



VLSI Design

tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

About the Tutorial

Over the past several years, Silicon CMOS technology has become the dominant fabrication process for relatively high performance and cost effective VLSI circuits. The revolutionary nature of these developments is understood by the rapid growth in which the number of transistors integrated on circuit on single chip. In this tutorial we are providing concept of MOS integrated circuits and coding of VHDL and Verilog language.

Audience

This reference has been prepared for the students who want to know about the VLSI Technology. The students will be able to know about the VHDL and Verilog program coding.

Prerequisites

Before you start proceeding with this tutorial, we make an assumption that you are already aware of the basic concepts of basic concept of Digital Electronics.

Copyright & Disclaimer

© Copyright 2015 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

About the Tutorial	i
Audience.....	i
Prerequisites.....	i
Copyright & Disclaimer	i
Table of Contents	ii
PART 1 – VLSI BASICS	1
1. VLSI – Digital System	2
VLSI Design Flow	2
Y Chart	4
Design Hierarchy-Structural	4
2. VLSI – FPGA Technology	7
FPGA – Introduction	7
Gate Array Design.....	8
Standard Cell Based Design	9
Full Custom Design	10
3. VLSI – MOS Transistor	11
Structure of a MOSFET	11
Working of a MOSFET.....	14
MOSFET Current – Voltage Characteristics	16
4. VLSI – MOS Inverter	19
Principle of Operation	19
Resistive Load Inverter	20
Inverter with N type MOSFET Load	22
Enhancement Load NMOS.....	22
Depletion Load NMOS	23
CMOS Inverter – Circuit, Operation and Description	24
5. VLSI – Combinational MOS Logic Circuits	28
CMOS Logic Circuits.....	28
Complex Logic Circuits.....	31
Complex CMOS Logic Gates.....	32
6. VLSI – Sequential MOS Logic Circuits.....	36
CMOS Logic Circuits.....	37
CMOS Logic Circuits.....	40
PART 2 – VHDL.....	44
7. VHDL – Introduction.....	45
Data Flow Modeling	46
Behavioral Modeling	46
Structural Modeling.....	46
Logic Operation – AND GATE.....	47
Logic Operation – OR Gate	48
Logic Operation – NOT Gate.....	48

Logic Operation – NAND Gate	49
Logic Operation – NOR Gate.....	50
Logic Operation – XOR Gate	51
Logic Operation – X-NOR Gate	52
8. VHDL – Programming for Combinational Circuits	54
VHDL Code for a Half-Adder	54
VHDL Code for a Full Adder	54
VHDL Code for a Half-Subtractor	55
VHDL Code for a Full Subtractor	56
VHDL Code for a Multiplexer	56
VHDL Code for a Demultiplexer.....	57
VHDL Code for a 8 x 3 Encoder:.....	58
VHDL Code for a 3 x 8 Decoder	58
VHDL Code – 4 bit Parallel adder.....	59
VHDL Code – 4 bit Parity Checker.....	61
VHDL Code – 4 bit Parity Generator	61
9. VHDL – Programming for Sequential Circuits	63
VHDL Code for an SR Latch	63
VHDL Code for a D Latch.....	63
VHDL Code for an SR Flip Flop	64
VHDL code for a JK Flip Flop	65
VHDL Code for a D Flip Flop.....	66
VHDL Code for a T Flip Flop	67
VHDL Code for a 4 - bit Up Counter.....	68
VHDL Code for a 4-bit Down Counter.....	69
PART 3 – VERILOG	71
10. Verilog – Introduction	72
Behavioral level	72
Register–Transfer Level	72
Gate Level	72
Lexical Tokens.....	72
Gate Level Modelling.....	73
Data Types	75
Operators	77
Operands	81
Modules.....	82
11. Verilog – Behavioral Modelling & Timing Control	84
Procedural Assignments	84
Delay in Assignment (not for synthesis)	85
Blocking Assignments	86
Nonblocking (RTL) Assignments	86
Conditions.....	87
Delay Controls	91
Procedures: Always and Initial Blocks	92

Part 1 – VLSI Basics

1. VLSI – Digital System

Very-large-scale integration (VLSI) is the process of creating an **integrated circuit** (IC) by combining thousands of **transistors** into a single chip. VLSI began in the 1970s when complex **semiconductor** and **communication** technologies were being developed. The **microprocessor** is a VLSI device.

