# Chapter 1

# VLSI Physical Design Automation

The information revolution has transformed our lives. It has changed our perspective of work, life at home and provided new tools for entertainment. The internet has emerged as a medium to distribute information, communication, event planning, and conducting E-commerce. The revolution is based on computing technology and communication technology, both of which are driven by a revolution in Integrated Circuit (IC) technology. ICs are used in computers for microprocessor, memory, and interface chips. ICs are also used in computer networking, switching systems, communication systems, cars, airplanes, even microwave ovens. ICs are now even used in toys, hearing aids and implants for human body. MEMs technology promises to develop mechanical devices on ICs thereby enabling integration of mechanical and electronic devices on a miniature scale. Many sensors, such as acceleration sensors for auto air bags, along with conversion circuitry are built on a chip. This revolutionary development and widespread use of ICs has been one of the greatest achievements of humankind.

IC technology has evolved in the 1960s from the integration of a few transistors (referred to as *Small Scale Integration* (SSI))o the integration of millions of transistors in *Very Large Scale Integration* (VLSI) chips currently in use. Early ICs were simple and only had a couple of gates or a flip-flop. Some ICs were simply a single transistor, along with a resistor network, performing a logic function. In a period of four decades there have been four generations of ICs with the number of transistors on a single chip growing from a few to over 20 million. It is clear that in the next decade, we will be able to build chips with billions of transistors running at several Ghz. We will also be able to build MEM chips with millions of electrical and mechanical devices. Such chips will enable a new era of devices which will make such exotic applications, such as tele-presence, augumented reality and implantable and wearable computers, possible. Cost effective world wide point-to-point communication will be common and available to all.

This rapid growth in integration technology has been (and continues to be) made possible by the automation of various steps involved in the design and fabrication of VLSI chips. Integrated circuits consist of a number of electronic components, built by layering several different materials in a well-defined fashion on a silicon base called a *wafer.* The designer of an IC transforms a circuit description into a geometric description, called the *layout.* A layout consists of a set of planar geometric shapes in several layers. The layout is checked to ensure that it meets all the design requirements. The result is a set of design files that describes the layout. An optical pattern generator is used to convert the design files into pattern generator files. These files are used to produce patterns called *masks.* During fabrication, these masks are used to pattern a silicon wafer using a sequence of photo-lithographic steps. The component formation requires very exacting details about geometric patterns and the separation between them. The process of converting the specification of an electrical circuit into a layout is called the *physical design process.* Due to the tight tolerance requirements and the extremely small size of the individual components, physical design is an extremely tedious and error prone process. Currently, the smallest geometric feature of a component can be as small as 0.25 micron (one micron, written as $\mu$m is equal to $1.0 \times 10^{-6}$m). For the sake of comparison, a human hair is 75 $\mu$m in diameter. It is expected that the feature size can be reduced below 0.1 micron within five years. This small feature size allows fabrication of as many as 200 million transistors on a 25 mm $\times$ 25 mm chip. Due to the large number of components, and the exacting details required by the fabrication process, physical design is not practical without the help of computers. As a result, almost all phases of physical design extensively use Computer Aided Design (CAD) tools, and many phases have already been partially or fully automated.

VLSI Physical Design Automation is essentially the research, development and productization of algorithms and data structures related to the physical design process. The objective is to investigate optimal arrangements of devices on a plane (or in three dimensions) and efficient interconnection schemes between these devices to obtain the desired functionality and performance. Since space on a wafer is very expensive real estate, algorithms must use the space very efficiently to lower costs and improve yield. In addition, the arrangement of devices plays a key role in determining the performance of a chip. Algorithms for physical design must also ensure that the layout generated abides by all the rules required by the fabrication process. Fabrication rules establish the tolerance limits of the fabrication process. Finally, algorithms must be efficient and should be able to handle very large designs. Efficient algorithms not only lead to fast turn-around time, but also permit designers to make iterative improvements to the layouts. The VLSI physical design process manipulates very simple geometric objects, such as polygons and lines. As a result, physical design algorithms tend to be very intuitive in nature, and have significant overlap with graph algorithms and combinatorial optimization algorithms. In view of this observation, many consider physical design automation the study of graph theoretic and combinatorial algorithms for manipulation of geometric

objects in two and three dimensions. However, a pure geometric point of view ignores the electrical (both digital and analog) aspect of the physical design problem. In a VLSI circuit, polygons and lines have inter-related electrical properties, which exhibit a very complex behavior and depend on a host of variables. Therefore, it is necessary to keep the electrical aspects of the geometric objects in perspective while developing algorithms for VLSI physical design automation. With the introduction of Very Deep Sub-Micron (VDSM), which provides very small features and allows dramatic increases in the clock frequency, the effect of electrical parameters on physical design will play a more dominant role in the design and development of new algorithms.

In this chapter, we present an overview of the fundamental concepts of VLSI physical design automation. Section 1.1 discusses the design cycle of a VLSI circuit. New trends in the VLSI design cycle are discussed in Section 1.2. In Section 1.3, different steps of the physical design cycle are discussed. New trends in the physical design cycle are discussed in Section 1.4. Different design styles are discussed in Section 1.5 and Section 1.6 presents different packaging styles. Section 1.7 presents a brief history of physical design automation and Section 1.8 lists some existing design tools.

## 1.1 VLSI Design Cycle

The VLSI design cycle starts with a formal specification of a VLSI chip, follows a series of steps, and eventually produces a packaged chip. A typical design cycle may be represented by the flow chart shown in Figure 1.1. Our emphasis is on the physical design step of the VLSI design cycle. However, to gain a global perspective, we briefly outline all the steps of the VLSI design cycle.

1. **System Specification:** The first step of any design process is to lay down the specifications of the system. System specification is a high level representation of the system. The factors to be considered in this process include: performance, functionality, and physical dimensions (size of the die (chip)). The fabrication technology and design techniques are also considered. The specification of a system is a compromise between market requirements, technology and economical viability. The end results are specifications for the size, speed, power, and functionality of the VLSI system.

2. **Architectural Design:** The basic architecture of the system is designed in this step. This includes, such decisions as RISC (Reduced Instruction Set Computer) versus CISC (Complex Instruction Set Computer), number of ALUs, Floating Point units, number and structure of pipelines, and size of caches among others. The outcome of architectural design is a Micro-Architectural Specification (MAS). While MAS is a textual (English like) description, architects can accurately predict the performance, power and die size of the design based on such a description.

Such estimates are based on the scaling of existing design or components of existing designs. Since many designs (especially microprocessors) are based on modifications or extensions to existing designs, such a method can provide fairly accurate early estimates. These early estimates are critical to determine the viability of a product for a market segment. For example, for mobile computing (such as lap top computer), low power consumption is a critical factor, due to limited battery life. Early estimates based on architecture can be used to determine if the design is likely to meet its power spec.

3. **Behavioral or Functional Design:** In this step, main functional units of the system are identified. This also identifies the interconnect requirements between the units. The area, power, and other parameters of each unit are estimated. The behavioral aspects of the system are considered without implementation specific information. For example, it may specify that a multiplication is required, but exactly in which mode such multiplication may be executed is not specified. We may use a variety of multiplication hardware depending on the speed and word size requirements. The key idea is to specify behavior, in terms of input, output and timing of each unit, without specifying its internal structure. The outcome of functional design is usually a timing diagram or other relationships between units. This information leads to improvement of the overall design process and reduction of the complexity of subsequent phases. Functional or behavioral design provides quick emulation of the system and allows fast debugging of the full system. Behavioral design is largely a manual step with little or no automation help available.

4. **Logic Design:** In this step the control flow, word widths, register allocation, arithmetic operations, and logic operations of the design that represent the functional design are derived and tested. This description is called Register Transfer Level (RTL) description. RTL is expressed in a Hardware Description Language (HDL), such as VHDL or Verilog. This description can be used in simulation and verification. This description consists of Boolean expressions and timing information. The Boolean expressions are minimized to achieve the smallest logic design which conforms to the functional design. This logic design of the system is simulated and tested to verify its correctness. In some special cases, logic design can be automated using *high level synthesis* tools. These tools produce a RTL description from a behavioral description of the design.

5. **Circuit Design:** The purpose of circuit design is to develop a circuit representation based on the logic design. The Boolean expressions are converted into a circuit representation by taking into consideration the speed and power requirements of the original design. *Circuit Simulation* is used to verify the correctness and timing of each component. The circuit design is usually expressed in a detailed circuit diagram. This diagram shows the circuit elements (cells, macros, gates, transistors) and interconnec-

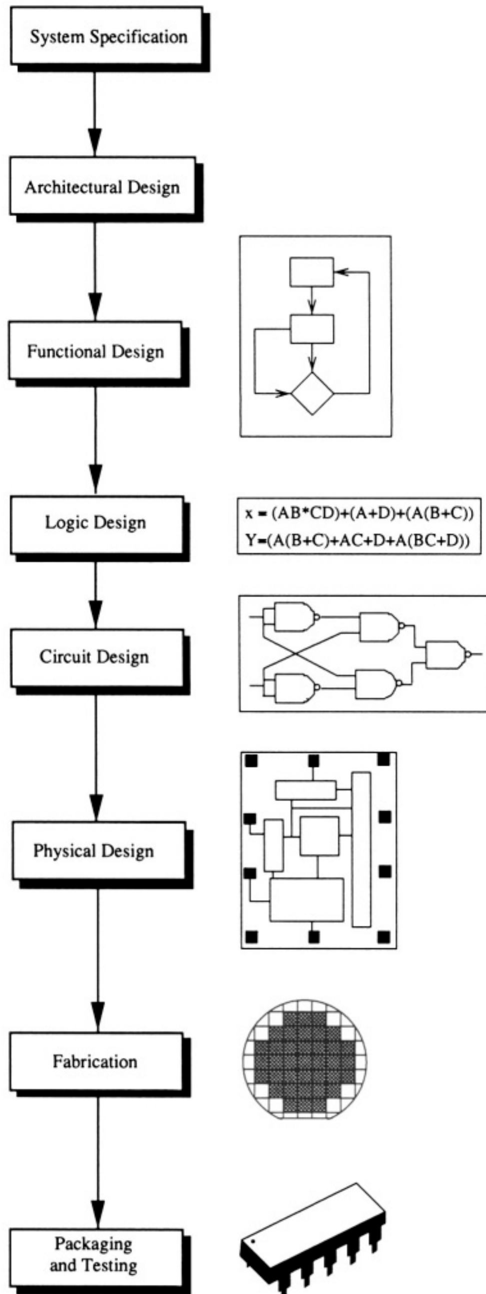



Figure 1.1: A simple VLSI design cycle.

tion between these elements. This representation is also called a *netlist.* Tools used to manually enter such description are called *schematic capture tools.* In many cases, a netlist can be created automatically from logic (RTL) description by using *logic synthesis* tools.

