All of these tasks are not required for every design. It depends on the purpose and complexity of the design. A typical CAD/CAM system, as shown in Figure 1, features tools which are partly software. It includes interactive graphics terminal or work station with CRT display, function and alphanumeric keyboard, digitizer and other operator input devices, the plotter, various peripherals, a communication channel to talk to the computer, one or more hardcopy devices, and in some cases, its own local CPU.

Design Automation

Total Design Automation includes all the software packages, hardware facilities and tools for verifying the design. This type of total design automation needs no human intervention, ideally. The

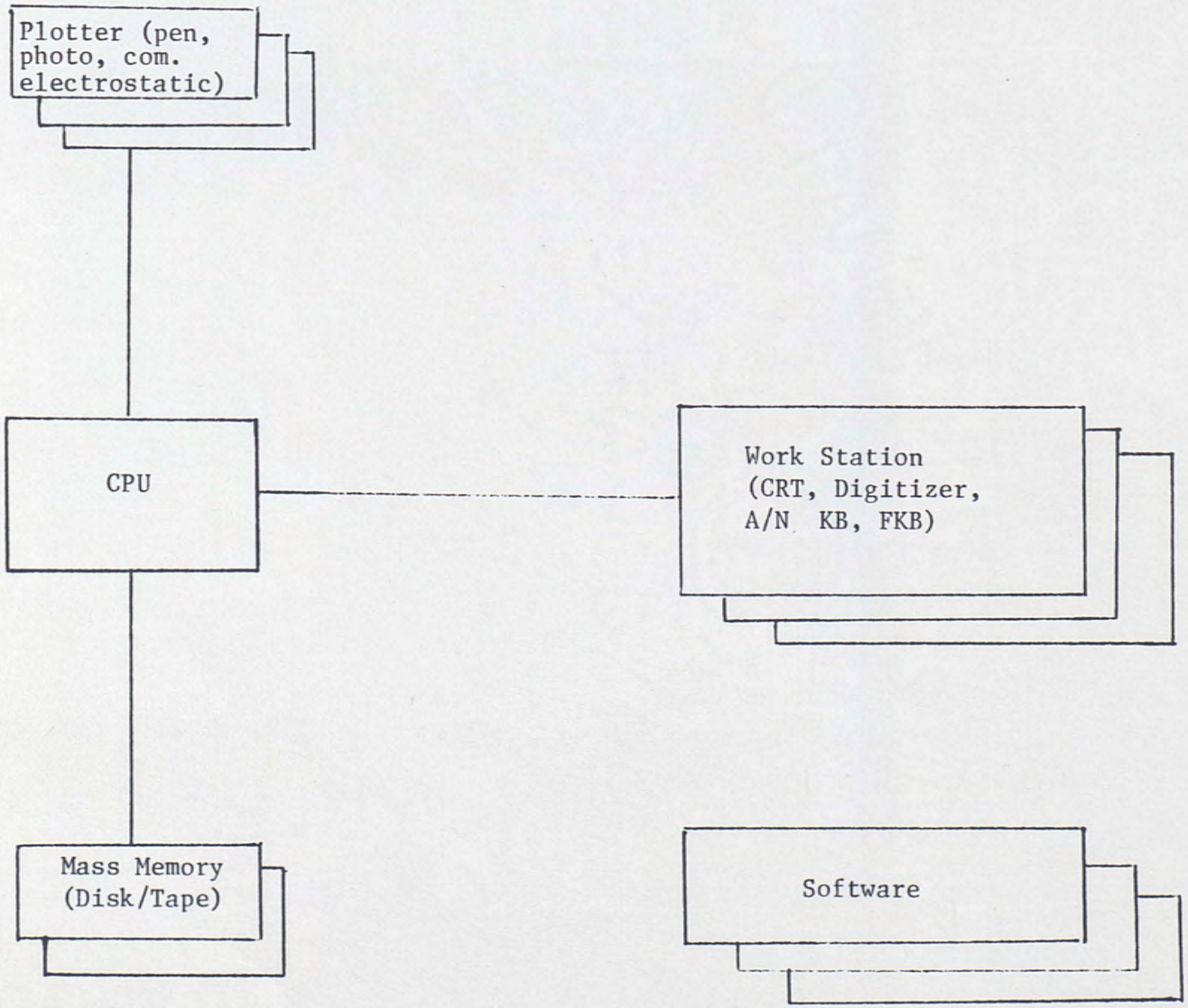

Fig. 1. Typical stand-alone CAD configuration.

input to this system is the description of the desired design and the output is the final product. In the case of VLSI, Design Automation (DA) are software systems that participate in the translation of a design to a hardware environment. Different companies like General Electric, Motorola, IBM, Harris and many others use DA which are different from each other in some respects, depending upon the type of process and design, to implement custom VLSI design. The DAs feature automatic tools to aid in design, improving quality and yield.

In the case of VLSI Total Design Automation System, human intervention is preferred at levels where decision making and expertise is essential, especially at the initial level where the design description is to be verified for all details and error possibilities. Another good reason for providing gaps for human intervention in the total Design Automation system is the fact that VLSI technology is subject to a number of changes in the processing techniques and reduce scaling. If a Total Design Automation System were to be designed at every step in this change, it would become too expensive.

In an ideal Total Design Automation system, the description of the desired VLSI design is first written and verified by the VLSI designer for all details and error possibilities. Timing, power dissipation, types and number of pads, size of the chip, type of technology are all decided upon at this level. Some DA tools can be

used to perform these computations and estimates but are less useful. A great deal of creativity and expertise is involved at this stage.

After this stage, the design description is translated into a high level hardware language, like RTL (Register Transfer Language). It forms an input to the design CAD. The CAD, in turn, carries out the process of translating the input into the logic level and then the circuit level. After a series of checks and verifications, the next translation is to the layout of the circuit. The layout is then checked for adherence to the design rules. This is performed by the design rule checker, to ensure that no last-minute errors have crept into the layout. If the layout is acceptable, an electrical network is derived from the mask information by the network extractor. This network is also checked for certain electrical rule violations and then simulated with a variety of inputs to verify correct operations. Most commonly used simulation packages are Tegas and Spice.

Before entering simulation, a number of small checks are made which help discover "trivial" errors. A Static Checker is used to examine the whole circuit once or any desired number of times and aids in automatic error tracking. This Checker gives faster and better error messages. After having performed these checks, the automated system provides facilities for simulating the design before it is fabricated. This facility features "Static" and "Dynamic" Simulator software packages. The Static Simulator checks for correct impedence ratios and reports the number, type and state of the

different transistors used. The Dynamic Simulator helps to verify the proper logical operation of the design layout. Various inputs are used and the outputs are verified against expected results. After the simulation is completed, the CAD checks for the power requirements of the chip. This is done with the help of a software package (like "Powest") which helps to estimate the power dissipation of the chip.

At any point in the design, if the designer is not convinced with the results, the design process can be repeated by variations in the associated influencing parameters. The function of the CAD terminates at the point of power estimate unless the CAD features some other facilities. The CAD outputs the layer description in a certain text file format, like CIF (Caltech Intermediate Form), Calma, or Applicon. The text file is now passed on to the CAM (Computer Aided Manufacturing), which aids in the fabrication of the chip. This type of facility, CAM, is installed in a fabrication house and includes software tools which generate outputs which, in turn, are directly used in the fabrication of the IC (Integrated Circuit). The IC is normally tested and packaged in the fabrication house.

At most universities, VLSI CAD facilities utilize the Berkeley tools and design starts at the layout level. The translation from the design description to the logic and circuit levels is done manually. The layouts are generated in an interactive mode.

Gate Array Circuits

One of the main problems faced in VLSI custom design is routing. A design engineer, with his expertise, can successfully layout the circuit in a minimum of area, obeying all the design constraints, but feels helpless at times when it comes to routing the various subsections of the complex circuit design layout. The primary source of this problem is the irregularity of the custom design circuit. The circuit irregularity sometimes makes it impossible to connect, that is, route between certain portions of the circuit. In such cases, the routing leads to non-compact circuits. This approach leads to greater delays, greater power dissipation, timing problems (owing to skewing) and need for more chip area. Such is not the case with Gate Arrays, which refer to circuits which have highly regular structure and, thus, have minimum routing problems.

Gate arrays, though very much desired, depend upon the type of the circuit to be implemented and they cannot be used in all custom designs. In the cases of memory design, in which identical memory cells are configured in rows and columns, gate arrays can be used. This is a general feature of gate arrays. A single memory cell is designed, according to the design specifications, and is iterated in a desired number of rows and columns. Thus, owing to their regularity, the gate arrays make routing easier and all the routing associated problems are solved.

Owing to the regularity in structure and easy routing, DA offers automated routing tools which take the form of software algorithms for gate arrays. Such DA routing tools route the design

automatically, thus saving time of planning and doing the routing of the layout. Like most DA tools, automatic routing tools fail to cope with the continuing increase in the complexity and get slower and error prone. In order to overcome these problems, hardware routing tools have been proposed.

Total System Design and Production Using CAD/CAM

Evolution of CAD/CAM in IC Design and Production

Initially, IC (Integrated Circuits) were exclusively used in military and avionics applications to offer advantages like improved reliability, reduced size and lower power consumption. The process of design and production was very expensive and time-consuming which made it rather difficult to use ICs for prototyping and commercial usage. The maximum of thirty transistors were fabricated on a chip.

Advances in design and manufacturing techniques gave birth to the use of CAD/CAM for IC design. The use of CAD relieved the IC designer of many tedious, non-creative tasks and enabled him to create otherwise impossible designs. The CAD also featured the ability to examine design alternatives without the delay and cost of actually fabricating experimental ICs. The complexity of ICs increased from 30 transistors to tens of million transistors.

CAD Supported IC Design

Integrated circuit design and construction has three main and distinct components: Circuit Design, Process Design and Mask Design.

All three components are interrelated and each depends upon the other two. Figure 2 shows the interrelationships of the three design components.

Circuit Design

Circuit design involves the preparations of schematics and logic diagrams describing the manner in which the circuit function would be implemented. Initially, this was done manually and involved a large number of draftsmen. It was very time-consuming and inefficient as some modifications needed totally new drafts of the design.

The present use of CADs for circuit design has partly automated and totally streamlined the circuit design process. CAD for circuit design features an interactive graphics terminal which interfaces the designer comfortably with a computer. It provides the engineer with a terminal with a CRT display. This is used as a "scratch pad" to manipulate circuit elements, develop the design, and view the results. On the completion of the final design, total design data is stored in the computer and is readily retrieved for examination, modification or processing for further analysis. When the circuit designer has completed the design, he can command the system to generate a "Hard Copy" of the design and this is made possible with high speed plotters. Figure 3 illustrates the general circuit design chart.

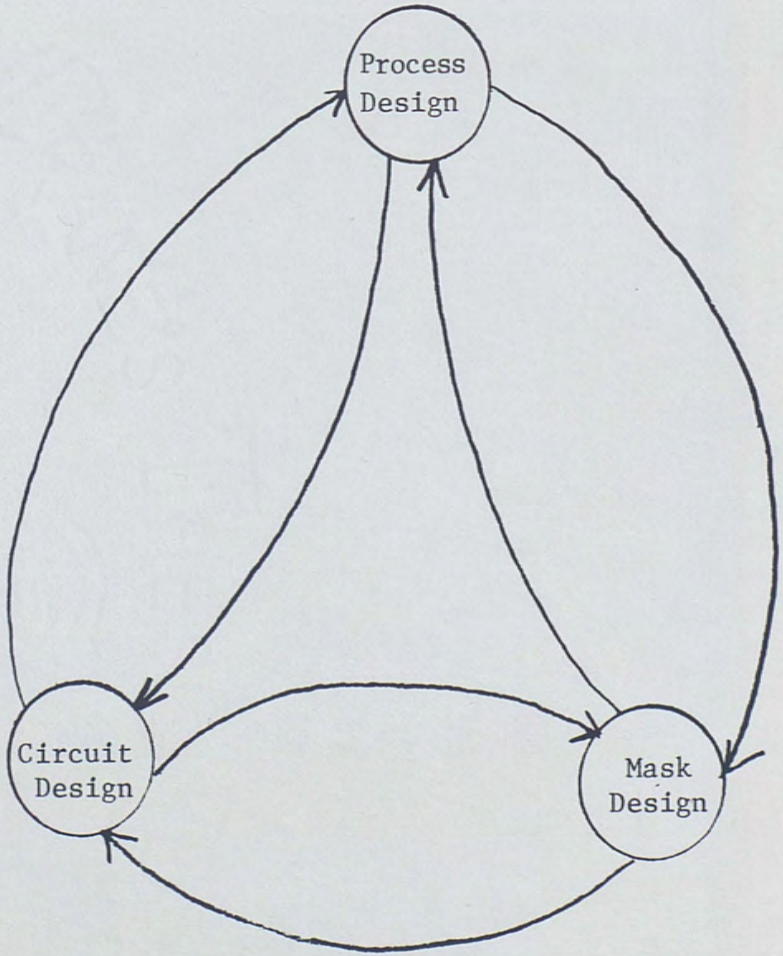

Fig. 2. Interrelationship of design components.

