Journal of
VLSI Design Tools & Technology
https://journals.stmjournals.com/jovdtt

ISSN: 2249-474X (Online)
ISSN: 2321-6492 (Print)
Volume 16, Issue 1, 2026
January–April
DOI (Journal): 10.37591/JoVDTT

A Well-Planned Strategy for a Career Path in VLSI Design

V. Basil Hans*

Abstract

Very large scale integration (VLSI) design is still one of the most active and important areas of the semiconductor industry. It is behind new developments in artificial intelligence (AI), 5G, automotive electronics, and high-performance computing. As companies like Intel, Taiwan Semiconductor Manufacturing Company (TSMC), and NVIDIA work to improve advanced-node technology, the need for qualified VLSI engineers is growing. However, people who want to work in this industry typically do not know what technical skills, tools, and career paths they need to get there. This article gives you a clear plan for how to start a career in VLSI design. It covers the basics of digital electronics, Complementary Metal-Oxide-Semiconductor (CMOS) technology, and semiconductor physics, as well as the important skills you need to know hardware description language (HDL) languages like Verilog and System Verilog. It shows the VLSI design cycle, from register transfer level (RTL) design and verification to physical design and tape-out, while also pointing out tools and methods that are important in the industry. It also talks about internship tactics, project development, certification alternatives, and new trends including low-power design, chiplet designs, and AI-driven electronic design automation (EDA) tools. The goal of this post is to give students and engineers just starting out in their careers clear, tangible steps to take to effectively enter and grow in the competitive VLSI ecosystem.

Keywords: Application-Specific Integrated Circuit (ASIC) design flow, CMOS technology, semiconductor industry, Verilog/SystemVerilog, VLSI design

INTRODUCTION

Very large scale integration (VLSI) has significantly contributed to the growth of microelectronics. Today, VLSI enables chip designers to create complex integrated circuits (ICs) that can perform a variety of functions. Semiconductors, the materials used to make ICs, remain important for VLSI design. This article provides an overview of the VLSI design process and examines the skills and education required to work in VLSI design.

For a relevant description of the design options currently available to circuit designers, VLSI is much better than the large-scale integration that was previously utilized. Chip manufacturing has become more cost-effective, allowing thousands of transistors to be placed on a single chip to meet the strict timing and power requirements of high-performance circuit designers. Therefore, IC

***Author for Correspondence**
V. Basil Hans
E-mail: vhans2011@gmail.com

Research Professor, Department of Commerce & Management and Humanities & Social Sciences, Srinivas University, Mangalore, Karnataka, India

Received Date: March 02, 2026
Accepted Date: March 9, 2026
Accepted Date: March 20, 2026

Citation: V. Basil Hans. A Well-Planned Strategy for a Career Path in VLSI Design. Journal of VLSI Design Tools & Technology. 2026; 16(1): 28–39p.

design procedures must differ significantly from logic design techniques. Ultra-large-scale integration (ULSI) is still the technique used in advanced chip design for IC design, and it remains the main focus of research into new technologies. VLSI is a technique that allows many transistors to link to each other on one chip. It also has a common design process for making complicated circuits, from logic to high-frequency analog [1].

BASICS OF VLSI DESIGN

Digital logic provides ways to show and change the most basic type of information: binary, discrete data. Although digital logic systems can be very different in size and complexity, all digital designs are composed of parts that are connected to each other and produce signal values based on other signal values according to a clear protocol or logic. For instance, combinational designs determine certain signal values solely and deterministically from other signal values without using memory. In contrast, sequential designs use internal memory states whose values depend on earlier values and events. In a sequential design, the signal values change over time. These changes in discrete binary data and their timing are closely related to the analog waveforms that modulate the electromagnetic impulses that pass along the connected connections of the design [2]. A comprehensive foundation for computer arithmetic that addresses stored numeric representations is fundamentally reliant on digital logic. Many mathematical problems are examples of high-performance computing.

Semiconductor physics explains how modern technologies for manufacturing electronic devices work and what materials they use. Transistors and other electronic devices use the nonlinear transfer properties and state-dependent behavior of semiconductor materials to switch and amplify the electrical signals. Even if a designer mostly works with all digital tools, knowing the basic physical principles can help them grasp how realistic a design can be and how certain gadgets can make it less so. You must know how to lay out a design, which is a geometric representation of each device as it was made on a physical die. Electronic design automation (EDA) facilitates the integration of advanced digital systems into chips and circuit boards (CBs) through design, simulation, validation, analysis, and implementation. These products connect the quantum-mechanical behavior of devices, electromagnetic propagation of signals, and binary system of information representation. All design work occurs at the behavioral level, and there are many industry-standard languages and modelling frameworks that may be used to describe this.