Before the introduction of VLSI technology, most ICs had a limited set of functions they could perform. An **electronic circuit** might consist of a **CPU, ROM, RAM** and other **glue logic**. VLSI lets IC designers add all of these into one chip.

The electronics industry has achieved a phenomenal growth over the last few decades, mainly due to the rapid advances in large scale integration technologies and system design applications. With the advent of very large scale integration (VLSI) designs, the number of applications of integrated circuits (ICs) in high-performance computing, controls, telecommunications, image and video processing, and consumer electronics has been rising at a very fast pace.

The current cutting-edge technologies such as high resolution and low bit-rate video and cellular communications provide the end-users a marvelous amount of applications, processing power and portability. This trend is expected to grow rapidly, with very important implications on VLSI design and systems design.

VLSI Design Flow

The VLSI IC circuits design flow is shown in the figure below. The various levels of design are numbered and the blocks show processes in the design flow.

Specifications comes first, they describe abstractly, the functionality, interface, and the architecture of the digital IC circuit to be designed.

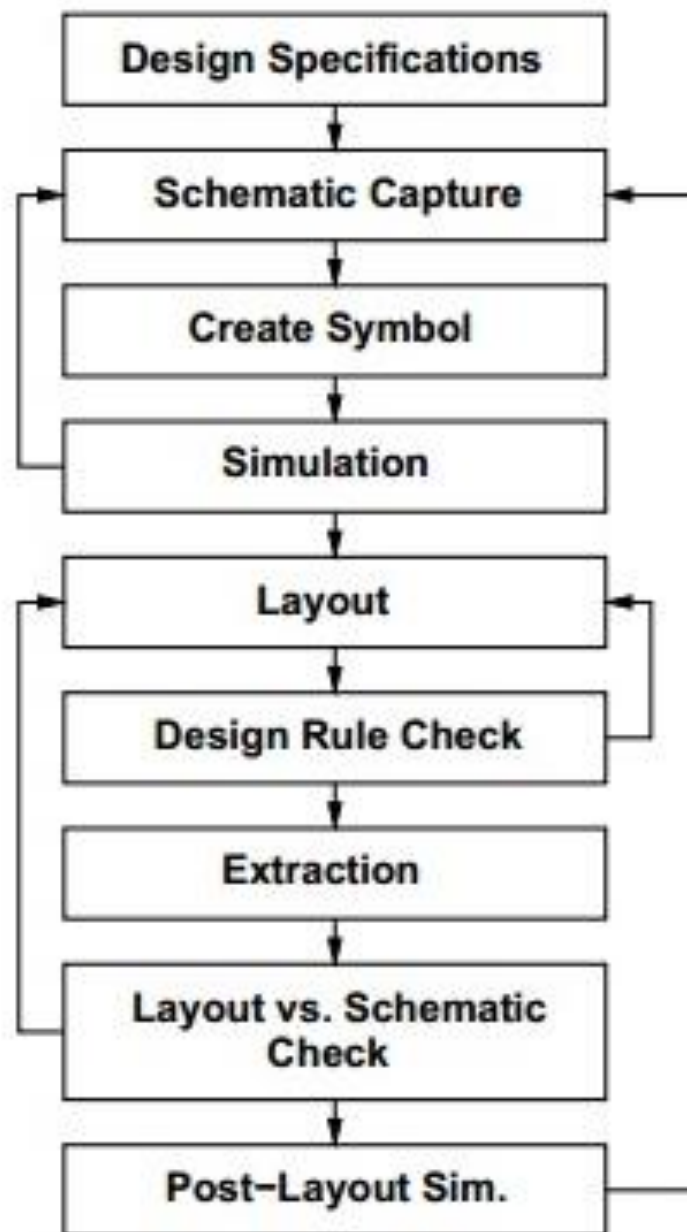


Figure: Simplified VLSI Design Flow

Behavioral description is then created to analyze the design in terms of functionality, performance, compliance to given standards, and other specifications.

RTL description is done using HDLs. This RTL description is simulated to test functionality. From here onwards we need the help of EDA tools.

