Chapter-1

A. Introduction

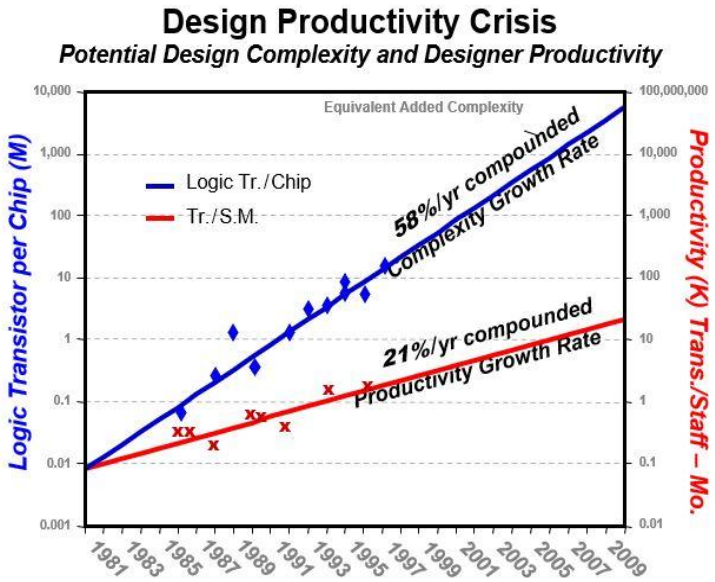
The electronics industry has achieved a tremendous growth over the last few decades, mainly due to the rapid advances in integration technologies and large-scale systems design. The use of integrated circuits in high-performance computing, telecommunications, and consumer electronics has been growing at a very fast pace. Typically, the required computational and information processing power of these applications is the driving force for the fast development of this field. This trend is expected to continue, with very important implications for VLSI and systems design.

As more and more complex functions are required in various data processing and telecommunications devices, the need to integrate these functions in a small package is also increasing. The level of integration as measured by the number of logic gates in a monolithic chip has been steadily rising for almost three decades, mainly due to the rapid progress in processing technology and interconnect technology. Table shows the evolution of logic complexity in integrated circuit over last couple of decades.

Term	Year	Complexity Level
Single Transistor	1958	less than 1
Unit logic	1960	1
Single function	1962	2-4
Multi & complex function	1964	5-20
Medium Scale Integration (MSI)	1967	20-200
Large scale integration (LSI)	1972	200-2000
Very large scale integration (VLSI)	1978	2000-20,000
Ultra large scale integration (ULSI)	1989	>20,000

Moore's Law:

Moore's law is basically a prediction on the growth rate of logic complexity made by Gordon Moore. It says that the transistor count per chip increases exponentially with years (approximately two times every eighteen months).

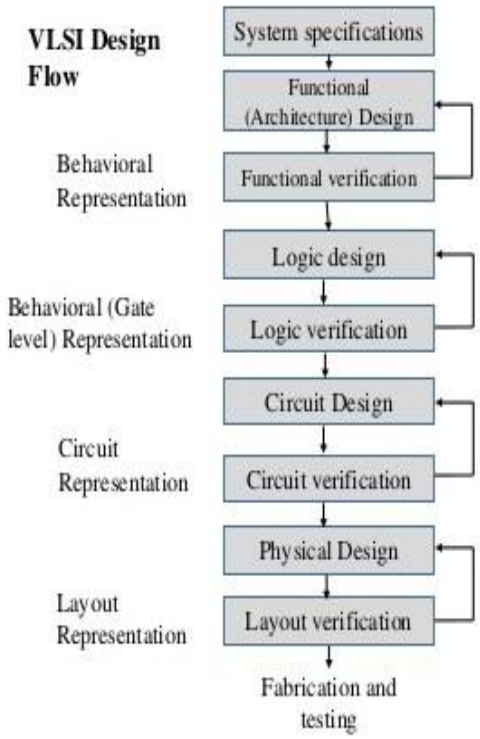


It may be noted that, since the Y-axis is on a logarithmic scale hence the variation is displayed as a straight line. The reduction in transistor size is basically due to the invention of matured fabrication facility, advanced computer aided design tools and libraries, etc. This greatly reduces design time and complexity. Due to the advancement in modern fabrication process flow, more number of transistor can be fabricated inside same silicon area and thus helps the Moore's prediction to became true.