Computer Arithmetic and Digital Logic

All digital systems, from supercomputers to small devices, rely on fast logic circuits. In practice, these circuits use Boolean functions on binary integers (0 and 1), which represent different voltage levels. Digital computers are devices that follow a set of instructions to change the binary values. The main parts of a digital computer are registers that store these numbers and arithmetic operations that change the numbers.

People typically use binary-coded decimal (BCD) to represent digits because it allows them to choose and send decimal numbers to other devices. For processing to work well, on-chip storage and manipulation must be efficient. Compact formats, such as ASCII and Extended Binary Coded Decimal Interchange Code (EBCDIC), reduce the number of digits that need to be moved, stored, and processed. Therefore, VLSI students need to learn about digital circuitry, computer math, and BCD representations. A simple Arithmetic Logic Unit (ALU) can process one-digit decimal values using basic gates and registers or a more advanced computer-aided design (CAD) program [3]. Decimal addition, incrementation, and transfer to a BCD register are examples of how circuits use algorithms to perform basic tasks such as extracting and replacing digits [4].

Physics and Devices for Semiconductors

ICs comprise semiconductor devices. This is basic knowledge that anyone who wants to work in the VLSI business must have. These devices are made of materials that can change their conductivity to switch the flow of electric current. In the early 1900s, Hall showed that the conductivity of metals can change significantly with temperature, even in a simple linear resistor [5]. Semiconductor materials are composed of crystals that have either free electrons (for n-type conductivity) or holes (for p-type conductivity). Silicon and germanium, which form crystals that resemble diamonds, are the main materials used in VLSI technology.

The energy bandgap, carrier mobility, and dielectric constant are the three fundamental parameters that define a semiconductor. You may also utilize the temperature dependence of resistivity and noise figure can also be used to determine the properties of semiconductors [6]. Additionally, a comprehensive understanding of carrier transport, encompassing carrier production, recombination, and transport events, is crucial for the accurate characterization of the pulse response in two- or three-terminal semiconductor devices.

Tools for Electronic Design Automation

Electronic design automation tools [7] are required to check whether hardware designs are becoming increasingly complicated. Generating test patterns (ATPG) is an important step for complicated designs. Hierarchical methods are used to simplify ATPG less complicated by using modular design frameworks. To ensure that a design meets the standards, design verification must consider the entire design space. Symbolic simulation is a technology that employs symbols instead of real data to model hardware designs quickly. At the design interface, a signal is given a symbolic representation, and a symbolic simulation run creates the reachability graph. Symbolic trajectory evaluation (STE) is a method for assessing the correctness of bounded state sequences and invariant behaviors by checking their properties. This can be performed without generating test patterns, which makes it easier to check the entire design more thoroughly.

In the physical design automation of VLSI systems, layout analysis involves design-rule-check (DRC) verification, netlist extraction, and electrical parameter extraction for both electrical and physical verification before production [8]. After the mask layout design, layout-versus-schematic (LVS) extraction gets the circuit netlist from the layout and compares it to the original schematic to see if they are the same. The circuit extraction process turns the layout into a machine-readable network. The comparison operation checks for differences between the extracted and schematic netlists. One of the biggest problems with LVS is that the labels for devices and nets are varied between the original and extracted schematics. This typically results in multiple netlists with varying item ordering. As a result, the netlist graph-matching issue, which discovers a quick way to match schematic and extracted netlists, has gotten a lot of interest. You can use LVS in a flat or hierarchical way. Layout extraction gives you a flat netlist that you can use to compare flat against flat. The extraction and matching algorithms' ability to work quickly and the accuracy of the circuit equivalence identification are two of the most important things in LVS. Since 1983, a number of LVS algorithms have been suggested. These include methods for extracting analog circuits and using graph-theory techniques to extract subcircuits.

WAYS TO LEARN

VLSI design is a complicated process that requires both digital and analog design expertise. A design document is a set of specifications that summarizes what the developer believes the needs are. It is at the heart of any VLSI design. Every year, new gadgets are developed, such as system-on-chip devices, microcontrollers, and radio frequency front-end chips. These devices can be used in biomedical applications, consumer electronics, and automotive systems. The first step in making a

VLSI design that is easy to understand is to perform an engineering design. Companies want candidates to know a lot about and be good at front-end and back-end design flows based on what they already know and what they have done well in the past [7].