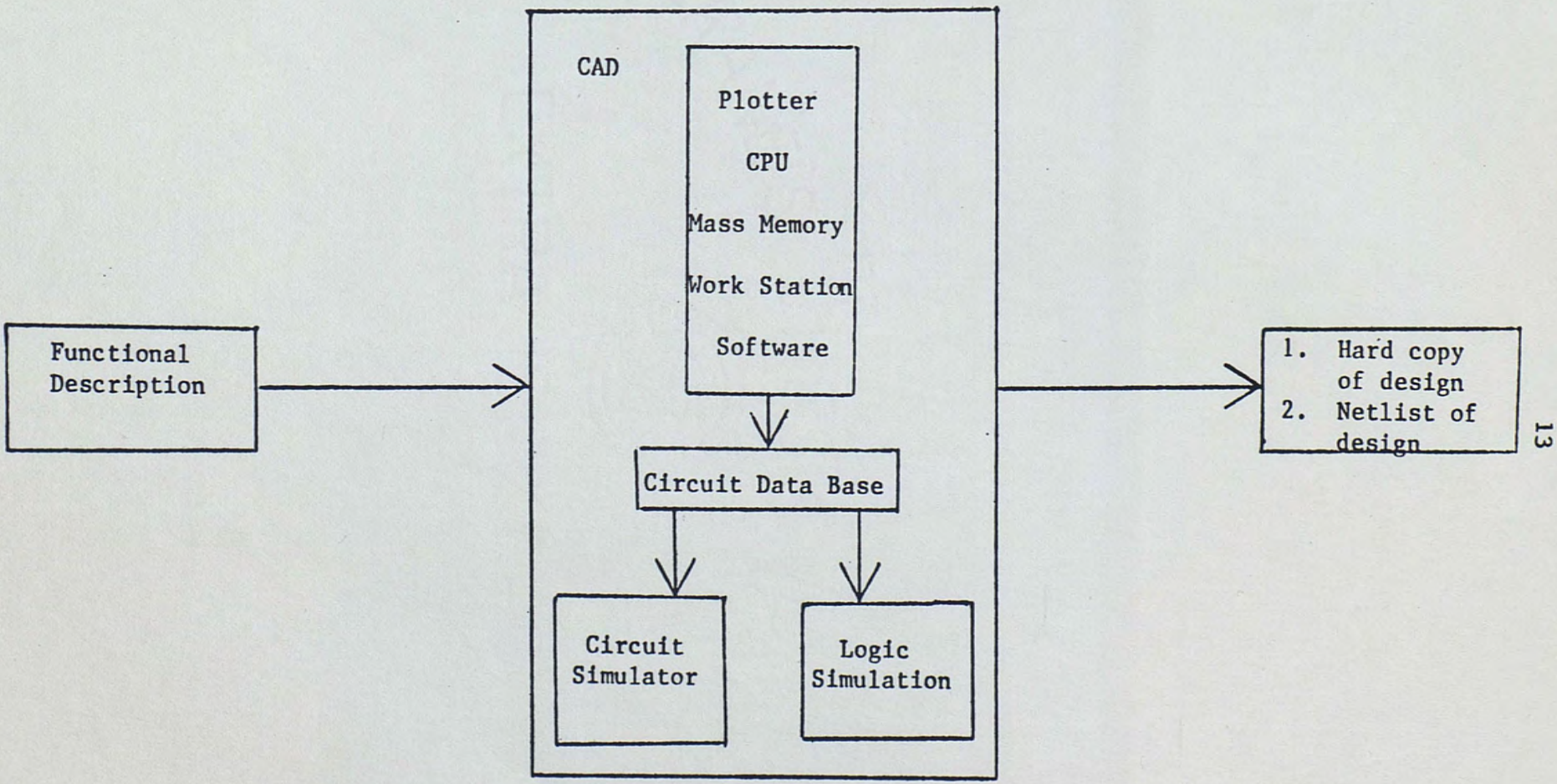

Fig. 3. Circuit design chart.

Process Design

This component of IC design deals with the process details used in manufacturing. The primary consideration in processing are the temperature of the wafers, the concentration of dopant atoms and effects of previous and subsequent processing steps.

CAD used for process design helps to compute the values for all parameters applicable to the process. The data base of process design is an input to a mathematical model to accurately portray what is happening in the wafer. The CAD used for process design emphasizes simulation programs which provide user interaction as well as graphical and statistical output. The CAD equipment for process design features a graphics terminal which enables the process engineer to view exactly what is taking place both in the wafer and on the wafer.

All the above facilities featured by the CAD enable a process engineer to contour a process for optimal result and to foresee problems that may occur in mask design. Figure 4 illustrates the process design steps.

Mask Design

This component of the IC design realizes the circuit design on a wafer. CAD that supports this design step partly automates and totally streamlines the mask design process.

In the mask design process, a number of masks used to define a circuit must align exactly with one another, must contain all the

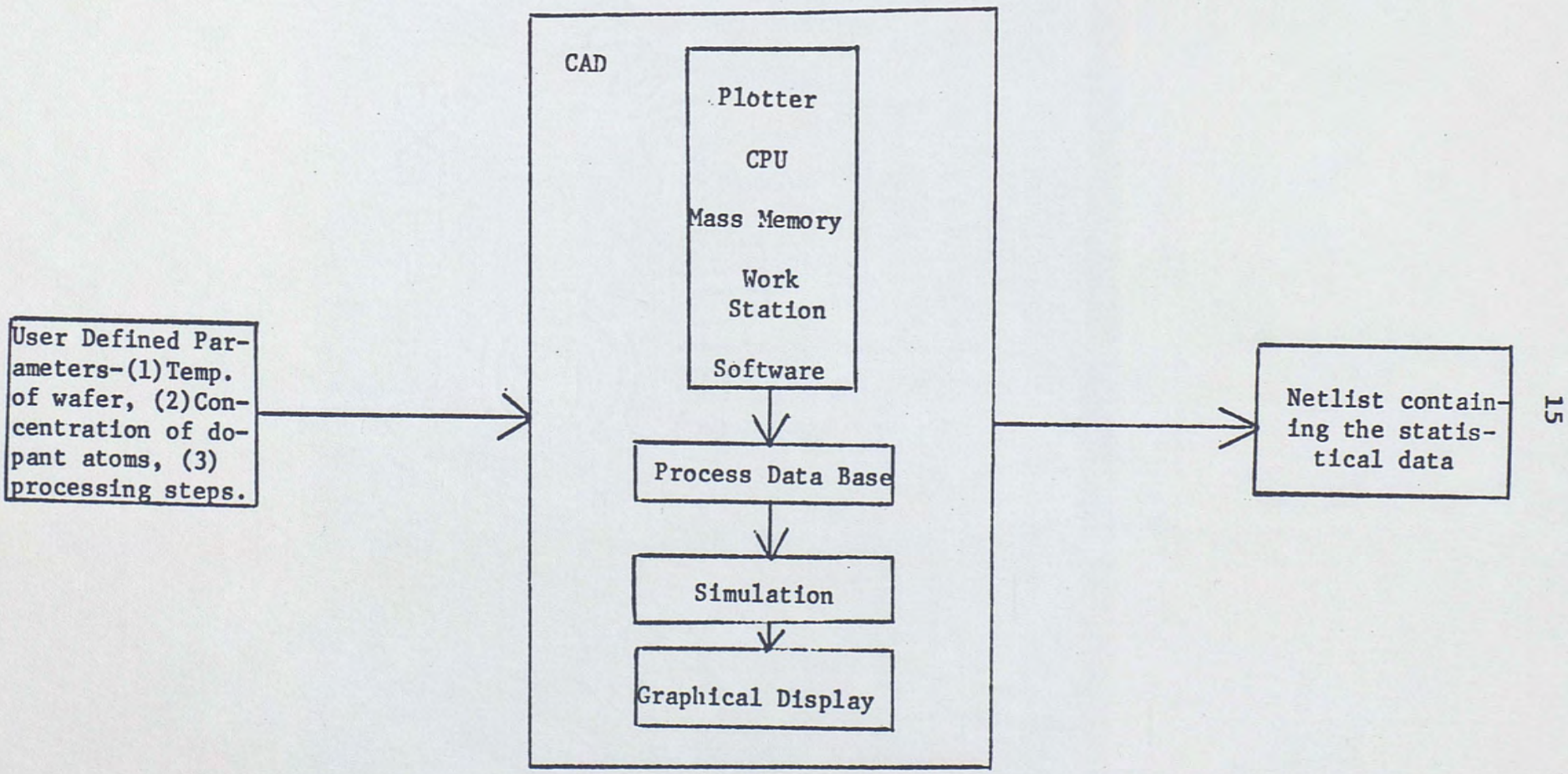

Fig. 4. Process design chart.

regions of dopant atoms, precisely sized and located, and must be designed and manufactured with a minimum of delay. The mask designer is relieved from most of these requirements by CAD. CAD for mask design features software tools which preserve the electrical and logical functions of the circuit, observe all process related constraints, and minimize the area the circuit will occupy.

Similar to the previous two design steps, the CAD for mask design features an interactive graphics terminal and an electronic pen to help create designs. Mask related data for a single component can be stored in a well defined "Stack" or region and can be used again and again throughout a circuit design.

The mask design data base contains a massive amount of data and, owing to the CAD's ability to manipulate this data in a very short time, designers can bend, twist or otherwise manipulate the design in real time to optimize placement of all regions. The sophisticated software for design rules checking guards against errors. The designer is prompted at every step, thus, the CAD ensures that the desired circuit does not violate process-imposed constraints before it is manufactured.

During the mask design step the netlist from the mask design data base and the netlist from the circuit design data base are compared and any electrical discrepancies are noted. The netlist extracted from the mask design data base is also used to check for any topological errors. Figure 5 illustrates the mask design steps.

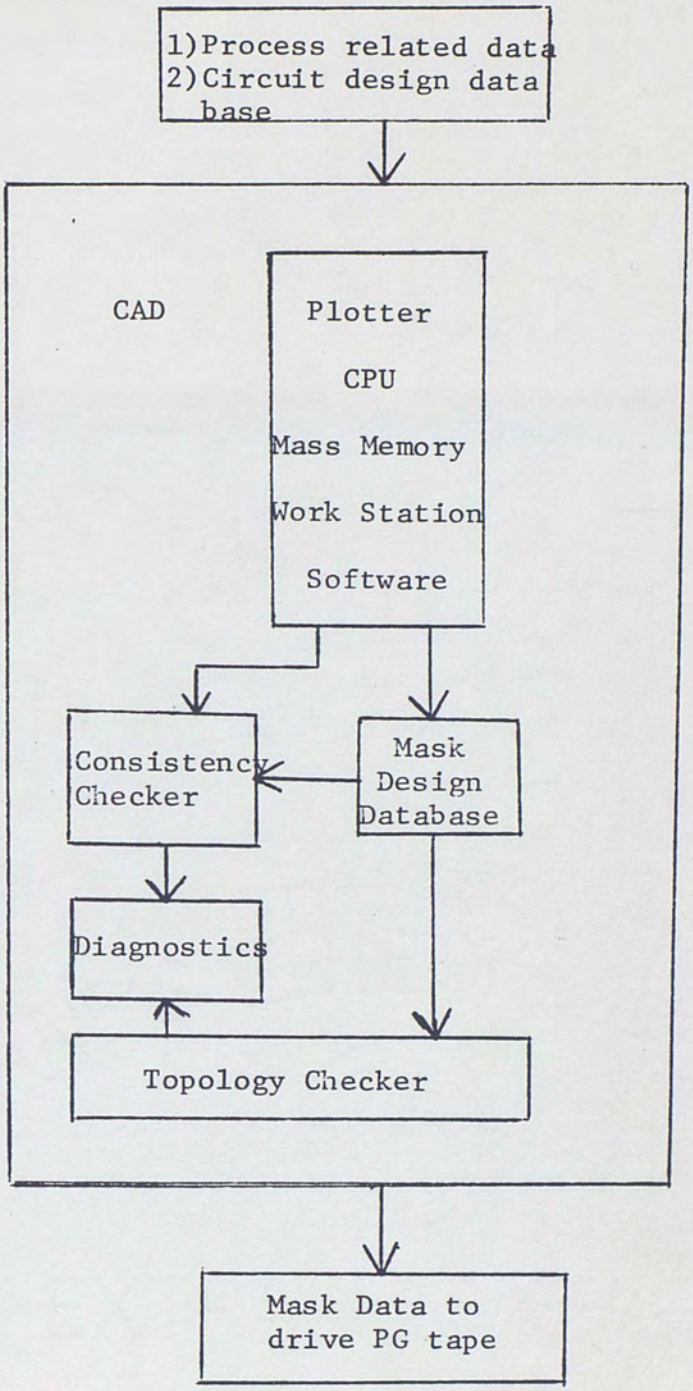

Fig. 5. Mask design chart.

The final step is the IC construction and testing. Here, all the three component netlists play a role. The process design component determines the various parameters to be used at all steps of fabrication. The mask design component netlist forms an input to drive the PG (pattern generation) machine, that paint the regions in the mask design on a mask. The circuit design component netlist forms input to the program that tests the circuits at completion of fabrication.