6. **Physical Design:** In this step the circuit representation (or netlist) is converted into a geometric representation. As stated earlier, this geometric representation of a circuit is called a *layout.* Layout is created by converting each logic component (cells, macros, gates, transistors) into a geometric representation (specific shapes in multiple layers), which perform the intended logic function of the corresponding component. Connections between different components are also expressed as geometric patterns typically lines in multiple layers. The exact details of the layout also depend on design rules, which are guidelines based on the limitations of the fabrication process and the electrical properties of the fabrication materials. Physical design is a very complex process and therefore it is usually broken down into various sub-steps. Various verification and validation checks are performed on the layout during physical design. In many cases, physical design can be completely or partially automated and layout can be generated directly from netlist by *Layout Synthesis* tools. Most of the layout of a high performance design (such as a microprocessor) may be done using manual design, while many low to medium performance design or designs which need faster time-to-market may be done automatically. Layout synthesis tools, while fast, do have an area and performance penalty, which limit their use to some designs. Manual layout, while slow and manually intensive, does have better area and performance as compared to synthesized layout. However this advantage may dissipate as larger and larger designs may undermine human capability to comprehend and obtain globally optimized solutions.

7. **Fabrication:** After layout and verification, the design is ready for fabrication. Since layout data is typically sent to fabrication on a tape, the event of release of data is called *Tape Out.* Layout data is converted (or fractured) into photo-lithographic masks, one for each layer. Masks identify spaces on the wafer, where certain materials need to be deposited, diffused or even removed. Silicon crystals are grown and sliced to produce wafers. Extremely small dimensions of VLSI devices require that the wafers be polished to near perfection. The fabrication process consists of several steps involving deposition, and diffusion of various materials on the wafer. During each step one mask is used. Several dozen masks may be used to complete the fabrication process. A large wafer is 20 cm (8 inch) in diameter and can be used to produce hundreds of chips, depending of the size of the chip. Before the chip is mass produced, a prototype is made and tested. Industry is rapidly moving towards a 30 cm (12 inch) wafer allowing even more chips per wafer leading to lower cost per chip.

8. **Packaging, Testing and Debugging:** Finally, the wafer is fabricated and diced into individual chips in a fabrication facility. Each chip is then packaged and tested to ensure that it meets all the design specifications and that it functions properly. Chips used in Printed Circuit Boards (PCBs) are packaged in Dual In-line Package (DIP), Pin Grid Array (PGA), Ball Grid Array (BGA), and Quad Flat Package (QFP). Chips used in Multi-Chip Modules (MCM) are not packaged, since MCMs use bare or naked chips.

It is important to note that design of a complex VLSI chip is a complex human power management project as well. Several hundred engineers may work on a large design project for two to three years. This includes architecture designers, circuit designers, physical design specialists, and design automation engineers. As a result, design is usually partitioned along functionality, and different units are designed by different teams. At any given time, each unit may not be at the same level of design. While one unit may be in logic design phase, another unit may be completing its physical design phase. This imposes a serious problem for chip level design tools, since these tools must work with partial data at the chip level.

The VLSI design cycle involves iterations, both within a step and between different steps. The entire design cycle may be viewed as transformations of representations in various steps. In each step, a new representation of the system is created and analyzed. The representation is iteratively improved to meet system specifications. For example, a layout is iteratively improved so that it meets the timing specifications of the system. Another example may be detection of design rule violations during design verification. If such violations are detected, the physical design step needs to be repeated to correct the error. The objectives of VLSI CAD tools are to minimize the time for each iteration and the total number of iterations, thus reducing time-to-market.

## 1.2 New Trends in VLSI Design Cycle

The design flow described in the previous section is conceptually simple and illustrates the basic ideas of the VLSI design cycle. However, there are many new trends in the industry, which seek to significantly alter this flow. The major contributing factors are:

1. **Increasing interconnect delay:** As the fabrication process improves, the interconnect is not scaling at the same rate as the devices. Devices are becoming smaller and faster, and interconnect has not kept up with that pace. As a result, almost 60% of a path delay may be due to interconnect. One solution to interconnect delay and signal integrity issue is insertion of repeaters in long wires. In fact, repeaters are now necessary for most chip level nets. This techniques requires advanced planning since area for repeaters must be allocated upfront.

2. **Increasing interconnect area:** It has been estimated that a microprocessor die has only 60%-70% of its area covered with active devices. The rest of the area is needed to accommodate the interconnect. This area also leads to performance degradation. In early ICs, a few hundred transistors were interconnected using one layer of metal. As the number of transistors grew, the interconnect area increased. However, with the introduction of a second metal layer, the interconnect area decreased. This has been the trend between design complexity and the number of metal layers. In current designs, with approximately ten million transistors and four to six layers of metal, one finds about 40% of the chips real estate dedicated to its interconnect. While more metal layers help in reducing the die size, it should be noted that more metal layers (after a certain number of layers) do not necessarily mean less interconnect area. This is due to the space taken up by the vias on the lower layers.

3. **Increasing number of metal layers:** To meet the increasing needs of interconnect, the number of metal layers available for interconnect is increasing. Currently, a three layer process is commonly used for most designs, while four layer and five layer processes are used mainly for microprocessors. As a result, a three dimensional view of the interconnect is necessary.

4. **Increasing planning requirements:** The most important implication of increasing interconnect delay, area of the die dedicated to interconnect, and a large number of metal layers is that the relative location of devices is very important. Physical design considerations have to enter into design at a much earlier phase. In fact, functional design should include *chip planning*. This includes two new key steps; block planning and signal planning. Block planning assigns shapes and locations to main functional blocks. Signal planning refers to assignment of the three dimensional regions through which major busses and signals will be routed. Timing should be estimated to verify the validity of the chip plan. This plan should be used to create timing constraints for later stages of design.

5. **Synthesis:** The time required to design any block can be reduced if layout can be directly generated or *synthesized* from a higher level description. This not only reduces design time, it also eliminates human errors. The biggest disadvantage is the area used by synthesized blocks. Such blocks take larger areas than hand crafted blocks. Depending upon the level of design on which synthesis is introduced, we have two types of synthesis.

   **Logic Synthesis:** This process converts an HDL description of a block into schematics (circuit description) and then produces its layout. Logic synthesis is an established technology for blocks in a chip design, and for complete Application Specific Integrated Circuits (ASICs). Logic synthesis is not applicable for large regular blocks, such as RAMs, ROMs, PLAs and Datapaths, and complete microprocessor chips for two reasons;

speed and area. Logic synthesis tools are too slow and too area inefficient to deal with such blocks.

**High Level Synthesis:** This process converts a functional or micro-architectural description into a layout or RTL description. In high level synthesis, input is a description which captures only the behavioral aspects of the system. The synthesis tools form a spectrum. The synthesis system described above can be called general synthesis. A more restricted type synthesizes some constrained architectures. For example, Digital Signal Processing (DSP) architectures have been successfully synthesized. These synthesis systems are sometimes called *Silicon Compilers.* An even more restricted type of synthesis tools are called *Module Generators,* which work on smaller size problems. The basic idea is to simplify the synthesis task, either by restricting the architecture or restricting the size of the problem. Silicon compilers sometimes use the output of module generators. High level synthesis is an area of current research and is not used in actual chip development [GDWL92]. In summary, high level synthesis systems provide very good implementations for specialized classes of systems, and they will continue to gain acceptance as they become more generalized.

In order to accommodate the factors discussed above, the VLSI design cycle is changing. In Figure 1.2, we show a VLSI design flow which is *closer to reality.* Due to increasing interconnect delay, the physical design starts very early in the design cycle to get improved estimates of the performance of the chip, The early floor physical design activities lead to increasingly improved chip layout as each block is refined. This also allows better utilization of the chip area to distribute the interconnect in three dimensions. This distribution helps in reducing the die size, improving yield and reducing cost. Essentially, the VLSI design cycle produces increasingly better defined descriptions of the given chip. Each description is verified and, if it fails to meet the specification, the step is repeated.

## 1.3 Physical Design Cycle

The input to the physical design cycle is a circuit diagram and the output is the layout of the circuit. This is accomplished in several stages such as partitioning, floorplanning, placement, routing, and compaction. The different stages of physical design cycle are shown in Figure 1.3. Each of these stages will be discussed in detail in various chapters; however, to give a global perspective, we present a brief description of all the stages here.

1. **Partitioning:** A chip may contain several million transistors. Due to the limitations of memory space and computation power available it may not be possible to layout the entire chip (or generically speaking any large circuit) in the same step. Therefore, the chip (circuit) is normally partitioned into sub-chips (sub-circuits). These sub-partitions are called

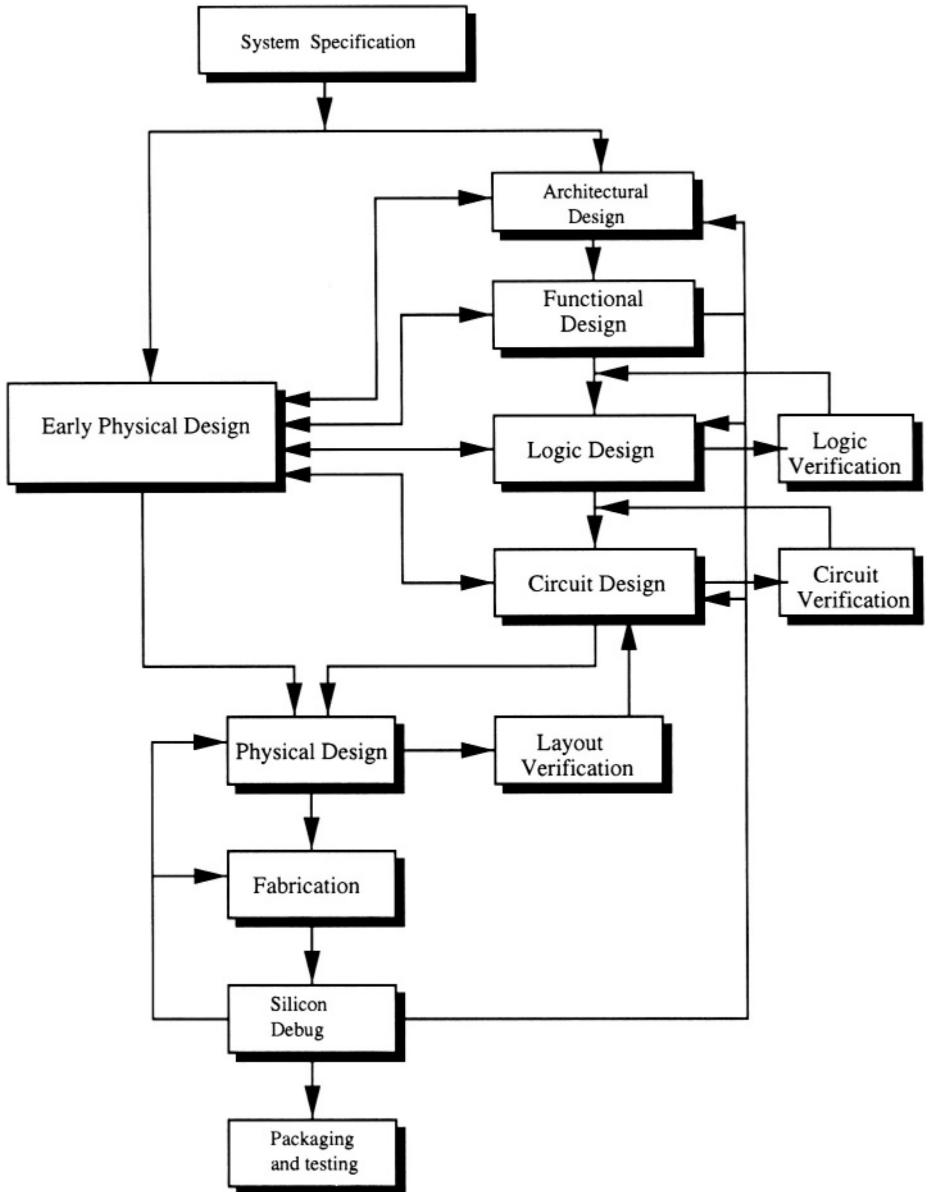



Figure 1.2: VLSI design cycle.