There will be a list of graduated-designated programming improvements for VLSI design that will help people become proficient in back-end design. The agreed-upon EDA tools used during parking-lot Graphic Data System (GDS) extraction of an analog circuit help people understand how other EDA tools work inside, which leads to Identity and Access Management (IAM) insights on supply voltage, transistor width, and layout topology. Hardware description language (HDL) CAD classes in graduate school provide a focused look at how to use the real basic HDL, which is helpful for beginners, while also covering a wide range of uses. Having sufficient knowledge of the subject matter before you go can help you get the most out of what you learn. Some HDL projects show how to use Chisel and Verilog in depth, both of which are great options for HDL experience.

Basic Degrees and Classes

The most frequent route to becoming a VLSI designer is obtaining a bachelor's degree in a related field [7]. Electronics and Communication Engineering (ECE) is the clear choice for a master's degree. Many schools offer ECE programs that focus on VLSI, microelectronics, or other similar topics; however, this is not necessary.

Digital logic design and computer arithmetic, semiconductor physics, device theory, and circuit theory are some of the most commonly taught subjects. Other common subjects include electronic design automation tools and methods. Depending on the school, the tools may include software programs for Simulation Program with Integrated Circuit Emphasis (SPICE) simulation, HDL simulation, circuit design, schematic capture, and standard cell libraries.

Certifications and Programs for Certain Fields

Many universities and colleges worldwide provide courses that lead to diplomas, degrees, or certificates. Each offers public web portals where one can easily find and view the structure, length, and deliverables of these structured programs. It is important to consider doing a lot of outreaches to these schools starting in the third to last year of the foundational degree programme. People usually think this a few months before they graduate from the basic degree programme or after they obtain the qualification. Every country and area has its own list of the most important, well-known, and respected institutions in the field. Courses offered at any of these schools have a much better chance of graduating with cutting-edge material or curriculum than courses offered at schools that are not on the list, which probably do not get enough industry participation. It is becoming more common to participate in Virtual/Global Design Contests organized by companies such as Cadence, Synopsys, Mentor Graphics, and Artemis. It is easier to connect with other participants and organizers if you start taking the courses well before you finish the core programme. Several individuals have articulated their perspectives regarding the extent to which the pursuit of universities, companies, or external domains may still be enticing for course enrolment; however, tangible outcomes or discourse on the matter remain constrained [7].

Hands-On Projects and Chances to do Research

There are a lot of hands-on projects and research opportunities in integrated circuit design and verification. These include designing, testing, and prototyping a software-programmable I2C/SPI intellectual property (IP) on the Advanced Microcontroller Bus Architecture (AMBA) bus; making dynamic clock switches for car systems; CMOS self-calibrating oscillators; high-precision quartz oscillators; and modelling in hardware description languages like Very High-Speed Integrated Circuit Hardware Description Language - Analog and Mixed-Signal (VHDL-AMS) and Verilog-AMS. The portfolio also talks about ways to find problems early in the design process, as well as ways to check the behavior of mixed-signal systems quickly, especially in automotive applications [9]. Supervised design space exploration is another thing that might happen in projects and research.

This is because modern VLSI technology has problems with long synthesis runtimes, tuning issues, and restricted design iterations. Supervised design space exploration is an abstraction layer between designers and synthesis tools. It manages synthesis jobs for design space exploration independently by iteratively revising knob settings based on machine learning models. This approach has been used to create heterogeneous accelerator cores and high-performance processors at the high-level, logic, and physical synthesis stages. It can also be used for other designs of outside accelerators and programmable cores [10].

CORE SKILLS FOR BUSINESS

Designing ICs at the physical level in VLSI requires skills that can only be learned through extensive training and study. Entry-level design jobs focus on three main areas: front-end design, back-end design, and verification and validation [11].

Register transfer level (RTL) specifications that explain the desired relationships between inputs and outputs are the first step in the front-end design. Logic-level design is based on how things work at the circuit or system level. If automatic synthesis is to be effective, RTL standards must be clear. To ensure that circuits function as intended, synthesis must be performed correctly. People utilize many different standard notations, such as Verilog and VHDL, along with simulators that go with them. To create a design that meets functional, performance, and cost requirements, the real design process uses several data formats, CAD tools, and human knowledge [10].

After synthesis and Layout Schematic Driven (LSD), back-end design work begins. Because user and automated parts are often intermingled, knowing how to use the synthesized RTL and why it works is quite helpful for back-end work. Good timing distribution and custom-designed glue aid result in good performance with a small chip area and low cost.

RTL and Synthesis for Front-End Design

The modern VLSI design process usually starts with schematic diagrams and then makes RTL descriptions to ensure that the design remains the same when it is implemented at the transistor level. RTLs show digital parts correctly, which helps with simulation, formal verification, and hardware synthesis [1]. Hardware description languages, let the description of data and structure in a way that does not interfere with industry-standard EDA tools. RTL's basic idea of RTL, which is seen in both circuit and software designs, makes it easier to perform high-level abstraction synthesis, which is when VLSI circuits may be defined using their delimiters, operators, and transformers [12].