CAD supports the creative nature of man and, together, they turn out a product that neither is capable of alone. CAD also helps to conduct product development experiments that are cost-effective and yield accurate results. There is no doubt that CAD provides superior designs and better yield.

Commercial CAD Services

A neutral third party service function has arisen in the design and realization of custom VLSI. The Silicon Broker interfaces with both the design community and the manufacturing community, mask shops, and silicon foundaries in the implementation of VLSI systems. The primary services and the advantages offered by the Silicon Broker are discussed below.

The Silicon Broker acts as an interface between the designer and processing vendor (foundary services) to oversee the implementation of the designer's circuit. This is shown in Figure 6 in a

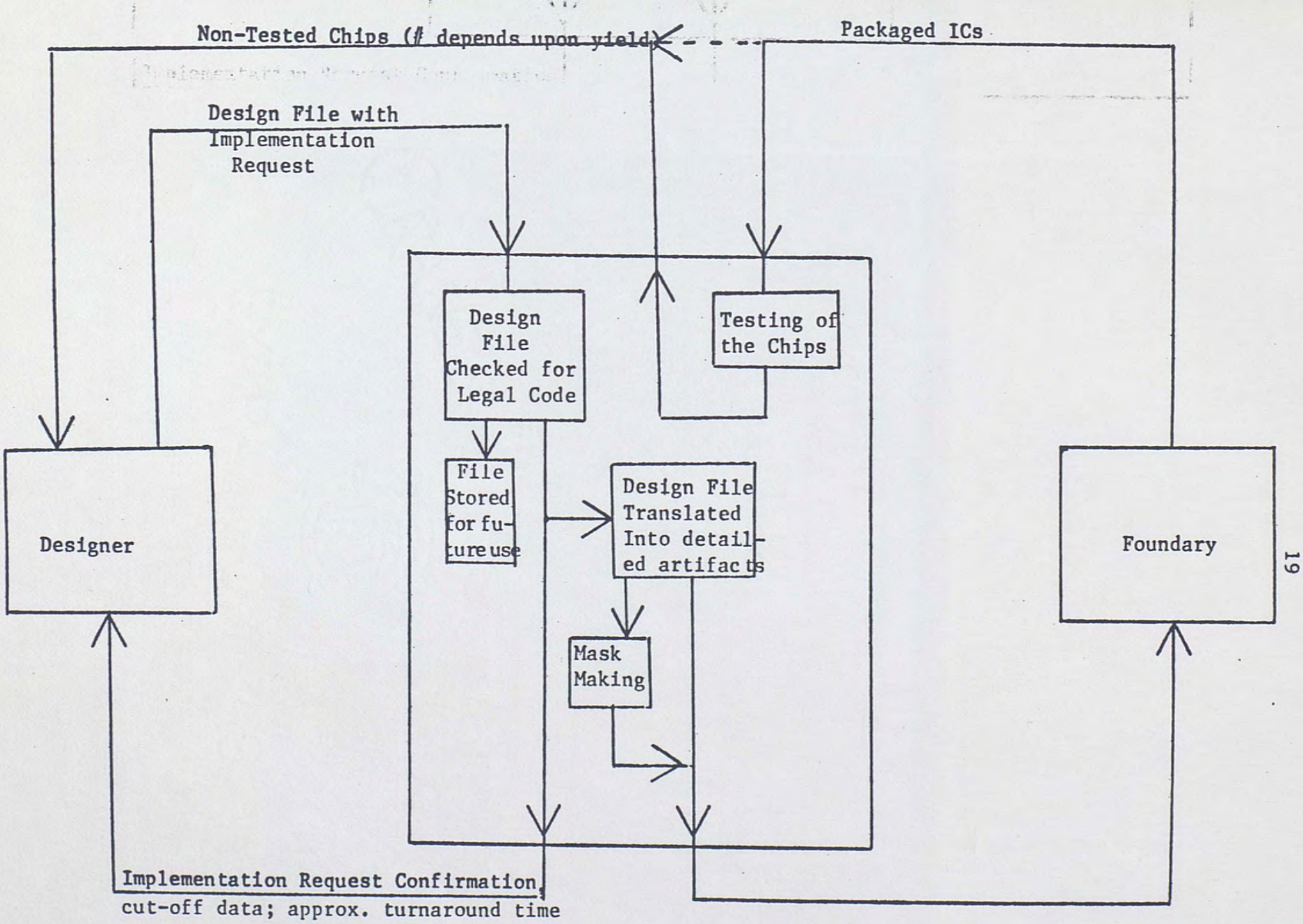

Fig. 6. Interface.

block diagram form. The primary goal of the broker is to minimize the designer's involvement with process interfacing details and to allow the designer to concentrate on functional chip design. As a result of this, the designer sees the implementation system as something to which he sends designs and receives packaged chips. The primary function of the broker is that of information management. In order to handle this large amount of information, the broker services use standardized formats and all communications with the designers are done over an electronic network with electronic mail and file transfer.

The designer generates his design file in a standardized format, like CIF, Calma, Applicon, etc., and sends it to the broker along with an implementation request message. The broker receives the design file and checks it to ensure that it is in legal code format, and determines the manufacturing parameters. The design file is stored away for future use and the designer is notified that the request was successful. Again, all message processing is automatic unless the user asks for human intervention.

The broker acts as a translator. He translates the standardized circuit design, submitted by the designer, into the detailed information required by the fabrication vendors. The broker accepts design requests from different designers and places them in one fabrication run. This reduces the cost per design dramatically. The masks are either provided by the broker, or are made in the foundary. After complete fabrication, the good wafers are sent out

for packaging. These packaged chips are returned to the designers by the broker or the packaging vendor, depending on the understanding. The number of chips returned is based on a simple yield estimate, determined by the broker. Enough chips are returned to be sure that at least two will work. The unbonded chips and the bonding diagrams are kept on file in case the designer needs more chips.

The Silicon Broker provides a number of other services in order to cover all aspects of VLSI implementation from design support through production of custom VLSI circuits. In this way, the Silicon Broker is set up as a central organization to support the needs of a VLSI designer. The flow diagram in Figure 7 shows the various services offered by the Silicon Broker to the VLSI designer. It can be seen from this flow diagram that the broker provides the services that the designer will need to take his design from concept to reality yet allowing the designer to concentrate fully on functional chip design.

The broker's services make low cost, fast turn-around prototyping a reality and also supports low volume production needs. The design services provide the needed aids for the evaluation of designs. The CAD tools provide for design simulation and layout. The production services free the designers from the intricacies of interfacing between various vendors. The broker's efforts in standardizing design rules, process parameters, library cells, and test structures benefit both designers and process vendors by maximizing

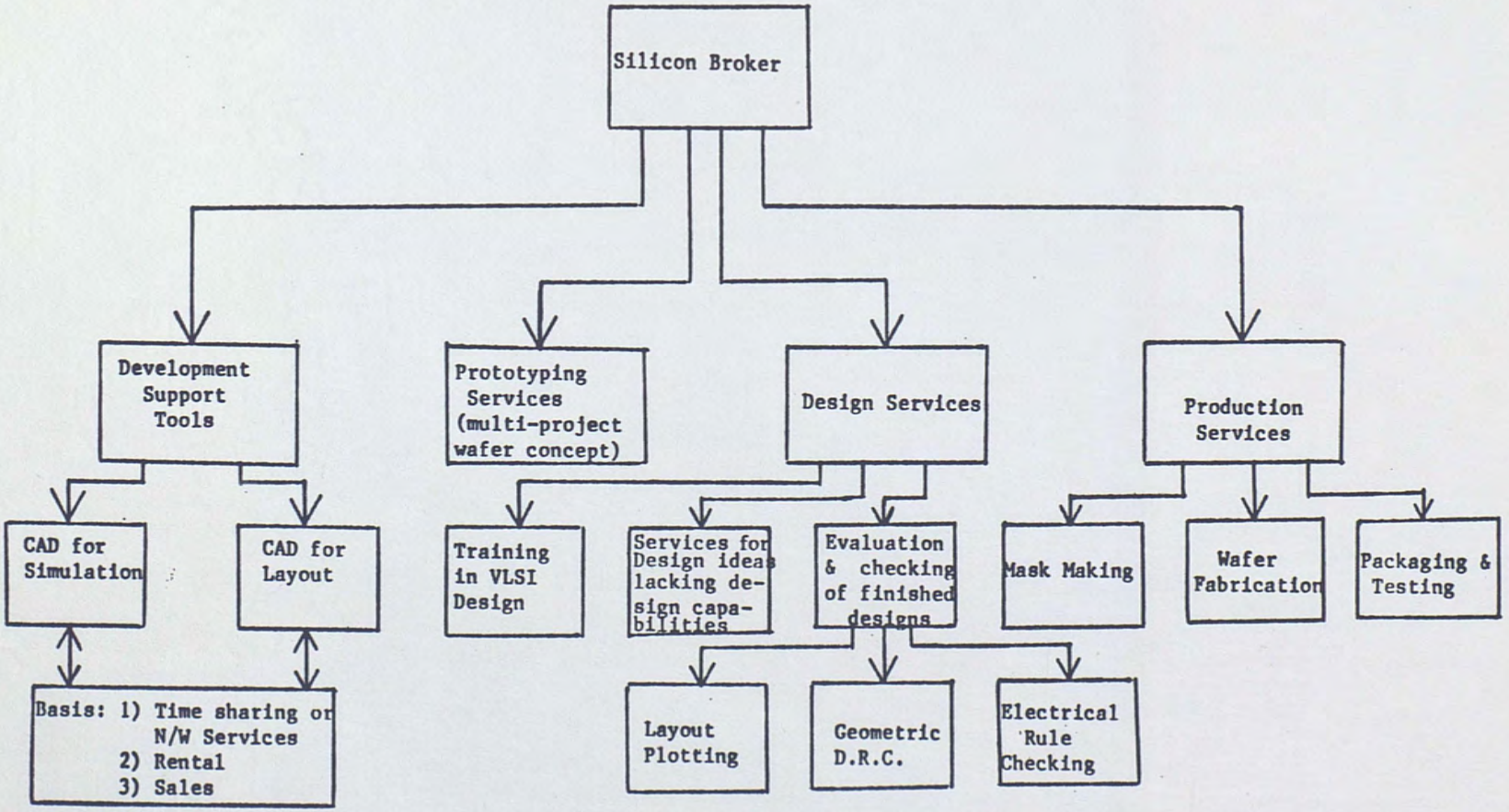

Fig. 7. Services offered by silicon broker.

the interface compatibility between the two groups. Due to the fracturing of the design and processing functions, the requirement of the IC designers to be well-versed in device physics is removed. As a result of this, large number of computer scientists, logic designers, and architects are now participating in VLSI design. This is in addition to electrical engineers.

Most importantly, the Silicon Broker concept helps integrate the design and processing functions. The role of the Silicon Broker is instrumental in bringing VLSI to the small user.

CHAPTER III

REPRESENTATIVE CAD SYSTEMS

The spectrum of CAD for VLSI extends from software tools which can be placed on a general purpose minicomputer to special purpose integrated hardware and software systems. The Caesar System is representative of the former and Calma is representative of the latter. These two spectrum ends are discussed in the following section. The design procedure using common notational levels and forms is also discussed.

The Caesar System

Caesar is one of the many interactive systems for creating and modifying VLSI circuit designs. It is a set of software tools, often referred to as the "Berkeley Tools". It includes a geometry editor enabling one to paint pictures of VLSI circuits and to combine pictures hierarchically into larger designs. It has no intelligence built into it to understand any design rules, electrical properties or even connectivity. It is based on Mead and Conway style of design and produces Caltech Intermediate Form (CIF) descriptions, which are suitable for chip fabrication.

Caesar runs under the 4th Berkeley Distribution of VAX Unix. It features two screens, called the text display screen and the

graphics display screen. The text display screen takes the form of any standard CRT terminal which has the capability of running the screen editor. It is from this terminal that Caesar is evoked. A command menu and several statistics about the chip design are displayed after having typed the proper commands at its keyboard. The graphics display screen is used to display a piece of the circuit being designed, in color. The graphics display features an AED 512 color display with eight memory planes. It is connected to the VAX through its serial line port and the connection runs at 9600 bauds or faster. A graphics tablet attached to the AED display is featured by Caesar versions 4 and later. The tablet may feature four-button cursor. Caesar can be run without a graphics display, also.

The paper, named "Editing VLSI Circuits with Caesar", by John Ousterhout, of the University of California, illustrates the various editing capabilities and commands used for Caesar. As previously mentioned, Caesar is evoked from the text display after typing the appropriate command for Caesar.