blocks. The actual partitioning process considers many factors such as the size of the blocks, number of blocks, and number of interconnections between the blocks. The output of partitioning is a set of blocks and the interconnections required between blocks. Figure 1.3(a) shows that the input circuit has been partitioned into three blocks. In large circuits, the partitioning process is hierarchical and at the topmost level a chip may have 5 to 25 blocks. Each block is then partitioned recursively into smaller blocks.

2. **Floorplanning and Placement:** This step is concerned with selecting good layout alternatives for each block, as well as the entire chip. The area of each block can be estimated after partitioning and is based approximately on the number and the type of components in that block. In addition, interconnect area required within the block must be considered. The actual rectangular shape of the block, which is determined by the aspect ratio may, however, be varied within a pre-specified range. Many blocks may have more general rectilinear shapes. Floorplanning is a critical step, as it sets up the ground work for a good layout. However, it is computationally quite hard. Very often the task of floorplanning is done by a design engineer, rather than a CAD tool. This is due to the fact that a human is better at 'visualizing' the entire floorplan and taking into account the information flow. Manual floorplanning is sometimes necessary as the major components of an IC need to be placed in accordance with the signal flow of the chip. In addition, certain components are often required to be located at specific positions on the chip.

   During placement, the blocks are exactly positioned on the chip. The goal of placement is to find a minimum area arrangement for the blocks that allows completion of interconnections between the blocks, while meeting the performance constraints. That is, we want to avoid a placement which is routable but does not allow certain nets to meet their timing goals. Placement is typically done in two phases. In the first phase an initial placement is created. In the second phase, the initial placement is evaluated and iterative improvements are made until the layout has minimum area or best performance and conforms to design specifications. Figure 1.3(b) shows that three blocks have been placed. It should be noted that some space between the blocks is intentionally left empty to allow interconnections between blocks.

   The quality of the placement will not be evident until the routing phase has been completed. Placement may lead to an unroutable design, i.e., routing may not be possible in the space provided. In that case, another iteration of placement is necessary. To limit the number of iterations of the placement algorithm, an estimate of the required routing space is used during the placement phase. Good routing and circuit performance depend heavily on a good placement algorithm. This is due to the fact that once the position of each block is fixed, very little can be done to

improve the routing and the overall circuit performance. Late placement changes lead to increased die size and lower quality designs.

3. **Routing:** The objective of the routing phase is to complete the interconnections between blocks according to the specified netlist. First, the space not occupied by the blocks (called the routing space) is partitioned into rectangular regions called *channels* and *switchboxes.* This includes the space between the blocks as well the as the space on top of the blocks. The goal of a router is to complete all circuit connections using the shortest possible wire length and using only the channel and switch boxes. This is usually done in two phases, referred to as the *Global Routing* and *Detailed Routing* phases. In global routing, connections are completed between the proper blocks of the circuit disregarding the exact geometric details of each wire and pin. For each wire, the global router finds a list of channels and switchboxes which are to be used as a passageway for that wire. In other words, global routing specifies the different regions in the routing space through which a wire should be routed. Global routing is followed by detailed routing which completes point-to-point connections between pins on the blocks. Global routing is converted into exact routing by specifying geometric information such as the location and spacing of wires and their layer assignments. Detailed routing includes channel routing and switchbox routing, and is done for each channel and switchbox. Routing is a very well studied problem, and several hundred articles have been published about all its aspects. Since almost all problems in routing are computationally hard, the researchers have focused on heuristic algorithms. As a result, experimental evaluation has become an integral part of all algorithms and several benchmarks have been standardized. Due to the very nature of the routing algorithms, complete routing of all the connections cannot be guaranteed in many cases. As a result, a technique called *rip-up and re-route* is used, which basically removes troublesome connections and reroutes them in a different order. The routing phase of Figure 1.3(c) shows that all the interconnections between the three blocks have been routed.

4. **Compaction:** Compaction is simply the task of compressing the layout in all directions such that the total area is reduced. By making the chip smaller, wire lengths are reduced, which in turn reduces the signal delay between components of the circuit. At the same time, a smaller area may imply more chips can be produced on a wafer, which in turn reduces the cost of manufacturing. However, the expense of computing time mandates that extensive compaction is used only for large volume applications, such as microprocessors. Compaction must ensure that no rules regarding the design and fabrication process are violated during the process. Figure 1.3(d) shows the compacted layout.

5. **Extraction and Verification:** Design Rule Checking (DRC) is a process which verifies that all geometric patterns meet the design rules imposed

by the fabrication process. For example, one typical design rule is the wire separation rule. That is, the fabrication process requires a specific separation (in microns) between two adjacent wires. DRC must check such separation for millions of wires on the chip. There may be several dozen design rules, some of them are quite complicated to check. After checking the layout for design rule violations and removing the design rule violations, the functionality of the layout is verified by *Circuit Extraction.* This is a reverse engineering process, and generates the circuit representation from the layout. The extracted description is compared with the circuit description to verify its correctness. This process is called *Layout Versus Schematics* (LVS) verification. Geometric information is extracted to compute Resistance and Capacitance. This allows accurate calculation of the timing of each component, including interconnect. This process is called *Performance Verification.* The extracted information is also used to check the reliability aspects of the layout. This process is called *Reliability Verification* and it ensures that layout will not fail due to electro-migration, self-heat and other effects [Bak90].

Physical design, like VLSI design, is iterative in nature and many steps, such as global routing and channel routing, are repeated several times to obtain a better layout. In addition, the quality of results obtained in a step depends on the quality of the solution obtained in earlier steps. For example, a poor quality placement cannot be 'cured' by high quality routing. As a result, earlier steps have more influence on the overall quality of the solution. In this sense, partitioning, floorplanning, and placement problems play a more important role in determining the area and chip performance, as compared to routing and compaction. Since placement may produce an 'unroutable' layout, the chip might need to be re-placed or re-partitioned before another routing is attempted. In general, the whole design cycle may be repeated several times to accomplish the design objectives. The complexity of each step varies, depending on the design constraints as well as the design style used. Each step of the design cycle will be discussed in greater detail in a later chapter.

## 1.4 New Trends in Physical Design Cycle

As fabrication technology improves and process enters the deep sub-micron range, it is clear that interconnect delay is not scaling at the same rate as the gate delay. Therefore, interconnect delay is a more significant part of overall delay. As a result, in high performance chips, interconnect delay must be considered from very early design stages. In order to reduce interconnect delay several methods can be employed.

1. **Chip level signal planning:** At the chip level, routing of major signals and buses must be planned from early design stages, so that interconnect distances can be minimized. In addition, these global signals must be routed in the top metal layers, which have low delay per unit length.

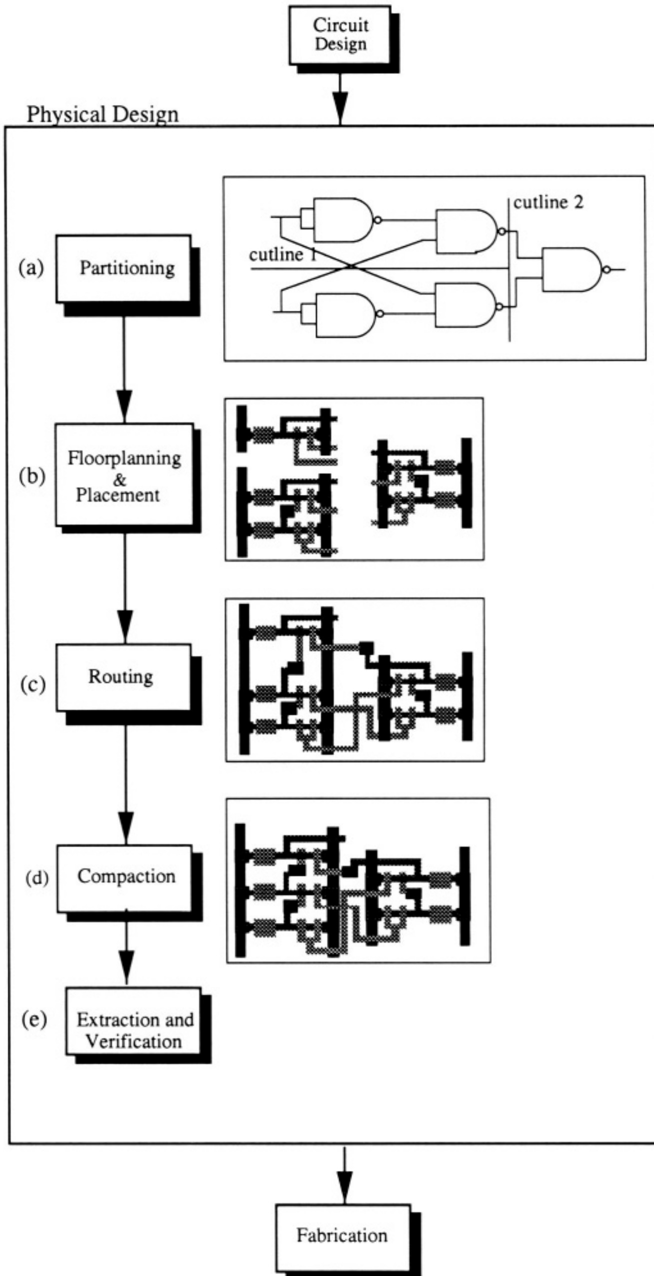



Figure 1.3: Physical design cycle.