Synthesis is a method that converts HDL into a netlist that does not depend on any specific technology or libraries. It first adds higher-level definitions of operations and then makes lower-level descriptions of how to make commercial ICs, boards, and systems. During synthesis, flexible restrictions are supported, which means that performance, power, precision, and other aspects can be set. However, this is difficult because there is so much freedom [13]. As a result, VLSI design includes many different methods, tools, techniques, settings, and objectives. This means that you must have a good grasp of the underlying principles for all your projects.

Designing the Back End: Place-and-Route and Timing

The layout design starts with the creation of a top-level transistor netlist after the RTL design and synthesis procedure. The digital layout's job is to place these transistors in the right places and wire them up so that they work properly within the timing, area, and power limits [14].

The layout design process usually involves three steps: global placement, routing, and detailed placement. Global placement finds the best starting point for parts so that the overall wire length is minimized. The designer provides a working block-level circuit topology as input. The global routing task provides the actual wiring tracks that connect these nets without breaking the design criteria [15].

Checking and Confirming

Formal verification and simulation are two common methods for checking RTL digital designs. Formal verification provides comprehensive design coverage, making it superior to simulation. A waypoint-based semiformal verification method increases the capacity of formal tools for large designs by examining traces that hit waypoints. This reduces memory and computation time. An automatic waypoint generation program uses synthesized netlist data to find key flip-flops, whereas a neural network scores flip-flops to create waypoints. This method was tested using a tiny First In, First Out (FIFO) example and RISC-V core framework. This shows that waypoint-based semiformal verification and automatic waypoint generation can be very useful for RTL verification [16].

Verification and validation are necessary in design verification to ensure that complicated designs are correct. Hierarchical methods make it easier to automatically generate test patterns by taking advantage of the modular structure built into the designs. Instead of using real values, symbolic simulation employs symbols to quickly check the response of hardware designs. STE method is used to check the properties of bounded state sequences and invariant behavior [7].

Guidelines and Limitations for Physical Design

When designing a VLSI, there are many rules regarding how to build an integrated circuit because there are many gates, the wiring is complicated, and the device must work well. Physicists, circuit designers, system designers, and architects each have their own design tasks. Distinct design fields use distinct design notations (such as transistor, logic, and schematic diagrams), which means that diverse design tools are required. One methodical way to deal with some parts of VLSI design is to examine the design from several levels of abstraction. The physical characteristics of a VLSI circuit affect its construction. Because of these physical limits, architects typically have to consider different topologies that are far off in the design space. One course cannot cover the entire design because VLSI circuits are extremely complicated. Therefore, only a few principles have been considered, with the expectation that they will be helpful when the issue arises in another class or when people actually work on VLSI design [2].

Data Sheets, Standards, and Other Paperwork

VLSI design is the process of making electronic circuits that are more and more complicated. Because these designs are so complicated, there needs to be a way for the people who made them to talk to the people who will utilize them later. To accurately specify electronic circuits, there needs to be a good balance between how detailed the descriptions are and how abstract they are. The datasheet is now the standard in the industry. The IEEE Standards set by the IEEE-Std (Institute of Electrical and Electronics Engineers Standard, the world's largest, oldest, and most respected standards body) have been very important in setting and keeping the levels of clarity that are expected from these kinds of documents. In fact, international students of IC design always ask what the equivalent reference standard is for their native languages. The Jaeger book is easier to get to than the IEEE Standards and has clear examples of appropriate color and style. It also explains the basics of how to write a good description in the context of IC design. Eventually, specialized monographs or graduate theses [7] that talk or graduate theses [7] that talk about the newest circuits from top design centers will take the place of these works. This extra material has made a big difference between what is considered acceptable form in IC design and what is considered good documentation in other engineering or scientific fields, which are now only looked at and much more carefully.

It is clear that huge amounts of money spent on technology are wasted since there is far more congestion than is normal in much of the developed world. In big companies, the better a datasheet is written, the more it goes against what some people still think is good form for other fields. This is when natural selection starts. Dead projects continue to exist because they use up too many resources compared to a lot of current research in areas where they have no chance of getting big benefits. The datasheets that go along with them, in one field after another, are slowly being changed with good results during normal office hours. So, expansion keeps going in the direction of applicants for projects that cannot be patented yet would still make a tiny market if they were close enough to what the person wanted. However, the resources that are put into those alternatives keep being used in different fields, which leads to more growth in technically unpatentable non-dead projects.