If the CAD facility features a bit pad with the AED terminal, then an extra effort is encountered in order to enable the Caesar to read characters from the AED's port. At the University of Central Florida (UCF) and Berkeley, the job "sleeper" has to be logged in on the AED terminal. This job does not require a password. It (the sleeper) is an interface between Caesar and the terminal. When Caesar begins the text screen is filled with characters, as shown in

Figure 8. The text display, as can be seen from Figure 8, has three sections: the short commands, the long commands and statistics display. They appear in the lower right, lower left and the top portions of the display, respectively. The short commands are displayed along with brief descriptions of their functions. Apart from their functional differences, the short and long commands differ in the way they are evoked. Each short command is evoked by typing a single letter on a keyboard whereas a long command is evoked by typing a colon (:) followed by a line of text, followed by a carriage return. The line of text is typed entirely in lower case and it contains the name of a command and any parameters that are needed by the command. The statistics display displays the name of the file being edited, the current cell, if any, being used, the coordinates of the cursor, the size of the editing file and current cell, the scale value, grid spacing and the types of layers visible on the screen at a particular time. These statistics inform the designer as to the area being used and other dimension-related details.

Designs can be created, modified, viewed, saved, arrayed, labelled, and re-edited with the help of the approximate editing commands. Caesar features a very helpful short command named undo. Whenever any modifications are made to the cell data base by Caesar, it saves enough information to undo the effects of the most recent modification. This is done by typing the short command "u" (undo), which voids the last change made.

```
Editing file:                    CAM1                0      15      0.5*   120
Current cell:                    COMP                0      15      0.5*    40
Current view:                                        0       0      15  *   143
Cursor (alignment:  1)                             -16      10      32.5*     2
Scale:                  1 visible layers:pdmicol   Grid Spacing     1*        1
```

Long Commands:			Short Commands:	
Align	GRIPe	RETurn	a - Cursor left	^L - Redisplay
Array	Height	SAvecell	s - Cursor right	^F - Find cell
CIf	Identifyc	SEarch	dw - Cursor up	^ - Find parent
CLOAd	Label	SUbedit	fz - Dursor down	C - Expand cell
CLOCKwise	Mark	Technolog	j - Cursor narrower	c - Unexpand cell
COLormap	MOvecell	Upsidedow	k - Cursor wider	V - Larger view
COPycell	PAth	VIEw	1i - Cursor taller	v - Smaller view
CSave	PEek	VISiblela	;m - Cursor shorter	G - See grid
Deletecel	POpcursor	WIdth	1234 - Move picture	g - Don't see grid
EDitcell	PUShcursor	WRiteall	left, right, up, down	E - Expand area
ERasepain	PUT	YAnk	5 - Center view on LL	e - Unexpand area
Fill	Quit	YCell	6 - Center view on UR	! - Shell command
GEtcell	RESet	YSave	7 - Move LL to center	+ - Larger scale
GRIDspaci			8 - Move UR to center	- - Smaller scale
			9 - Center cursor LL+UR	u - Undo
			^P - Paint under cursor	

Fig. 8. Caesar text display.

Caesar features a rectangular cursor as the graphics cursor and this type of a cursor selects an area, not just a point. A blinking black-and-white crosshair also appears on the graphics screen and its position is controlled by moving the tablet cursor around on the tablet. The main purpose of the crosshair and tablet is to position the graphics cursor. The graphics cursor, when of zero size, appears as a crosshair rather than a rectangle.

Pre-designed cells and cells from the library can be added to any current editing file providing a hierarchical capability. A cell from the library cannot be edited for deletion of any of its section. Quite complex designs can be implemented using Caesar with little insight into the reason for the design rules and fabrication details. Caesar is a highly recommended system for beginners.

The editing commands are few and are displayed on the text screen, avoiding the burden on the designer to memorize them. Apart from these and the previously described advantages of Caesar, it has a number of drawbacks.

The main drawback of Caesar system is that it is slow. The reason for its slow speed is the general purpose nature of the host processor and the use of the processor in a shared environment. The Caesar system is hard to reconfigure for group projects. It does provide the facility to transfer the designs between different designers, but does not feature the capability to edit them in the transferred file.

The capacity of the memory is not very large and this restricts the storage of a large data base, thus discouraging very complex design projects. Caesar lacks the intelligence to sense electrical violations in design rules, electrical properties, and errors in connectivity. This slows the design process. The design to be checked for the above constraints has to be saved and then checked using other software tools in the CAD system which is under the UNIX operating system. The design rule checker, the circuit extractor, the static and dynamic simulator that accompany the Caesar program add to its disadvantages as they prove to be slow and inefficient in certain cases. The design rule checker gives redundant errors, the dynamic simulator, at times, fails to simulate detailed sections of a complex design, especially if the design is dynamic. Caesar's AED display terminal is prone to electrical disturbances and it forces the designer to exit Caesar during such occurrences.

CALMA Systems

This section of the report discusses a few CADs offered by CALMA, a subsidiary of General Electric. The services provided by this company are intended for large users, which feature a line of machines or products, two of which are CIRCUITS and CHIPS.

"Circuits" - CALMA Logic Designing CAD

The purpose of the CALMA CAD "Circuits" is to increase engineering productivity and help the logic designers to manage complex VLSI designs and digital systems. It also helps to integrate CAD into the

overall design cycle. It has a number of special features. These include:

High-Speed Network Links

Circuits features a high-speed network that links individual nodes, giving each user access to a reservoir of shared information and facilitates team problem solving efforts. The Circuits network can support up to 22 optional nodes. Each node offers its user mainframe capabilities, a large virtual address space, virtual memory management, large disk capacity, a high-quality display system and the benefits of the high-performance network.

Dedicated CPU

Each node employs a dedicated CPU. As a result, the engineers do not waste time waiting for the system to process demands from many terminals. Active nodes can be placed up to 1000 meters apart, thus allowing for more and wider working area. Circuits can easily be reconfigured. A ring of 20 connected nodes could be configured into two rings of eight and 12 nodes for two new smaller design projects, increasing system flexibility. Each node CPU has 1 megabyte of memory. Expandable memory capacity allows one to add up to 3 megabytes of main memory per node, further increasing system performance. A typical node supports one 19-inch black-and-white graphics display screen for fine graphics resolution, one alphanumeric keyboard, and one 11" x 11" tablet with four function pack for accurate cursor control.

Circuits offers multi-window processing, full screen viewing and user-definable windowing capabilities.

Operating System

Circuits features AEGIS Operation System (AOS) which allows the engineer to access multiple windows. These windows can be moved, changed in size or overlayed. Additionally, AOS facilitates performance of up to 16 concurrent processes without slowing overall system performance.

Interactive on-screen help files, on-screen menus and on-screen error messages prompt the engineer so that the system is easy to learn and fast to use. These advantages minimize the need for command memorization.

True Hierarchy

Circuits features a true hierarchical data base which gives freedom for top-down and bottom-up design freedom and access to nested components. The number of levels which can be added in the hierarchy are limitless and new levels can be added at any time. Connectivity for logic symbols is maintained during editing, resulting in fewer connectivity errors and greater speed, especially since component connections are never broken.

Format Features

Circuits features automatic block symbol creation for fast creation of parts. The system allows the engineer to create a

standard symbol library which provides a source of consistent documentation for the entire design team. On-screen rules allow quick, accurate, visual component sharing while pan and zoom features offer fast data display.

Text Editing Feature

Circuits text editing feature creates ASCII files for engineering documentation on text entry, allowing quick access to documentation by other members of the design team.

Automatic NETLIST Generation for Logic Simulation

Circuits also dramatically improves the speed and accuracy of netlist generation for logic simulation. Each node automatically generates and down loads netlists to simulations such as SPICE and TEGAS, by-passing the tedious, error-prone manual method of keying in and coding.

CALMA CAD Chips

The purpose of the CALMA CAD "chips" is to put the entire VLSI design process on line from schematic development all the way through pattern generator or e-beam tape output. It has a number of special features. These include:

Data Base Used

GDSII design data management system offers the precision to resolve graphics to one part in 4 billion. GDSII gives additional

advantages of color graphics terminals for better design discrimination. Chips incorporates an expandable data base that allows addition of user-defined properties to the standard graphics information.

Disk Storage Capacity

300 megabyte disk storage system is the largest on today's market. In addition, up to four 300-megabyte disks can be used. Thus, more drawings can be stored as structures in libraries for use over and over again. This storage system provides the room and feasibility to handle data in the most efficient manner for any particular operation.

Edit-In-Place Facility

This facilitates design changes, cuts into half the editing time as compared to other graphic systems. This technique os Edit-In-Place provides the fastest, easiest access to data, simplifying the designer's job, dramatically boosting his overall productivity.

Vector Memory Display (VMD)

This feature of Calma's Chip offers many unique design advantages. The designer may zoom in or out, and pan across a design for faster, more discriminate viewing. VMD also provides multi-view capabilitiys so that one can simultaneously view a chip and any selected segment of it.

Calma Memory Mapping and Disk Operating System

This provides a multi-task capability by separating foreground and background operations. This allows multiple background operations to run concurrently while designing in the foreground.

Library Facilities

The designer can assemble drawing in his own working library just as a draftsman creates drawings from ready-made templates. Each designer's working library usually contains the entire drawing of a large chip. Libraries may be cross-referenced so that the pieces of previously designed cells can be reused in other layouts or chip designs.

Packaging Density

Calma Chips has the precision to pack maximum circuitry into minimum space. The GDSII data base provides a grid with over 4 billion data points in each axis, providing a "drawing board" to accommodate the largest, densest circuit drawing.

Calma's chips offer many more features for speeding throughput, controlling costs and quality, and achieving substantial productivity gains. Design rule checks, high-speed electrostatic plotting and output formatting tailor Calma's chip to any particular operation.

Chips Design Station

The Calma Chips design station features two displays. A graphics screen displays the drawing, an alphanumeric screen echos all command input, telling the designer exactly what task the system is performing at any given time. The alphanumeric screen displays x/y coordinate information, the data, time and station number. It prompts the designer during interactivity, it displays error messages in bright characters accompanied by an audible alert, it gives status reports on all grounds in the system, digitizing status, operating modes and library drawing status. There are help files to assist designers, and they may edit these files or create their own application notes and procedures. This dual display keeps a designer in constant communicaton.

Input to Chips is by electronic pen and table using on-screen command menus. A standard on-screen menu is provided and is designed to support IC layout; but the system allows for the designers to create their own custom menus to fit their unique process requirements. Different design stations may use different menus. Up to four menu levels can contain as many as 512 commands each. Command menus are displayed on the graphics screen, allowing the designer to enter commands without looking away from his drawing. The command sequences frequently used could easily be compiled into a GPLII (Graphics Programming Language) program and assigned to a single menu button. Calma chips offers a choice of graphic displays including the black-and-white master display and the medium high-

resolution display (RC500). RC500 color makes work easier to see, assigning different colors to different work layers improves discrimination and reduces eye strain, circuit viewing is less confusing.

The system gives 4,096 different color choices, any 7 of which may be displayed at one time. The RC500's microprocessor controller handles display functions, unburdening the main CPU. Also, refresh graphics technology allows selective erase without time-consuming repainting of the entire screen, saving both operator and CPU time.

Calma system enables designers to quickly create drawings right on the screen using chips commands instead of pencils, mylar and erasers. Display grids and Calma's display ruler permit precise free-hand geometric placement. The system minimizes errors by prompting the designer to complete unfinished shapes.

Typical CAD Design Procedure for the Circuit Design

This section of the report discusses the different levels of symbolic representation and design procedures which are common to CAD for MOS circuits and subsystems. The various levels of symbolic representation are illustrated using a NAND gate, shown in Figure 9, as an example.

As the first step in the design methodology, the design description is properly scrutinized for details and various required facilities, like memory, shift register, etc., are decided upon. The description is then translated in different levels of symbolic representation depending upon the requirement. There are different

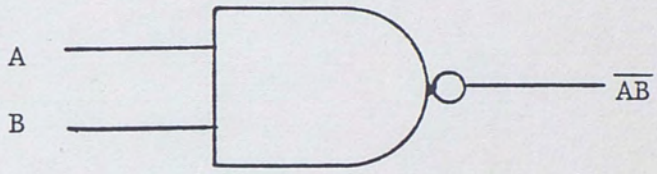

Fig. 9. NAND gate logic symbol.

levels of symbolic representation: stick diagram, circuit diagram, logic symbol, layout geometry and mixed notation.

A stick diagram is a color-coded graphical representation of an integrated circuit layout. This diagram shows as much information as a detailed layout. A stick diagram can be shown with important circuit parameters, such as the length-to-width ratios, if necessary. The main reasons for the color codes used in the stick diagram representation are for clarity and layout information. The color coding in the stick diagrams is the same as in the layout geometries: diffusion and transistor channel are symbolized by green color, ion implantation and depletion mode transistor by yellow, the red color symbolizes polysilicon, blue indicates a contact cut and the buried contact is symbolized by brown color. Figure 10 shows the stick diagram of a NAND gate along with the respective length-to-width ratios.

Circuit diagrams or logic symbols are used to represent MOS circuit when the details of neither geometry nor topology are needed. Figures 9 and 11 represent the logic symbols and circuit diagram, respectively.