RTL description is then converted to a gate-level netlist using logic synthesis tools. A gate-level netlist is a description of the circuit in terms of gates and connections between them, which are made in such a way that they meet the timing, power and area specifications.

Finally, a physical layout is made, which will be verified and then sent to fabrication.

Y Chart

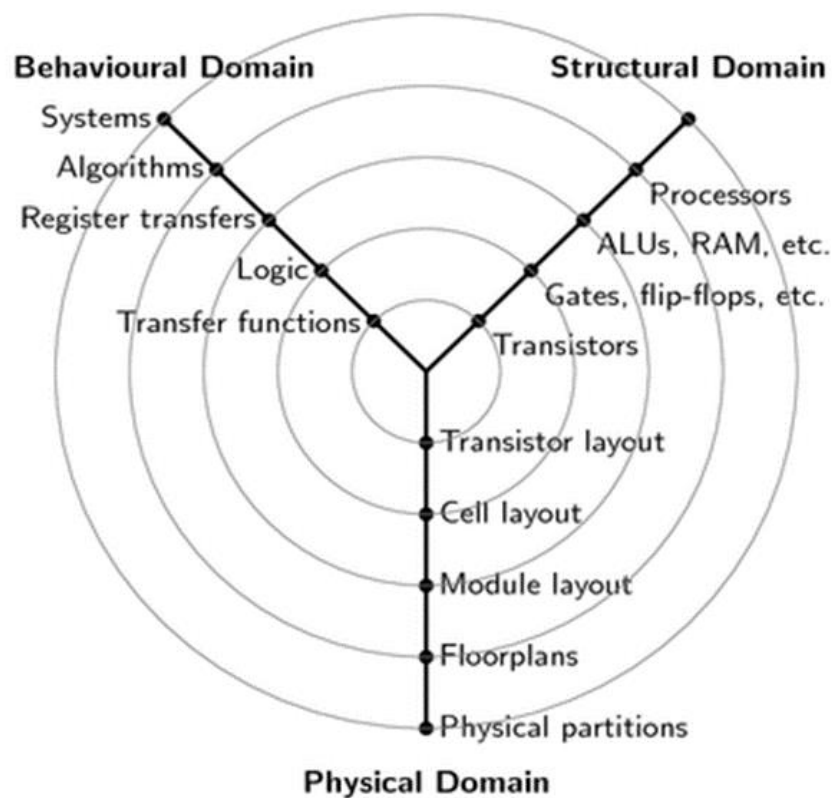
The Gajski-Kuhn Y-chart is a model, which captures the considerations in designing semiconductor devices.

The three domains of the Gajski-Kuhn Y-chart are on radial axes. Each of the domains can be divided into levels of abstraction, using concentric rings.

At the top level (outer ring), we consider the architecture of the chip; at the lower levels (inner rings), we successively refine the design into finer detailed implementation:

Creating a structural description from a behavioral one is achieved through the processes of high-level synthesis or logical synthesis.

Creating a physical description from a structural one is achieved through layout synthesis.



Gajski-Kuhn Y-chart

Figure: Y Chart

Design Hierarchy-Structural

The design hierarchy involves the principle of "Divide and Conquer." It is nothing but dividing the task into smaller tasks until it reaches to its simplest level. This process is most suitable because the last evolution of design has become so simple that its manufacturing becomes easier.

We can design the given task into the design flow process's domain (Behavioral, Structural, and Geometrical). To understand this, let's take an example of designing a 16-bit adder, as shown in the figure below.