2. **OTC routing:** Over-the-Cell (OTC) routing is a term used to describe routing over blocks and active areas. This is a departure from conventional channel and switchbox routing approach. Actually, chip level signal planning is OTC routing on the entire chip. The OTC approach can also be used within a block to reduce area and improve performance. The OTC routing approach essentially makes routing a three dimensional problem. Another effect of the OTC routing approach is that the pins are not brought to the block boundaries for connections to other blocks. Instead, pins are brought to the top of the block as a sea-of-pins. This concept, technically called the *Arbitrary Terminal Model* (ATM), will be discussed in a later chapter.

The conventional decomposition of physical design into partitioning, placement and routing phases is conceptually simple. However, it is increasingly clear that each phase is interdependent on other phases, and an integrated approach to partitioning, placement, and routing is required.

Figure 1.4 shows the physical design cycle with emphasis on timing. The figure shows that timing is estimated after floorplaning and placement, and these steps are iterated if some connections fail to meet the timing requirements. After the layout is complete, resistance and capacitance effects of one component on another can be extracted and accurate timing for each component can be calculated. If some connections or components fail to meet their timing requirements, or fail due to the effect of one component on another, then some or all phases of physical design need to be repeated. Typically, these 'repeat-or-not-to-repeat' decisions are made by experts rather than tools. This is due to the complex nature of these decisions, as they depend on a host of parameters.

## 1.5 Design Styles

Physical design is an extremely complex process. Even after breaking the entire process into several conceptually easier steps, it has been shown that each step is computationally very hard. However, market requirements demand quick time-to-market and high yield. As a result, restricted models and design styles are used in order to reduce the complexity of physical design. This practice began in the late 1960s and led to the development of several restricted design styles [Feu83]. The design styles can be broadly classified as either full-custom or semi-custom. In a full-custom layout, different blocks of a circuit can be placed at any location on a silicon wafer as long as all the blocks are non-overlapping. On the other hand, in semi-custom layout, some parts of a circuit are predesigned and placed on some specific place on the silicon wafer. Selection of a layout style depends on many factors including the type of chip, cost, and time-to-market. Full-custom layout is a preferred style for mass produced chips, since the time required to produce a highly optimized layout can be justified. On the other hand, to design an Application Specific Integrated Circuit (ASIC),

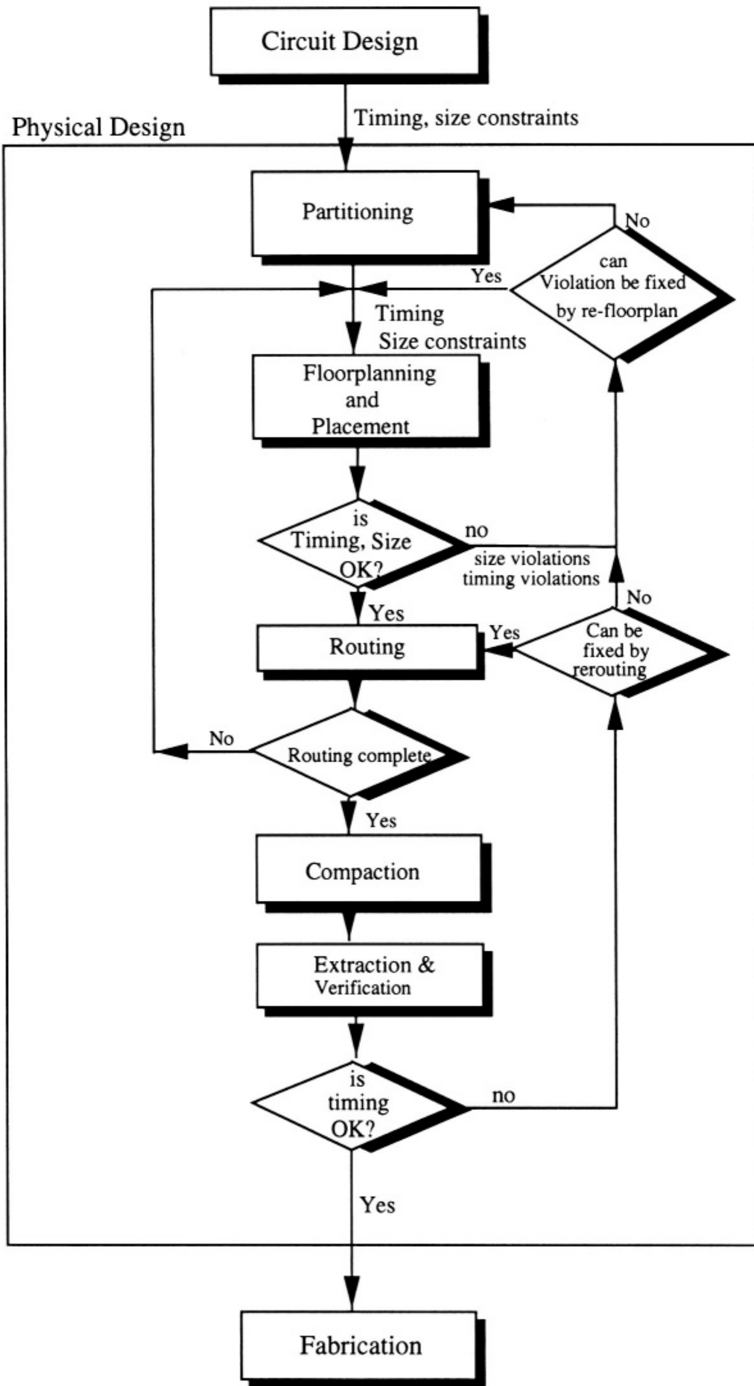



Figure 1.4: New physical design cycle.

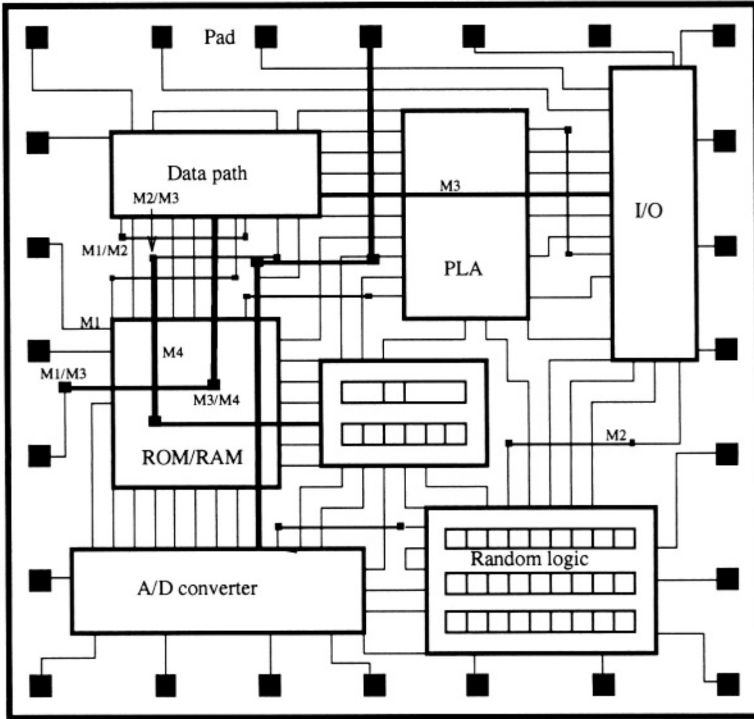



Figure 1.5: Full-custom structure.

a semi-custom layout style is usually preferred. On a large chip, each block may use a different layout design style.

### 1.5.1 Full-Custom

In its most general form of design style, the circuit is partitioned into a collection of sub-circuits according to some criteria such as functionality of each sub-circuit. The process is done hierarchically and thus full-custom designs have several levels of hierarchy. The chip is organized in clusters, clusters consist of units, and units are composed of *functional blocks* (in short, blocks). For sake of simplicity, we use the term blocks for units, blocks, and clusters. The full-custom design style allows functional blocks to be of any size. Figure 1.5 shows an example of a very simple circuit with few blocks. Other levels of hierarchy are not shown for this simple example. Internal routing in each block is not shown for the sake of clarity. In the full-custom design style, blocks can be placed at any location on the chip surface without any restrictions. In other words, this style is characterized by the absence of any constraints on the physical design process. This design style allows for very compact designs.

However, the process of automating a full-custom design style has a much higher complexity than other restricted models. For this reason it is used only when the final design must have minimum area and design time is less of a factor. The automation process for a full-custom layout is still a topic of intensive research. Some phases of physical design of a full-custom chip may be done manually to optimize the layout. Layout compaction is a very important aspect in full-custom design. The rectangular solid boxes around the boundary of the circuit are called *I/O pads*. Pads are used to complete interconnections between different chips or interconnections between the chip and the board. The spaces not occupied by blocks are used for routing of interconnecting wires. Initially all the blocks are placed within the chip area with the objective of minimizing the total area. However, there must be enough space left between the blocks so that routing can be completed using this space and the space on top of the blocks. Usually several metal layers are used for routing of interconnections. Currently, three metal layers are common for routing. A four metal layer process is being used for microprocessors, and a six layer process is gaining acceptance, as fabrication costs become more feasible. In Figure 1.5, note that width of the M1 wire is smaller than the width of the M2 wire. Also note that the size of the via between M1 and M2 is smaller than the size of the via between higher layers. Typically, metal widths and via sizes are larger for higher layers. The figure also shows that some routing has been completed on top of the blocks. The routing area needed between the blocks is becoming smaller and smaller as more routing layers are used. This is due to the fact that more routing is done on top of the transistors in the additional metal layers. If all the routing can be done on top of the transistors, the total chip area is determined by the area of the transistors. However, as circuits become more complex and interconnect requirements increase, the die size is determined by the interconnect area and the total transistor area serves as a lower bound on the die size of the chip.

In a hierarchical design of a circuit, each block in a full-custom design may be very complex and may consist of several sub-blocks, which in turn may be designed using the full-custom design style or other design styles. It is easy to see that since any block is allowed to be placed anywhere on the chip, the problem of optimizing area and the interconnection of wires becomes difficult. Full custom design is very time consuming; thus the method is inappropriate for very large circuits, unless performance or chip size is of utmost importance. Full custom is usually used for the layout of microprocessors and other performance and cost sensitive designs.