In some cases, a mixture of several levels are preferred in one diagram for convenience. The most commonly used mixture is (a) stick diagram in portions where topological properties are desired to be illustrated, (b) circuit symbols for depletion mode transistors or pullups, and (c) logic symbols for remaining portions of the

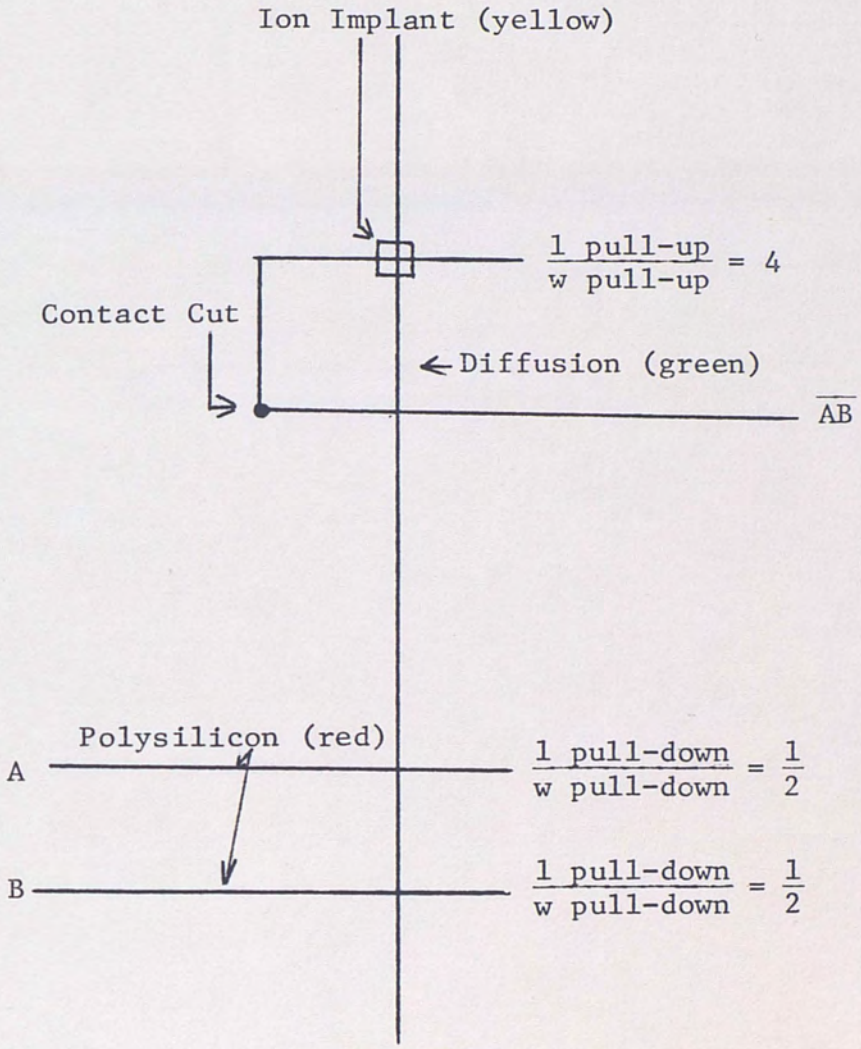

Fig. 10. NAND gate stick diagram.

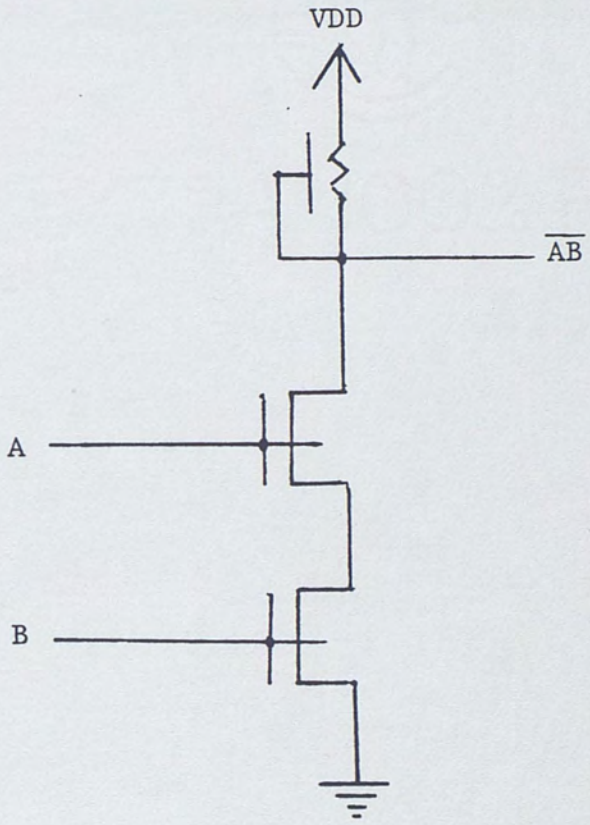

Fig. 11. NAND gate circuit diagram.

circuit or system. This type of representation is called mixed notation. Figure 12 illustrates the mixed notation of a NAND gate.

The stick diagram is transformed into circuit layout considering all the design rule constraints discussed in a latter section of this report. Figure 13 illustrates the NAND gate layout geometry. Layout geometry is generated automatically from the circuit description in a VLSI design automation system.

The best way to proceed for a beginner is to first represent the design description at the logic level, then translate the logic level into the circuit level. The circuit diagram can then be transformed into the stick diagram or the mixed diagram, as preferred. The final presentation, the layout, can then be created from the mixed or stick diagram presentations.

After the creation of a complete layout, the source file (usually stored in Caltech Intermediate Form) is checked by the design rule checker to ensure that the layout is free of all errors. After the layout passes design rule checks, an electrical network is derived from the mask information by the network extractor. The network extractor checks for violations of certain electrical rules (Baker 1980).

The next design step, after circuit extraction, is simulation. There are two types of simulations: static and dynamic; and are performed by the static and dynamic simulators, respectively. The static simulator performs basic checks, on the extracted circuit, to

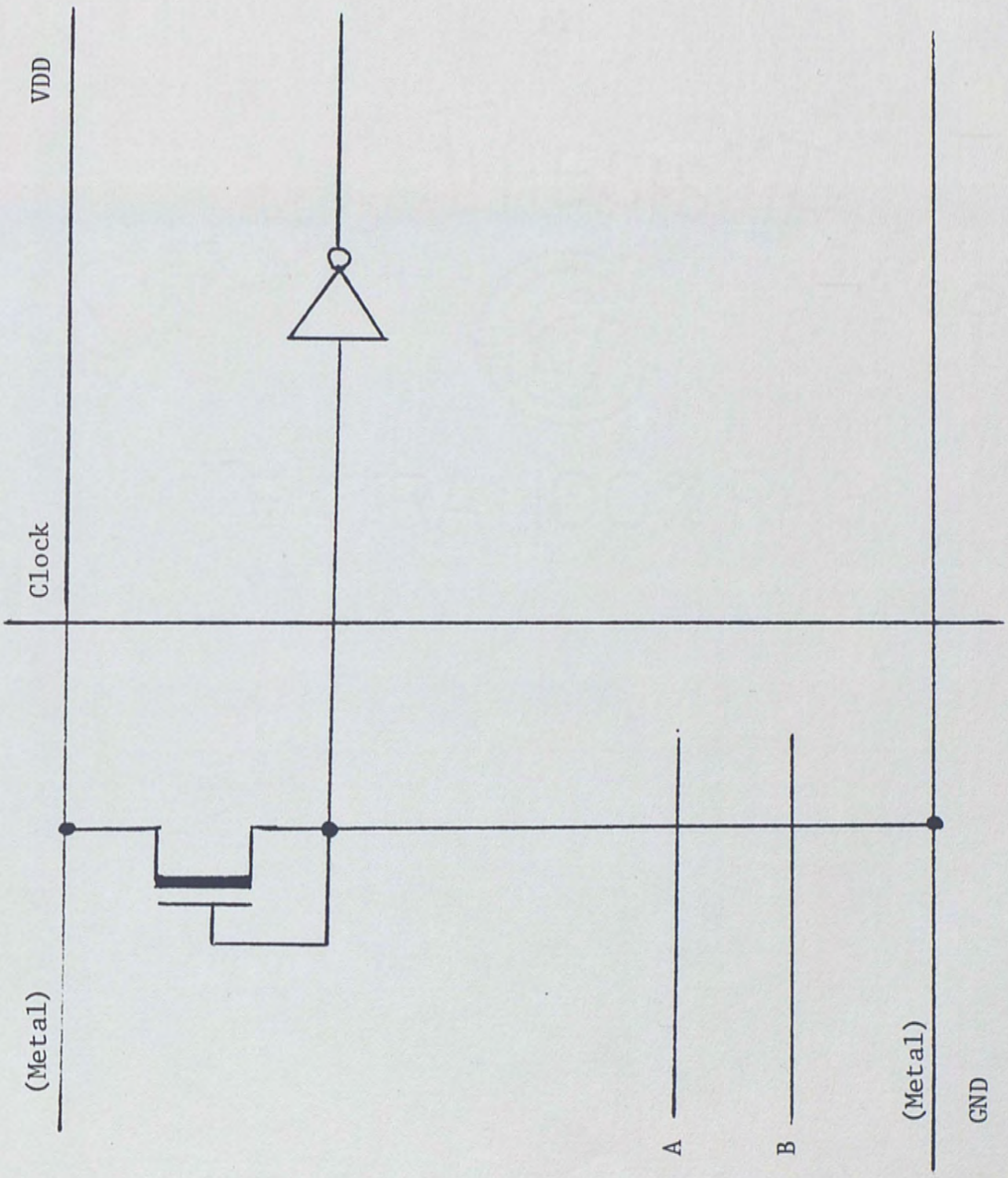

Fig. 12. NAND gate mixed notation.

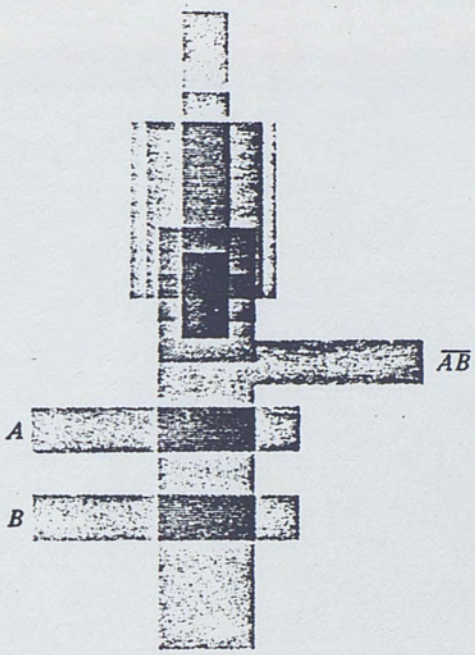

Fig. 13. NAND gate layout geometry (Mead 1980).

ensure that each node in the circuit can be potentially pulled up and pulled down, each depletion mode transistor is checked to be sure it is used in an appropriate manner. It also checks for the threshold drops, power supply and ground errors of the transistors.

The dynamic simulator or the logical simulator is used to simulate the layout with a variety of inputs to verify correct operation. The next CAD tool that can follow is the Power Estimator. This software tool estimates the total and individual power dissipations of the various pullups in the circuit layout. Table 1 summarizes all the discussed CAD tools used for the verification of integrated circuit design.

TABLE 1

VERIFICATION DESIGN TOOLS

Design Tool Type	Function
Design Rule Checker (DRC)	Checks for design rule violations which can occur during the layout (Mead and Conway 1980).
Circuit Extractor	Extracts the actual electrical circuit from the mask description and checks for violations of certain electrical rules.
Static Checker or Static Simulator	Makes use of the extracted electrical circuit to check, provide statistics and flag errors about the pullup-pulldown potentials of the transistors; threshold drop occurrences; power supply and ground connections of the transistors and the mode of depletion (pullup) transistors.
Dynamic or Logic Simulator	Simulates the network with a variety of inputs and verifies correct operation. It allows the designer to specify random input values and checks the respective outputs for proper results.
Power Estimator	Estimates the average and maximum power dissipation of the various pullups in the circuit layout.

CHAPTER IV

A COMPLETE DESIGN EXAMPLE USING THE CAESAR SYSTEM

An illustration of the steps involved in VLSI design follows. The CAD is the Caesar System running under the fourth Berkeley distribution of VAX-Unix.

Design of a CAM (Content Addressable Memory) Cell

The first step is to define and then derive the block diagram of the target circuit, the CAM cell, A CAM or associative memory is a memory unit accessed on the basis of data content rather than by specific address or location. A CAM cell is made up of two sub-cells, namely a storage cell and a compare cell, and the circuits for enabling read, write and match modes. Figure 14 shows the block diagram of this CAM cell.