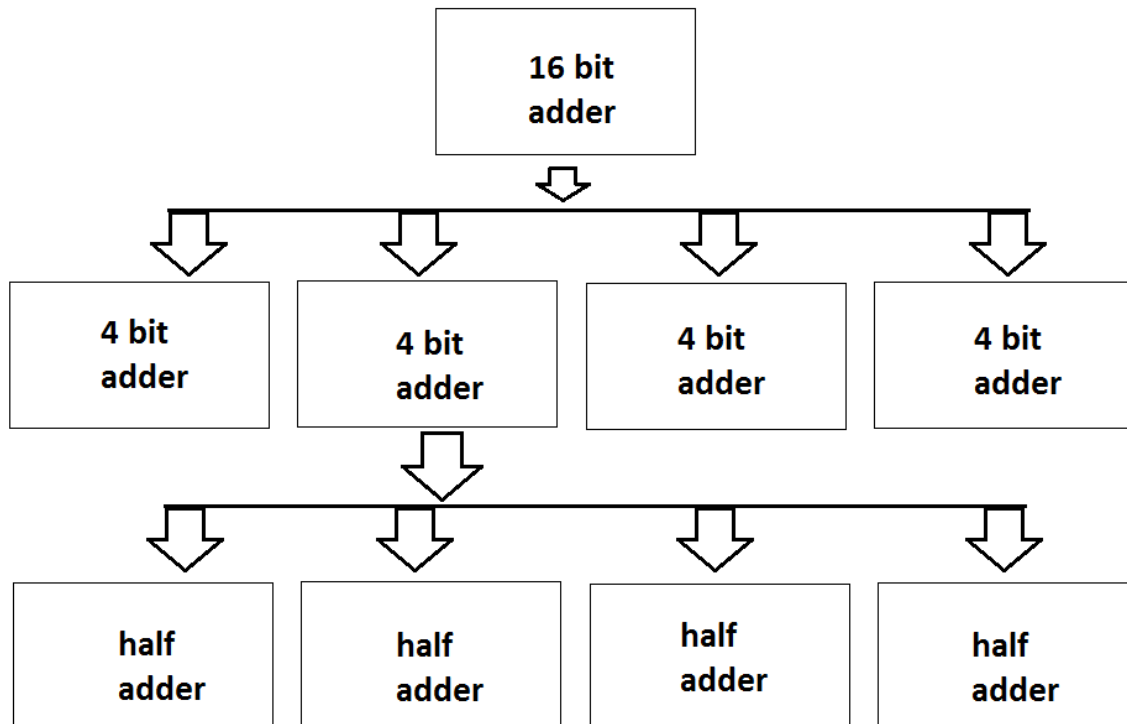


Figure: Structural hierarchy of 16 bit adder circuit

Here, the whole chip of 16 bit adder is divided into four modules of 4-bit adders. Further, dividing the 4-bit adder into 1-bit adder or half adder. 1 bit addition is the simplest designing process and its internal circuit is also easy to fabricate on the chip. Now, connecting all the last four adders, we can design a 4-bit adder and moving on, we can design a 16-bit adder.