### 1.5.2 Standard Cell

The design process in the standard cell design style is somewhat simpler than full-custom design style. Standard cell architecture considers the layout to consist of rectangular cells of the same height. Initially, a circuit is partitioned into several smaller blocks, each of which is equivalent to some predefined subcircuit (cell). The functionality and the electrical characteristics of each

predefined cell are tested, analyzed, and specified. A collection of these cells is called a *cell library.* Usually a cell library consists of 500-1200 cells. Terminals on cells may be located either on the boundary or distributed throughout the cell area. Cells are placed in rows and the space between two rows is called a *channel.* These channels and the space above and between cells is used to perform interconnections between cells. If two cells to be interconnected lie in the same row or in adjacent rows, then the channel between the rows is used for interconnection. However, if two cells to be connected lie in two non-adjacent rows, then their interconnection wire passes through empty space between any two cells or passes on top of the cells. This empty space between cells in a row is called a *feedthrough.* The interconnections are done in two steps. In the first step, the feedthroughs are assigned for the interconnections of non-adjacent cells. Feedthrough assignment is followed by routing. The cells typically use only one metal layer for connections inside the cells. As a result, in a two metal process, the second metal layer can be used for routing in over-the-cell regions. In a three metal layer process, almost all the channels can be removed and all routing can be completed over the cells. However, this is a function of the density of cells and distribution of pins on the cells. It is difficult to obtain a channelless layout for chips which use highly packed dense cells with poor pin distribution. Figure 1.6 shows an example of a standard cell layout. A cell library is shown, along with the complete circuit with all the interconnections, feedthroughs, and power and ground routing. In the figure, the library consists of four logic cells and one feedthrough cell. The layout shown consists of several instances of cells in the library. Note that representation of a layout in the standard cell design style is greatly simplified as it is not necessary to duplicate the cell information.

The standard cell layout is inherently non-hierarchical. The hierarchical circuits, therefore, have to undergo some transformation before this design style can be used. This design style is well-suited for moderate size circuits and medium production volumes. Physical design using standard cells is somewhat simpler as compared to full-custom, and is efficient using modern design tools. The standard cell design style is also widely used to implement the 'random or control logic' part of the full-custom design as shown in Figure 1.5.

Logic Synthesis usually uses the standard cell design style. The synthesized circuit is mapped to cell circuits. Then cells are placed and routed.

While standard cell designs are quicker to develop, a substantial initial investment is needed in the development of the cell library, which may consist of several hundred cells. Each cell in the cell library is 'hand crafted' and requires highly skilled physical design specialists. Each type of cell must be created with several transistor sizes. Each cell must then be tested by simulation and its performance must be characterized. Cell library development is a significant project with enormous manpower and financial resource requirements.

A standard cell design usually takes more area than a full-custom or a hand-crafted design. However, as more and more metal layers become available for routing and design tools improve, the difference in area between the two design styles will gradually reduce.

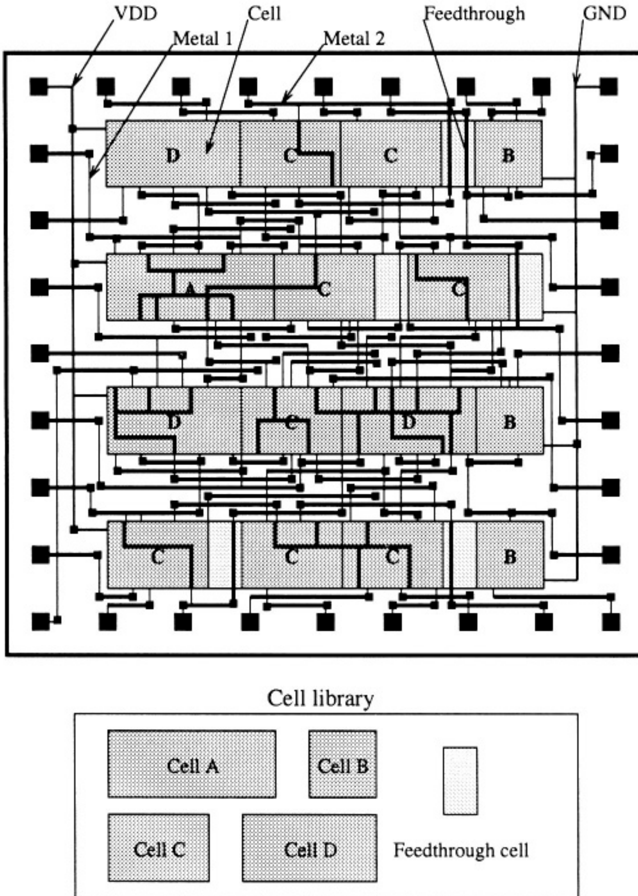



Figure 1.6: Standard cell structure.

### 1.5.3 Gate Arrays

This design style is a simplification of standard cell design. Unlike standard cell design, all the cells in gate array are identical. Each chip is an array of identical gates or cells. These cells are separated by both vertical and horizontal spaces called vertical and horizontal channels. The circuit design is modified such that it can be partitioned into a number of identical blocks. Each block must be logically equivalent to a cell on the gate array. The name 'gate array' signifies the fact that each cell may simply be a gate, such as a three input NAND gate. Each block in design is mapped or placed onto a prefabricated cell on the chip during the partitioning/placement phase, which is reduced to a block to cell assignment problem. The number of partitioned blocks must be less than or equal to the total number of cells on the chip. Once the circuit

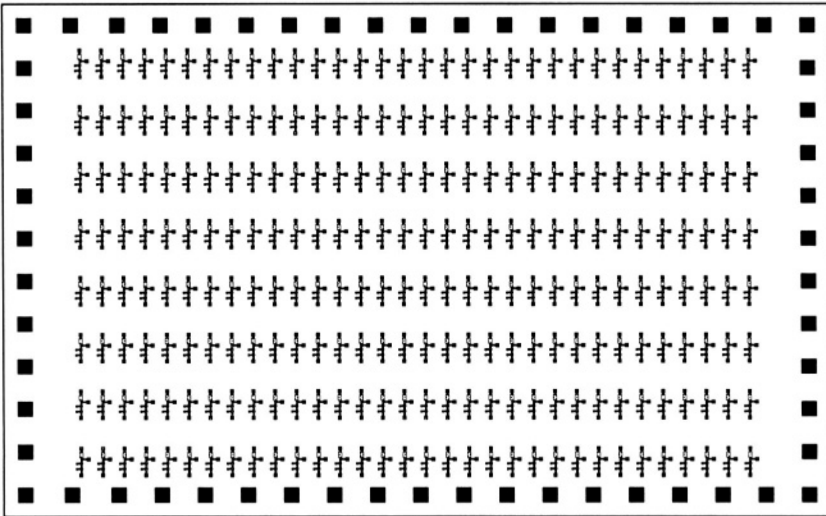


Figure 1.7: A conceptual uncommitted gate array.

is partitioned into identical blocks, the task is to make the interconnections between the prefabricated cells on the chip using horizontal and vertical channels to form the actual circuit. Figure 1.7 shows an 'uncommitted' gate array, which is simply a term used for a prefabricated chip. The gate array wafer is taken into a fabrication facility and routing layers are fabricated on top of the wafer. The completed wafer is also called a 'customized wafer'. It should be noted that the number of tracks allowed for routing in each channel is fixed. As a result, the purpose of the routing phase is simply to complete the connections rather than minimize the area. Two layers of interconnections are most common; though one and three layers are also used. Figure 1.8 illustrates a committed gate array design. Like standard cell designs, synthesis can also use the gate array style. In gate array design the entire wafer, consisting of several dozen chips, is prefabricated.

This simplicity of gate array design is gained at the cost of rigidity imposed upon the circuit both by the technology and the prefabricated wafers. The advantage of gate arrays is that the steps involved for creating any prefabricated wafer are the same and only the last few steps in the fabrication process actually depend on the application for which the design will be used. Hence gate arrays are cheaper and easier to produce than full-custom or standard cell. Similar to standard cell design, gate array is also a non-hierarchical structure.

The gate array architecture is the most restricted form of layout. This also means that it is the simplest for algorithms to work with. For example, the task of routing in gate array is to determine if a given placement is routable. The routability problem is conceptually simpler as compared to the routing

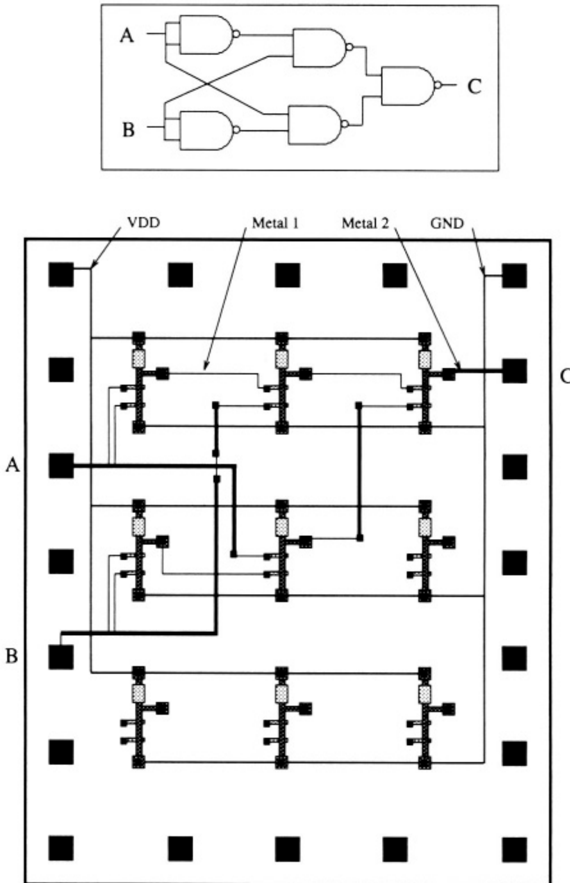
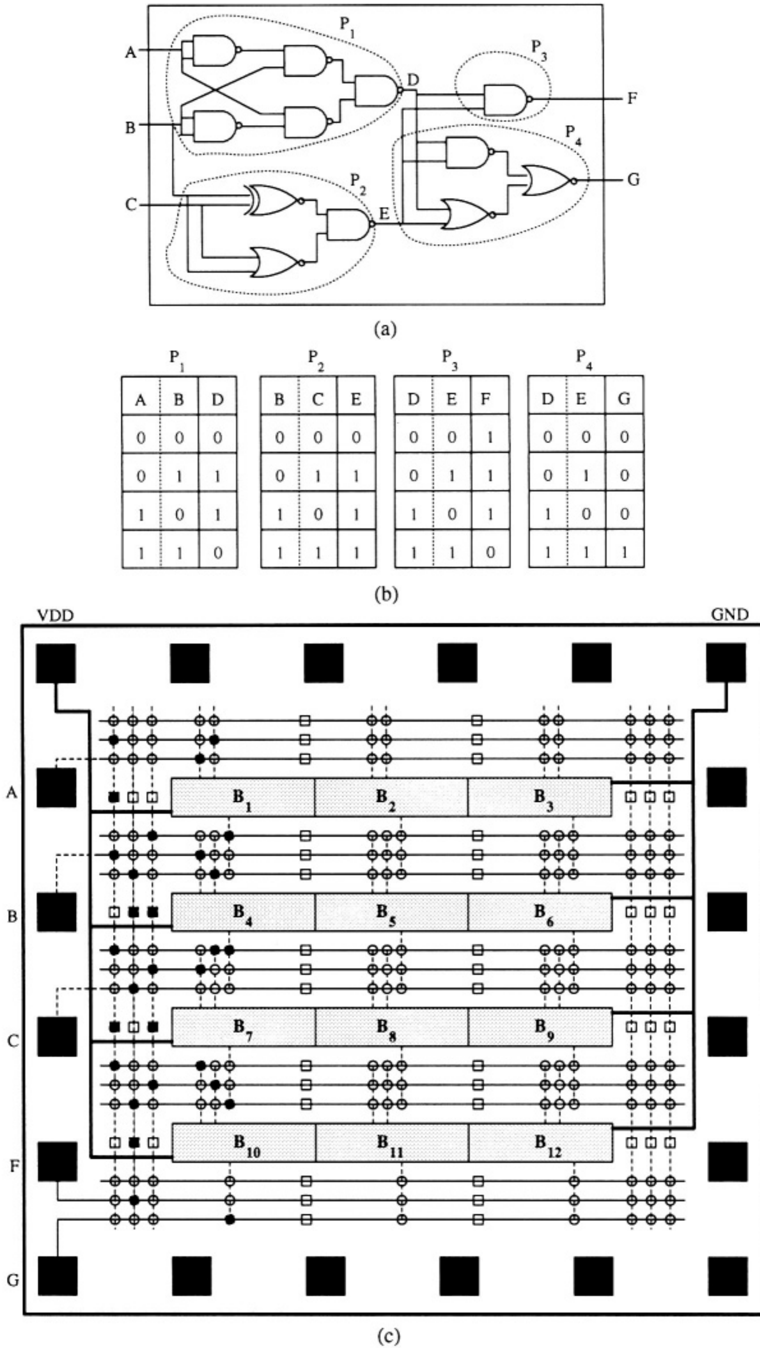