As seen from Figure 14, the CAM cell is capable of performing "WRITE", "READ", and "MATCH" modes. The details of these operations follow in the circuit description of the storage cell and match cell. The "READ", "WRITE" and "MATCH" are the control signal inputs for the respective modes. The "MATCH Output" (M) indicates the state of the match. DA and DA' are the input bits, data-argument and its complement, respectively, to the cell. D and D' are the data output bits. Having defined the block diagram, the next step is to

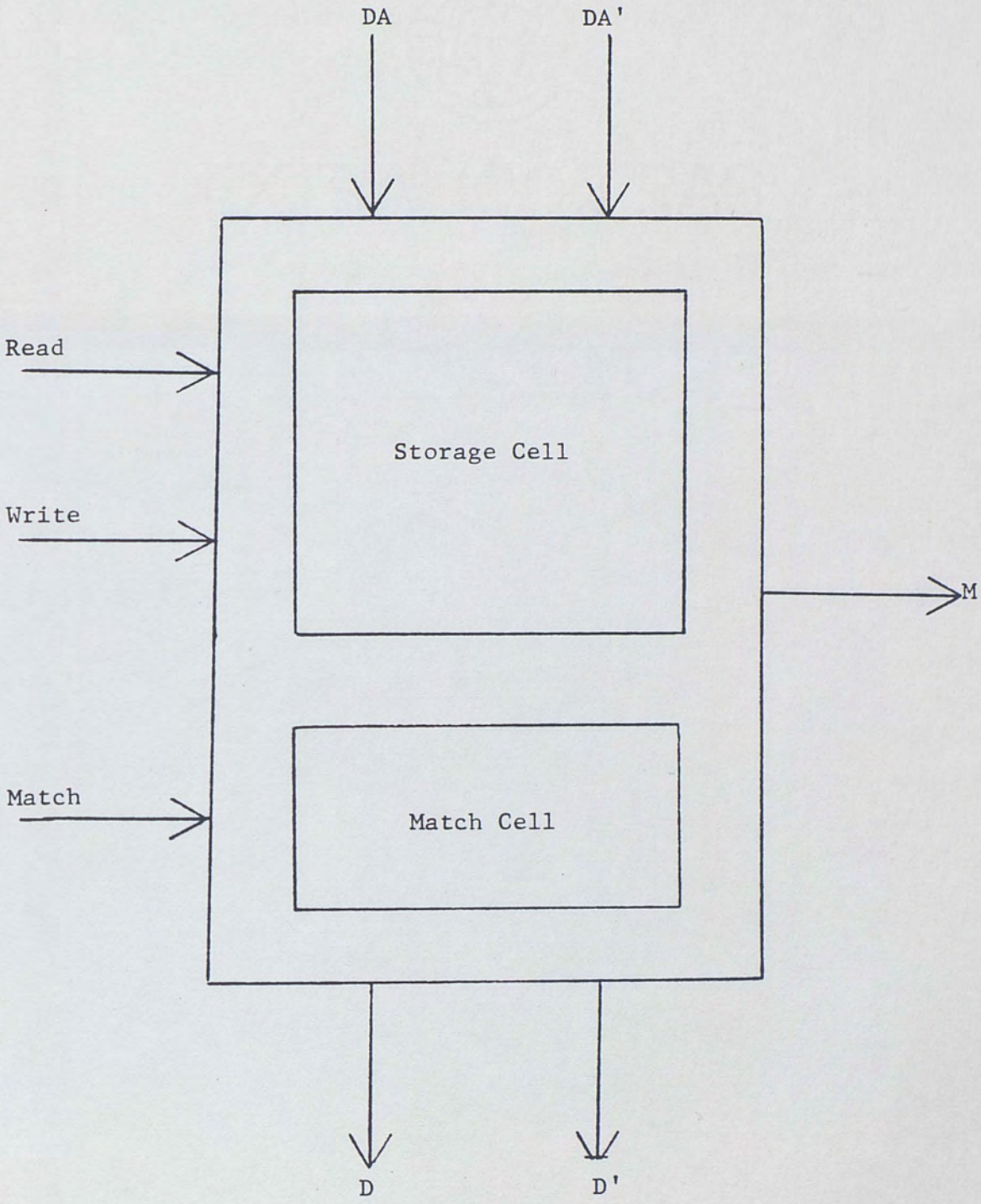

Fig. 14. CAM cell block.

select the cell types for the storage and match cells and evaluate the circuit's need of the particular cell type.

The CAM Storage Cell

A standard 6-transistor static RAM cell can be employed as a CAM memory cell. Figure 15 shows the circuit diagram of this cell. The cell is capable of performing "WRITE" and "READ" operations. During the "WRITE" operation, the address select line is driven high and the data and data' lines are loaded with the information and its complement, respectively. Sufficient time is allowed for the circuit to settle. As shown in Figure 15, data is stored at the base of Q4 and data' at the base of Q3.

During the "READ" mode, the data and data' lines are tristated and on the application of a high voltage (+VDD) to the address select line, cell information and its complement are sensed on the data and data' bit lines, respectively.

The CAM Compare Cell

A modified NXOR (not exclusive or) gate circuit, shown in Figure 16, is employed to perform the compare function. This circuit has 6 transistors. As shown in Figure 16, the stored data bit (D) is compared with the argument bit (A) and the result of the comparison is valid only when the key control bit (K) is high.

When the key bit (K) is low, the output is pulled up to VDD, irrespective of the result of comparison. When the key control bit (K) is driven high, the output depends on the result of the

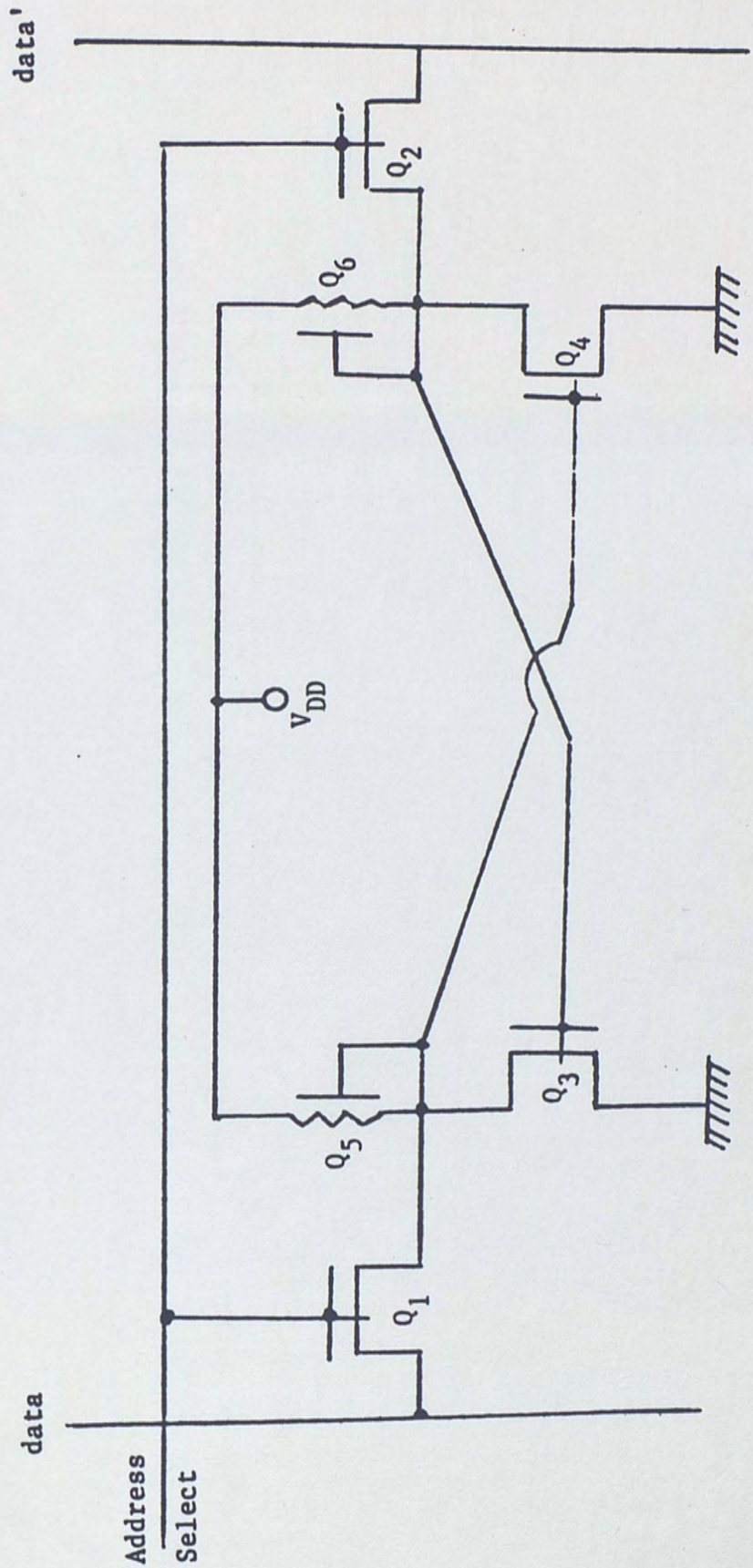

Fig. 15. 6-T static memory cell.

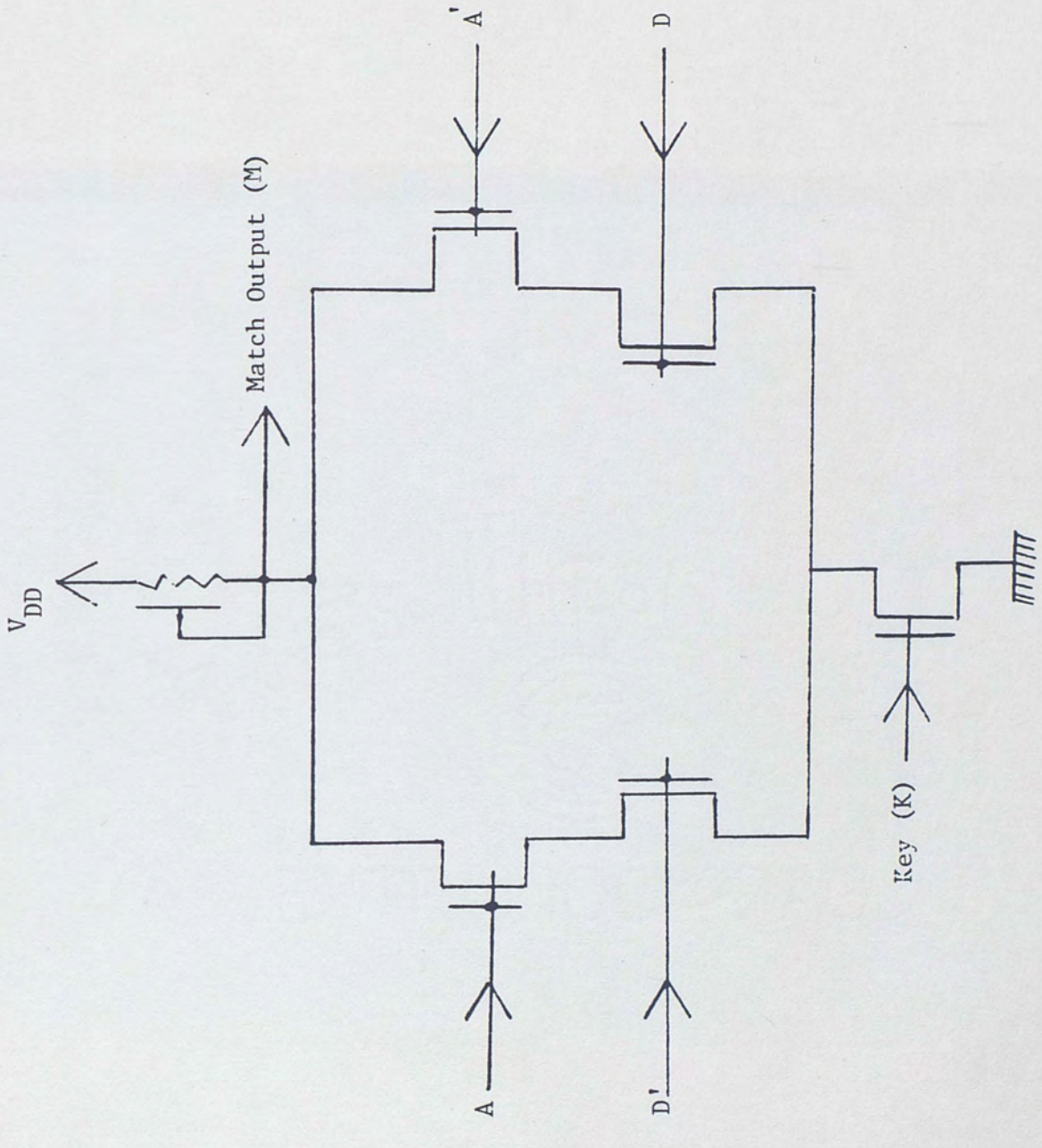

Fig. 16. NXOR.

comparison. If the comparison is yes, i.e., if data bit (D) = argument bit (A), then the output is pulled up to VDD. This is because the path to ground is blocked by this particular configuration. However, if "A" <> "D", i.e., if (A) = (D') or (A') = (D), then the output is pulled down to ground, as seen from Figure 16. Thus, a low value on the match output indicates that a match does not exist. A high value can mean either that match exists or that the match operation is not being performed or that the agrument bit is masked. The doubt can be clarified by keeping track of the control to the match mode.

CAM Cell Circuit

Storage and compare cells, discussed in the previous subsections, when connected together, in a manner as shown in Figure 17, constitute a CAM cell which is capable of performing (1) "WRITE" and "READ" and (2)"Compare" or "Match" During the "READ" and "WRITE" modes, the key bit (K) is held low. During the match mode, (K) is set in accordance with the corresponding bit in the key register.