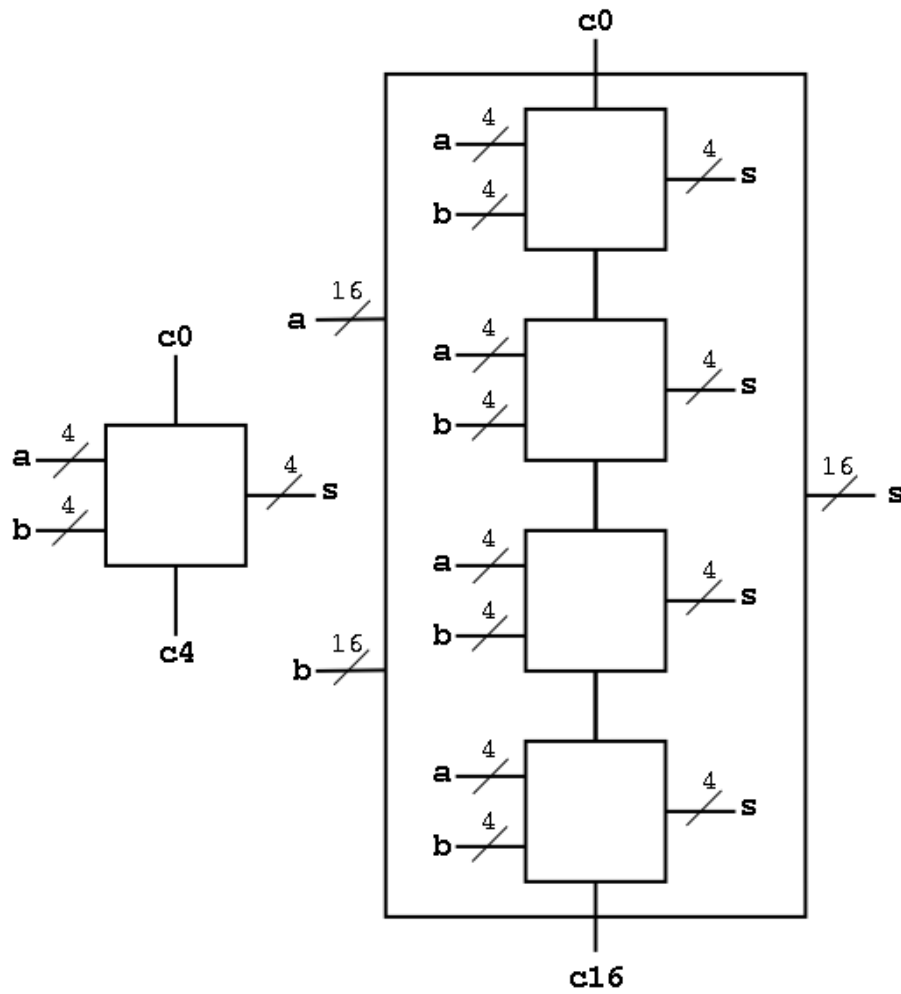


Figure: Decomposition of a 4 bit adder

2. VLSI – FPGA Technology

FPGA–Introduction

The full form of **FPGA** is “**Field Programmable Gate Array**”. It contains ten thousand to more than a million logic gates with programmable interconnection. Programmable interconnections are available for users or designers to perform given functions easily. A typical model FPGA chip is shown in the given figure. There are I/O blocks, which are designed and numbered according to function. For each module of logic level composition, there are **CLB’s (Configurable Logic Blocks)**.

CLB performs the logic operation given to the module. The inter connection between CLB and I/O blocks are made with the help of horizontal routing channels, vertical routing channels and PSM (Programmable Multiplexers).

The number of CLB it contains only decides the complexity of FPGA. The functionality of CLB’s and PSM are designed by VHDL or any other hardware descriptive language. After programming, CLB and PSM are placed on chip and connected with each other with routing channels.

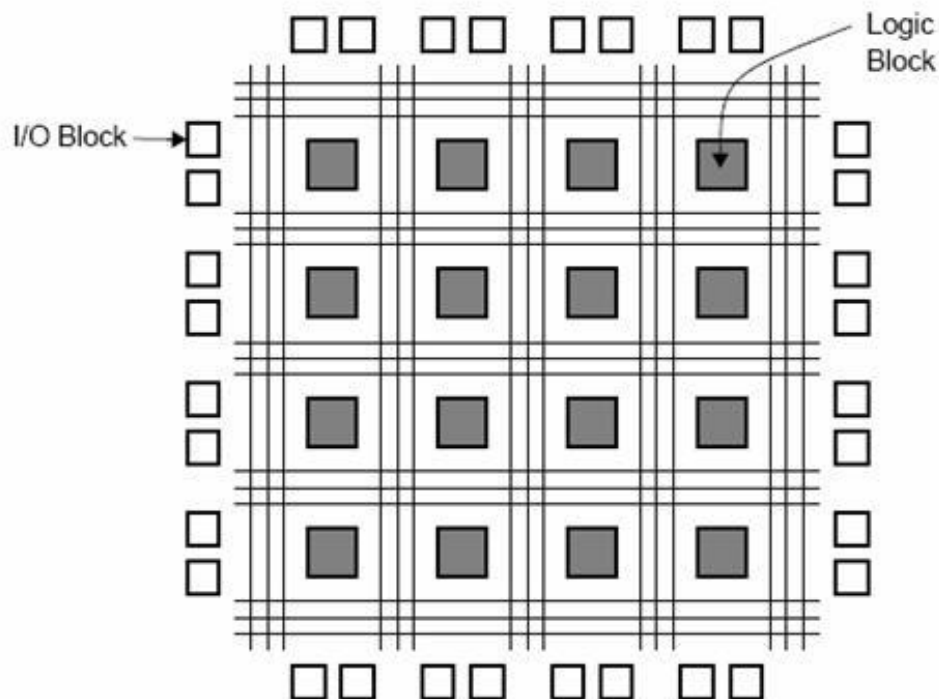


Figure: Interconnection between routing and CLB

Advantages

- It requires very small time; starting from design process to functional chip.
- No physical manufacturing steps are involved in it.
- The only disadvantage is, it is costly than other styles.