Figure 1.8: A conceptual gate array.

problem in standard cell and full-custom design styles.

### 1.5.4 Field Programmable Gate Arrays

The Field Programmable Gate Array (FPGA) is a new approach to ASIC design that can dramatically reduce manufacturing turn-around time and cost for low volume manufacturing [Gam89, Hse88, Won89]. In FPGAs, cells and interconnect are prefabricated. The user simply 'programs' the interconnect. FPGA designs provide large scale integration and user programmability. A FPGA consists of horizontal rows of programmable logic blocks which can be interconnected by a programmable routing network. FPGA cells are more complex than standard cells. However, almost all the cells have the same layout. In its simplistic form, a logic block is simply a memory block which can be pro-




P1

| A | B | D |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

P2

| B | C | E |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

P3

| D | E | F |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

P4

| D | E | G |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Figure 1.9: A committed FPGA.

grammed to remember the logic table of a function. Given a certain input, the logic block 'looks up' the corresponding output from the logic table and sets its output line accordingly. Thus by loading different look-up tables, a logic block can be programmed to perform different functions. It is clear that $2^K$ bits are required in a logic block to represent a $K$-bit input, 1-bit output combinational logic function. Obviously, logic blocks are only feasible for small values of $K$. Typically, the value of $K$ is 5 or 6. For multiple outputs and sequential circuits the value of $K$ is even less. The rows of logic blocks are separated by horizontal routing channels. The channels are not simply empty areas in which metal lines can be arranged for a specific design. Rather, they contain predefined wiring 'segments' of fixed lengths. Each input and output of a logic block is connected to a dedicated vertical segment. Other vertical segments merely pass through the blocks, serving as feedthroughs between channels. Connection between horizontal segments is provided through *antifuses,* whereas the connection between a horizontal segment and a vertical segment is provided through a *cross fuse.* Figure 1.9(c) shows the general architecture of a FPGA, which consists of four rows of logic blocks. The cross fuses are shown as circles, while antifuses are shown as rectangles. One disadvantage of fuse based FPGAs is that they are not reprogrammable. There are other types of FPGAs which allow re-programming, and use pass gates rather than programmable fuses.

Since there are no user specific fabrication steps in a FPGA, the fabrication process can be set up in a cost effective manner to produce large quantities of generic (unprogrammed) FPGAs. The customization (programming) of a FPGA is rather simple. Given a circuit, it is decomposed into smaller subcircuits, such that each subcircuit can be mapped to a logic block. The interconnections between any two subcircuits is achieved by programming the FPGA interconnects between their corresponding logic blocks. Programming (blowing) one of the fuses (antifuse or cross fuse) provides a low resistance bidirectional connection between two segments. When blown, antifuses connect the two segments to form a longer one. In order to program a fuse, a high voltage is applied across it. FPGAs have special circuitry to program the fuses. The circuitry consists of the wiring segments and control logic at the periphery of the chip. Fuse addresses are shifted into the fuse programming circuitry serially. Figure 1.9(a) shows a circuit partitioned into four subcircuits, $P_1, P_2, P_3$, and $P_4$. Note that each of these four subcircuits have two inputs and one output. The truth table for each of the subcircuits is shown in Figure 1.9(b). In Figure 1.9(c), $P_1, P_2, P_3$, and $P_4$ are mapped to logic blocks $B_1, B_4, B_7$, and $B_{10}$ respectively and appropriate antifuses and cross fuses are programmed (burnt) to implement the entire circuit. The programmed fuses are shown as filled circles and rectangles. We have described the 'once-program' type of FPGAs. Many FPGAs allow the user to re-program the interconnect, as many times as needed. These FPGAs use non-destructive methods of programming, such as pass-transistors.

The programmable nature of these FPGAs requires new CAD algorithms to make effective use of logic and routing resources. The problems involved in customization of a FPGA are somewhat different from those of other design

styles; however, many steps are common. For example, the partition problem of FPGAs is different than partitioning the problem in all design style while the placement and the routing is similar to gate array approach. These problems will be discussed in detail in Chapter 11.

### 1.5.5 Sea of Gates

The sea of gates is an improved gate array in which the master is filled completely with transistors. The master of the sea-of-gates has a much higher density of logic implemented on the chip, and allows a designer to fabricate complex circuits, such as RAMs, to be built. In the absence of routing channels, interconnects have to be completed either by routing through gates, or by adding more metal or polysilicon interconnection layers. There are problems associated with either solution. The former reduces the gate utilization; the latter increases the mask count and increases fabrication time and cost.

### 1.5.6 Comparison of Different Design Styles

The choice of design style depends on the intended functionality of the chip, time-to-market and total number of chips to be manufactured. It is common to use full-custom design style for microprocessors and other complex high volume applications, while FPGAs may be used for simple and low volume applications. However, there are several chips which have been manufactured by using a mix of design styles. For large circuits, it is common to partition the circuit into several small circuits which are then designed by different teams. Each team may use a different design style or a number of design styles. Another factor complicating the issue of design style is re-usability of existing designs. It is a common practice to re-use complete or partial layout from existing chips for new chips to reduce the cost of a new design. It is quite typical to use standard cell and gate array design styles for smaller and less complex Application Specific ICs (ASICs), while microprocessors are typically full-custom with several standard cell blocks. Standard cell blocks can be laid out using logic synthesis tools.

Design styles can be seen as a continuum from very flexible (full-custom) to a rather rigid design style (FPGA) to cater to differing needs. Table 1.1 summarizes the differences in cell size, cell type, cell placement and interconnections in full-custom, standard cell, gate array and FPGA design styles. Another comparison may be on the basis of area, performance, and the number of fabrication layers needed. (See Table 1.2). As can be seen from the table, full-custom provides compact layouts for high performance designs but requires a considerable fabrication effort. On the other hand, a FPGA is completely pre-fabricated and does not require any user specific fabrication steps. However, FPGAs can only be used for small, general purpose designs.

| | style | | | |
|---|---|---|---|---|
| | full-custom | standard cell | gate array | FPGA |
| cell size | variable | fixed height* | fixed | fixed |
| cell type | variable | variable | fixed | programmable |
| cell placement | variable | in row | fixed | fixed |
| interconnections | variable | variable | variable | programmable |
| design cost | high | medium | medium | low |

Table 1.1: Comparison of different design styles.
* uneven height cells are also used.

| | style | | | |
|---|---|---|---|---|
| | full-custom | standard cell | gate array | FPGA |
| Area | compact | compact to moderate | moderate | large |
| Performance | high | high to moderate | moderate | low |
| Fabricate | All layers | All layers | Routing layers only | No layers |

Table 1.2: Area, Performance and Fabrication layers for different design styles.

## 1.6 System Packaging Styles

The increasing complexity and density of semiconductor devices are the key driving forces behind the development of more advanced VLSI packaging and interconnection approaches. Two key packaging technologies being used currently are Printed Circuit Boards (PCB) and Multi-Chip Modules (MCMs). Let us first start with die packaging techniques.

### 1.6.1 Die Packaging and Attachment Styles

Dies can be packaged in a variety of styles depending on cost, performance and area requirements. Other considerations include heat removal, testing and repair.

#### 1.6.1.1 Die Package Styles

ICs are packaged into ceramic or plastic carriers called Dual In-Line Packages (DIPs), then mounted on a PCB. These packages have leads on 2.54 mm centers on two sides of a rectangular package. PGA (Pin Grid Array) is a package in which pins are organized in several concentric rectangular rows. DIPs and PGAs require large thru-holes to mount them on boards. As a result, thru-hole assemblies were replaced by Surface Mount Assemblies (SMAs). In SMA,

pins of the device do not go through the board, they are soldered to the surface of the board. As a result, devices can be placed on both sides of the board. There are two types of SMAs; leaded and leadless. Both are available in quad packages with leads on 1.27, 1.00, or 0.635 mm centers. Yet another variation of SMA is the Ball Grid Array (BGA), which is an array of solder balls. The balls are pressed on to the PCB. When a BGA device is placed and pressed the balls melt forming a connection to the PCB. All the packages discussed above suffer from performance degradation due to delays in the package. In some applications, a naked die is used directly to avoid package delays.

#### 1.6.1.2 Package and Die Attachment Styles

The chips need to be attached to the next level of packaging, called system level packaging. The leads of pin based packages are bent down and are soldered into plated holes which go inside the printed circuit board. (see Figure 1.10). SMAs such as BGA do not need thru holes but still require a relatively large footprint.

In the case of naked dies, die to board connections are made by attaching wires from the I/O pads on the edge of the die to the board. This is called the wire bond method, and uses a robotic wire bonding machine. The active side of the die faces away from the board. Although package delays are avoided in wire bonded dies, the delay in the wires is still significant as compared to the interconnect delay on the chip.

*Controlled Collapsed Chip Connection* (C4) is another method of attaching a naked die. This method aims to eliminate the delays associated with the wires in the wire bond method. The I/O pins are distributed over the die (ATM style) and a solder ball is placed over the I/O pad. The die is then turned over, such that the active side is facing the board, then pressure is applied to fuse the balls to the board.

The exact layout of chips on PCBs and MCMs is somewhat equivalent to the layout of various components in a VLSI chip. As a result, many layout problems such as partitioning, placement, and routing are similar in VLSI and packaging. In this section, we briefly outline the two commonly used packaging styles and the layout problems with these styles.