VLSI Design Flow

The flowchart of simplified VLSI design process is shown in figure below. The design process starts with a given specification of system such as which type of job can be performed by the system, whether the system is fully analog or digital or mixed-signal type etc. Also, there may be some specifications regarding the area, power dissipation etc. By keeping all these specifications in mind, the designer decides the type of blocks/modules that will be used and the specification or nature of each module. Sometimes, this type of design approach is termed as "top-down design methodology".

After defining the system level module descriptions, the architecture of each module should be clearly specified. For example, if one designer going to implement one ALU, the components such as adder, multiplier, etc. and their architecture must be clearly defined. There may be different types of adder architecture such as CLA adder, carry-save adder, Manchester carry adder, etc. To simplify the design, the designer may divide larger blocks to smaller sub-blocks. Sometimes the architectural design is also referred as functional design blocks.



VLSI Design Flow Contd...

The architectural or functional design also tells about the input-output pins of the system. Generally, some verification steps are used after every design step. If the design does not satisfy the system specifications, then the designer changes the functional block's architecture to get a more efficient design.

The next step after the functional design is logic design. This step basically explores all the functional blocks and they are represented by their logic level descriptions i.e. in terms of gates, flip-flops, registers, etc. If required, the designer may divide large logical blocks into several smaller parts to manage the design complexity. After the successful logic design, testing and verification are done to ensure the proper operation of the system. For digital systems, generally, hardware description languages (HDL) are involved.

In the next steps, the logic level description of the system is represented by the circuit level description. For example, the logic gates are now represented using MOS transistors at the circuit level. Again there is a testing and verification steps are followed to ensure correctness and also to meet the design metrics such as delay, power, area requirement, etc. Several CAD tools use SPICE to simulate the circuit and to verify the circuit level description.

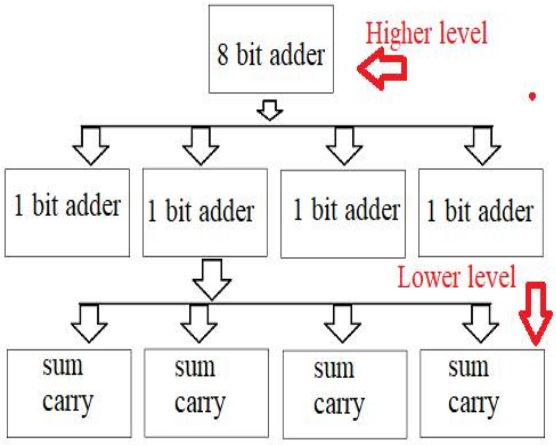
Finally, in the physical design step, all transistors are being replaced by their physical description i.e. layout. This step is very crucial and design metrics such as delay and area requirement strongly depend upon this step. A layout vs schematic analysis is performed to compare layout generated circuit results with schematic level circuit performance. If the post-layout simulation does not meet the design criteria which were mentioned in 1st step i.e. architecture level, the designer may modify the layout to satisfy these matrices.

Although the design process is described in a very brief way with a couple of steps followed by top-down design methodology, the actual design flow involved several steps. In between two steps as shown in the flowchart, there exist several smaller steps. Most of the steps are iterative in nature and for a successful design, a top-down with the bottom-up approach is followed.

Design Strategy

From our previous discussions, it is clear that the VLSI design process is complex in nature. There is some useful technique to reduce the design complexity. These are discussed below.