Implementing the CAM Cell Using Caesar

After having decided on the circuit diagram, the stick diagram is drawn. Figure 17 shows the mixed notation presentation of the CAM cell. This is a mixture of the circuit diagram and the stick diagram. This diagram indicates the various pull-up and pull-down transitions used for correct operations. The next step is layout.

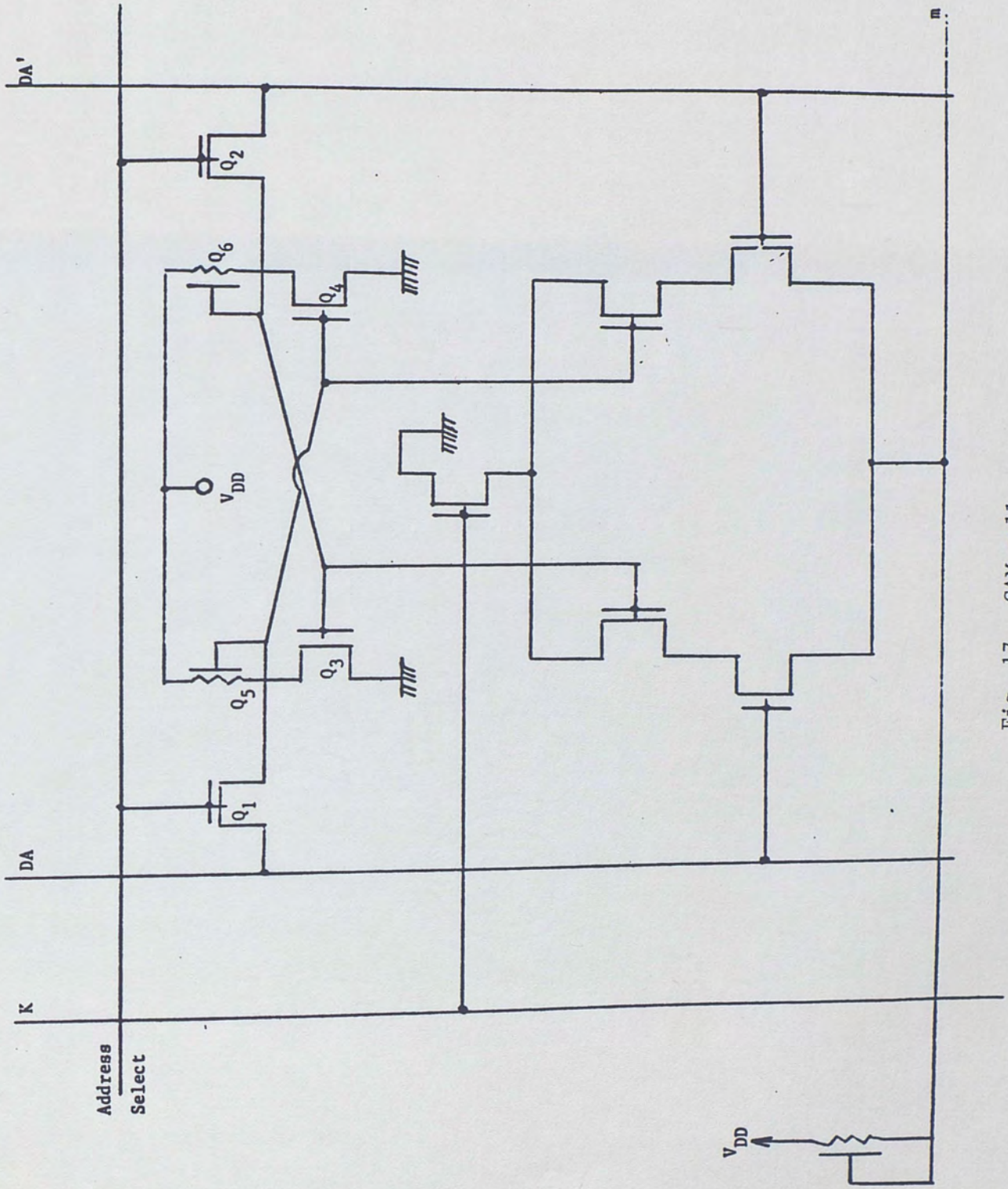

Fig. 17. CAM cell.

Depending on the level of experience, one may transfer a proper design to the CAD system, or one may start directly with the CAD using the stick diagram. The layout is done following the applicable n-MOS design rules (Mead and Conway 1980). The designer starts the layout step by giving the cell a name. If the CAM cell is named CAM1, then the first command after getting into Caesar is:

: EDITCELL CAM1

The layout is completed with the help of painting commands. Having completed the layout, the long command:

: CIF 250

is typed which creates the CIF file and sets the lambda value to 2.5 microns. The cell is saved by the long command:

: SAVECELL

The Caesar is now exited with the long command:

: QUIT

The next step is to check the circuit for adherence to the design rules. This is done under the UNIX operating system. The sequence of commands typed are:

% TOCED CAM1

% MAKE-REC CAM1

%DRC CAM1

%MORE CAM1.DRC.

The TOCED command generates a rectangular definition of every symbol in the CIF file. "MAKE-REC" combines all the definitions

into one complete description. The DRC command runs the design rule checker over the whole cell layout. The last command, "MORE CAM1.DRC", is typed to look into the error message file. The error message file displays the type of error and the (X.Y) coordinates of its occurrence, if any. If any error is displayed, then it is required to go back into Caesar and edit the cell to correct the error. In order to find the (X,Y) coordinate, the long command used is:

:PUSH X Y

This command places the cross-hair of the graphics terminal at the X, Y coordinates.

The next step is to perform dynamic simulation to check the logical operation of the cell. A "simulation block" diagram is drawn on paper, which indicates the various input, outputs and control variables. Figure 18 shows such a block diagram. This block diagram helps the designer to get a better feel of the simulation commands that will be needed and the proper sequence of the commands.

All three modes of operation are simulated by defining input, output and control variables, as stated below:

Input Data-argument and Data-argument' bits : DA DA'

Output Data and Data' Bits: D D'

Key or Mask Input :K

Match Output :M

Input Address :A

The input and control variables are set or given random variables and the outputs are checked for expected results.

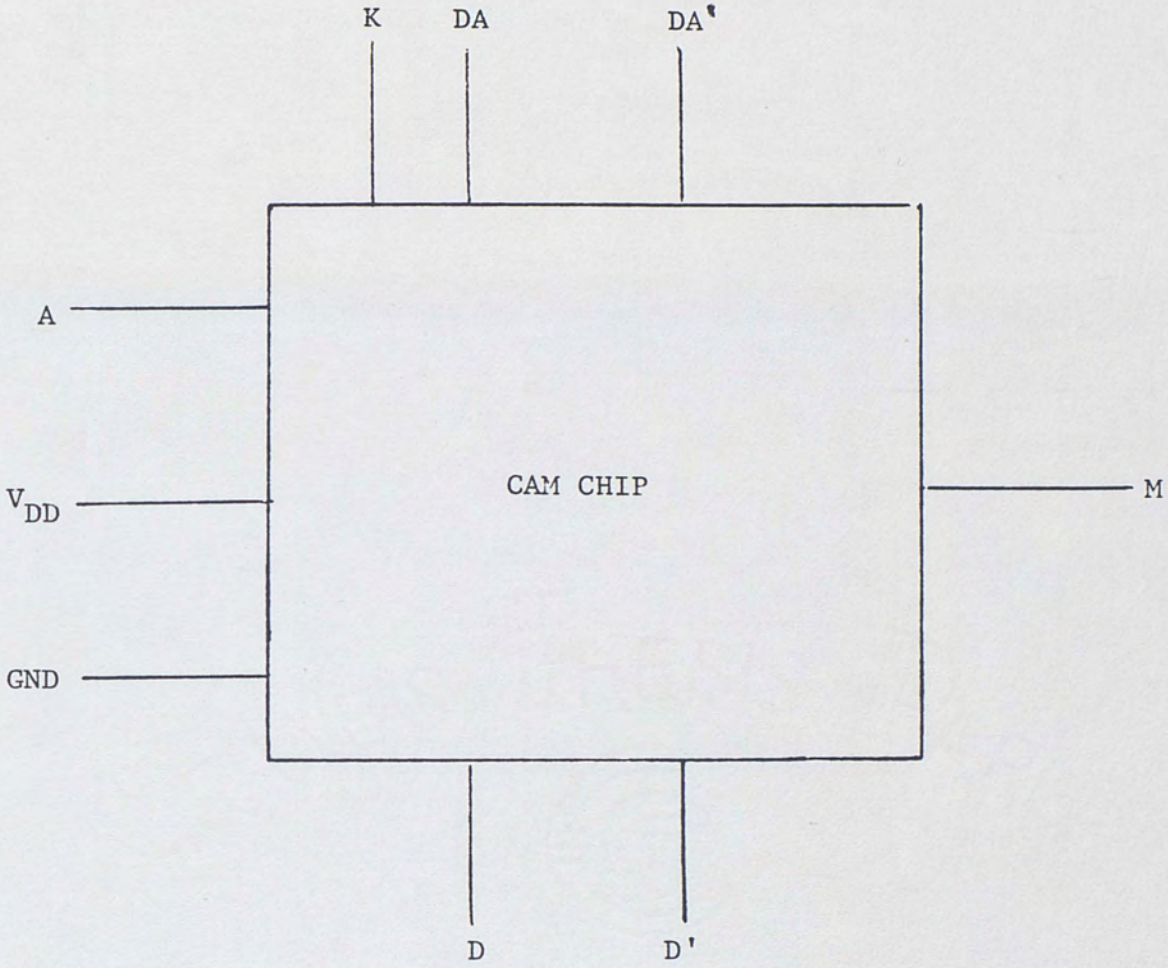

Fig. 18. Simulation block.

Having decided on the simulation variables, the cell is edited again to label the desired variables at their respective layers or locations. In order to save the labels, the command

:CIF-P 250

is used, instead of :CIF 250. This includes the labels as part of the cell.

Now, in UNIX, command %LS is typed which lists all the cell files generated thus far. The highest definition file number is selected from this list and the following steps are typed in the following sequence, for dynamic simulation. Let the highest definition number be denoted by "#".

% MAKE-REC #

%CNODE-EXTR CAM1 #

This command sequence extracts the circuit from the graphic layout. After this command, the system asks questions regarding the type of layer to which each label corresponds. The questions are answered by typing M (if the label corresponds to the metal layer), P (if polysicon layer) and D (if diffusion layer).

%MAKE-SIM #

% MAKE-BITS #

% STIPOUT/TMP/ #.STIP A*

This last command carries the nodes of the circuit to be printed out.

% ESIM #.SIM

This command calls "ESIM" the dynamic (logic) simulator. After this command, the system responds with the "SIM>" symbol. The

simulation variables are given values. After the simulator command is executed, the outputs are obtained. The outputs are then checked for expected values.

The following example presents the dynamic simulation of CAM1, the "ESIM" prompt is SIM>.

Procedure for Dynamic Simulation of CAM1

1. To write logic "1" in the cell:

Command	Comments
SIM > I	This step initializes the cell
SIM > h DA A	This step sets DA = A = logic "1"
SIM > 1 DA' K	This step sets DA' = K = logic "0"
SIM > w DA DA' M	DA, DA' and M are made to be watched
SIM > S	This step simulates with the stated values of the variables
DA = 1 DA' = 0 M = 1	If this is what is displayed by the simulator, then the answer is correct, otherwise, NO

2. To read the cell:

SIM > X DA DA'	This tri states the DA and DA' bit lines for reading
SIM > h A	This enables the address line
SIM > 1 K	This disables the match line
SIM > S	This simulates the cell with defined values of the variables
DA = 1 DA' = 0 M = 1	If this result is displayed, then the cell is declared to read and write properly, otherwise it is not

3. To perform an associative match with the contents of the cell:

SIM > h DA' K	The content of the cell is compared with DA' = 1
SIM > 1 DA A	DA = 0 A = 0 (previous step puts K=1)
SIM > S	Simulate command
DA = 0 DA' = 1 M = 0	If this result is displayed, then correct; otherwise no. This is because the cell was stored with DA = logic "1" and is compared with logic "0", thus, as expected, M should go low.

Different values are assigned to the input, output and control variables to make sure that the cell operates as desired. A script file can be obtained for this simulation through the "vi" text editor.

After completing the logical analysis using the "ESIM" Simulator, static simulation is performed using the "STAT" simulator. This simulator performs the electrical analysis of the circuit of the layout. The command to evoke this simulator is % STAT followed by the name of the file which in this case is CAM1.

∴ The command is: % STAT CAM1

The STAT simulator responds to this command with information on the various pull-up, pull-down ratios with their node numbers, the number of transistors along with their types, the number of bad transistors, the total number and the identifiers of the floating nodes, etc. This information helps to verify the expected results.