TOOLS AND WAYS TO DO THINGS

VHDL and Verilog are the two main HDLs. Their basic ideas and grammar are simple, but their real-life use requires a lot of practice with complicated procedures and methods. A whole VLSI chip can normally be described in less than thousand words. However, cleanup, synthesis, simulation, timing, and other fixes and changes may add up to several thousand lines of HDL. Some good books to start with and use as references are P. Gray's summary of Silicon Compilation [1] and M. Vedula's study of verification and testing methodologies [7].

Academic institutions involved in the Microelectronics Advanced Research Corporation (MARCO) and its Focus Center on VLSI Technology Development have demonstrated the ability to systematically and coherently evaluate numerous EDA tools for designing VLSI chips. Another great place to start is with one or two free open-source or academic design tools, such as block design and layout, schematic capture, place-and-route, or back-end setups. Tutorials for these programs are also available. These products encourage their use in classes and seminars, making them even more helpful for students. Major suppliers also offer these summarized features in different ways.

Languages for Describing Hardware and Simulation

HDL are important for converting digital designs into simulation models. This allows the design to be checked before making a physical prototype. VHDL, an early standard created by the US Department of Defense, allows detailed descriptions of systems at the architectural and gate levels, as well as structural and behavioral representations [17]. HDLs are used to record, model, and simulate designs so that problems can be identified early, which saves a lot of money.

CAD and EDA Packages

CAD and EDA design tools are important for simulation, verification, and validation. They allow engineers to check for logical completeness, functionality, and timing to ensure that the specifications are correct. Formal verification faces significant hurdles because of the complexity of contemporary designs, particularly when it becomes unfeasible to exhaust all combinations of inputs and states [7]. A hierarchical method, on the other hand, breaks down complexity into the basic modular structure of designs. Higher inputs and states are symbolically expressed, enabling valid, finite, lower representations.

Higher functionality can be verified by tracing the output ranges on certain input sequences without trying every possible combination [18]. Therefore, when state-space explosions make it impossible to fully check safety attributes, designers can still obtain important information about remaining-safe pieces along limited paths. This helps to improve the style and lower the total risk. Symbolic simulations, such as STE, ensure invariant actions, such as never holding, by demonstrating persistence using iterative closures. For a safety property, unfeasible paths indicate that there are provisions that confirm the absence of errors or the retention of assurances. All safety properties validated using the STE serve as proof of correctness for cyclic systems.

Integration of High-Level Synthesis and System-on-Chip

High-level synthesis (HLS) takes high-level design specifications written in C, C++, SystemC, or SystemVerilog and turns them into RTL implementations in Verilog or VHDL. HLS makes you far more productive than creating RTL code. When you switch from RTL to high-level specifications, the code density decreases by 7 to 10 times. When many programmers work on the same project, high-level specifications make it easier to combine their work and find bugs. HLS enables the design of software and hardware together sooner in embedded systems by allowing virtual platforms that emulate the target architecture at levels of abstraction above RTL. HLS makes it possible to automatically create "components" that perform tasks such as filtering, encryption, and video processing from a high-level specification.

This is what makes component-based electronic system level design possible. Electronic system level design involves systems by employing a mix of high-level modelling, architectural exploration, and simulation. HLS is an important part of component-based electronic system level design since it costs a lot of money to make reusable parts. HLS is a one-time effort instead of a design process if there are no automatic synthesis tools. This only shifts the focus of the specification problem.

A specification is only useful if a designer can implement it [10]. HLS involves three steps: first, changing a high-level specification (such as C, C++, SystemC, or SystemVerilog) into RTL (Verilog or VHDL); second, changing the RTL into a gate-level netlist (getting ready for silicon); and third, creating the physical design from the netlist. Scheduling, technology mapping, and register allocation are examples of the tasks that must be performed to construct a series of actions with certain dependencies. Many ICs include built-in memory, and discussing their use might include both high-level and RTL specifications. A good programming style makes it clear how to access memory by utilizing special operators [11].

GETTING TO KNOW THE INDUSTRY AND GROWING YOUR CAREER

Just because you take academic classes in VLSI design does not mean you will get a job in the field. All students must work hard to learn skills and knowledge that are not taught in the classroom by doing real-world projects such as internships and co-op assignments. Networking and meeting industry guests one-on-one during campus recruitment and through professional organizations are important for being ready for a job and moving up in your career. Internships and research assistantships considerably improve your chances of receiving a job offer, and joining professional groups encourages you to keep learning and updating your skills.