Gate Array Design

The **gate array (GA)** ranks second after the FPGA, in terms of fast prototyping capability. While user programming is important to the design implementation of the FPGA chip, metal mask design and processing is used for GA. Gate array implementation requires a two-step manufacturing process.

The first phase results in an array of uncommitted transistors on each GA chip. These uncommitted chips can be stored for later customization, which is completed by defining the metal interconnects between the transistors of the array. The patterning of metallic interconnects is done at the end of the chip fabrication process, so that the turn-around time can still be short, a few days to a few weeks. The figure given below shows the basic processing steps for gate array implementation.

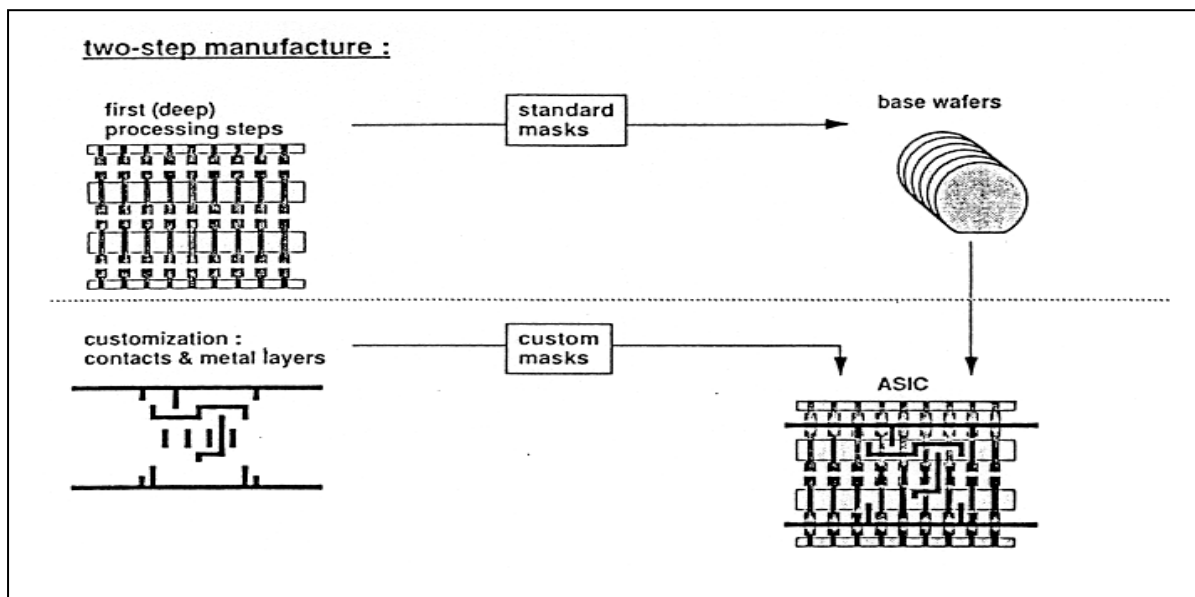


Figure: Basic Processing Steps for gate array implementation

Typical gate array platforms use dedicated areas called channels, for inter-cell routing between rows or columns of MOS transistors. They simplify the interconnections. Interconnection patterns that perform basic logic gates are stored in a library, which can then be used to customize rows of uncommitted transistors according to the netlist.

In most of the modern GAs, multiple metal layers are used for channel routing. With the use of multiple interconnected layers, the routing can be achieved over the active cell areas; so that the routing channels can be removed as in Sea-of-Gates (SOG) chips. Here, the entire chip surface is covered with uncommitted nMOS and pMOS transistors. The neighboring transistors can be customized using a metal mask to form basic logic gates.

For inter cell routing, some of the uncommitted transistors must be sacrificed. This design style results in more flexibility for interconnections and usually in a higher density. GA chip utilization factor is measured by the used chip area divided by the total chip area. It is higher than that of the FPGA and so is the chip speed.

Standard Cell Based Design

A standard cell based design requires development of a full custom mask set. The standard cell is also known as the polycell. In this approach, all of the commonly used logic cells are developed, characterized and stored in a standard cell library.