### 1.6.2 Printed Circuit Boards

A Printed Circuit Board (PCB) is a multi-layer sandwich of routing layers. Current PCB technology offers as many as 30 or more routing layers. Via specifications are also very flexible and vary, such that a wide variety of combinations is possible. For example, a set of layers can be connected by a single via called the *stacked via.* The traditional approach of single chip packages on a PCB have intrinsic limitations in terms of silicon density, system size, and contribution to propagation delay. For example, the typical inner lead bond pitch on VLSI chips is 0.0152 cm. The finest pitch for a leaded chip carrier is 0.0635 cm. The ratio of the area of the silicon inside the package to the package area

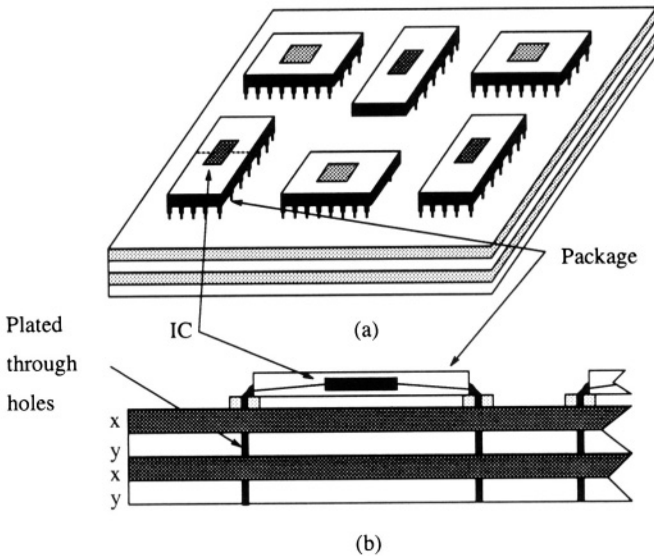



Figure 1.10: Chip placement on a printed circuit board.

is about 6%. If a PCB were completely covered with chip carriers, the board would only have at most a 6% efficiency of holding silicon. In other words, 94% or more of the board area would be wasted space, unavailable to active silicon and contributing to increased propagation delays. Thru-hole assemblies gave way to Surface Mount Assemblies (SMAs). SMAs eliminated the need for large diameter plated-thru-holes, allowing finer pitch packages and increasing routing density. SMAs reduce the package footprint and improve performance.

The SMA structure reduces package footprints, decreases chip-to-chip distances and permits higher pin count ICs. A 64 pin leadless chip carrier requires only a 12.7 mm × 12.7 mm footprint with a 0.635 mm pitch. This space conservation represents a twelve fold density improvement, or a four fold reduction in interconnection distances, over DIP assemblies.

The basic package selection parameter is the pin count. DIPs are used for chips with no more than 48 pins. PGAs are used for higher pin count chips. BGAs are used for even higher pin count chips. Other parameters include power consumption, heat dissipation and size of the system desired.

The layout problems for printed circuit boards are similar to layout problems in VLSI design, although printed circuit boards offer more flexibility and a wider variety of technologies. The routing problem is much easier for PCBs due to the availability of many routing layers. The planarity of wires in each layer is a requirement in a PCB as it is in a chip. There is little distinction between global routing and detailed routing in the case of circuit boards. In fact, due to the availability of many layers, the routing algorithm has to be

modified to adapt to this three dimensional problem. Compaction has no place in PCB layout due to the constraints caused by the fixed location of the pins on packages.

For more complex VLSI devices, with 120 to 196 I/Os, even the surface mounted approach becomes inefficient and begins to limit system performance. A 132 pin device in a $635\ \mu m$ pitch carrier requires a 25.4 to $38.1\ mm^2$ footprint. This represents a four to six fold density loss, and a two fold increase in interconnect distances as opposed to a 64 pin device. It has been shown that the interconnect density for current packaging technology is at least one order of magnitude lower than the interconnect density at the chip level. This translates into long interconnection lengths between devices and a corresponding increase in propagation delay. For high performance systems, the propagation delay is unacceptable. It can be reduced to a great extent by using SMAs such as BGAs. However, a higher performance packaging and interconnection approach is necessary to achieve the performance improvements promised by VLSI technologies. This has led to the development of multi-chip modules.

### 1.6.3 Multichip Modules

Current packaging and interconnection technology is not complementing the advances taking place in the IC. The key to semiconductor device improvements is the shrinking feature size, i.e., the minimum gate or line width on a device. The shrinking feature size provides increased gate density, increased gates per chip and increased clock rates. These benefits are offset by an increase in the number of I/Os and an increase in chip power dissipation. The increased clock rate is directly related to device feature size. With reduced feature sizes each on-chip device is smaller, thereby having reduced parasitics, allowing for faster switching. Furthermore, the scaling has reduced on-chip gate distances and, consequently, interconnect delays. However, much of the improvement in system performance promised by the ever increasing semiconductor device performance has not been realized. This is due to the performance barriers imposed by todays packaging and interconnection technologies.

Increasingly more complex and dense semiconductor devices are driving the development of advanced VLSI packaging and interconnection technology to meet increasingly more demanding system performance requirements. The alternative approach to the interconnect and packaging limits of conventional chip carrier/PCB assemblies is to eliminate packaging levels between the chip and PCB. One such approach uses MCMs. The MCM approach eliminates the single chip package and, instead, mounts and interconnects the chips directly onto a higher density, fine pitch interconnection substrate. Dies are wire bonded to the substrate or use a C4 bonding. In some MCM technologies, the substrate is simply a silicon wafer, on which layers of metal lines have been patterned. This substrate provides all of the chip-to-chip interconnections within the MCM. Since the chips are only one tenth of the area of the packages, they can be placed closer together on an MCM. This provides for both higher density assemblies, as well as shorter and faster interconnects. Figure 1.11 shows

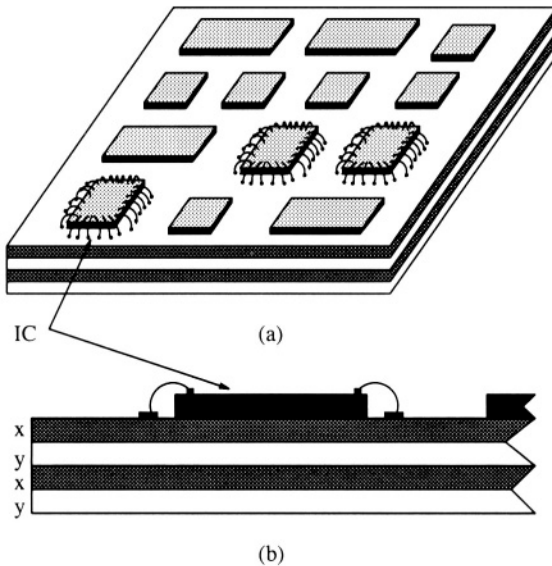



Figure 1.11: A multi-chip module with wire bonded dies.

diagram of an MCM package with wire bonded dies. One significant problem with MCMs is heat dissipation. Due to close placement of potentially several hundred chips, a large amount of heat needs to be dissipated. This may require special, and potentially expensive heat removal methods.

At first glance, it appears that it is easy to place bare chips closer and closer together. There are, however, limits to how close the chips can be placed together on the substrate. There is, for example, a certain peripheral area around the chip which is normally required for bonding, engineering change pads, and chip removal and replacement.

It is predicted that multichip modules will have a major impact on all aspects of electronic system design. Multichip module technology offers advantages for all types of electronic assemblies. Mainframes will need to interconnect the high numbers of custom chips needed for the new systems. Cost-performance systems will use the high density interconnect to assemble new chips with a collection of currently available chips, to achieve high performance without time-consuming custom design, allowing quick time-to-market.

In the long term, the significant benefits of multichip modules are: reduction in size, reduction in number of packaging levels, reduced complexity of the interconnection interfaces and the fact that the assemblies will clearly be cheaper and more efficient. However, MCMs are currently expensive to manufacture due to immature technology. As a result, MCMs are only used in high performance applications. The multichip revolution in the 1990s will have an impact on electronics as great or greater than the impact of surface mount

technology in the 1980s.

The layout problems in MCMs are essentially performance driven. The partitioning problem minimizes the delay in the longest wire. Although placement in MCM is simple as compared to VLSI, global routing and detailed routing are more complex in MCM because of the large number of layers present in MCM. The critical issues in routing include the effect of cross-talk, and delay modeling of long interconnect wires. These problems will be discussed in more detail in Chapter 12.

### 1.6.4 Wafer Scale Integration

MCM packaging technology does not completely remove all the barriers of the IC packaging technology. Wafer Scale Integration (WSI) is considered as the next major step, bringing with it the removal of a large number of barriers. In WSI, the entire wafer is fabricated with several types of circuits, the circuits are tested, and the defect-free circuits are interconnected to realize the entire system on the wafer.

The attractiveness of WSI lies in its promise of greatly reduced cost, high performance, high level of integration, greatly increased reliability, and significant application potential. However, there are still major problems with WSI technology, such as redundancy and yield, that are unlikely to be solved in the near future. Another significant disadvantage of the WSI approach is its inability to mix and match dies from different fabrication processes. The fabrication process for microprocessors is significantly different than the one for memories. WSI would force a microprocessor and the system memory to be fabricated on the same process. This is significant sacrifice in microprocessor performance or memory density, depending on the process chosen.

### 1.6.5 Comparison of Different Packaging Styles

In this section, we compare different packaging styles which are either being used today or might be used in future. In [Sag89] a figure of merit has been derived for various technologies, using the product of the propagation speed (inches/$10^{-9}$sec) and the interconnection density (inches/sq. in). The typical figures are reproduced here in Table 1.3. The figure of merit for VLSI will need to be partially adjusted (downward) to account for line resistance and capacitance. This effect is not significant in MCMs due to higher line conductivity, lower drive currents, and lower output capacitance from the drivers.

MCM technology provides a density, performance, and cost comparable to or better than, WSI. State-of-the-art chips can be multiple-sourced and technologies can be mixed on the same substrate in MCM technology. Another advantage of MCM technology is that all chips are pretestable and replaceable. Furthermore, the substrate interconnection matrix itself can be pretested and repaired before chip assembly; and test, repair, and engineering changes are possible even after final assembly. However, MCM technology is not free of all problems. The large number of required metallurgical bonds and heat removal

| Technology | Figure of Merit (inches/psec . density inches/sq in) |
|---|---|
| WSI | 28.0 |
| MCM | 14.6 |
| PCB | 2.2 |

Table 1.3: Figure of merit speed-density.

are two of the existing problems. While WSI has higher density than MCM, its yield problem makes it currently unfeasible. The principal conclusion that can be drawn from this comparison is that WSI cannot easily compete with technology already more or less well established in terms of performance, density, and cost.