Internships, co-op jobs, and research assistantships provide a taste of both academic and industry research and help you learn more about specific projects. Internships show that someone can use their talents in the real world, while summer and semester breaks allow them to learn new things while their classes are on hold. Internships are important for students graduating at the conclusion of an economic cycle and wanting to work in another country. A good VLSI industry mentor helps students become ready for the job market by sharing their experiences, telling them what skills and knowledge they need, and motivating them to keep learning and growing.

Internships, Co-Ops, and Research Assistantships

Students considering a full-time profession in chip design and verification must learn about the ideas and procedures used in the field. You can gain this type of experience through university connections with VLSI businesses or personal connections with VLSI graduates. The two best methods to achieve this are through short summer internships or extended co-op projects. Most students have a summer internship for 16 weeks, which is the summer before their senior year. Co-op work can run for one semester or a full year and can even occur after the senior year. Some students apply for and get full-time jobs in VLSI after their summer internships, while others look for jobs in other fields. If a student is unsure whether they will receive a job offer at the end of their summer internship, they can return to school to complete their degree and seek a permanent job after graduation. It is not uncommon for co-op students to receive two or more job offers before they finish their studies.

During the academic year, working as a research assistant is another excellent way to learn about the sector. Most of the time, professors have research contracts that pay students to conduct thorough project work. The research area can be anything from looking into design rules to testing IC products for or with a corporation. This research necessitates practical engagement in industry-related design and/or verification activities. Students who want to gain hands-on experience in designing and testing new goods that will soon be made in a factory often choose projects that provide them with such experience.

Networking, Getting Help from Others, and Joining Professional Groups

Getting past the first problems by looking at the employment market can lead to other discoveries that make you even more interested in VLSI. Students in the middle and late stages of their studies who are thinking about a career outside of VLSI can still benefit from talking with instructors and other industry experts. Without a VLSI design group, where would the firm that made the chip on your Printed Circuit Board (PCB) be? There must be a database somewhere that has a list of all the open-source and closed-source IP blocks that are available, together with their specs. Networking in engineering is always helpful, whether it is through labs, summer internships, conferences, seminars, or visits to companies. Freshmen should look for mentors who have been assigned to them and learn as much as they can about how to choose a major. Sophomores, juniors, and seniors should participate in work-study programs outside of their major; the broader experience is what matters most. Graduate students should look for Research Assistantships during the summer and while they are working on their degrees. Graduate students looking for work should talk to their mentors and alumni of their degree program. Seniors going out for technical interviews should also talk to their advisors, who know the people who are doing the interviews very well.

Engaging in hobbies and non-technical pursuits that invigorate creative impulses provides an equilibrium in life. Writing articles for student papers can help improve your writing and technical skills. Joining a professional society, on the other hand, might be the best course of action. IEEE and Association for Computing Machinery (ACM) both have student chapters at big colleges, and being a part of one can make a big difference. A prepared resume should be included with your membership application. The cost of membership is often less than a trip to a local fast-food restaurant, but it is one of the best investments you can make. As a member, you can read IEEE journals and attend seminars and speeches, some of which the institution does not offer. The subscription also gives access to Resume Central, where businesses post jobs that are good for student co-ops and interns with sponsor financing.

Career Paths: IP, Foundry, and IC Design

Most students want to work as IC designers for companies that create chips, mobile phones, and the internet. An IC designer usually creates the chip logic, synthesizes it, checks that it fits the verification goals, and then gives it to a physical designer. Therefore, the key goal for students should be to excel in these front-end jobs. A summer job with an IC design group is the best way to learn this. Chip design cycles are rather long; therefore, designers who have been doing it for a while can stay at the same company for a long time. In chip design, people who frequently change jobs are looked down upon more than in other fields, where a chip may be considered a star at one time and a failure at another.

Some people want to work as design engineers in large foundries such as Intel, Taiwan Semiconductor Manufacturing Company (TSMC), and GlobalFoundries. more want to work as mask or fabrication engineers in IC design businesses. More students want to work for companies like Advanced Semiconductor Materials Lithography(ASML) and KLA-Tencor, which make lithography and other technology for making and testing chips. Heavy investments in cutting-edge foundry techniques every year make it difficult for resources to compete with each other. Many design organizations are working on cheap chips using outdated techniques. Others rent silicon from foundries to test their novel architectures and operating systems at a low cost. Students should carefully research the companies they are interested in to ensure that strong and stable customers support their business. Large and small companies also make specialized parts, such as sensors, ASICs, and Digital Signal Processors (DSPs) or EDA tools, such as IP, synthesis, placement-and-route, or other EDA tools.