- **Design Hierarchy:** This is similar to the divide and rule approach. A larger system with unmanageable complexity is further divided into smaller parts. This decomposition process will continue until and unless the complexity of the generated blocks is manageable. For example, one eight bit full adder can be decomposed into eight single bit adder. Now it is much more easier to design a single bit adder. Each single bit adder can be further divided in to sum and carry part. Finally, at the end of design process, all eight adders are connected and form the eight bit adder. The hierarchy process is shown in figure below.



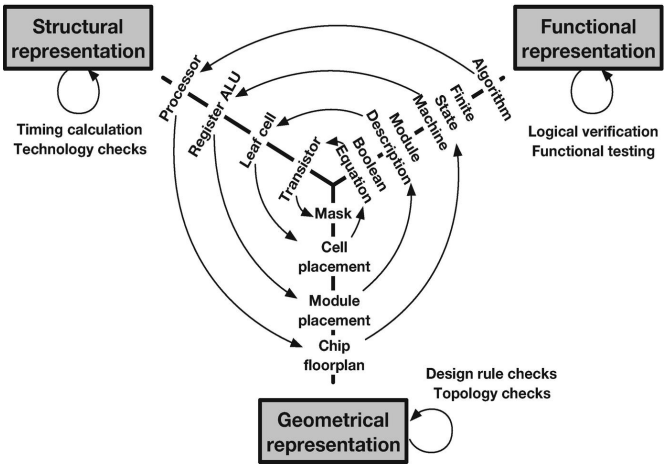
- **Regularity:** At the time of decomposition, the designer must look for not only simple blocks but also similar types of block as much as possible. This strategy reduces the design time. For example, if one designer design a single bit adder, then same design will be repeated to form complete eight bit adder quickly.

- **Modularity:** After decomposition, the generated sub modules must have their well defined , independent functions and interfaces. This property permits the designer to test each sub module independently and to allow design of multiple component parallelly (since each modules are independent to each other).

- **Locality:** This property ensure that the interconnection between sub-modules should be within local places. That means, the modules should be placed closely to each other to avoid long distance wiring and thus reduces interconnect delays and also fabrication cost.

Gajski's Y-Chart

In VLSI design flow, there are three design domains: Functional or Behavioral, structural, and geometric or physical domain. They are strongly correlated to each other. The relationship between the three domains is represented by Gajski's Y chart. The three-arm (design domain) of the chart forms the letter "Y". The pictorial representation of the Y-chart is shown in the figure below.



The behavioral description of an algorithm can be implemented by using a processor. The process can be designed using the layout of the chip i.e. floor planning. Similarly, in the behavioral domain, one algorithm may consist of several finite state machines (FSM) to execute the different tasks. At the structural level, the FSM can be realized using some memory elements such as registers. The same can be designed as a module during chip layout and floor planning.

The next level of abstraction starts with module description, which can be represented by leaf cell i.e. a group of basic logic gates, flip flops, etc. The same can be placed using a cell placement inside the chip. In the next level of abstraction, the Boolean equation is used to describe the functionalities of a module, which can be structurally implemented using transistors followed by the mask/layout of a single transistor.

It is interesting to observe that at any design domain i.e. at any particular arm of "Y", the level of complexity reduces from outer towards the center of "Y", which indicates that not only three domains are correlated but the components present in a particular design domain is also related to each other.

Design Style

A VLSI system sometimes is referred to as Application Specific Integrated Circuit (ASIC). There are several methodologies to implement a VLSI system. Each one having its own advantages and drawbacks. They are (1) Full custom design (2) Semi-custom design and (3) Programmable design style.