## 1.7 Historical Perspectives

During the 1950s the photolithographic process was commonly used in the design of circuits. With this technology, an IC was created by fabricating transistors on crystalline silicon. The design process was completely manual. An engineer would create a circuit on paper and assemble it on a breadboard to check the validity of the design. The design was then given to a layout designer, who would draw the silicon-level implementation. This drawing was cut out on rubylith plastic, and carefully inspected for compliance with the original design. Photolithographic masks were produced by optically reducing the rubylith design and these masks were used to fabricate the circuit [Feu83].

In the 1970s there was a tremendous growth in circuit design needs. The commonly used rubylith patterns became too large for the laboratories. This technology was no longer useful. Numerically controlled pattern generation machinery was implemented to replace the rubylith patterns. This was the first major step towards design automation. The layouts were transferred to data tapes and for the first time, design rule checking could be successfully automated [Feu83].

By the 1970s a few large companies developed interactive layout software which portrayed the designs graphically. Soon thereafter commercial layout systems became available. This interactive graphics capability provided rapid layout of IC designs because components could quickly be replicated and edited, rather than redrawn as in the past [Feu83]. For example, L-Edit is one such circuit layout editor commercially available. In the next phase, the role of computers was explored to help perform the manually tedious layout process. As the layout was already in the computer, routing tools were developed initially to help perform the connections on this layout, subject to the design rules specified for that particular design.

As the technology and tools are improving, the VLSI physical design is

moving towards high performance circuit design. The high-performance circuit design is of highest priority in physical design. Current technology allows us to interconnect over the cells/blocks to reduce the total chip area, thereby reducing the signal delay for high performance circuits. Research on parallel algorithms for physical design has also drawn great interest since the mid 80s. The emergence of parallel computers promises the feasibility of automating many time consuming steps of physical design.

In the early decades, most aspects of VLSI design were done manually. This elongated the design process, since any changes to improve any design step would require a revamping of the previously performed steps, thus resulting in a very inefficient design. The introduction of computers in this area accelerated some aspects of design, and increased efficiency and accuracy. However, many other parts could not be done using computers, due to the lack of high speed computers or faster algorithms. The emergence of workstations led to the development of CAD tools which made designers more productive by providing the designers with 'what if' scenarios. As a result, the designers could analyze various options for a specific design and choose the optimal one. But there are some features of the design process which are not only expensive, but also too difficult to automate. In these cases the use of certain knowledge based systems is being considered. VLSI design became interactive with the availability of faster workstations with larger storage and high-resolution graphics, thus breaking away from the traditional batch processing environment. The workstations also have helped in the advancement of integrated circuit technology by providing the capabilities to create complex designs. Table 1.4 lists the development of design tools over the years.

## 1.8 Existing Design Tools

Design tools are essential for the *correct-by-construction* approach, that is get the design right the very first time. Any design tool should have the following capabilities.

- layout the physical design for which the tool should provide some means of schematic capture of the information. For this either a textual or interactive graphic mode should be provided.
- physical verification which means that the tool should have design rule checking capability.
- some form of simulation to verify the behavior of the design.

There are tools available with some of the above mentioned capabilities. For example, BELLE (Basic Embedded Layout Language) is a language embedded in PASCAL in which the layout can be designed by textual entry. ABCD (A Better Circuit Description) is also a language for CMOS and nMOS designs. The graphical entry tools, on the other hand, are very convenient for the designers, since such tools operate mostly through menus. KIC, developed at

| Year | Design Tools |
|---|---|
| 1950-1965 | Manual design |
| 1965-1975 | Layout editors<br>Automatic routers (for PCB)<br>Efficient partitioning algorithm |
| 1975-1985 | Automatic placement tools<br>Well defined phases of design of circuits<br>Significant theoretical development in all phases |
| 1985-1990 | Performance driven placement and routing tools<br>Parallel algorithms for physical design<br>Significant development in underlying graph theory<br>Combinatorial optimization problems for layout |
| 1990-1995 | Over-the-Cell Routing tools<br>Three dimensional interconnect based physical design<br>Synthesis tools mature and gain widespread acceptance |
| 1995 - Present | Interconnect design and Modelling dominates physical design<br>Process related tools (reliability, Electro-migration) |

Table 1.4: Design Tools Development.

the University of California, Berkeley and PLAN, developed at the University of Adelaide, are examples of such tools. Along with the workstations came peripherals, such as plotters and printers with high-resolution graphics output facilities which gave the designer the ability to translate the designs generated on the workstation into hardcopies.

The rapid development of design automation has led to the proliferation of CAD tools for this purpose. Some tools are oriented towards the teaching of design automation to the educational community, while the majority are designed for actual design work. Some of the commercially available software is also available in educational versions, to encourage research and development in the academic community. Some of the design automation CAD software available for educational purposes are L-Edit, MAGIC, SPICE etc. We shall briefly discuss some of the features of L-Edit and MAGIC.

L-Edit is a graphical layout editor that allows the creation and modification of IC mask geometry. It runs on most PC-family computers with a Graphics adapter. It supports files, cells, instances, and mask primitives. A file in L-Edit is made up of cells. An independent cell may contain any number of combinations of mask primitives and instances of other cells. An instance is a copy of a cell. If a change is made in an instanced cell, the change is reflected in

all instances of that cell. There may be any number of levels in the hierarchy.

In L-Edit files are self-contained, which means that all references made in a file relate only to that file. Designs made by L-Edit are only limited by the memory of the machine used. Portability of designs is facilitated by giving a facility to convert designs to CIF (Caltech Intermediate Format) and vice versa. L-Edit itself uses a SLY (Stack Layout Format) which can be used if working within the L-Edit domain. The SLY is like the CIF with more information about the last cell edited, last view and so on. L-edit exists at two levels, as a low-level full-custom mask editor and a high-level floor planning tool.

MAGIC is an interactive VLSI layout design software developed at the University of California, Berkeley. It is now available on a number of systems, including personal computers. It is based on the Mead and Conway design style. MAGIC is a fairly advanced editor. MAGIC allows automatic routing, stretching and compacting cells, and circuit extraction to name a few. All these functions are executed, as well as concurrent design rule checking which identifies violations of design rules when any change is made to the circuit layout. This reduces design time as design rule checking is done as an event based checking rather than doing it as a lengthy post-layout operation as in other editors. This carries along with it an overhead of time to check after every operation, but this is certainly very useful when a small change is introduced in a large layout and it can be known immediately if this change introduces errors in the layout rather than performing a design rule check for the whole layout.

MAGIC is based on the corner stitched data structure proposed by Ousterhout [SO84]. This data structure greatly reduces the complexity of many editing functions, including design rule checking. Because of the ease of design using MAGIC, the resulting circuits are 5-10% denser than those using conventional layout editors. This density tradeoff is a result of the improved layout editing which results in a lesser design time. MAGIC permits only Manhattan designs and only rectilinear paths in designing circuits. It has a built-in hierarchical circuit extractor which can be used to verify the design, and has an on-line help feature.

## 1.9 Summary

The sheer size of the VLSI circuit, the complexity of the overall design process, the desired performance of the circuit and the cost of designing a chip dictate that CAD tools should be developed for all the phases. Also, the design process must be divided into different stages because of the complexity of entire process. Physical design is one of the steps in the VLSI design cycle. In this step, each component of a circuit is converted into a set of geometric patterns which achieves the functionality of the component. The physical design step can further be divided into several substeps. All the substeps of physical design step are interrelated. Efficient and effective algorithms are required to solve different problems in each of the substeps. Good solutions at each step

are required, since a poor solution at an earlier stage prevents a good solution at a later stage. Despite significant research efforts in this field, CAD tools still lag behind the technological advances in fabrication. This calls for the development of efficient algorithms for physical design automation.

## Bibliographic Notes

Physical design automation is an active area of research where over 200 papers are published each year. There are several conferences and journals which deal with all aspects physical design automation in several different technologies. Just like in other fields, the Internet is playing a key role in Physical design research and development. We will indicate the URL of all key conferences, journals and bodies in the following to faciliate the search for information.

The key conference for physical design is *International Symposium on Physical Design* (ISPD), held annually in April. ISPD covers all aspects of physical design. The most prominent conference is EDA is the *ACM/IEEE Design Automation Conference* (DAC), (www.dac.com) which has been held annually for the last thirtyfive years. In addition to a very extensive technical program, this conference features an exhibit program consisting of the latest design tools from leading companies in VLSI design automation. The *International Conference on Computer Aided Design* (ICCAD) (www.iccad.com) is held yearly in Santa Clara and is more theoretical in nature than DAC. Several other conferences, such as the *IEEE International Symposium on Circuits and Systems* (ISCAS) (www.iscas.nps.navy.mil) and the *International Conference on Computer Design* (ICCD), include significant developments in physical design automation in their technical programs. Several regional conferences have been introduced to further this field in different regions of the world. These include the *IEEE Midwest Symposium on Circuits and Systems* (MSCAS), the *IEEE Great Lakes Symposium on VLSI* (GLSVLSI) (www.eecs.umich.edu/glsvlsi/) the *European Design Automation Conference* (EDAC), and the *International Conference on VLSI Design* (vcapp.csee.usf.edu/vlsi99/) in India. There are several journals which are dedicated to the field of VLSI Design Automation which include broad coverage of all topics in physical design. The premier journal is the *IEEE Transactions on CAD of Circuits and Systems* (akebono.stanford.edu/users/nanni/tcad). Other journals such as, *Integration,* the *IEEE Transactions on Circuits and Systems,* and the *Journal of Circuits, Systems and Computers* also publish significant papers in physical design automation. Many other journals occasionally publish articles of interest to physical design. These journals include *Algorithmica, Networks,* the *SIAM journal of Discrete and Applied Mathematics,* and the *IEEE Transactions on Computers.*

The access to literature in Design automation has been recently enhanced by the availability of the Design Automation Library (DAL), which is developed by the ACM Special interest Group on Design Automation (SIGDA). This library is available on CDs and contains all papers published in DAC, ICCAD, ICCD, and IEEE Transactions on CAD of Circuits and Systems.

An important role of the Internet is through the forum of newsgroups, comp. lsi. cad is a newsgroup dedicated to CAD issues, while specialized groups such as comp. lsi. testing and comp. cad. synthesis discuss testing and synthesis topics. Since there are very large number of newsgroups and they keep evolving, the reader is encouraged to search the Internet for the latest topics.

Several online newslines and magazines have been started in last few years. EE Times (www.eet.com) provides news about EDA industry in general. Integrated system design (www.isdmag.com) provides articles on EDA tools in general, but covers physical design as well.

ACM SIGDA (www.acm.org/sigda/) and Design Automation Technical Committee (DATC) (www.computer.org/tab/DATC) of IEEE Computer Society are two representative societies dealing with professional development of the people involved, and technical aspects of the design automation field. These committees hold conferences, publish journals, develop standards, and support research in VLSI design automation.