WAYS TO GET READY FOR A JOB

Internships, co-ops, and research assistantships are examples of dedicated programs that help students gain experience in the field before they graduate. These are very important for students who wish to work in VLSI design. However, it is crucial to remember the big picture. If arranged correctly, students can conduct hands-on research projects or personal explorations in the same academic semester and region. Getting to know people in your field, obtaining a research assistantship, or joining a professional group can all be helpful ways to get help, advice, and information.

When preparing for job interviews, you need to think carefully about your Curriculum Vitae (CV) and portfolio. The portfolio should show that the person has a good academic record and their areas of experience. Most significantly, it should demonstrate originality in design, depth of learning, and capacity to handle real-world problems. The way you prepare for an interview is also important, although it is less noticeable. You should start preparing for an interview long before it occurs. Technical interviews are frequently the hardest form of interview; therefore, you need to put in a lot of work to get ready for them. You should also try to practice using as many different resources as possible. Even if they do not seem to be connected, you should not ignore simple, everyday technical questions because they are typically used to test your thought process. To complete the process of preparing for interviews and job hunts, you must continue learning and changing your abilities.

Résumé, Portfolio, and Project Stories

The search for a job in integrated circuit design or VLSI starts long before graduation day. Therefore, college and graduate school students need to prepare materials when job applications

ISSN: 2249-474X (Online), ISSN: 2321-6492 (Print)

open. To do this, you need a clear, short, and professional résumé to showcase your skills. An electronic portfolio, which can be housed on a personal website or shared through a file-exchange service, shows important project work that adds to the résumé. This usually includes two or three sample projects that show a wide range of skills and are done as part of formal classes or outside of school. Each project description, which can be as short as a few sentences or as long as a full page, should include the project's main goals and purpose, the main reasons for choosing the project, any relevant technical specifications or limitations, an overview of the implementation process, and a summary of what was learned from it. The following pictures provide examples of a résumé, a project description focused on programming, and a project description focused on circuit design [19].

Technical Interviews and Tests

Technical interviews are a significant part of the job search for technical jobs such as VLSI design. They differ from regular interviews. Technical interviewers look for individuals who have the abilities and potential to do well in the position, rather than only their past experience and soft skills.

Candidates should be ready for interviews that test their knowledge and skills in several important areas. Candidates may choose to focus on two main skills: being able to solve easy problems in the main skill areas of the work and being able to use core knowledge to tackle known or unknown challenges. Candidates should strive for extensive and profound knowledge that encompasses fundamental concepts: breadth, as no individual possesses complete knowledge, hence enabling the identification of knowledge gaps, and depth, sufficient to facilitate the efficient and accurate resolution of component issues. Candidates should always remember that new information is constantly being added to the pool of humanistic knowledge. They should also remember the importance of being flexible, looking for solutions, and using all available resources wisely, both during development and when needed unexpectedly. This broke down any essentialist or fatalistic view of the world.

One typical expectation from technical interviews is to perform live coding on a whiteboard or in an Integrated Development Environment (IDE). This means addressing issues with formal mathematical accuracy, not the cute, smart programming shortcuts that allow you to write code quickly. Many people think that the orange ponytail in a T-shirt who drinks Pepsi and rides a skateboard to the interview will need less serious concentration on the problem description and solution than a suit. Before starting to design and code, candidates should ensure that they grasp all parts of the issue statement. As they go along, they should also explain their thought processes clearly and simply.

Continuing Education and Skill Adaptation

New job openings in microarchitecture, systems design, and optimization approaches require people to learn new skills and continue learning [10]. Research in design space exploration includes customizing processors and adding new instructions to the instruction set. Evolutionary multi-objective optimization, decision-theoretic approaches, and convex optimization help in trade-off analysis and make designs more efficient. Supervised learning, random forests, and cross-layer modelling are machine learning methods that can help predict performance and assess reliability. Continually investigating these paths keeps abilities up to date, which is necessary for designing flexible electronic systems.

ETHICAL, LEGAL, AND SOCIAL ISSUES IN VLSI

Automatic application-specific instruction set extensions constrained by microarchitecture are significant factors in VLSI. Design trade-offs, such as resource sharing in hierarchical synthesis and multi-objective optimization, affect the performance of processors. To evaluate design methodologies, the design space, trade-offs in customizable processors, and simulation of circuit reliability must be considered. It is also important to include legal and moral issues, including security, data privacy, and effects on society. Keeping up with new technologies, such as Moore's law and improvements in CMOS technology, helps make design choices that are both ethical and long-lasting.

It is still important to include legal and ethical issues, including security, data privacy, and effects on society. Keeping up with new technologies, such as Moore's law and improvements in CMOS technology, helps VLSI design experts work in an ethical and environmentally friendly manner.