(1) Full Custom Design

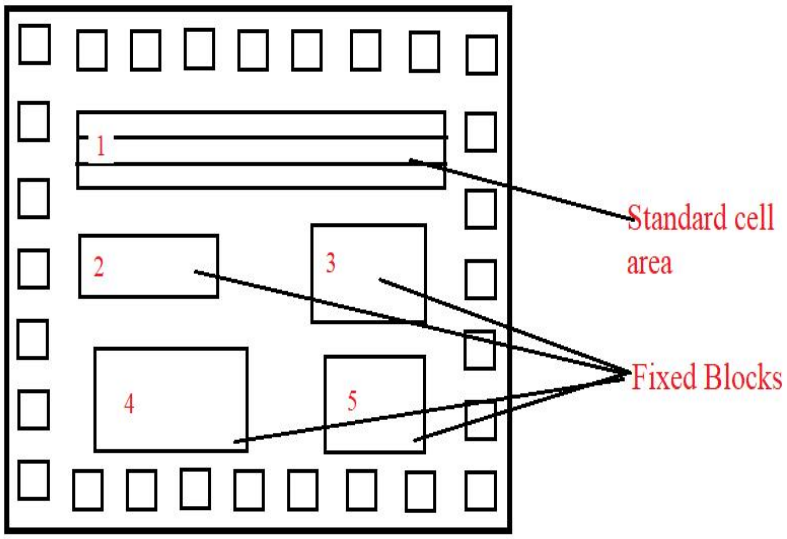
In this design, the designer starts from scratch. This means that the designer abandons the approach of pre-designed and pre-tested and pre-characterized cells for all or part of the design. This might be because existing cell libraries are not fast enough, or the logic cells are not small enough or consume too much power. You may need to use full-custom design if the ASIC technology is new or so specialized that there are no existing cell libraries or because the ASIC is so specialized that some circuits must be custom designed. Fewer and fewer full-custom ICs are being designed because of the problems with these special parts of the ASIC. As a designer, you can optimize your design for speed, power, and area requirements. The full custom design approach needs more time and the cost of prototyping is higher compared to other design styles.

(2) Semi-Custom Design

In this type of design process, some part of the design is customized. That means the designer can pick up a pre-designed and tested cell from the cell library. Each cell library may contain an adder, subtractor, multiplier, flip flops, etc. For a single design, category say adder, there may be different variants with different speed and power ratings. This drastically reduces design time and cost. This is further divided into two categories: standard cell-based and gate array-based design.

Standard Cell Based Design: In a standard cell-based design approach, the designer uses pre-designed logic cells such as gates (AND, OR, NOR, NAND, etc.), multiplexers, flip flops, etc., and they are commonly known as standard cells. Those pre-designed and tested cells are stored in a library. Different cells with different speed grades, power, and area requirements are also available. The chip designer only defines the placement and interconnections between different cells. By using stand cell, a designer can save time, and risk of failure by using a pre-designed tested, and characterized cell library.

The disadvantage of this type of design approach is the expense of purchasing the cell library. Also, it may take time to interconnect all layers. The average manufacturing lead time is six to eight weeks.



Gate Array Based Design:

In this design style, the transistors are predefined on a silicon wafer. These are called a base array or base cell. Only the top few layers of metal which are used to make interconnections are customized. The designer chooses a pre-verified gate array library of logic cells. These are termed as "macros". The designer needs to define the interconnections between the base array. Depending upon the configuration of base cells, there are three different types of gate array-based design, (1) channel-less (2) Channeled, and (3) Structured gate array.

(1) Channel-less gate array:

- Only some of the mask layers are customized.
- Manufacturing lead time between two days to two weeks

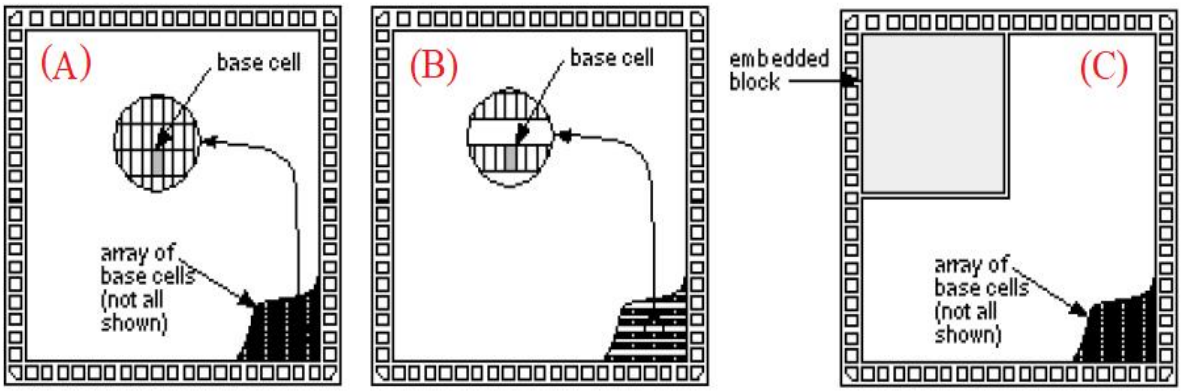
(2) Channeled gate array:

- Only interconnect is customized.
- Interconnect uses predefined spaces between row and base cell.
- Manufacturing time is the same as a channel-less gate array.

(3) Structured gate array:

- Only interconnect is customized.
- Custom blocks can be embedded by the designer.

- Manufacturing time is almost the same as other types of gate array-based design.



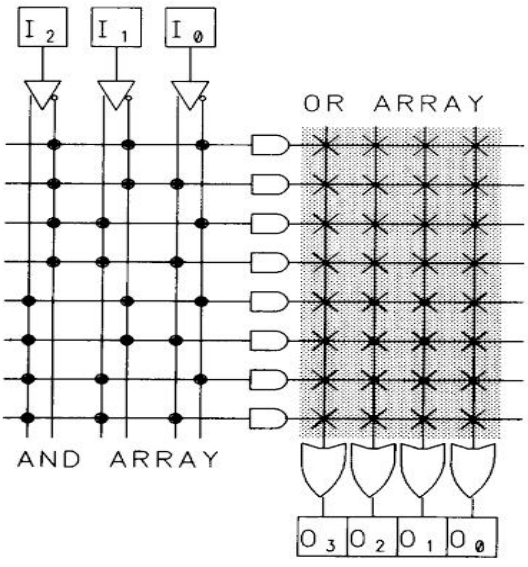
Schematic structure of (A) Channel-less (B) Channelled and (C) structured gate array

Programmable design style:

These are some of the most popular options among the designer and available in the market which are sold at a large volume at affordable cost. These options are adopted by designers for faster prototyping. Since they are designed for a specific application, sometimes they are also termed as "Programmable ASIC".

There are two basic types of programmable ASICs. One is a Programmable logic device (PLD) and another one is Field programmable gate array (FPGA).

- PLDs use programmable AND, OR gates where one programmable switch is placed. The connection is established by activating these switches. They may be one-time programmable or can be re-programmable in nature. The designer only implements the logic function in a CAD tool and after synthesis, a bit file (binary file) is generated. The software-generated bit file takes care of defining interconnection between different logic gates by activating corresponding MOS switches. They are suitable for small to medium-scale logic design. Due to the use of programmable switches, generally, PLDs are not suitable for high-speed operations. Examples of PLDs: Programmable logic array (PLA), Programmable array logic (PAL), Complex programmable logic devices (CPLD).



- Field Programmable Gate Array (FPGA) is another popular choice for programmable ASIC design. Generally, the logic capacity of FPGA is higher than PLDs and also suitable for applications where a large logic capacity is required. FPGA uses SRAM, multiplexers as core components. The logic is implemented using the lookup table

(LUT) approach. The combination of several LUTs, flip flops, and the mux is called a configurable logic block (CLB). Similar to PLDs, programmable interconnects are used to establish interconnections between different CLBs. Input/output buffers are used to connect data with external pins. Design time is faster than PLDs. Hardware description language is used to describe the logic function. After synthesis and successful placement, routing analysis, the bit file is transferred to FPGA and it is configured as the required logic functions. Xilinx, Altera, Actel are the manufacturer of FPGA. Popular FPGA categories are SPARTAN, VIRTEX series from Xilinx, and also others.