A library may contain a few hundred cells including inverters, NAND gates, NOR gates, complex AOI, OAI gates, D-latches and Flip-flops. Each gate type can be implemented in several versions to provide adequate driving capability for different fan-outs. The inverter gate can have standard size, double size, and quadruple size so that the chip designer can select the proper size to obtain high circuit speed and layout density.

Each cell is characterized according to several different characterization categories, such as,

- Delay time versus load capacitance
- Circuit simulation model
- Timing simulation model
- Fault simulation model
- Cell data for place-and-route
- Mask data

For automated placement of the cells and routing, each cell layout is designed with a fixed height, so that a number of cells can be bounded side-by-side to form rows. The power and ground rails run parallel to the upper and lower boundaries of the cell. So that, neighboring cells share a common power bus and a common ground bus. The figure shown below is a floorplan for standard-cell based design.

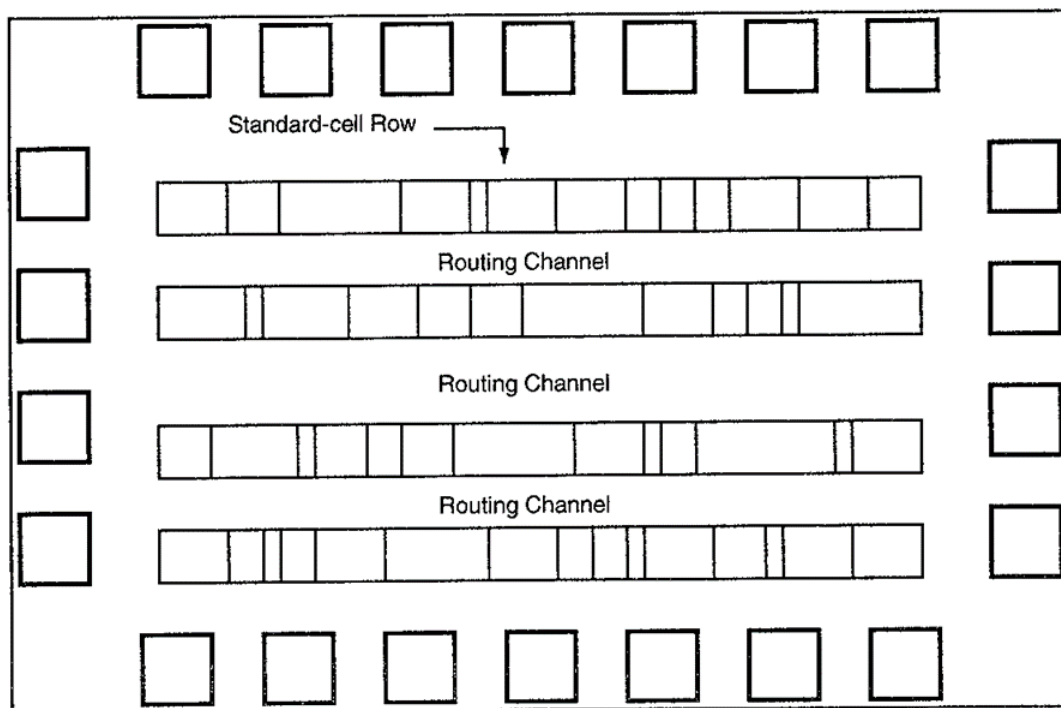


Figure: Floor Plan for Standard Cell Based Design

Full Custom Design

In a full-custom design, the entire mask design is made new, without the use of any library. The development cost of this design style is rising. Thus, the concept of design reuse is becoming famous to reduce design cycle time and development cost.

The hardest full custom design can be the design of a memory cell, be it static or dynamic. For logic chip design, a good negotiation can be obtained using a combination of different design styles on the same chip, i.e. standard cells, data-path cells, and **programmable logic arrays (PLAs)**.

Practically, the designer does the full custom layout, i.e. the geometry, orientation, and placement of every transistor. The design productivity is usually very low; typically a few tens of transistors per day, per designer. In digital CMOS VLSI, full-custom design is hardly used due to the high labor cost. These design styles include the design of high-volume products such as memory chips, high-performance microprocessors and FPGA.

End of ebook preview

If you liked what you saw...

Buy it from our store @ <https://store.tutorialspoint.com>