CONCLUSION

ICs constitute the main component of today's electronics. They also fuel many new technologies, such as the Internet of Things (IoT), artificial intelligence (AI), autonomous vehicles, and virtual reality (VR). As IC technology continues to miniaturize, integrated circuits have become increasingly complex. In addition to design and functionality, the industry now requires IC designers to ensure that performance, power, area, reliability, security, testability, manufacturability, and other design factors are also considered at the design stage. These growing expectations mean that designers need to know not only the basics of VLSI design very well, but also how to use that knowledge to address real-world design challenges.

Transitioning hard to go from college to work can be difficult. Therefore, there are proposals and recommendations for the educational path and the instruments needed. We also discussed how to enter the VLSI design field and what the industry is like. One can develop a strong foundation for a successful career in VLSI design by following a defined educational path and using the tools, industry knowledge, and job preparation tactics learned in college and at work.

REFERENCES

1. Gray JP. Introduction to silicon compilation. 16th Design Automation Conference, San Diego, CA, USA. 1979. p.305–306. doi:10.1109/DAC.1979.1600123.
2. Colocassides EM. Educational management: Contemporary strategic communication decisions for European higher education institutions. Int J Appl Sci Eng. 2024;12(1):29–45. doi:10.30954/2322-0465.1.2024.4.
3. Carlson S, Gadio CT. Teacher professional development in the use of technology. In: Haddad WD, Draxler A, editors. Technologies for Education: Potentials, Parameters, and Prospects. Paris (FR): UNESCO; Washington (DC): Academy for Educational Development; 2002. p.118–132.
4. Yıldırım B, Akcan AT, Öcal E. Teachers' perceptions and STEM teaching activities: Online teacher professional development and employment. J Balt Sci Educ. 2022;21(1):84–107. doi:10.33225/jbse/22.21.84.
5. Madishetty S. Formation and characterization of n/p shallow junctions in sub-micron MOSFETs [master's thesis]. Newark (NJ): New Jersey Institute of Technology; 2002.
6. Walke A, Schlenvogt G, Kurinec S. Design strategies for ultra-low power 10 nm FinFETs. Solid State Electron. 2017;136:75–80. doi:10.1016/j.sse.2017.06.012.
7. Vedula VM. HDL slicing for verification and test [dissertation]. Austin (TX): The University of Texas at Austin; 2003.
8. Bhatt AS. A career anchor perspective of employees in a large company. In: Proceedings of the 2011 International Conference on Advancements in Information Technology (ICAIT) with Workshop of ICBMG 2011. IPCSIT vol. 20. Singapore: IACSIT Press; 2011. p. 265–270.

9. Cecchini T. Platform-based design, test and fast verification flow for mixed-signal systems on chip [PhD thesis]. Pisa (IT): University of Pisa; 2011.
10. Liu HY. Supervised design-space exploration [dissertation]. New York (NY): Columbia University; 2015.
11. Johnsson L. VLSI architecture and design [technical report]. Pasadena (CA): California Institute of Technology; 1980. doi:10.7907/bmm7d-81x26.
12. Nemesszeghy G. Using logic synthesis tools for Texas Instruments FPGAs. Period Polytech Electr Eng. 1994;38(2):148–174.
13. Nieman LZ, Donoghue GD, Ross LL, Morahan PS. Implementing a comprehensive approach to managing faculty roles, rewards, and development in an era of change. Acad Med. 1997;72(6):496–504. doi:10.1097/00001888-199706000-00012.
14. Buddi N. Layout synthesis for datapath designs [master's thesis]. Portland (OR): Portland State University; 1996. doi:10.15760/etd.7113.
15. Yang Z. A multiple-objective ILP based global routing approach for VLSI ASIC design [PhD thesis]. Waterloo (CA): University of Waterloo; 2008.
16. Wang Y. Extending capability of formal tools: applying semiformal verification on large design [master's thesis]. Austin (TX): The University of Texas at Austin; 2019. doi:10.26153/tsw/5533.
17. Moustakas E. Design and simulation of a primitive RISC architecture using VHDL [master's thesis]. Rochester (NY): Rochester Institute of Technology; 1991.
18. Abas MA, Zazalinda NI. Experiencing layout design techniques from highly skilled IC design engineers. In: Ismail A, Dahalan WM, Öchsner A, editors. Advanced Materials and Engineering Technologies. Cham: Springer; 2022. p.283–294. doi:10.1007/978-3-030-92964-0_28.
19. Harris D. The microprocessor as a microcosm: a hands-on approach to VLSI design education. 32nd Annual Frontiers in Education, Boston, MA, USA. 2002. p.S2D–S2D. doi:10.1109/FIE.2002.1158657.