The last step in the design is to find how much power the cell will consume. This is done with the help of the "POWEST" software package which provides an estimate of power of the cell. The command to evoke this package is:

% POWEST CAM1

The output of the layout can be obtained by typing:

% Cifplot CAM1.Cif
↑
(name of cell)

This prints out the hard copy of the layout in CIF (i.e., Caltech Intermediate form).

The design procedure can be extended to design larger systems which may utilize cells from the cell library, test vectors for the

dynamic simulation, and the replication of cells and iteration of the design steps.

Figure 19 shows the Caltech Intermediate Form (CIF) plot of CAM1 cell, derived from the Caesar system. Figure 20 shows the CIF plot of a 4 words x 4 bits CAM design. This 4 words x 4 bits CAM design was implemented on the Caesar system and, after its design completion and verification, was fabricated through Mosis in California.

The Role of an Electrical Engineer in VLSI Design Implementation and Improvement

Until recently, the design of integrated circuitry has been the province of circuit and logic designers working within semiconductor firms. This group of Electrical Engineers provided services from design specifications straight through the fabrication processes. At this time, not many design tools were developed, which demanded further expertise of an Electrical Engineer to analyze and process integrated circuit designs.

The introduction of design tools, Design Automation (DA), sophisticated CAD/CAM, automated process control and many such facilities encouraged other professionals, e.g., computer scientists, to implement VLSI designs. The new facilities helped to reduce the mental baggage, the design engineer had to carry from one stage of VLSI design to another for non state-of-the-art applications. The expertise in device physics and electronic circuits were no longer a requirement for the implementation of a VLSI design.

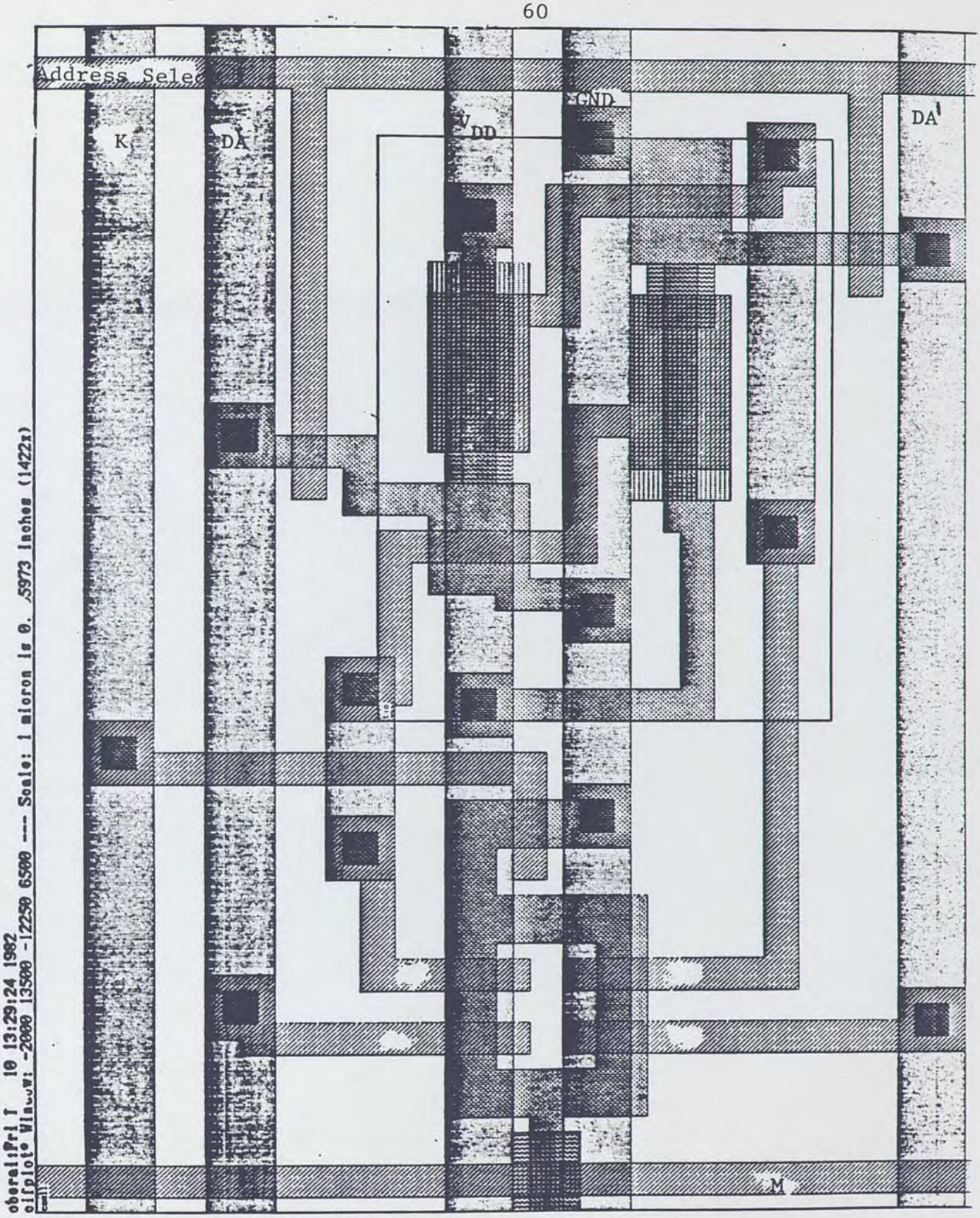

Fig. 19. CIF plot of CAM1 cell.

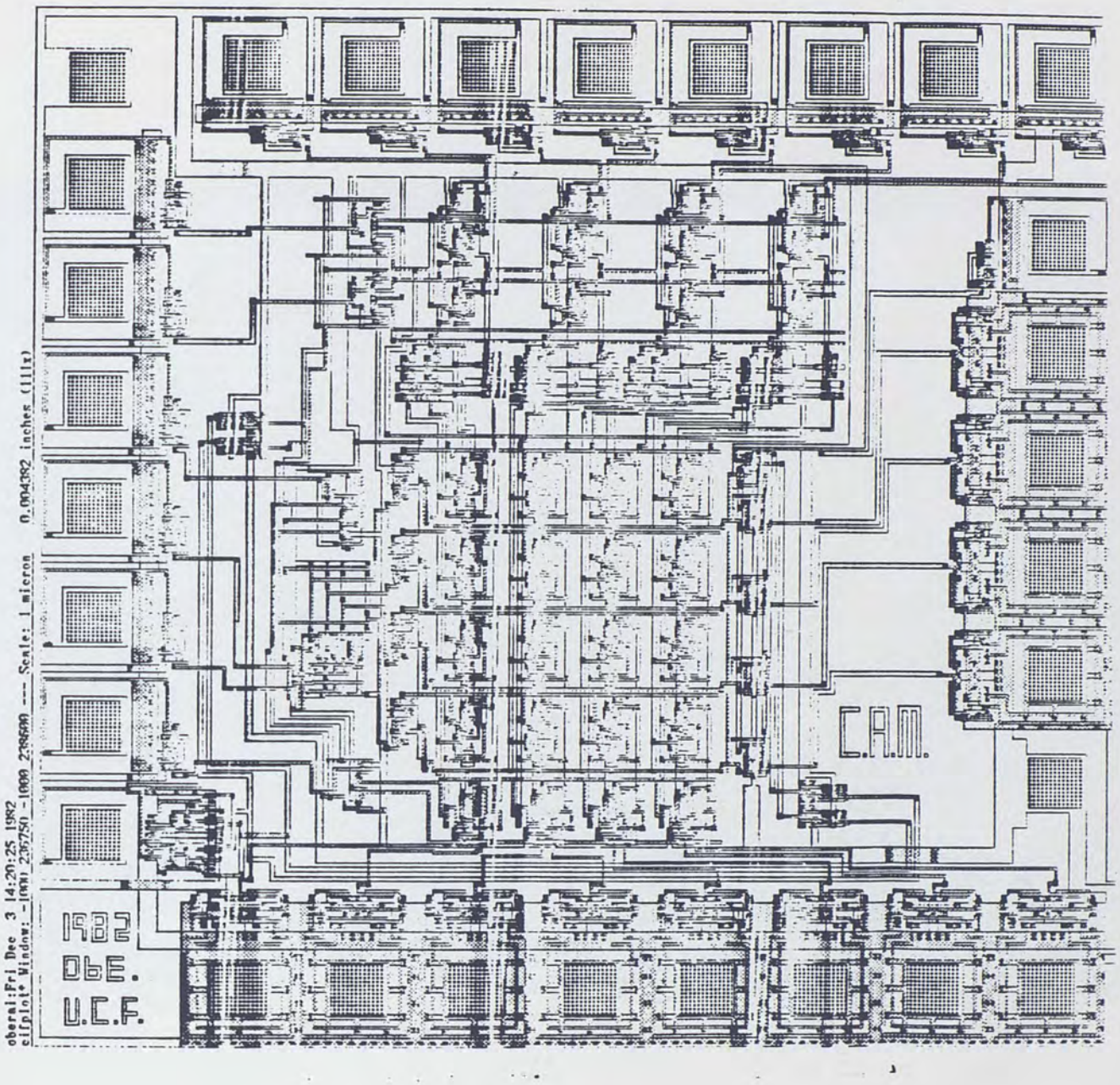

Fig. 20. CIF plot of 4 x 4 CAM.

Software packages were developed for design rule verification, simulation, circuit analysis and power estimates. This evolution in VLSI implementation reduced the electrical engineering expertise required for design layout and analysis.

At present, continuing advances in VLSI technology are putting pressure on DA applications. This is because the DA and other software design tools fail to cope with the increasing complexity of VLSI design. Hardware DA tools have been proposed and implemented. This demand for hardware DA tools, to enhance system speed and reliability, again call for the hardware design expertise of an electrical engineer in instances where present VLSI technology cannot be applied to provide the desired speed and complexity. For better speed and complexity, the circuit cleverness trends are a requirement. These trends further call for detailed research in the fields of device physics and in the electrical characteristics of transistors.

Advances in VLSI fabrication processes demand efficient and cost effective procedures and this task can be supported best by electrical engineers. This discipline group does, and will continue to, help in the design of sophisticated CAD/CAM systems. This sophistication will further boost the efficiency and turn-around times of VLSI designs.

Present masking techniques demand for a more efficient and precise masking methodology to attain better yield and quality. This, again, calls for the services of an electrical engineer to bring about this change.

An Electrical Engineer has and will always provide the VLSI technology with better and wider applications. The increasing applications of VLSI designs makes this technology more advanced and sophisticated. The Electrical Engineers will find themself mainly concentrated in its research and development of sophisticated VLSI design applications. Sophisticated VLSI design applications will dictate for greater design complexity which will further dictate for better design tools, to meet the complexity. This would, in turn, lead to the improvement of the VLSI technology.

Thus, it can be seen that electrical engineering has and will play a very important role in VLSI implementation and improvement.

CHAPTER V

CONCLUSIONS

This report discussed the VLSI design methodology and the various design tools that support this methodology at various steps in the design process. The evolution of CAD/CAM for implementation of ICs has made it possible for the complexity of circuits to grow exponentially. On the other hand, the growing complexity calls for better and faster design tools. The proposal and implementation of hardware design tools has increased the demand for Gate Arrays to custom design. This is owing to the modularity, ease of routing, and hierarchical design approach; featured by Gate Arrays.

Efforts are being made to further automate the VLSI design procedures. A total design automation will further minimize human intervention in VLSI design procedures. The essential services and human interface between the designer's group and the silicon foundaries provided by the Silicon Broker has commercialized the VLSI technology. Commercialization has lead to low cost of production, better quality design, higher yield and faster turn around times of integrated circuits. This has further encouraged more research in VLSI design and production, with partial incentives of making profits.

Caesar system, supported by the "Berkeley Tools", is a highly recommended system for the beginners in the field of VLSI designing. Quite complex designs can be implemented using Caesar with a minimum of design rules and process constraints. Caesar lacks the sophisticated features of the commercial CADs offered by CALMA. CALMA's CADS, "CIRCUITS" and "CHIPS", present to the VLSI designers designing capabilities which allow the designers to create systems so complex that the present fabrication process may fail to keep up with the complexity.

An Electrical Engineer is faced with the responsibilities of designing hardware design tools and more sophisticated CAD/CAM. His main focus in VLSI design methodology will be that of application research. He will provide the VLSI designers group, made up of various computer scientists, computer architects and logic designers, with ideas of VLSI application in various spectrums. His expertise will always be needed at every point in the evolution of VLSI technology.

REFERENCES

Association for Computing Machinery, Inc. Proceedings of the ACM Computer Science Conference. Orlando, FL, February 15-17, 1983.

Baker, Clark M., and Terman, Chris. "Tools for Verifying Integrated Circuit Designs," Lambda (4th Quarter, 1980): 22.

Machover, Carl, and Blauth, Robert E. The CAD/CAM Handbook. Bedford, MA: Computervision Corporation, 1980.

Mead, Carver, and Conway, Lynn. Introduction to VLSI Systems. Reading, MA: Addison-Wesley Publishing Company, 1980.

Ousterhout, John. "Editing VLSI Circuits with Caesar," User's Manual for Caesar 6, Berkeley, CA: University of California, 1980.